EnergyPATHWAYS User Guide

Release 2024.11.18

C. Farid, H. Luo, A. Gunawan, A. Pascale, and E. Larson

Nov 18, 2024

CONTENTS

CHAPTER

ONE

UI GUIDE OVERVIEW

EnergyPATHWAYS is a macro-energy demand and cost model that offers flexibility in terms of data input and geographical granularity. Users can input data in a variety of forms to project energy demand and cost into the future. Projections can be controlled by user measures that allow for the simulation and comparison of different test scenarios. This comprehensive guide is meant to supplement the documentation originally provided with the model, and aid users in maximizing its use. The guide begins with installation and first run instructions, before offering an overview of the model methodology and database setup. The latter part of this document includes a description of all relevant database files. This guide is a work in progress, and red text corresponds to unknown/uncertain information. Please email us at zerolab@princeton.edu if you would like to contribute to the guide.

CHAPTER TWO

INSTALLATION AND SETUP

2.1 Installation and Test Run Instructions (Windows)

The following installation and test run instructions are for Windows users of EnergyPATHWAYS. Mac use is possible but involves a different setup and use of EP through the command prompt (as opposed to macro functions in the scenariobuilder.xlsm). See https://energypathways.readthedocs.io/en/latest/interface.html#run-energy-pathways-on-macos for more information.

For a more detailed description and walkthrough of the EP Interface, please take advantage of the videos posted on https://energypathways.readthedocs.io/en/latest/interface.html.

1. Install the EnergyPathways Model

    a. Navigate to https://energypathways.readthedocs.io/en/latest/setup.html. Follow the main installation steps on the page (summarized in steps b-g below).

    b. Download the Anaconda (or miniconda) Distribution linked on the page.

    c. Clone the GitHub files.

        i. Download the GitHub desktop app.

        ii. Clone https://github.com/EvolvedEnergyResearch/EnergyPATHWAYS

        iii. Note: It is not recommended to download the .zip file and extract EP from there. It was unsuccessful for the group. Cloning was successful.

    d. Open the Anaconda Prompt app (found in Windows Search; may have to run as administrator).

    e. Change directory to the folder containing "environment.yml" using the "cd" command. For example, if the GitHub files were cloned into Documents, type:

    ```
    $ cd C:\\Users\\Username\\Documents\\GitHub\\EnergyPATHWAYS
    ```

f. Proceed with commands listed on website site (copied below):

```
$ cd EnergyPATHWAYS
$ conda env create -f environment.yml
$ conda activate ep
$ pip install -e .
```

g. EnergyPathways is now installed.

h. Setup Interface

    i. Go to the location where EnergyPATHWAYS is installed (where it was cloned to from GitHub). Copy the interface folder (EP interface) from the EP main folder to a new working directory (to keep the original copy preserved).

   ii. Open scenario_builder.xlsm from copied EP interface folder.

   iii. Open xlwings.config tab of sheet. Ensure the following three lines match:

      1. *Interpreter_Win*: python

      2. *Conda Path*: C:\Users\Username\anaconda3 (Note: This may be different depending on where anaconda3 is installed. See website for how to find folder location.)

      3. *Conda Env*: ep

2. Retrieve and Prepare Test Database

   a. Download the Australia test database from Dropbox.

      Reference: Davis, Dominic, Andrew C. Pascale, Bishal Bharadaj, Richard Bolt, Michael Brear, Brendan Cullen, Robin Batterham, et al. Net Zero Australia - Slide Pack and Input Data (Updated Version). Data Collection. The University of Queensland, 2024. https://doi.org/10.48610/e32f6e5.

   b. Put both folders (database and model_runs) into the new working directory (where the EP Interface copy from step h.i is).

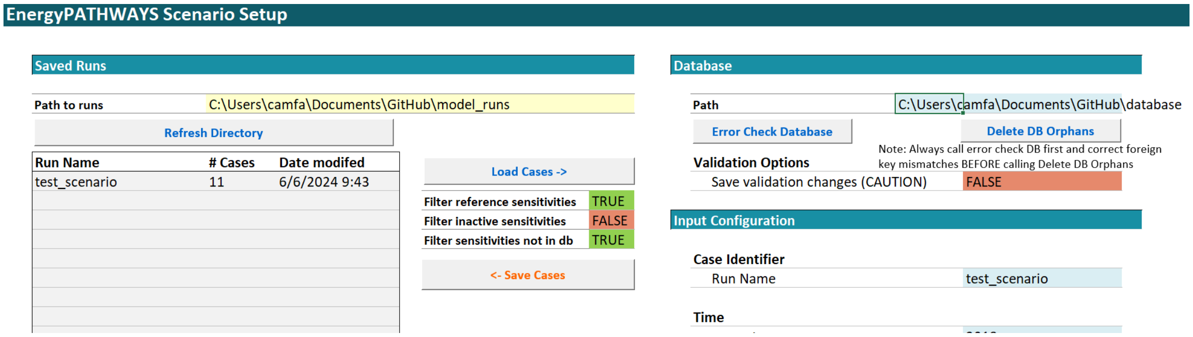
   c. Open the model_runs folder. In test_scenario, open config.INI. Edit the fourth line to include the directory of the database folder.

      i. Example: database_path = C:\Users\Username\Documents\GitHub\database

   d. Open the scenario_builder.xlsm sheet again.

   e. In the yellow box on the top left of **both** the cases and controls tab, put the model_runs folder directory.

   f. In the cases tab, under the "database" section, add the directory of the database folder to cell L6. (step may be unnecessary due to database directory specification in config.ini)



EnergyPATHWAYS Scenario Setup

Saved Runs

Path to runs: C:\Users\camfa\Documents\GitHub\model_runs

Run Name	# Cases	Date modifed
test_scenario	11	6/6/2024 9:43

Filter reference sensitivities	TRUE
Filter inactive sensitivities	FALSE
Filter sensitivities not in db	TRUE

Database

Path: C:\Users\camfa\Documents\GitHub\database

Note: Always call error check DB first and correct foreign key mismatches BEFORE calling Delete DB Orphans

Validation Options

Save validation changes (CAUTION): FALSE

Input Configuration

Case Identifier

Run Name: test_scenario

Time

Scenario builder cases tab. The yellow cell in the top left illustrates the cases tab change made in Step E. Make an identical change to the yellow cell in the controls tab.

3. Disable RIO Export Process

   a. To avoid an edge case that may crash the program, disable the RIO export component of the code (this will also speed things up). Note that this is a temporary fix since the long-term goal for EP is to replace the RIO model export component with MACRO supply side model export capabilities. Make the fix by opening the file pathways_model.py and commenting out the following two lines of code:

      i. Line 57: #export = ep2rio.RioExport(self)

      ii. Line 58: #export.write_all()

4. Load Cases into Model and Run

   a. In the cases tab, select "Refresh Directory" and let test_scenario appear in the run list (should have 11 cases). Then click on the test_scenario cell in the list and press "Load Cases" and let the data cells on the right-hand side of the sheet populate.

b. Next, go the controls tab and select "Refresh." In a similar fashion to before, test_scenario should appear in the list. Click on it again and select "Queue Selected Scenario."

c. Now, the different cases should appear in the "Case List." Remove any that you do not want to run.

    i. Note: You might just want to run one case (e.g., reference) at first to make sure it can go through without error.

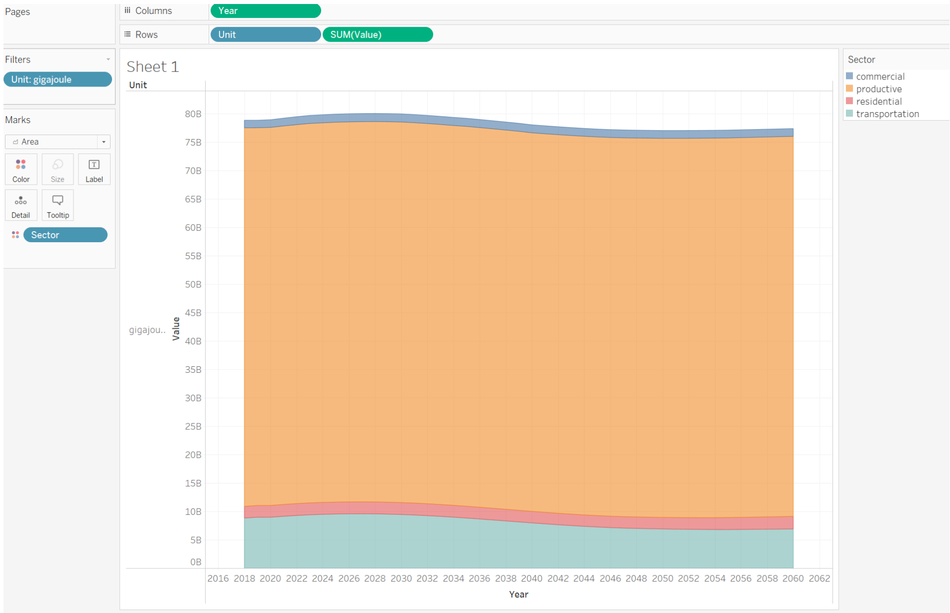
d. Adjust the number of cores and command windows in the "Cases run controls" section to control performance. Then click "Start Runs." This may take several hours.

5. Visualize Data

    a. At the conclusion of the run, select "Compile Finished Cases" to aggregate data and make it easier to examine.

    b. Download Tableau if you do not already have it and open one of the .csv files from the aggregated output folder (folder name: _aggregate_ouputs_EP).

    c. Example: To look at energy, open d_energy.csv in Tableau. Can use area chart with a setup like that in Figure below to visualize by sector or energy type.



Columns: Year

Rows: Unit, SUM(Value)

Filters: Unit: gigajoule

Marks: Area

Color: Sector

Results for energy demand in the test database reference case, color coded by sector in Tableau.

CHAPTER

THREE

RUNNING THE MODEL

3.1 Scenario Builder

“scenario_builder.xlsm” is a macro-enabled spreadsheet that serves as the main user interface for the EnergyPATHWAYS model. It simplifies the running of the python model with user-friendly buttons and controls. There are two main tabs: “controls” and “cases.”

- Controls Tab: the controls tab allows the user to “control” different components of the run. This includes specifying processing power, which subsectors will be active or inactive, and which loaded cases will ultimately be run. The following steps outline the setup process for this sheet. They are also exemplified in steps 2-4 of Section *Installation and Setup*.

  i. Paste the model_runs folder path into the yellow cell in the top left of the sheet.

  ii. Click the “Refresh” button.

  iii. Select the cell with the desired scenario (e.g., test_scenario from the test Australia database) and click “Queue Selected Scenario.” Let the “Case List” populate with the different cases specified by this scenario. See the description of runs_key.csv in Section *Key Run Files* for an explanation of cases.

  iv. Cases Control: Remove any unwanted cases from the Case List.

  v. Performance Control: In “Cases run controls,” alter the number of cores (num_cores) and command window count (cmd_window_count) to control performance allocation. parallel_process can also be set to TRUE to run multiple processes in the model simultaneously (higher performance demand).

  vi. Active/Inactive Subsectors Control: In the “Active Subsectors” section, change “Is active?” to TRUE or FALSE to enable or disable any subsectors in the run.

- Cases Tab: The cases tab loads in the different user-designed cases run by the model. As discussed in Section *Key Run Files*, cases are different demand scenarios for the future (e.g., business-as-usual, rapid electrification, rapid efficiency improvements). The following steps outline the setup process for this sheet. They are also exemplified in steps 2-4 of Section *Installation and Setup*.

  i. Paste the model_runs folder path into the yellow cell in the top left of the sheet.

  ii. Click the “Refresh Directory” button.

