# Surrogate Tools DB v2 User's Guide

Last updated: October 19, 2026


Introduction ..... 2

Background ..... 2

What is spatial allocation for emissions? ..... 2

How can emissions be spatially allocated? ..... 3

What are spatial surrogates? ..... 4

What types of data are used for calculating spatial surrogates? ..... 4

How to choose spatial data for emissions? ..... 5

What does spatial surrogate data look like? ..... 5

How is geographic data provided to Surrogate Tools DB? ..... 6

Example shapefile: cb_2017_us_county_500k ..... 6

Example shapefile: acs2016_5yr_bg ..... 7

How is spatial surrogate data generated? ..... 7

What is the data shapefile? ..... 7

What is the weight shapefile? ..... 7

Advanced weight shapefiles options: filter and weight functions ..... 8

How is the modeling grid specified? ..... 9

What is gap filling? ..... 9

What are the steps to defining a surrogate? ..... 10

Components of Surrogate Tools DB ..... 10

Technical details ..... 10

Overview of configuration files ..... 10

Steps to generating spatial surrogates ..... 11

Scripts ..... 12

Configuration File Reference ..... 12

Database config (database_config.csv) ..... 12

Control variables (control_variables.csv) ..... 13

Shapefile load list (shapefile_load.csv) ..... 14
Shapefile catalog (shapefile_catalog.csv) ..... 14
Surrogate specification (surrogate_specification.csv) ..... 14
Surrogate generation (surrogate_generation.csv) ..... 16
Script Reference ..... 16
utils.py ..... 16
load_shapefiles.py ..... 17
generate_grid.py ..... 17
reproject.py ..... 18
compute_surrogate.py ..... 18
Intermediate database tables created by compute_surrogate.py ..... 18
gapfill_surrogate.py ..... 19

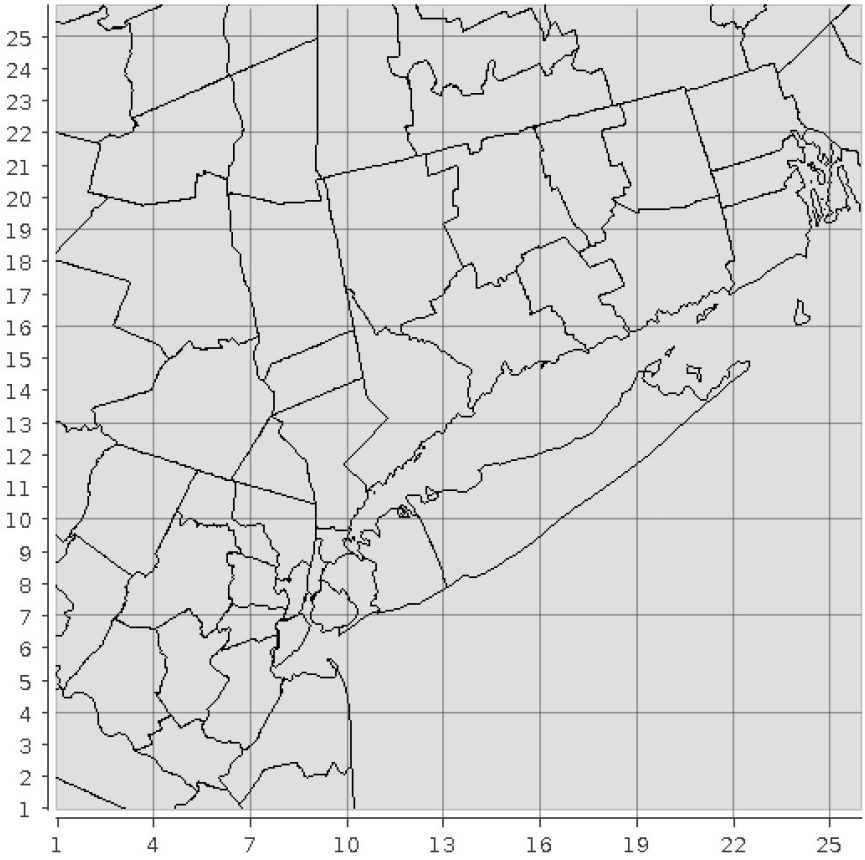
# Introduction

Surrogate Tools DB is a set of tools to develop spatial surrogate fractions, a key component of emissions processing. Spatial surrogates are used to allocate county-level (or other geographic designation) emissions estimates to modeling grid cells. Spatial surrogate fractions can be calculated using a variety of underlying spatial datasets to represent the geographical distribution of emissions, such as population and housing data, activity on road networks, or locations of individual facilities. The Surrogate Tools DB toolkit can be used to create individual surrogates and to perform gap-filling on surrogates.

# Background

## What is spatial allocation for emissions?

Emissions modeling often starts from county-level emissions estimates. The map below shows counties around the Long Island Sound (New York, Connecticut, etc.).




These emissions need to be allocated to smaller geographic areas for modeling. Typically, these smaller geographic areas are grid cells which are commonly 36-km (kilometer), 12-km, or 4-km in size. In the map above, the lines indicate 36-km grid cells while the row and column numbers correspond to 12-km grid cells. As seen in the map, individual counties can intersect multiple grid cells and a single grid cell can have contributions from multiple counties.

In addition to allocating emissions to grid cells, there may also be a need to allocate county-level emissions to census tracts or other geographic shapes.

## How can emissions be spatially allocated?

A simple approach to allocating emissions is to use the area fractions. If a grid cell is 10% of the county, it gets 10% of the emissions. All grid cells fully within the county boundaries will have the same emissions. Grid cells that aren't fully within the county get a smaller fraction of the emissions based on the amount of the grid cell within the county.

A better approach to allocating emissions is to use information about how the emissions are generated to allocate emissions more precisely. For example, emissions from extended truck idling could be concentrated in grid cells where truck stops or rest areas are located.

Similarly, emissions produced by residential buildings should be placed where homes are, not in commercial districts.

## What are spatial surrogates?

Spatial surrogates are a dataset that provides a mapping for each grid cell within the county and the fraction of emissions that should be allocated to that grid cell. Based on the underlying data used to generate the spatial surrogate dataset, the fraction for a specific county+grid-cell combination will vary.

Generically, "spatial surrogates" refers to a suite of data for the same grid with different types of underlying data. Some example surrogate types are listed below:

- Population
- Housing
- Extended Idle Locations
- Total Road Miles

EPA's emissions modeling platform uses over 70 different surrogates. Each surrogate gets a user-assigned unique surrogate code (e.g. Population = 100, Housing = 110).

When applying spatial surrogates, emission sources are assigned a surrogate based on the type of emissions using the Source Classification Code (SCC).

## What types of data are used for calculating spatial surrogates?

- Census population and number of housing units
- American Community Survey (ACS) housing unit heating data
- Potential idling locations
- Highway Performance Monitoring System (HPMS) roadway network and average annual daily traffic
- National Transportation Atlas Database (NTAD) bus terminal locations and railway networks
- National Land Cover Data (NLCD) land-use categories
- Federal Emergency Management Agency (FEMA) building square footage
- Oil refineries, tank farms, wells
- Commercial airports
- Port locations
- Shipping lanes and navigable waterways
- Golf course locations

## How to choose spatial data for emissions?

Determining which spatial data to use for a particular set of emissions depends on the source of the emissions and available spatial data that maps to where those emissions are produced.

Each sector in EPA's emissions modeling platform uses a variety of surrogates. Below are examples from the 2022v2 platform with the largest emissions:

- afdust: OSM Unpaved Roads, NLCD Total Agriculture, Total Road Miles
- fertilizer: NLCD Total Agriculture
- nonpt: Residential Heating - Natural Gas, NLCD Med + High
- np_oilgas: Well Count - Gas Wells, Well Count - Oil Wells
- np_solvents: Population, NLCD Med + High, OSM Asphalt Surfaces
- rail: NTAD Class 1 2 3 Railroad Density
- rwc: Single Family + Manufactured + Small Multi-family Residential Structures

## What does spatial surrogate data look like?

Generated spatial surrogate data consists of text files that provides a mapping for each grid cell within the county and the fraction of emissions that should be allocated to that grid cell. A sample output is shown below:

```
#GRID us12k_516x444 -2952000.000000 -2772000.000000 12000.000000 12000.000000 516 444
1 LAMBERT meters 33.000000 45.000000 -97.000000 -97.000000 40.000000
#SRGDESC=100,Population
100 01001 325 166 0.0000045092                    ! 0.2482 55052.8718
100 01001 325 167 0.0017151747                    ! 94.4253 55052.8718
100 01001 325 168 0.0020971938                    ! 115.4565 55052.8718
```

The first line of the file is a #GRID record detailing the grid definition including grid origin, grid cell size, number of rows and columns, and the map projection. Next is the #SRGDESC record with surrogate code and name. The surrogate data file can contain additional metadata records starting with a pound sign (#).