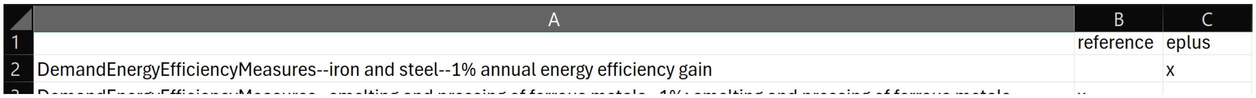
  iii. Select the cell with the desired scenario (e.g., test_scenario from the test Australia database) and click “Load Cases.” The “Database” section of the sheet should populate with information from the config.ini file, discussed more in Section *Key Run Files*. Edit any information as necessary (e.g., can change current year here if different year was specified in the config.ini file). The right-hand side of the sheet should populate with case information from runs_key.csv, also discussed in Section *Key Run Files*.

- Running: Return to the controls tab and select “Start Runs.” This will run the model. After the model is run, consider clicking “Compile finished cases” to aggregate data from different cases into one set of data sheets for easy viewing and comparing in Tableau. If data is not compiled, each case must be viewed individually from the output files in its folder (e.g., reference case output files are in “reference” folder of model_runs).

3.2 Key Run Files

The model_runs folder holds the “instructions” for running the model. Within the model_runs folder, there are subfolders corresponding to each scenario that can be loaded into the model and run. The test_scenario folder from the Net Zero Australia Dropbox is one such example. In this folder, there are three key files: logs, config.ini, and runs_keys.csv.

- logs folder: The logs folder simply stores the command line outputs that are generated when running the model. These can be useful for recalling run times or error messages when the model crashes.
- runs_key.csv: This file specifies the different “cases” that may be run by the model. Cases are different demand scenarios for the future (e.g., business-as-usual, rapid electrification, rapid efficiency improvements). The test database has 11 default cases. Each case can uniquely call on the user-induced measures placed in the database. The general format for such calls is: filename–subsector (or demand technology)–measure name.
  - For example, if the user created a measure in DemandEnergyEfficiencyMeasures.csv for the “iron and steel” subsector titled “1% annual energy efficiency gain,” it can be called as featured in Row 2 of Figure below. Note that the “x” in column C (but not in column B) denotes that the measure is only considered in the “eplus” case. The reference case is run without the efficiency measure for comparison.



	A	B	C
1		reference	eplus
2	DemandEnergyEfficiencyMeasures--iron and steel--1% annual energy efficiency gain		x

Sample Energy Efficiency Measure in Runs Key.

- The user can also put in different data for the same piece of information, perhaps at different sensitives or under different assumptions. For example, two efficiency values – a high and a low – may be put in for a gasoline car. The default selection is indicated by a [d] in the *sensitivity* column of the dataset. To call a non-default value for a case, it has to be specified in the runs_ key.csv file, in a parallel manner to that shown in Figure above.
- config.ini: This text file contains the information that will populate the cases tab when the scenario is loaded into the scenario_builder.xlsm. It includes critical information like the database file path, performance metrics, time range of the model (e.g., start year, end year), and geography specification (e.g., what geography level is the model run at, default map key for scaling data). This file should be appropriately configured for each run. See summary of user-controlled contents below:
  - DATABASE
    - ∗ database_path: database file path on computer (e.g., “C:\Users\Username\Documents\EnergyPathways\Net Zero Australia\test_stock_decay_database\database”)
  - CALCULATION_PARAMETERS (processing controls)
    - ∗ parallel_process: control whether parallel process running is enabled or not (True or False)
    - ∗ num_cores: specify number of cores to allocate from machine (e.g., 4)
    - ∗ shape_check: check demand shapes when model runs (leave True)
  - TIME (temporal controls)
    - ∗ current_year: start year of model (e.g., 2023)
    - ∗ end_year: simulate through this year (e.g., 2060)
    - ∗ weather_years: same as current_year in Australia test database

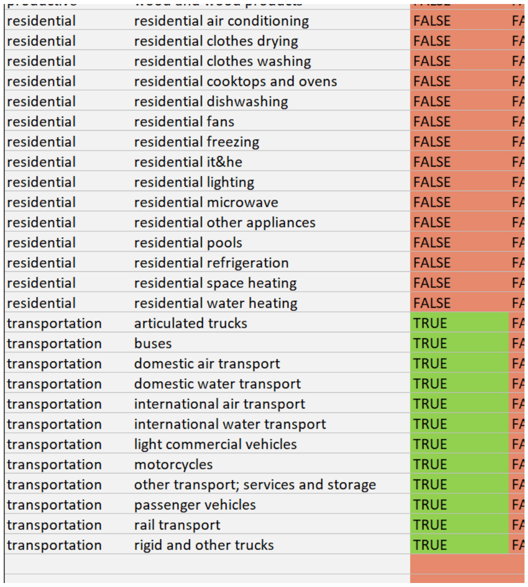
    - dispatch_outputs_timezone: time zone of output temporal demand files (from list of time zones in time_zones.csv file installed from GitHub with EnergyPATHWAYS model)
- GEOGRAPHY
    - default_geography_map_key: default map key (used to scale data across geographies if no map key is explicitly specified in data input files, e.g., population)
    - demand_primary_geography: primary geography (what geography level is the model run at)
    - primary_subset (blank for Australia test database)
    - breakout_geography (blank for Australia test database)
    - include_foreign_gaus (True for Australia test database)
    - diag_geography (sa4 – smallest region – for Net Zero Australia)
    - disagg_breakout_geography (blank for Australia test database)
- UNITS (output units)
    - energy_unit: energy output unit (e.g., gigajoule)
    - mass_unit: mass output unit (e.g., kilogram)
    - currency_name: currency output name (e.g., AUD)
    - currency_year: currency output year (e.g., 2023)
    - inflation_rate: inflation rate of currency (e.g., 0.027 for Australia test database)
- DEMAND_OUTPUT_DETAIL
    - dod_years_subset: blank for Australia test database
    - dod_vintage: track stock vintages in demand output (True or False)
    - dod_demand_technology: track demand technologies in demand output (True or False)
    - dod_cost_type: track cost type (e.g., capital, installation, etc.) in demand cost output (True or False)
    - dod_new_replacement: track whether technology was new or replacement for decayed technology in demand output (True or False)
    - dod_other_index_1: Track Other Index 1 (True or False)
    - dod_other_index_2: Track Other Index 2 (True or False)
        - See Section *OtherIndexes.csv* for more information on other indexes.
    - dod_output_hourly_profiles: Output hourly demand profiles (True or False)
    - dod_hourly_profile_final_energy_types: Control final-energy types that are reported on hourly profiles (e.g., electricity, pipeline gas; from list of final energy types specified in Section *FinalEnergy.csv*)
    - dod_hourly_profile_years: Control which years are reported for hourly profiles (e.g., "2020, 2030, 2040, 2050, 2060" would report hourly annual profiles every 10 years through 2060)
    - dod_hourly_profile_keep_subsector: Control whether hourly profiles are given on subsector level (True or False)
    - dod_hourly_profile_keep_feeder: Control whether hourly profiles are given on dispatch feeder level (True or False; see Section *DispatchFeedersAllocation.csv* for more information on dispatch feeders)
- DEMAND_CALCULATION_PARAMETERS

- use_service_demand_modifiers: Enable service demand modifiers ("True or False"). If "False," service demand is allocated to technologies in proportion to stock composition (e.g., if a technology makes up 50% of stock, it is allocated 50% of total service demand). If "True," modifiers are weighting numbers that change the allocation procedure. For example, electric vehicles may be driven more than gasoline vehicles. The electric vehicle can be weighted higher such that, even if the EV stock is equivalent to gasoline vehicles, it meets more of the service demand. See Net Zero America Annex A.2*, 2020, p. 49* for relevant equation and *p. 52-54* for more information on service demand modifiers.
- removed_demand_levels: blank for Australia test database

- RIO and RIO_DB sections are excluded since RIO export component of EnergyPATHWAYS model is disabled (as per Section *Installation and Setup*).
- LOG
  - log_level: type of information reported in log for a given run (available log levels are "CRITICAL", "ERROR", "WARNING", "INFO", or "DEBUG")
  - stdout (True or False)

3.3 Controlling Subsectors for Run

EnergyPATHWAYS has the capability to run a single sector or subsector at a time. This allows users to test run subsectors as they are inputted into the model to ensure appropriate input formatting and reasonable results. It also cuts down on time if only a certain sector or subsector's data is desired for examination. To run a single sector or subsector, open the "controls tab" and place "FALSE" under the "is active?" column for all non-desired subsectors. For example, to run only the transportation sector, disable all subsectors from the residential, productive, and commercial sectors, as shown in Figure below.



productive	wood and wood products	FALSE	F
residential	residential air conditioning	FALSE	FA
residential	residential clothes drying	FALSE	FA
residential	residential clothes washing	FALSE	FA
residential	residential cooktops and ovens	FALSE	FA
residential	residential dishwashing	FALSE	FA
residential	residential fans	FALSE	FA
residential	residential freezing	FALSE	FA
residential	residential it&he	FALSE	FA
residential	residential lighting	FALSE	FA
residential	residential microwave	FALSE	FA
residential	residential other appliances	FALSE	FA
residential	residential pools	FALSE	FA
residential	residential refrigeration	FALSE	FA
residential	residential space heating	FALSE	FA
residential	residential water heating	FALSE	FA
transportation	articulated trucks	TRUE	FA
transportation	buses	TRUE	FA
transportation	domestic air transport	TRUE	FA
transportation	domestic water transport	TRUE	FA
transportation	international air transport	TRUE	FA
transportation	international water transport	TRUE	FA
transportation	light commercial vehicles	TRUE	FA
transportation	motorcycles	TRUE	FA
transportation	other transport; services and storage	TRUE	FA
transportation	passenger vehicles	TRUE	FA
transportation	rail transport	TRUE	FA
transportation	rigid and other trucks	TRUE	FA

Controls tab subsector command window for run with only transportation sector.

CHAPTER

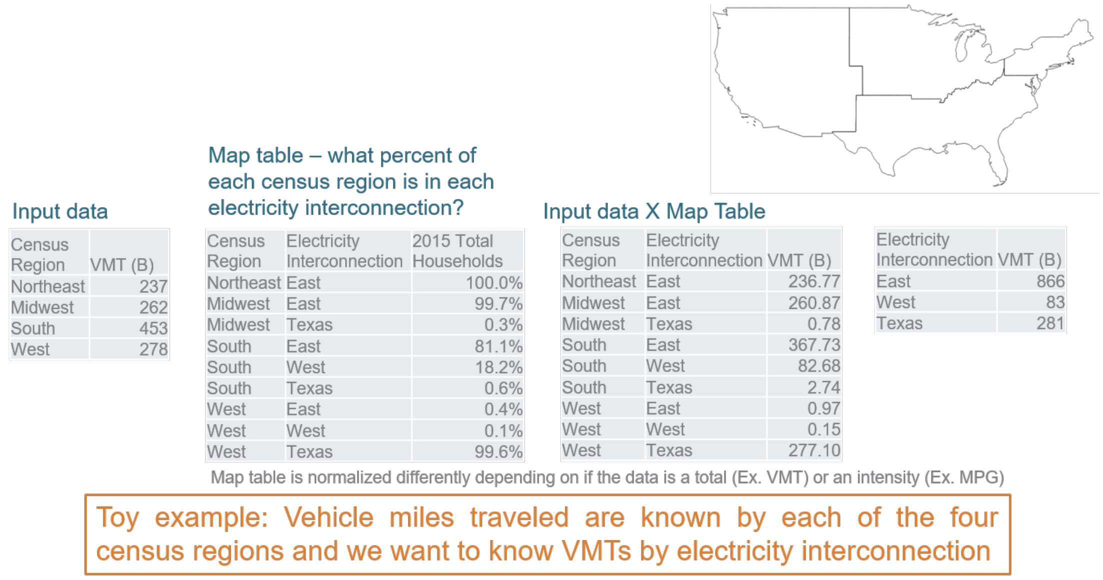
FOUR

INPUT DATABASE METHODOLOGY

At its core, the input database can be broken down into four fundamental parts: geographies, drivers, subsectors, and temporal shapes.