The data columns are detailed in the table below:

| Column # | Description | Data Type | Example |
|---|---|---|---|
| **1** | Spatial surrogate code | Integer | 100 |
| **2** | Country/state/county code | 6-digit YSSCCC | 01001 |
| **3** | Grid column number | Integer | 326 |
| **4** | Grid row number | Integer | 167 |
| **5** | Surrogate fraction | Real | 0.0124998228 |

The comment data after the exclamation mark (!) provides QA values. The first value is the county+cell-specific numerator and the second value is the county-specific denominator. The surrogate fraction is calculated by dividing the numerator by the denominator.

## How is geographic data provided to Surrogate Tools DB?

Currently, Surrogate Tools DB uses shapefiles as the source of input geographic data. A shapefile is stored as a group of files on disk with various file extensions (.shp, .shx, .dbf, .prj, etc.).

Shapefiles can contain point, line, or polygon shapes. Line and polygon shapes can be single or multi (e.g. polyline, multi-polygon). Each shapefile will only contain one shape type; for instance, a single shapefile can't contain both points and lines.

A shapefile has data attributes to store pieces of information for each shape. Example data attributes include:

- State and county code for an area, line segment, or point location
- Total population for an area
- Number of households in an area
- Average annual daily traffic on a line segment
- Number of berths at a port (e.g. point location)

Geographic data can come in other formats as well such as Keyhole Markup Language (KML), GeoJSON, Geodatabase, or GeoPackage (GPKG). At this time, Surrogate Tools DB doesn't have tools to load formats other than shapefiles.

*Example shapefile: cb_2017_us_county_500k*

Description: Cartographic Boundary Shapefile provided by the U.S. Census Bureau (https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.html)

This shapefile contains polygon shapes representing county boundaries.

Attributes in the shapefile include:

- STATEFP – String 2 characters (state code)
- COUNTYFP – String 3 characters (county code)
- GEOID – String 5 characters (combined state/county code)
- NAME – String 100 characters (county name)

### *Example shapefile: acs2016_5yr_bg*

Description: American Community Survey data at the census block group level (https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-data.html)

This shapefile contains polygon shapes representing census block groups.

Attributes in the shapefile include:

- STATEFP – String 2 characters (state code)
- COUNTYFP – String 3 characters (county code)
- GEOID – String 12 characters (census block group ID)
- POP2016 – Real (population)
- HU2016 – Real (housing units)
- UTIL_GAS, LP_GAS, ELEC, FUEL_OIL, COAL, WOOD – Real (number of housing units using different types of fuel for primary heating)

## How is spatial surrogate data generated?

Two shapefiles are used when generating a particular spatial surrogate: the "data shapefile" and the "weight shapefile".

### *What is the data shapefile?*

The data shapefile contains shapes corresponding to the base emissions data, typically counties. The "data attribute" is the name of the attribute in the data shapefile to use as the identifier for the base shape when creating the surrogates (e.g. the state and county code).

Example: cb_2017_us_county_500k can be used as the data shapefile, with GEOID as the data attribute.

Different data shapefiles may be used depending on the surrogate to be generated. For instance, a surrogate like Offshore Shipping may need a data shapefile that extends county boundaries over water.

### *What is the weight shapefile?*

The weight shapefile contains data at a finer geographic resolution than the data shapefile, such as:

- census block groups
- individual road segments
- point locations

This data is used to “weight” the grid-cell emissions based on the underlying data. The “weight attribute” is the name of the attribute in the weight shapefile to use as the data value when creating surrogates.