4.1 Geographies

Geographies: EnergyPATHWAYS allows for data input and running on a variety of geographical levels (country-wide, state-wide, city-wide, regional, etc). Map keys are used to map data between these different levels. Examples include population, number of households, land area, and steel production. For instance, if iron and steel sector energy demand is taken in on the national level, but the model is run on the state level, the demand can be broken down by state in proportion to the map key of steel production. All regions and map keys are outlined in the spatial join table (GeographiesSpatialJoin.csv). See Figure below, repurposed from the Evolved Energy Research EnergyPATHWAYS website, for another example.



Input data

Census Region	VMT (B)
Northeast	237
Midwest	262
South	453
West	278

Map table – what percent of each census region is in each electricity interconnection?

Census Region	Electricity Interconnection	2015 Total Households
Northeast	East	100.0%
Midwest	East	99.7%
Midwest	Texas	0.3%
South	East	81.1%
South	West	18.2%
South	Texas	0.6%
West	East	0.4%
West	West	0.1%
West	Texas	99.6%

Input data X Map Table

Census Region	Electricity Interconnection	VMT (B)
Northeast	East	236.77
Midwest	East	260.87
Midwest	Texas	0.78
South	East	367.73
South	West	82.68
South	Texas	2.74
West	East	0.97
West	West	0.15
West	Texas	277.10

Electricity Interconnection	VMT (B)
East	866
West	83
Texas	281

Map table is normalized differently depending on if the data is a total (Ex. VMT) or an intensity (Ex. MPG)

Toy example: Vehicle miles traveled are known by each of the four census regions and we want to know VMTs by electricity interconnection

Geography Rescaling Example

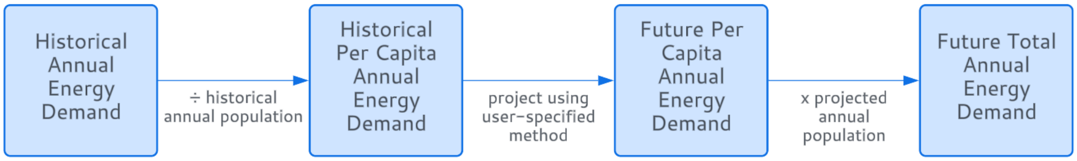
Here, VMTs known by census region are mapped to electricity interconnection region according to fraction of households in the census region that are in each electricity interconnection region.

Source: Evolved Energy Research EnergyPATHWAYS Website (Retrieved June 27, 2024)

4.2 Demand Drivers

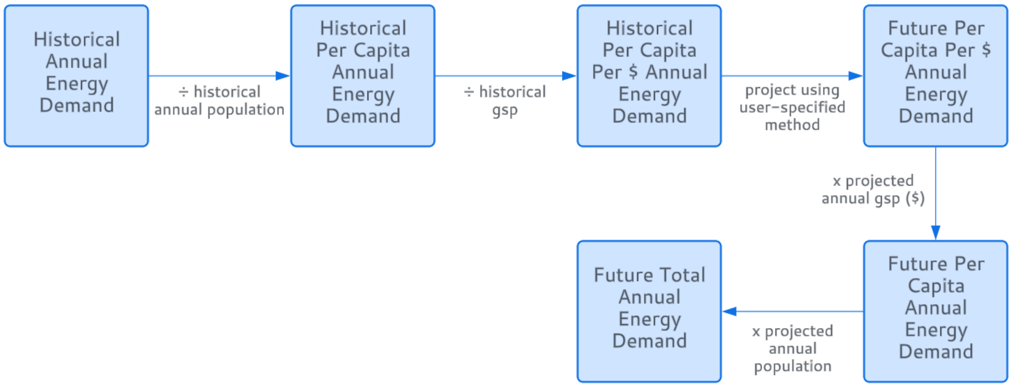
Demand drivers like population and gross state product are often correlated with energy demand. The file DemandDrivers.csv allows for historical and projected inputs of such drivers into the model.

Single Driver Example: Imagine we are projecting iron and steel energy demand to 2060 and specify a driver of population. Historical annual energy demand input for iron and steel will be divided by the population input for each year. These per capita values will then be projected into the future using the extrapolation method specified by the user (e.g., linear regression, exponential fit). Finally, the projected per capita values to 2060 will be multiplied by the respective population projection for each year to obtain a final-energy demand value. See Figure below.


Historical Annual Energy Demand
÷ historical annual population
Historical Per Capita Annual Energy Demand
project using user−specified method
Future Per Capita Annual Energy Demand
x projected annual population
Future Total Annual Energy Demand

Single Demand Driver Visual Example

Multiple Drivers Example: Multiple drivers can be employed together in a parallel manner. For each new driver, there is just an additional division and multiplication step. The model divides by each driver, projects forward, and then multiplies by each driver to get a total energy demand. For instance, if the drivers of population *and* gsp (gross state product) are specified, the model divides annual energy demand values by both population and gsp, obtaining values in units of energy per person per dollar. These values are then projected into the future and multiplied by the future population and gsp projections to obtain final-energy demand projections. See Figure below.


Historical Annual Energy Demand
÷ historical annual population
Historical Per Capita Annual Energy Demand
÷ historical gsp
Historical Per Capita Per $ Annual Energy Demand
project using user−specified method
Future Per Capita Per $ Annual Energy Demand
x projected annual gsp ($)
Future Per Capita Annual Energy Demand
x projected annual population
Future Total Annual Energy Demand

Multiple Demand Driver Visual Example

4.3 Subsector Data Input

The EnergyPATHWAYS model groups data on two main levels: sectors and subsectors. These can be defined by the user. Sectors are the primary groupings of energy demand and are often analogous to the different sectors of society. In the Net Zero Australia study, four sectors were used: transportation, residential, productive (industry), and commercial. Subsectors are the secondary groupings of energy demand and are located within each sector. For instance, in the Net Zero Australia study, the iron and steel industry was a subsector of the productive sector.

After loading in geographic and demand driver data, as outlined in Sections *Geographies* and *Demand Drivers*, the user can define the sectors and subsectors and begin to input individual subsector data. The model can be run independently for each subsector, so they can be tested as they are added. Subsector modeling is done using one of four core methods: (the first two being the most common)

1. **Energy Demand** (see Net Zero America Annex A.2*, 2020, p. 56*)**:** This method is the simplest employed by the model. It is commonly used for the productive sector, where subsector stock and service demand data are limited. Instead, users can input direct historical energy demand by fuel type in DemandEnergyDemands.csv. Historical energy demand data are used with user specified drivers and growth methods to project energy demand into the future.

    a. Users can further influence projection with efficiency measures (e.g., 1% annual growth in energy efficiency) or fuel switching measures (e.g., push a transition from coal to hydrogen in iron and steel sector). See Sections *DemandEnergyEfficiencyMeasures.csv* to *DemandFuelSwitchingMeasuresImpact.csv* for details on the configuration of such measures.

2. **Stock and Service Demand** (see Net Zero America Annex A.2*, 2020, p. 48-54*)**:** This method is the most explicit representation of energy demand in the model and requires the most amount of input data.

    a. The main data requirements include:

        1. Historical service demand data (e.g., passenger-km per year) input into the model and tied to relevant drivers in DemandServiceDemands.csv.

        2. Relevant technologies and efficiencies specified in files such as DemandTechs.csv. Each subsector using the Stock and Service Method has a set of user-inputted demand technologies that are used to meet the projected annual service demand. A demand technology can be defined as a technology that uses an energy carrier (e.g., electricity, coal) to meet a service demand (e.g., passenger-km). For example, in the passenger vehicles subsector, a light duty gasoline automobile and light duty electric vehicle are two examples of demand technologies.

        3. Initial technology stock specified in DemandStock.csv, with relevant sales measures specified in DemandSalesShareMeasures.csv. For an example of how sales share inputs are utilized, see Net Zero America Annex A.2*, 2020, p. 50-52.*

    b. Method: The model uses initial stock and sales measures to perform rollover calculations and obtain the technology stock for each future year. Service demand is allocated to stock technologies. Using technology efficiency (e.g., mpg) and service demand, final energy demand is calculated.

3. **Stock and Energy Demand** (see Net Zero America Annex A.2*, 2020, p. 54-55*)**:** This method accounts for subsectors where service demand data are not readily available, but technology stock and energy consumption data are. Instead, users can input direct energy demand and technology stock data, and the model will back solve for service demand and run in a similar method to Stock and Service.

    a. The main data requirements include:

        1. Historical annual energy demand data by demand technology input into DemandEnergyDemands.csv.

        2. Relevant technologies and efficiencies specified in files such as DemandTechs.csv and DemandTechsMainEfficiency.csv.

3. Initial technology stock specified in DemandStock.csv, with relevant sales measures specified in DemandSalesShareMeasures.csv. For an example of how sales share inputs are utilized, see Net Zero America Annex A.2*, 2020, p. 50-52.*

b. Model Calculation Method:

1. For each year that energy demand data is *explicitly* inputted into DemandEnergyDemands.csv, divide (or multiply, depending on units) by that year's technology efficiency (from DemandTechsMainEfficiency.csv) to obtain the service demand value for that year.

2. Project service demand values into missing years using the interpolation and extrapolation methods specified in DemandEnergyDemands.csv.

3. Run the "stock and service" method as usual to obtain energy demand projections.

4. **Service and Service Efficiency** (see Net Zero America Annex A.2*, 2020, p. 55*)**:** This method accounts for subsectors that have service demand data but lack the technology or stock data necessary to perform a full rollover calculation. Instead, service efficiency terms are input into DemandServiceEfficiency.csv and, combined with service demands, allow for an energy demand calculation.

4.4 Temporal Shape Data Input

EnergyPATHWAYS has the capability to craft temporal (hourly) demand curves for each energy carrier on the subsector level. The model can be instructed to do this from the config.ini file discussed in Section *Key Run Files*, where the user can specify which energy carriers to produce subsector-level temporal energy demand curves for. Demand shapes are based on input temporal shapes in the ShapeData folder of the database. All shapes are listed in Shapes.csv (Section *Shapes.csv*). For example, the file residential_electricity.csv contains residential electricity demand for every hour of the year for different geographical regions. Subsectors are assigned shapes in one of two ways:

1. **Subsector Level**: The user can specify a demand shape for an individual subsector in the "shape" column of DemandSubsectors.csv. The shape name is from the list in Shapes.csv (Section *Shapes.csv*). This shape will be used for ALL final-energy carriers' hourly profiles if generated. For example, if a shape is assigned to the iron and steel subsector, coal demand, electricity demand, and all other energy carriers' temporal demand will follow this shape. When a shape is forced on the subsector level, there is an additional data field (in the shape csv file) known as the dispatch_feeder. The user must use DispatchFeedersAllocation.csv (Section *DispatchFeedersAllocation.csv*) to specify dispatch_feeder rates. This controls what proportion of the energy demand is met by each type of dispatch_feeder shape values. For instance, imagine the dispatch_feeder types are "productive" and "residential" in the passenger vehicle subsector's shape. This means that separate shape data is available with the flag "productive" and "residential." The user must use DispathFeedersAllocation.csv to control what proportion of the subsector's shape is determined by the "productive" flagged data versus the "residential" flagged data ("1" means fully by one feeder; "0.5" and "0.5" means a 50-50 split, etc.).

2. **Sector Level**: If a subsector-specific shape is not specified, then it defaults to the sector level shape determined from DispatchFeedersAllocation.csv (Section *DispatchFeedersAllocation.csv*). Each sector is assigned a shape in the DemandSectors.csv file (e.g., Residential sector is assigned the shape residential_electricity from the list in Shapes.csv). In DispatchFeedersAllocation.csv, the user breaks down what proportion of energy in each subsector comes from each sector's shape. For example, the user can specify that, for the passenger vehicle subsector, 50% of the demand is met by the residential sector shape and 50% is met by the productive sector shape. The model will combine the two shape curves from those two sectors (which were specified in DemandSectors.csv) to produce one for the subsector. As in the first method, this shape will be used for all energy carriers in the subsector.

CHAPTER

FIVE

INPUT DATABASE CONSTITUENT FILE DETAILS

5.1 General Database File Structure

The .csv files featured below collectively represent the EP input database. Each file and its contents are explained in detail in Section 5.3. Section headers are included before file names for reference.

- Geographies
  - GeographiesSpatialJoin.csv
  - Geographies.csv
  - GeographiesMapKeys.csv
- Drivers
  - DemandDrivers.csv
- Subsector Data Input: Energy Demand Method
  - Demand Itself: DemandEnergyDemands.csv
  - Energy Efficiency Measures (e.g., 1% annual efficiency growth)
    - DemandEnergyEfficiencyMeasures.csv
    - DemandEnergyEfficiencyMeasuresCost.csv
  - Fuel Switching Measures (e.g., force a switch from gasoline to electricity)
    - DemandFuelSwitchingMeasures.csv
    - DemandFuelSwitchingMeasuresCost.csv
    - DemandFuelSwitchingMeasuresEnergyIntensity.csv
    - DemandFuelSwitchingMeasuresImpact.csv
- Subsector Data Input: Stock and Service Method
  - Service Demand
    - DemandSerivceDemands.csv
    - DemandServiceDemandMeasures.csv
    - DemandServiceDemandMeasuresCost.csv
    - DemandServiceEfficiency.csv
    - DemandServiceLink.csv
  - Technology Stock/Sales

        - DemandStock.csv
        - DemandStockMeasures.csv
        - DemandSales.csv
        - DemandSalesShareMeasures.csv
    - Technology Details
        - DemandTechs.csv
        - DemandTechsAirPollution.csv
        - DemandTechsAuxEfficiency.csv
        - DemandTechsCapitalCost.csv
        - DemandTechsFixedMaintenanceCost.csv
        - DemandTechsFuelSwitchCost.csv
        - DemandTechsIncentive.csv
        - DemandTechsInstallationCost.csv
        - DemandTechsMainEfficiency.csv
        - DemandTechsParasiticEnergy.csv
        - DemandTechsSerivceDemandModifier.csv
        - DemandTechsServiceLink.csv
- Energy Demand Temporal Patterns
    - Shapes.csv
    - ShapeData folder (energy demand temporal patterns)
        - E.g., residential_electricity.csv contains residential electricity demand for every hour of the year for different geographical regions.
    - DispatchFeedersAllocation.csv
- Other
    - DemandSectors.csv
    - DemandSubsectors.csv
    - FinalEnergy.csv
    - CurrenciesConversion.csv
    - InflationConversion.csv
    - OtherIndexes.csv

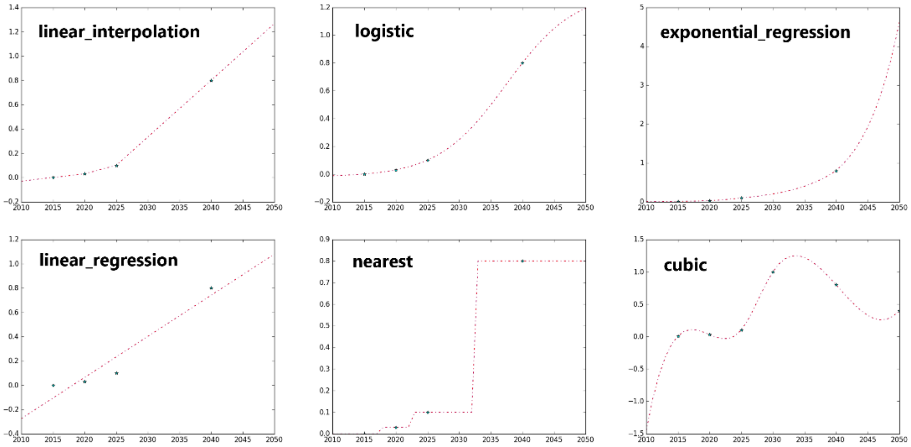
5.2 Common Terms in Database Files

The following terms appear in multiple database files and are defined here to avoid redundancy. Other terms that are uniquely used in only one database file are defined in the section of this reposit where that file is explicated.

- *age_growth_or_decay_type*: This control allows quantities to grow or decay over the lifetime of a technology. For example, a vintage 2030 gasoline car (a car that entered the technology stock in 2030) may experience a loss in efficiency as it ages. The initial efficiency value in 2030 < the value in 2031 < the value in 2032… The two options for this decay type are "linear" and "exponential." Respective growth rates are specified in *age_growth_or_decay*, described in the next term.
  - It is important to understand that this control influences how a vintage of technology changes **across its lifetime**. For instance, in the efficiency example above, this control **does not** influence the efficiency of new gasoline cars sold over time. It only controls how the efficiency of a car added to the stock in year "x" changes as that vintage car enters year "x+1", "x+2," … of its lifespan.
- *age_growth_or_decay*: This control specifies the rate at which quantities change over the lifetime of the technology according to the "linear" or "exponential" function specified in *age_growth_or_decay_type*.
  - If the "exponential" type is selected, this value is the yearly rate of change. For example, "0.01" indicates a growth of 1% each year. Similarly, "-0.01" indicates a decay of 1% each year.
  - If the "linear" type is selected, this value is what portion of the initial value the function increases or decreases by each year. For example, a value of "0.01" indicates that the quantity increases linearly by 1% of the initial value each year. If the initial efficiency value is 10 miles per gallon and a value of "0.01" is selected, the efficiency value would linearly increase by 0.01 x 10 = 0.1 miles per gallon each year.
- *cost_denominator_unit*: cost is in $/x. What is x? (e.g., GJ for gigajoules)
- *cost_of_capital*: Weighted Average Cost of Capital (WACC) (in decimal value)
- *demand_technology*: a technology that uses energy to provide a service (e.g., an electric vehicle, a gasoline vehicle, a refrigerator, a heat pump, etc.)
- *driver_1*
- *driver_2*
- *driver_3*
  - Relevant drivers (if any) for projections, as explained in Section 4.2.
- *driver_denominator_1*
- *driver_denominator_2*
  - indicates the driver that a data input has been normalized by. For example, light duty vehicle service demand may be input as annual vehicle miles traveled per person. In this case, the driver denominator would be 'population'.
- *extrapolation_method*: method used to extrapolate data beyond range of years explicitly defined in the database. (same input options as *interpolation_method*, featured below)
- *extrapolation_growth*: if (and only if) the "exponential_interpolation" method is specified in *extrapolation_method,* the user can specify a growth rate for extrapolated values here. For example, putting 0.01 for *extrapolation_growth* will force a 1% increase in extrapolated values each year. Otherwise, the interpolated curve from the previous two data points will be projected forward to obtain extrapolated values.
- *final_energy*: final-energy type of data input (any from list in FinalEnergy.csv – see Section *FinalEnergy.csv*)
- *gau*: 'geographical analysis unit' – references a single zone within a *geography* category.
  - E.g., Within a *geography* of "state," each individual state (e.g., "new jersey") is a *gau*.

  - There is a special hardcoded *geography-gau* pair of "global" (*geography*) and "all" (*gau*) used for data inputs that cover all geographical regions (e.g., national level data for Australia).

- *geography*: geographical category of data input (e.g., state, province, city)
- *geography_map_key*: used to upscale or downscale data inputs between geographies.
  - E.g., if iron and steel sector energy demand is taken in on the national level, but the model is run on the state level, the demand may be broken down by state in proportion to the map key of steel production. See additional discussion in Section *Geographies*.
- *input_type*: specifies that input data is an "intensity" or "total" value. (these are the only two options)
  - E.g., is the data for all people (total) or per person (intensity)?
- *interpolation_method*: method used to interpolate data between given years (e.g., if data is given for 2030 and 2040, how do we get data for 2035?) See list of options below. Figure below offers a visual description for several of the methods.
  - *linear_interpolation*: Adjacent data points are connected with a line to generate missing data.
  - *linear_regression*: A line of best fit is generated from all provided data points. Original data are replaced with data along this line. Missing data are found from this line. (i.e., all data is along a line of best fit)
  - *logistic*: Fits an s-curve function to the provided data points. Original data are replaced with data along this curve, and missing data are found from the curve.
  - *cubic*: Fits a cubic spline to the provided data points using the scipy library. The spline contains all the provided data points, and missing data are found along the spline.
  - *quadratic*: Fits a quadratic spline to the provided data points using the scipy library. The spline contains all provided data points, and missing data are found along the spline.
  - *nearest*: Missing data are filled to be equal to the nearest provided data point.
  - *exponential_interpolation*: Adjacent data points are connected with an exponential curve to generate missing data, similar to how linear interpolation works but exponential curves instead of straight lines. In extrapolation, there is no final point to connect a curve to. The model will use the previous curve (from the previous two points) and project it forward unless an exponential growth rate is given in the *extrapolation_growth* column. A value of 0.01 given in the *extrapolation_growth* column forces 1% annual growth in extrapolated values.
  - *exponential_regression*: Creates a best fit exponential curve for all the provided data points using the polyfit function. Original data are replaced with data along this curve. Missing data are found from the curve.
  - *average*: replaces provided values with a mean (flat line at mean value).
  - *decay_towards_linear_regression*: Only an option for extrapolation. After the final provided data point, values approach the line of best fit (the same line used in linear_regression method).
  - *forward_fill*: missing years are made equal to the closest historical year.
  - *back_fill*: missing years are made equal to the closest future year.



Graphical example of several interpolation/extrapolation methods. Provided data points are blue stars. Source: EnergyPATHWAYS Model Website.