Example: acs2016_5yr_bg can be used as a weight shapefile; the specific weight attribute used changes the generated surrogate:

- Weight attribute = POP2016 ==> Population surrogate
- Weight attribute = HU2016 ==> Housing surrogate
- Weight attribute = UTIL_GAS ==> Residential Heating - Natural Gas surrogate

If the weight attribute is set to NONE, then the surrogate is calculated based on the weight shape’s area (for polygons), length (for lines), or count (for points).

A surrogate can be generated using a single shapefile for both the data shapefile and weight shapefiles. For instance, a road network shapefile may already have the line segments broken up by the county boundaries. Suppose the shapefile has attributes “fips” identifying the county code corresponding to the line segment, and “aadt” reporting the average annual daily traffic on the line segment. In this case, a Total Road AADT surrogate could be calculated by specifying the single shapefile as both the data shapefile and the weight shapefile, with “fips” as the data attribute and “aadt” as the weight attribute.

### *Advanced weight shapefiles options: filter and weight functions*

An optional “filter function” can be used to filter out shapes in the weight shapefile. For example, the ntad_2016_ipcd shapefile provides point locations for transportation facilities. In this shapefile:

- Shapes where the attribute BUS_I = 1 correspond to intercity bus terminals
- Shapes where the attribute BUS_T = 1 correspond to transit bus terminals

By using filter functions, this shapefile can be used to generate two different surrogates:

- Filter function = “bus_i=1” ==> Intercity Bus Terminals surrogate
- Filter function = “bus_t=1” ==> Transit Bus Terminals surrogate

An optional "weight function" can be used to build a weight attribute based on multiple attributes in the weight shapefile. This is an alternative to using a single weight attribute. For example, suppose a weight shapefile has the individual attributes IND1, IND2 and IND3. The weight function could be set to “IND1+IND2+IND3” and the surrogate will be calculated using the summed IND1, IND2, and IND3 attributes.

## How is the modeling grid specified?

A modeling grid is specified by the coordinates of the grid's origin (lower left corner), the grid cell size, the number of rows and columns in the grid, and the map projection.

A map projection indicates how the curved surface of the Earth should be transformed into 2-D coordinates. An example map projection used for modeling is a Lambert-Conformal projection, with Alpha = 33, Beta = 45 and Gamma = -97, with a center of X = -97 and Y = 40.

Surrogate Tools DB uses a "grid description" file, GRIDDESC, to specify map projections and modeling grids. A sample GRIDDESC file is shown below:

```
'LAM_40N97W'
  2        33.000        45.000       -97.000       -97.000        40.000
'us12k_444x336'
'LAM_40N97W'    -2736000.000   -2088000.00    12000.000     12000.000  444 336   1
```

Details for the GRIDDESC format can be found at https://cmascenter.org/ioapi/documentation/all_versions/html/GRIDDESC.html.

In the PostgreSQL database engine used by Surrogate Tools DB, the map projection becomes a "proj-string", like the following:

```
+proj=lcc +lat_1=33 +lat_2=45 +lat_0=40 +lon_0=-97 +x_0=0 +y_0=0 +a=6370000
+b=6370000 +units=m +no_defs
```

When a new map projection is defined, the proj-sting needs to be added to the "spatial_ref_sys" database table with a unique numerical spatial reference ID (SRID).

## What is gap filling?

After generating an individual surrogate, there may not be any fractions calculated for a particular county if there wasn't data in the weight shapefile corresponding to that county.

For instance, suppose a surrogate was generated based on transit bus terminal locations, but the weight shapefile didn't have any terminals located in a particular county. The grid cell fractions for the county would all be zero. However, if there are transit bus emissions associated with that county, those emissions need to be allocated somehow, otherwise the emissions would be dropped.

Gap filling is used to identify locations with a data gap and then use a secondary surrogate in those locations. The gap filling process is done after all the individual surrogates are generated.

For each surrogate, a secondary, tertiary, and quaternary surrogate can be specified. The gap filling process will first identify counties with no data. Then the missing counties will

attempt to be filled with the secondary surrogate. If additional filling is needed, the tertiary and then quaternary surrogates will be used.

The surrogates used for gap filling should get increasingly generic. For example, the Golf Courses surrogate is initially generated based on point locations for golf courses. Here are the secondary, tertiary, and quaternary surrogates used for Golf Courses:

- Secondary surrogate for the Golf Courses surrogate = Housing
- Tertiary surrogate for the Golf Courses surrogate = Population
- Quaternary surrogate for the Golf Courses surrogate = NLCD Land

### What are the steps to defining a surrogate?

1. Assign a name and code for the surrogate, e.g. Population (100) or Housing (110)

   The surrogate code will be used for assigning the surrogate to emission sources (via SCC) during the emissions processing.