- *is_levelized*: is the capital cost annualized? (TRUE/FALSE)
- *is_stock_dependent*: is total service demand dependent on size of stock? (TRUE or FALSE)
- *lifetime_variance:* variance in lifetime of a technology in years
  - Note: user must either specify min and max lifetime *or* mean and variance
- *max_lifetime:* maximum lifetime of technology in years
  - Note: user must either specify min and max lifetime *or* mean and variance
- *mean_lifetime:* mean lifetime of technology in years
  - Note: user must either specify min and max lifetime *or* mean and variance
- *min_lifetime:* minimum lifetime of technology in years
  - Note: user must either specify min and max lifetime *or* mean and variance
- *notes*: user notes to document assumptions (not used for computations in model)
- *other_index_1*
- *other_index_2*
  - Optional additional index categories to maintain during calculations, e.g., in the residential sector, building type can be an additional level of grouping data and viewing results.
  - See the description of OtherIndexes.csv (Section *OtherIndexes.csv*) for more information on other indexes.
- *oth_1:* one element from the *other_index_1* category. Ex. If *other_index_1* is 'building_type', *oth_1* may be 'detached single family'.
- *oth_2*: analogous to *oth_1* for *other_index_2*
- *sector*: main groupings of energy consumption (e.g., in Net Zero Australia, these are productive, commercial, residential, and transportation)
- *sensitivity:* sensitivity tag for the data input
  - E.g., for 2040 population, may have a high, medium, and low estimate depending on sensitivity ("low", "medium [d]", "high")
  - [d] indicates default option for model to select (if not specified in runs_key file)

- *shape*: hourly shape of the service demand for a sector, subsector, or technology
  - e.g., commercial_electricity, industry_electricity (productive sector), residential_elecricity
- *source*: where the data/information came from (not used in model calculations)
- *stock_decay_function:* models the rate at which vintaged stock decays and needs to be replaced. Three options are "linear," "exponential," and "weibull." See Section *Stock Rollover Calculation* for more information on the stock rollover calculations.
- *subsector*: sub-groupings of energy consumption (e.g., in Net Zero Australia, subsectors of the productive sector are specific industries, like the iron and steel industry).
- *vintage*: sales year of technology

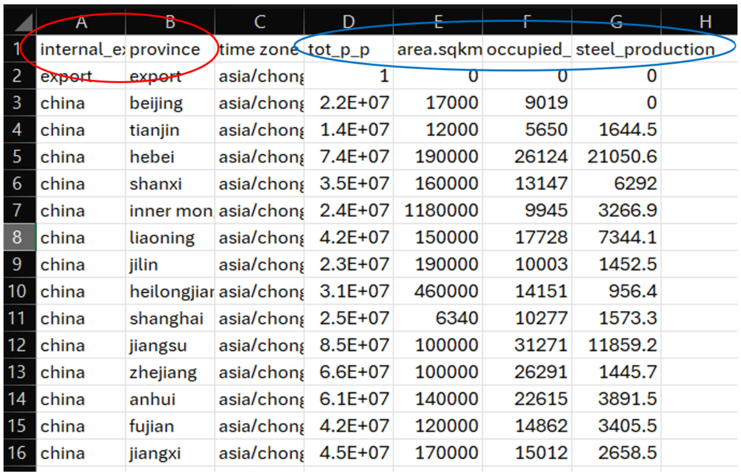
5.3 Database File-by-File Details

This section explores every database file, its purpose, and contents. Column headers without a description are featured as common terms in Section *Common Terms in Database Files*.

5.3.1 GeographiesSpatialJoin.csv

Purpose: Fundamental geography table. Indicates relevant geography types (*geography*), names (*gau*), time zones, and map keys to scale data between different granularities.

Figure below depicts a sample spatial join table for a preliminary China study. The column headers circled in red are the geography types (*geography*). In this case, data is only input on the national or provincial level. The analysis units (*gau*) associated with each type are listed below the headers. These include each individual province as well as the country as a whole ("china"). The column headers circled in blue are map keys (e.g., population, land area, occupied dwellings, steel production) with values for each region listed below them. Time zones are chosen from the time_zones.csv file that was installed from GitHub with the EnergyPATHWAYS model. It is featured in the energyPATHWAYS folder of the installation.



	A	B	C	D	E	F	G	H
1	internal_e:	province	time zone	tot_p_p	area.sqkm	occupied_	steel_production	
2	export	export	asia/chong	1	0	0	0	
3	china	beijing	asia/chong	2.2E+07	17000	9019	0	
4	china	tianjin	asia/chong	1.4E+07	12000	5650	1644.5	
5	china	hebei	asia/chong	7.4E+07	190000	26124	21050.6	
6	china	shanxi	asia/chong	3.5E+07	160000	13147	6292	
7	china	inner mon	asia/chong	2.4E+07	1180000	9945	3266.9	
8	china	liaoning	asia/chong	4.2E+07	150000	17728	7344.1	
9	china	jilin	asia/chong	2.3E+07	190000	10003	1452.5	
10	china	heilongjiar	asia/chong	3.1E+07	460000	14151	956.4	
11	china	shanghai	asia/chong	2.5E+07	6340	10277	1573.3	
12	china	jiangsu	asia/chong	8.5E+07	100000	31271	11859.2	
13	china	zhejiang	asia/chong	6.6E+07	100000	26291	1445.7	
14	china	anhui	asia/chong	6.1E+07	140000	22615	3891.5	
15	china	fujian	asia/chong	4.2E+07	120000	14862	3405.5	
16	china	jiangxi	asia/chong	4.5E+07	170000	15012	2658.5	

Sample geography spatial join table for a China study. Geography types are circled in red. Geography map keys are circled in blue.

5.3.2 Geographies.csv

Purpose: List relevant geographic types (*geography*) and regions (*gau*) for use in the database and model run.

Column Breakdown:

A. *geography*

B. *gau*

5.3.3 GeographyMapKeys.csv

Purpose: List relevant geography map keys for scaling data cross granularities.

Column Breakdown:

A. *name*: map key name (e.g., steel production 2023)

5.3.4 DemandDrivers.csv

Purpose: factors used to project energy and service demands over time. E.g., population, gross domestic product, square footage of commercial building types, vehicle miles traveled.

Column Breakdown:

A. *name*: driver name

B. *base driver*: is the driver specified in column A tied to a more fundamental driver?

   i. E.g., a driver of households may have a base driver of population. To project household size into the future, the method explained in Section *Demand Drivers* will be used, with the driver being population.

C. *input_type*

D. *unit_prefix*: value of a single unit

   i. E.g., does a value of "1" represent 1 person, 2 people, 3 people?

E. *unit_base*: what is the unit in question?

   i. E.g., people, households, square meters of floor

F. *geography*

G. *other_index_1*

H. *other_index_2*

I. *geography_map_key*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*

M. *source*

N. *notes*

O. *gau*

P. *oth_1*

Q. *oth_2*

R. *year*: year of data input

S. *value*: demand driver value (in relative or absolute units, as specified in Columns C-E)

   i. E.g., annual population value

T. *sensitivity*

5.3.5 DemandEnergyDemands.csv

Purpose: Accounts for energy demand in subsectors without enough information for a stock/service demand calculation.

Column Breakdown:

A. *subsector*

B. *is_stock_dependent*

C. *input_type*

D. *unit*

E. *driver_denominator_1*

F. *driver_denominator_2*

G. *driver_1*

H. *driver_2*

I. *driver_3*

J. *geography*

K. final_energy_index: generally blank but unsure of exact purpose

L. demand_technology_index: generally blank but unsure of exact purpose

M. *other_index_1*

N. *other_index_2*

O. *interpolation_method*

P. *extrapolation_method*

Q. *extrapolation_growth*

R. *geography_map_key*

S. *source*

T. *notes*

U. *gau*

V. *oth_1*

W. *oth_2*

X. *final_energy*

Y. *demand_technology*

Z. *year*

A. *value*: energy demand value for subsector (Column A), year (Column Z), and energy type (Column X) specified

   AB. *sensitivity*

5.3.6 DemandEnergyEfficiencyMeasures.csv

Purpose: Allows user to change energy efficiency over time for subsectors that run with the simple energy demand method (those in DemandEnergyDemands.csv).

Column Breakdown:

A. *name:* measure name (a specific name like “DDP: food; beverages and tobacco” is helpful for understanding what is represented; DDP stands for “Deep Decarbonization Pathway” as opposed to REF which stands for “Reference.”)

B. *subsector*

C. *input_type*: efficiencies are always an intensity

D. *unit*: energy efficiencies are generally unitless (blank box)

E. *geography*

F. *other_index_1*

G. *other_index_2*

H. *interpolation_method*

I. *extrapolation_method*

J. *extrapolation_growth*

K. *stock_decay_function*: not used/outdated

L. *min_lifetime*: not used/outdated

M. *max_lifetime*: not used/outdated

N. *mean_lifetime*: not used/outdated

O. *lifetime_variance*: not used/outdated

P. *source*

Q. *notes*

R. *final_energy*

S. *gau*

T. *oth_1*.

U. *oth_2*

V. *year*

W. *value*: energy efficiency gain by year specified. For example, a typical value is 0.310550914 in 2060 and 0 in 2023. This represents 1% annual efficiency gain from 2023 to 2060, since 1-0.99$^{2060\text{-}2023}$ = 0.310550914. The *interpolation_method* is used to fill in values for the intermediate years.

5.3.7 DemandEnergyEfficiencyMeasuresCost.csv

Purpose: Specification of the cost of induced efficiency improvement measures specified in DemandEnergyEfficiencyMeasures.csv.

Column Breakdown:

A. *parent*: measure name (from DemandEnergyEfficiencyMeausures.csv)

B. *currency*

C. *currency_year*

D. *cost_denominator_unit*

E. *cost_of_capital*

F. *is_levelized*

G. *geography*

H. *other_index_1*

I. *other_index_2*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*

M. *source*

N. *notes*

O. *gau*

P. *oth_1*

Q. *oth_2*

R. *vintage*

S. *value*: cost of induced efficiency improvement measure in units of *currency*/*cost_denominator_unit* (e.g., 10 $/GJ)

T. *final_energy*: (empty for NZAu)

5.3.8 DemandFuelSwitchingMeasures.csv

Purpose: Allows user to control energy type (ex. force a transition from coal to electricity) in sectors that run with the simple energy demand method (those in DemandEnergyDemands.csv).

Column Breakdown:

A. *name:* measure name (a specific name like "DDP: food; beverages and tobacco" is helpful for understanding what is represented; DDP stands for "Deep Decarbonization Pathway" as opposed to REF which stands for "Reference.")

B. *subsector*

C. *final_energy_from*: original energy type (e.g., diesel)

D. *final_energy_to*: new energy type (e.g., electricity)

E. *stock_decay_function*

F. *min_lifetime*

G. *max_lifetime*

H. *mean_lifetime*

I. *lifetime_variance*

5.3.9 DemandFuelSwitchingMeasuresCost.csv

Purpose: Incorporate cost of fuel switching measures specified in DemandFuelSwitchingMeasures.csv.

Column Breakdown:

A. *parent*: measure name (from DemandFuelSwitchingMeasures.csv)

B. *currency*

C. *currency_year*

D. *cost_denominator_unit*

E. *cost_of_capital*

F. *is_levelized*

G. *geography*

H. *other_index_1*

I. *other_index_2*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*:

M. *source*

N. *notes*

O. *gau*

P. *oth_1*

Q. *oth_2*

R. *vintage*

S. *value*: cost of fuel switching measure in units of *currency/cost_denominator_unit* (e.g., 10 $/GJ)

5.3.10 DemandFuelSwitchingMeasuresEnergyIntensity.csv

Purpose: Specify change in energy intensity as a result of fuel switching measures specified in DemandFuelSwitchingMeasures.csv.

Column Breakdown:

A. *parent*: measure name (from DemandFuelSwitchingMeasures.csv)

B. *geography*

C. *other_index_1*

D. *other_index_2*

E. *interpolation_method*

F. *extrapolation_method*

G. *extrapolation_growth*

H. *source*

I. *notes*

J. *gau*

K. *oth_1*

L. *oth_2*

M. *year*

N. *value*: new energy intensity (relative to original fuel). For example, 0.5 indicates that total energy demand for this process/subsector dropped by 50% as a result of switching fuels. This could represent switching from coal blast furnaces to electric arc furnaces (EAFs) in the iron and steel sector, since EAFs consume less energy to produce the same amount of steel.

5.3.11 DemandFuelSwitchingMeasuresImpact.csv

Purpose: Allow users to control rate at which fuel switching measures in DemandFuelSwitchingMeasures.csv occur.

Column Breakdown:

A. *parent*: measure name (from DemandFuelSwitchingMeasures.csv)

B. *input_type*: intensities are always an intensity

C. *unit*: intensities are generally unitless (blank box)

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *interpolation_method*

H. *extrapolation_method*

I. *extrapolation_growth*

J. *source*

K. *notes*

L. *gau*

M. *oth_1*

N. *oth_2*

O. *year*

P. *value*: what proportion of old fuel has been replaced by new fuel in year specified

    i. Ex. 0 indicates no replacement. 0.5 indicates 50% replacement of old fuel use with new fuel. 1 indicates full replacement of old fuel use with new fuel.

5.3.12 DemandServiceDemands.csv

Purpose: Outline annual service demands and projection methods.

Column Breakdown:

A. *subsector*

B. *is_stock_dependent*

C. *input_type*

D. *unit*: demand unit, e.g., available_seat_kilometer

E. *driver_denominator_1*

F. *driver_denominator_2*

G. *driver_1*

H. *driver_2*

I. *driver_3*

J. *geography*

K. final_energy_index: generally blank but unsure of exact purpose

L. demand_technology_index: generally blank but unsure of exact purpose

M. *other_index_1*

N. *other_index_2*

O. *interpolation_method*

P. *extrapolation_method*

Q. *extrapolation_growth*

R. *geography_map_key*

S. *source*

T. *notes*

U. *gau*

V. *final_energy*

W. *demand_technology*

X. *oth_1*

Y. *oth_2*

Z. *year*

A. *value*: service demand value

AB. sensitivity

5.3.13 DemandServiceDemandMeasures.csv

Purpose: Allow user to alter the projection of service demand in a subsector. For example, if we project a 25% decline in vehicle miles traveled, we could enter that as a measure here.

Column Breakdown:

A. *name*: measure name (user is recommended to use informative name like "DDP_lowdemand: commercial air conditioning")

B. *subsector*

C. *input_type*

D. *unit*

E. *geography*

F. *other_index_1*

G. *other_index_2*

H. *interpolation_method*

I. *extrapolation_method*

J. *extrapolation_growth*

K. *stock_decay_function*: not used/outdated

L. *min_lifetime*: not used/outdated

M. *max_lifetime*: not used/outdated

N. *mean_lifetime*: not used/outdated

O. *lifetime_variance*: not used/outdated

P. *geography_map_key*

   i. Blank for NZAu

Q. *source*

R. *notes*

S. *gau*

T. *oth_1*

U. *oth_2*

V. *year*

W. value: relative service demand growth (relative to 2020 baseline (0) in Australia)

   i. e.g., value of 0.2 in 2050 and 0 in 2020 indicates a 20% **decline** in service demand from 2020-2050.

5.3.14 DemandServiceDemandMeasuresCost.csv

Purpose: Incorporate cost of service demand measures specified in DemandServiceDemandMeasures.csv.

Column Breakdown: *Note: Empty spreadsheet for NZAu

A. *parent*: relevant service measure
B. *currency*
C. *currency_year*
D. *cost_denominator_unit*
E. *cost_of_capital*
F. *is_levelized*
G. *geography*
H. *other_index_1*
I. *other_index_2*
J. *interpolation_method*
K. *extrapolation_method*
L. *extrapolation_growth*
M. *source*
N. *notes*
O. *gau*
P. *oth_1*
Q. *oth_2*
R. *vintage*
S. *value*: cost of service demand measure in units of *currency*/*cost_denominator_unit* (e.g., $/km)

5.3.15 DemandServiceEfficiency.csv

Purpose: Allows for input of service efficiency for subsectors that run on the Service Demand and Efficiency method (when not enough information available for full stock rollover calculation).

Column Breakdown:

A. *subsector*: only used for “cement CO2 capture” for NZAu
B. *energy_unit*: ex. mmbtu
C. *denominator_unit*: ex. tonne (mmbtu/tonne overall unit)
D. *geography*
E. *other_index_1*
F. *other_index_2*
G. *interpolation_method*
H. *extrapolation_method*
I. *extrapolation_growth*

J. *geography_map_key*

K. *source*

L. *notes*

M. *final_energy*

N. *gau*

O. *oth_1*

P. *oth_2*

Q. *year*

R. *value*: service efficiency in units of *energy_unit/denominator_unit* (e.g., J/km)

S. *sensitivity*

5.3.16 DemandServiceLink.csv

Purpose: Specify relationships in service demand between subsectors. For example, the user can control what percentage of hot water service demand comes from clothes washing. Thus, if service demand for clothes washing drops, hot water service demand also drops according to the service demand ratio.

Column Breakdown:

A. *name*: user-created name of service demand linkage (e.g., "dem_svc_link_1", "dem_svc_link_2," . . .)

B. *subsector*: affecting subsector (e.g., if clothes washing contributes to hot water demand, clothes washing would be the affecting subsector since it affects hot water demand)

C. *linked_subsector:* affected subsector (e.g., if clothes washing contributes to hot water demand, hot water would be the affected subsector since its service demand was impacted by clothes washing service demand.)

D. *service_demand_share:* Value is ratio of shared service demand. For example, If the subsector is "residential clothes washing" and the linked subsector is "residential water heating," a value of "0.25" specifies that 25% of hot water demand is from clothes washing. So, if clothes washing goes down 100% in service demand (is eliminated), hot water service demand drops 25%.

E. *year*

5.3.17 DemandStock.csv

Purpose: Define stock of technologies for each subsector.

Column Breakdown:

A. *subsector*

B. *is_stock_dependent*

C. *driver_denominator_1*

D. *driver_denominator_2*

E. *driver_1*

F. *driver_2*

G. *driver_3*

H. *geography*: geographical category of data input

I. *other_index_1*

J. *other_index_2*

K. *geography_map_key*

L. *input_type*

M. *demand_stock_unit_type: unit type (generally "equipment")*

N. *unit*: stock unit (ex. truck)

O. time_unit: if time is relevant to stock unit

P. *interpolation_method*

Q. *extrapolation_method*

R. *extrapolation_growth*

S. *specify_stocks_past_current_year*: do you want to specify stock past the current year? (ex. specify 2050 stock breakdown) (TRUE/FALSE)

T. *source*

U. *notes*

V. *gau*

W. *oth_1*

X. *oth_2*

Y. *demand_technology*

Z. *year*

A. *value*: stock quantity

AB. sensitivity

5.3.18 DemandStockMeasures.csv

Purpose: Allows user to directly control stock composition (as opposed to sales measures) (only used for "2017 AEO Res Distillate Boiler" in NZAu)

Column Breakdown:

A. *name*: technology name

B. *subsector*

C. *geography*

D. *other_index1*

E. *demand_technology*

F. *interpolation_method*

G. *extrapolation_method*

H. *extrapolation_growth*

I. *source*

J. *notes*

K. *gau*

L. *oth_1*

M. *year*

N. *value*: stock size in year specified (only zeros used in NZAu)

5.3.19 DemandSales.csv

Purpose: If there are no technological classifications of historical stock data, the model will not know how to replace decayed technology in the baseline rollover (the rollover without the use of any sales share measures). An input to DemandSales.csv specifies this baseline rollover rate.

Column Breakdown:

A. *subsector*

B. *geography*

C. *other_index_1*

D. *other_index_2*

E. *input_type*

F. *interpolation_method*

G. *extrapolation_method*

H. *extrapolation_growth*

I. *source*

J. *notes*

K. *gau*

L. *oth_1*

M. *oth_2*

N. *demand_technology*

O. *vintage*

P. *value*: proportion of sales made up by *demand_technology* (Column N) in vintage year (Column O).

Q. *sensitivity*

5.3.20 DemandSalesShareMeasures.csv

Purpose: Directly control sales rates in a certain year (e.g., can specify what percent of cars sold in 2050 are EVs) to influence stock.

Column Breakdown:

A. *name*: measure name (a descriptive name is helpful. For example, “REF: CFL Exterior -> Incandescent Exterior” is referring to the replacement rate of decayed incandescent exterior lights with new CFL exterior lights in the reference case.)

B. *subsector*

C. *geography*

D. *other_index_1*

E. *demand_technology*: new technology

   i. e.g., CFL Exterior for example in A

F. *replaced_demand_tech*: technology that decayed and will be replaced by *demand_technology*

   i. e.g., Incandescent Exterior for example in A

G. *input_type*

H. *interpolation_method*

I. *extrapolation_method*

J. *extrapolation_growth*

K. *source*

L. *notes*

M. *gau*

N. *oth_1*

O. *vintage*

P. *value:* what proportion of sales in subsector are for *demand_technology*? If replacing, what proportion of *replaced_demand_tech* should be replaced by *demand_technology* upon decay in stock rollover?

5.3.21 DemandTechs.csv

Purpose: Specify relevant demand technologies for the model. A demand technology is a device that uses energy to meet a service demand.

Column Breakdown:

A. *name*: relevant technology (e.g., Electric Transit Bus)

B. linked: technology that the current technology is linked to. For example, a heat pump that provides heating is also linked to a heat pump that provides cooling.

C. stock_link_ratio: controls ratio of stock between linked technologies

D. *subsector*

E. *source*

F. *additional_description:* user notes (not used by model)

G. *demand_tech_unit_type*: technology unit type (only “equipment” used in NZAu)

H. *unit*: technology unit (e.g., bus, truck, motorcycle)

I. time_unit: time unit if relevant (blank for NZAu)

J. *cost_of_capital*

K. *stock_decay_function*

L. *min_lifetime*

M. *max_lifetime*

N. *mean_lifetime*

O. *lifetime_variance*

P. *shape*

5.3.22 DemandTechsAirPollution.csv

Purpose: Specify air pollutant emissions for technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition*: is the value "absolute" or "relative" to another demand technology

C. *reference_tech*: reference technology (if *definition* is relative)

D. *mass_unit*: unit of mass for pollutant ex. kilogram

E. *energy_unit*: ex. mmbtu

F. *geography*

G. *other_index1*: "air pollution" for this data sheet

H. *other_index2*: blank for this sheet

I. *interpolation_method*

J. *extrapolation_method*

K. *extrapolation_growth*

L. *source*

M. *notes*

N. *gau*

O. *oth_1*: specific pollutant for this data row (ex. Nox, PM2.5, Sox)

P. *oth_2*: blank for this sheet

Q. *final_energy*

R. *vintage*

S. *year*

T. *value*: pollutant value in units of mass_unit/energy_unit (ex. 0.35 kg Nox/mmbtu for gasoline LDV vintage 1960 in 1990 Australia)

U. *sensitivity*

5.3.23 DemandTechsCapitalCost.csv

Purpose: Specification of capital costs for demand technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition*: absolute or relative

   a. If "relative" definition, specify a reference technology in column C and a conversion operation in column M. For example, an operation of "multiply" in column M will multiply the reference technology's value by the relative value in column U to obtain this technology's absolute value.

C. *reference_tech*

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *currency*

H. *currency_year*

I. *is_levelized*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*

M. *reference_tech_operation*: only “multiply” used in NZAu.

N. *source*

O. *notes*

P. *new_or_replacement*: is technology cost in column U for a “new” installation or “replacement” for a decayed technology? (replacements may have lower costs)

Q. *gau*

R. *oth_1*

S. *oth_2*

T. *vintage*

U. *value*: see note under B for specifics on relative value (relative to specified reference technology by operation specified in M)

V. *sensitivity*

5.3.24 DemandTechFixedMaintenanceCost.csv

Purpose: Specification of maintenance costs for demand technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition*: “absolute” or “relative” (relative to reference technology in column C)

C. *reference_tech*

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *currency*

H. *currency_year*

I. *interpolation_method*

J. *extrapolation_method*

K. *extrapolation_growth*

L. *age_growth_or_decay_type*

M. *age_growth_or_decay*

N. *additional_description*: same as *notes*

O. *source*

P. *notes*

Q. *gau*

R. *oth_1*

S. *oth_2*

T. *vintage*

U. *value*: cost value in absolute units or relative to reference technology, as specified by Column B

V. *sensitivity*

5.3.25 DemandTechsFuelSwitchCost.csv

Purpose: Specification of fuel switching costs for demand technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition*: absolute or relative (to reference technology specified in column C)

C. *reference_tech*

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *currency*

H. *currency_year*

I. *is_levelized*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*

M. *source*

N. *notes*

O. *gau*

P. *oth_1*

Q. *oth_2*

R. *vintage*

S. *value*: fuel switch cost in absolute units or relative to reference tech, as specified in Column B

T. *sensitivity*

5.3.26 DemandTechsIncentive.csv

Purpose: Specification of incentives for demand technologies listed in DemandTechs.csv (blank for NZAu study)

Column Breakdown:

A. *demand_technology*

B. incentive_type: no examples to base on in NZAu

C. apply_lesser_of_incentives

D. *include_capital_cost:* TRUE/FALSE

E. *include_installation_cost:* TRUE/FALSE

F. *include_fuel_switching_cost:* TRUE/FALSE

G. *source*

H. *notes*

I. *geography*

J. *other_index_1*

K. *other_index_2*

L. *currency*

M. *currency_year*

N. *is_levelied*

O. *interpolation_method*

P. *extrapolation_method*

Q. *extrapolation_growth*

R. *gau*

S. *oth_1*

T. *oth_2*

U. *vintage*

V. value (in units of currency?)

W. *sensitivity*

5.3.27 DemandTechsInstallationCost.csv

Purpose: Specify installation costs for demand technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition*: absolute or relative (to reference technology in C)

C. *reference_tech*

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *currency*

H. *currency_year*

I. *is_levelized*

J. *interpolation_method*

K. *extrapolation_method*

L. *extrapolation_growth*

M. *source*

N. *notes*

O. *new_or_replacement*: is technology new or replacing an existing technology?

P. *gau*

Q. *oth_1*

R. *oth_2*

S. *vintage*

T. *value*: installation cost in currency specified (and absolute or relative units as specified in Column B)

U. *sensitivity*

5.3.28 DemandTechsMainEfficiency.csv

Purpose: Specify primary efficiencies of demand technologies listed in DemandTechs.csv.

Column Breakdown:

A. *demand_technology*

B. *definition:* absolute or relative (to reference technology in C)

C. *reference_tech*

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *final_energy*

H. *utility_factor*: energy carrier utility factor. This value allocates a share of a technology's service demand to a specific energy type. For example, a dual fuel technology might use one fuel with one efficiency value half the time and another fuel with a different efficiency value the rest of the time. In this case, the technology would have one line in DemandTechsMainEfficiency.csv and another line in DemandTechsAuxEfficiency.csv to account for both efficiency values. The utility factor x service demand will determine how much of the service demand is met by the "main" efficiency and energy type. The remaining service demand is allocated to the "auxiliary" efficiency and energy type. See Net Zero America Annex A.2*, 2020, p. 54* for more information.

I. *is_numerator_service*: TRUE/FALSE - is numerator (instead of denominator) the service unit?

   a. Ex: lightbulb service demand is in lumens. Efficiency is recorded in lumens/watt. Therefore, the numerator (lumens) is the service unit à TRUE.

   b. Counter-Ex: refrigerator service demand is cubic feet. Efficiency is recorded in kwh/cubic foot. Therefore, the denominator (cubic feet) is the service unit à FALSE

J. *numerator_unit*: ex. if efficiency in lumens/watt, numerator_unit is lumen or lumen_hour

K. *denominator_unit* ex. if efficiency in lumens/watt, deominator_unit is watt or watt_hour

L. *interpolation_method*

M. *extrapolation_method*

N. *extrapolation_growth*

O. *age_growth_or_decay_type*

P. *age_growth_or_decay*

Q. *geography_map_key*

R. *source*

S. *notes*

T. *gau*

U. *oth_1*

V. *oth_2*

W. *vintage*

X. *value*: efficiency value in units of numerator_unit/denominator_unit

Y. *sensitivity*

5.3.29 DemandTechsAuxEfficiency.csv

Purpose: Secondary efficiency specification for technologies with two relevant efficiencies (e.g., dual fuel technologies). See note under *utility_factor* in Section *DemandTechsMainEfficiency.csv* for more information.

Column Breakdown:

A. *demand_technology*

B. *definition*: absolute for both technologies used in NZAu

C. *reference_tech*: reference technology specification for a non-reference technology (blank for NZAu study)

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *final_energy*

H. *demand_tech_efficiency_types*: blank in NZAu

I. *is_numerator_service*: TRUE/FALSE - is numerator (instead of denominator) the service unit?

    a. E.g., lightbulb service demand is in lumens. Efficiency is recorded in lumens/watt. Therefore, the numerator (lumens) is the service unit à TRUE.

    b. Counter-Example: refrigerator service demand is cubic feet. Efficiency is recorded in kwh/cubic foot. Therefore, the denominator (cubic feet) is the service unit à FALSE

J. *numerator_unit*: ex. if efficiency in lumens/watt, *numerator_unit* is lumen

K. *denominator_unit* ex. if efficiency in lumens/watt, *deominator_unit* is watt

L. *interpolation_method*

M. *extrapolation_method*

N. *extrapolation_growth*

O. *age_growth_or_decay_type*

P. *age_growth_or_decay*

Q. *shape*

R. *source*

S. *notes*

T. *gau*

U. *oth_1*

V. *oth_2*

W. *vintage*

X. *value*: auxiliary efficiency value in units of *numerator_unit*/*denominator_unit*

   a. if *numerator_unit* = *denominator_unit* (e.g., both btu), then efficiency is unitless

Y. *sensitivity*

5.3.30 DemandTechsParasiticEnergy.csv

Purpose: Factor in parasitic energy losses for technologies listed in DemandTechs.csv. Parasitic energy loss is energy that the technology produces that does not directly contribute to meeting service demand. In NZAu, this sheet was only used for Heat Pump and Natural Gas Furnace. Why can we not just factor this into efficiency?

Column Breakdown:

A. *demand_technology*

B. *definition*: absolute or relative (to reference technology in C)

C. *reference*_tech

D. *geography*

E. *other_index_1*

F. *other_index_2*

G. *energy_unit*

H. *time_unit*

I. *interpolation_method*

J. *extrapolation_method*

K. *extrapolation_growth*

L. *age_growth_or_decay_type*

M. *age_growth_or_decay*

N. *source*

O. *notes*

P. *gau*

Q. *oth_1*

R. *oth_2*

S. *final_energy*

T. *vintage*

U. *value*: energy loss rate in units of energy_unit/time_unit

V. *sensitivity*

5.3.31 DemandTechsServiceDemandModifier.csv

Purpose: Without service demand modifiers, service demand is allocated to technologies in proportion to stock. This input allows the user to control how technologies in DemandTechs.csv absorb the service demand (e.g., new cars are driven more than old cars).

Column Breakdown:

A. *demand_technology*

B. *geography*

C. *definition*: absolute or relative (to reference technology in D)

D. *reference_tech*

E. *other_index_1*

F. *other_index_2*

G. *interpolation_method*

H. *extrapolation_method*

I. *extrapolation_growth*

J. *source*

K. *notes*

L. *gau*

M. *oth_1*

N. *oth_2*

O. *vintage*

P. *year*

Q. *value*: Without service demand modifiers, service demand is allocated to technologies in proportion to stock. This value is a weighting factor for allocating service demand to technology stock. See Net Zero America Annex A.2, *2020, p. 49* for use in equation and *p. 52-54* for more information on service demand modifiers.

R. *Sensitivity*

5.3.32 DemandTechsServiceLink.csv

Purpose: Ryan Jones will follow-up on this control.

Column Breakdown:

A. *Name*

B. *service_link*

C. *demand_technology*

D. *definition*: all absolute in NZAu study

E. *reference*

F. *geography*

G. *other_index_1*

H. *other_index_2*

I. *interpolation_method*

J. *extrapolation_method*

K. *extrapolation_growth*

L. *age_growth_or_decay_type*

M. *age_growth_or_decay*

N. *source*

O. *notes*

P. *gau*

Q. *oth_1*

R. *oth_2*

S. *vintage*

T. *value*

U. *sensitivity*

5.3.33 Shapes.csv

Purpose: Specify demand temporal shapes for use in data sheets.

Column Breakdown:

A. *name*: shape name

   a. correspond to file names in the ShapeData folder of the database, which has a .csv file for each temporal demand shape. For example, the file residential_electricity.csv has normalized power demand values for every hour of the year for several geographical regions.

B. *shape_type*: “time slice” or “weather date”

   a. “weather date” indicates that the shape has specific values for every hour of the year (e.g., residential_power.csv)

   b. “time slice” indicates different temporal scaling (e.g., EV.csv contains demand data by workday and non-workday).

C. *shape_unit_type*: all “power” in NZAu
D. *time_zone*: time zones are chosen from the time_zones.csv file that was installed from GitHub with the EnergyPATHWAYS model. It is featured in the energyPATHWAYS folder of the installation.
E. *geography*
F. *other_index_1*
G. *other_index_2*
H. *geography_map_key*
I. *interpolation_method*
J. *extrapolation_method*
K. input_type
L. *supply_or_demand*: “d” for demand in all NZAu shapes
M. *is_active*: TRUE for all NZAu shapes (you can probably disable these if desired)

5.3.34 DispatchFeedersAllocation.csv

Purpose: Breakdown of electricity supply to each subsector (e.g., 100% residential, 50% residential and 50% productive)

Column Breakdown:

A. *subsector*
B. *geography*
C. *geography_map_key*
D. *input_type*
E. *interpolation_method*
F. *extrapolation_method*
G. *source*
H. *notes*
I. *gau*
J. *dispatch_feeder*: which sector’s electricity demand shape to follow
K. *year*
L. *value:* proportion of electricity that comes from *dispatch_feeder* (e.g., “1” indicates all electricity is from specified demand shape in *dispatch_feeder*, 0.5 indicates 50% is from *dispatch_feeder*)
M. *sensitivity*

5.3.35 DemandSectors.csv

Purpose: list primary groupings of demand (commercial, productive, residential, transportation in Net Zero Australia)

Column Breakdown:

A. *name*

B. *shape*

C. *max_lead_hours*: not sure (empty for NZAu)

D. *max_lag_hours*: not sure (empty for NZAu)

5.3.36 DemandSubsectors.csv

Purpose: Outline secondary energy demand groupings used for data input/analysis in model.

Column Breakdown:

A. *name*: subsector name

B. *sector*: (productive, commercial, residential, commercial used in NZAu)

C. *cost_of_capital*

D. *sub_type*: method for projecting demand (based on data availability)

   i. Stock and Service (most preferred)

   ii. Stock and Energy Demand

   iii. Service and Energy Demand

   iv. Energy Demand (least preferred)

E. *shape*

F. *override_service_demand_unit*: specify a unit for a specific subsector (ex. residential freezing/refrigeration can be forced into units of cubic_foot)

5.3.37 FinalEnergy.csv

Purpose: Specify final energy types/details

Column Breakdown:

A. *name*: final energy type name

B. *blend_group*: relevant blend group (ex. A: gasoline à B: gasoline blend)

C. *unit_type*: mainly mass or energy (mass mainly for CO2)

D. *reverse_sign_EP2RIO*: not sure but FALSE for all NZAu. This is likely no concern if not exporting results to RIO.

E. *shape*: if relevant (for electricity used bulk_electricity in NZAu)

5.3.38 CurrenciesConversion.csv

Purpose: Convert different currencies used in input data to USD (or other common currency).

Column Breakdown:

A. currency (x)

B. year

C. value relative to USD in respective year (x/USD)

   i. USD has value of 1 for each year.

5.3.39 InflationConversion.csv

Purpose: Allow older currencies to be adjusted to account for inflation in comparison.

Column Breakdown:

A. *currency*: only USD used for NZAu (other currencies can be used in the model but are adjusted for inflation according to the USD inflation and the currency's annual exchange rate with the USD).

B. *year*

C. *value*: not sure what relative to (perhaps each other?)

5.3.40 OtherIndexes.csv

Purpose: Outline additional index (data grouping) options for use in data sheets. For example, it may be useful to view results of the residential building sector by building type. Here, "building_type" can be created as an index by listing it in the *other_index* column. The name of each specific building type, like "high-rise," would be in *name*.

Data input into other sheets can be classified in this way by putting the "building_type" option in the *other_index_1* or *other_index_2* column and the specific *name* in the *oth_1* or *oth_2* column. These do not affect the model run and just help to classify and sort data and results. A prominent index example in the model is the index of *geography* where each individual element (*name*) is a *gau*.

Column Breakdown:

A. *name*: see "purpose" for explanation

B. *other_index*: see "purpose" for explanation

CHAPTER

SIX

MODEL OUTPUT FILE DESCRIPTIONS

After the model finishes its run and results folders are saved to the model_runs folder, it is recommended to aggregate the results using the "compile finished cases" button in the controls tab. After this process is completed, several core data files are featured in the _aggregate_outputs_EP folder (in the loaded scenario folder of model_runs). These files and their contents are summarized below. Some files may be too large for viewing in Microsoft Excel. For this reason, Tableu is recommended for easy viewing and chart construction.

d_air_pollution.csv: annual air pollutant emissions for each subsector broken down by pollutant type, demand technology, energy type, geography, and case (e.g., reference).

d_annual_costs.csv: outlines annual costs for each subsector by cost type (e.g., capital, installation), demand technology, geography, and case (e.g., reference).

d_annual_costs_documentation.csv: outlines stock costs for different technologies (e.g., cost to replace a bus) by cost type (e.g., capital, OM), geography, whether item is new or replacement, case (e.g., reference), and vintage (make year).

d_driver.csv: outlines all drivers and respective projections (e.g., population) by geography, case (e.g., reference), and year.

d_energy.csv: outlines annual energy consumption for each subsector by demand technology, energy type, geography, case (e.g., reference).

d_levelized_costs.csv: outlines annual levelized costs for each subsector by cost type (e.g., capital, OM), demand technology, geography, and case (e.g., reference).

d_sales.csv: outlines annual sales of different technologies by technology type, geography, case (e.g., reference), and sector/subsector.

d_service_demand.csv: outlines annual service demand (e.g., kilometers) allocation by technology type, energy type, geography, case (e.g., reference), and sector/subsector.

d_stock.csv: outlines annual technology stock by technology type, geography, case (e.g., reference), and sector/subsector.

electricity_reconciliation.csv: used for debugging; represents mismatch between bottom-up and top-down shapes, which can help guide data updates to the bottom up shapes.

subsector_electricity_profiles.csv: outline subsector electricity demand in GJ for every hour of the year broken down by geography, case (e.g., reference), sector/subsector, and year (NZAu did 10 year increments for this output file: 20201, 2030, 2040, 2050, 2060).

subsector_liquid_hydrogen_profiles.csv: same as subsector_electricity_profiles.csv but for liquid hydrogen demand.

subsector_on-site_hydrogen_profiles.csv: same as subsector_electricity_profiles.csv but for on-site hydrogen demand.

subsector_pipeline_gas_profiles.csv: same as subsector_electricity_profiles.csv but for pipeline gas demand.

CHAPTER SEVEN

REFERENCES

1. Davis, D., Pascale, A., Vecchi, A., Bharadwaj, B., Jones, R., Strawhorn, T., Tabatabaei, M., Lopez Peralta, M., Zhang, Y., Beiraghi, J., Kiri, U., Vosshage, Finch, B., Batterham, R., Bolt, R., Brear, M., Cullen, B., Domansky, K., Eckard, R., Greig, C., Keenan, R., Smart, S. (2023). *Methods, Assumptions, Scenarios & Sensitivities, Net Zero Australia*. ISBN 978 0 7340 5704 4. https://www.netzeroaustralia.net.au/wp-content/uploads/2023/04/Net-Zero-Australia-Methods-Assumptions-Scenarios-Sensitivities.pdf

2. Jones, R., & Haley, B. (2020). *Net-Zero America by 2050 Technical Supplement (Appendix A.2)*. Evolved Energy Research. https://netzeroamerica.princeton.edu/img/NZA%20Annex%20A2%20-%20Technical%20appendix%20to%20EER%20report.pdf

3. Jones, R. (n.d.). *energyPATHWAYS Model Website*. Retrieved June 28, 2024, from https://energypathways.readthedocs.io/en

4. Davis, Dominic, Andrew C. Pascale, Bishal Bharadaj, Richard Bolt, Michael Brear, Brendan Cullen, Robin Batterham, et al. Net Zero Australia - Slide Pack and Input Data (Updated Version). Data Collection. The University of Queensland, 2024. https://doi.org/10.48610/e32f6e5.

CHAPTER

EIGHT

APPENDIX A: FURTHER MODEL METHODOLOGY EXPLANATION

8.1 Stock Rollover Calculation

For both the “stock and service” and “stock and energy” calculation methods, the model performs a stock rollover, in which vintaged (aged) technology stock is decayed and replaced with new technology stock (i.e., technology whose vintage is the current year). Stock is added each year to meet the current year’s DemandStock.csv stock size input (or projection), considering the decay of older stock. The three main decay types, as specified on the technology-level in DemandTechs.csv, include linear, exponential, and Weibull. Other relevant inputs include the maximum and minimum lifetime (*max_lifetime* and *min_lifetime* columns), or (alternatively) mean lifetime and lifetime variance (*mean_lifetime* and *lifetime_variance* columns).

Decay Type Breakdown (see shared_classes.py for decay function code):

- linear: Up until the *min_lifetime*, there is no decay in the stock, making the survival probability (the fraction of original stock still in use) 1. From the *min_lifetime* to *max_lifetime*, the survival probability drops linearly each year from 1 (full survival) to 0 (no survival). After the *max_lifetime*, survival probability remains at 0, indicating that no more stock of that vintage remains. If a *mean_lifetime* and *lifetime_variance* are inputted instead of maximum and minimum lifetime values, these will be used internally to approximate the maximum and minimum values, as per the equation below.
  - $minimum\ lifetime = mean\ lifetime - 2 \cdot lifetime\ variance^{1/2}$
  - $maximum\ lifetime = mean\ lifetime + 2 \cdot lifetime\ variance^{1/2}$
- exponential: Survival probability is determined from the exponential decay function e^{-ct}, where $c = 1/mean\_lifetime$ and *t* is a time unit (e.g., year). If max and min lifetime values are specified rather than mean lifetime and variance, the model will internally approximate the mean lifetime and lifetime variance using the equations below.
  - $mean\ lifetime = minimum\ lifetime + \frac{maximum\ lifetime - minimum\ lifetime}{2}$
  - $lifetime\ variance = \left(\frac{maximum\ lifetime - minimum\ lifetime}{4}\right)^2$
- Weibull: Survival probability is determined from the Weibull function $e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$. *t* is a time unit (e.g., year). β is a shape parameter that determines the failure rate behavior over time (note: if $\beta = 1$, the decay rate is constant (exponential decay)). α is a time scaling coefficient. Both coefficients are found internally using the mean lifetime and variance. If max and min lifetime values are specified rather than mean lifetime and variance, the model will internally approximate the mean lifetime and variance (using the same equations as above). This function has a longer tail, corresponding to small quantities of technology stock that remain in use for long periods of time (e.g., a small number of people still drive 30+ vintage cars regularly).

8.2 Initial Stock Calculation

EnergyPATHWAYS computes an initial vintaged technology stock composition at the start of a run. This accounts for the fact that, at the start year of the model, not all technology is brand new. For instance, if the current year is set to 2024, the model should not assume that all cars on the road in 2024 began their lifetimes that year. Instead, some cars made in 2023 will be on the road, along with others vintaged 2022, and so on.

To compute the initial stock, EnergyPATHWAYS uses the input stock values (in DemandStock.csv) along with technology lifetimes and decay functions (from DemandTechs.csv). Vintage 1999 and earlier vintages of technology are grouped together.

Pre-2000 vintage vehicles are grouped in the following manner: The model takes the 1999 stock and evenly distributes it across "x" years before 2000. "x" can be defined differently depending on the decay function:

- linear: $x = maximum\ technology\ lifetime$
- Weibull: $x = mean\ lifetime + 10\sqrt{lifetime\ variance}$
- exponential: $x = mean\ lifetime + 10\sqrt{lifetime\ variance}$

The model then creates a single decay function such that, each year, the decay value is equivalent to the sum of all decay of technologies vintage (2000-x) to 1999, as per the specified decay function (linear, exponential, or Weibull). For instance, if x = 10, the model will create a single decay function that is the cumulative decay of all technology vintage 1990-1999. This decay function is used for pre-2000 stock.

Each year after 2000, the model will add stock to meet the new stock size, as specified in DemandStocks.csv (or interpolated/extrapolated from values within), while accounting for the decay in previous years' stock. Stock will follow the specified decay function (linear, Weibull, or exponential) computed using the lifetime data input, as detailed in Section *Stock Rollover Calculation*.

CHAPTER NINE

AUTHORS

Cameron Farid, Hongxi Luo, Arya Gunawan, Andrew Pascale, and Eric Larson

Energy Systems Analysis Group, Andlinger Center for Energy and the Environment, Princeton University

CHAPTER

TEN

SUGGESTION CITATIONS

This manual:

- C. Farid, H. Luo, A. Gunawan, A. Pascale, and E. Larson, “EnergyPATHWAYS Model User Guide, Revision 1,” Andlinger Center for Energy and the Environment, Princeton University, August 1, 2024. (`User Guide`)

Test EP database for Australia:

- Davis, Dominic, Andrew C. Pascale, Bishal Bharadaj, Richard Bolt, Michael Brear, Brendan Cullen, Robin Batterham, et al. Net Zero Australia - Slide Pack and Input Data (Updated Version). Data Collection. The University of Queensland, 2024. https://doi.org/10.48610/e32f6e5.

CHAPTER

ELEVEN

LAST UPDATED

August 1, 2024

CHAPTER TWELVE

PDF VERSION

Link to PDF version of this document: `User Guide`.