2. Specify the data shapefile and data attribute
3. Specify the weight shapefile and weight attribute
4. Note the weight shapefile's geometry type
5. Determine if a filter or weight function is needed
6. Determine gap-filling fallbacks

## Components of Surrogate Tools DB

### Technical details

Surrogate Tools DB is a set of scripts written for Python 3. The scripts use the Ibis dataframe library (https://ibis-project.org).

Currently, Surrogate Tools DB uses the database engine PostgreSQL with the PostGIS 3 extension. The database engine is used to store geographic data and to calculate shape + grid cell intersections and weightings. The Surrogate Tools DB scripts need a database user with creation permissions (e.g. create scheme, create tables). Surrogate Tools DB can make use of user-specific schemas for data separation.

Surrogate Tools DB uses the GDAL (https://gdal.org/) utility ogr2ogr to read shapefile data and load into PostgreSQL.

### Overview of configuration files

All configuration files used by Surrogate Tools DB are text-based, CSV (comma-separated values) files and can be edited in Excel.

Database config

- Stores database connection information including user and database name

Control variables

- Sets up parameters needed for running Surrogate Tools DB including input and output file locations and the output grid

Shapefile load list

- List of shapefile locations on disk to load into the database

Shapefile catalog

- List of all shapefiles available (both data and weight) for use when generating surrogates

Surrogate specification

- Lists all defined surrogates including name, weights to use, and gap-filling fallbacks
- References shapefiles from the shapefile catalog input file
- The surrogate specification file isn't specific to a particular grid or run of the Surrogate Tools DB

Surrogate generation

- Lists which surrogates should be created during a particular run of the Surrogate Tools DB

## Steps to generating spatial surrogates

1. Load shapefiles from disk into the database (1 time task)
2. Set up the modeling grid (once per grid)
3. Reproject shapefiles (once per grid **projection**)
4. Generate surrogates

   This step is grid-dependent. Multiple surrogates can be generated depending on how many different weighting options are specified.

5. Gap-filling

   After individual surrogates are generated, identify gaps (counties with no grid cell fractions) and fill in with secondary surrogates. This prevents the loss of emissions during gridding step of emissions processing.

### Scripts

utils.py

This utility script can query the database to see what shapefiles and grids are loaded. It can also show shapefile reprojections and density attributes that have been added.

load_shapefiles.py

This script loads shapefiles into the database. It reads a CSV file indicating what shapefiles to load and where the files are located.

generate_grid.py

This script is used to create a modeling grid. It reads the GRIDDESC grid description file; this same file is used in SMOKE and CMAQ. The script can automatically handle Lambert-Conformal projections. For other projections, the database spatial reference will need to be set up directly.

reproject.py

This script handles reprojecting shapefiles to the modeling grid projection and also calculates density attributes.

compute_surrogate.py

This script is used to compute surrogates. The script reads the surrogate generation file to determine which individual surrogates to create. The script references the surrogate specification file for details on how to calculate each surrogate.

gapfill_surrogate.py

This script handles the gap-filling process. The script checks each surrogate in the surrogate generation file for data gaps. The script supports gap-filling with secondary, tertiary, and quaternary surrogates.

## Configuration File Reference

### Database config (database_config.csv)

Description: stores database connection information including user and database name

Used by: all Surrogate Tools DB scripts

The database config file contains two columns named “key” and “value”. Recognized keys are listed below.

| Key | Description | Default value |
|---|---|---|
| backend | Database backend type; only “postgres” support currently | postgres |
| postgres_host | PostgreSQL host | localhost |
| postgres_port | PostgreSQL port | 5432 |
| postgres_database | PostgreSQL database name | |
| postgres_user | PostgreSQL username | |
| postgres_password | PostgreSQL password | |

The PostgreSQL database user must have creation permissions (e.g. create scheme, create tables).

## Control variables (control_variables.csv)

Description: sets up parameters needed for running Surrogate Tools DB including input and output file locations and the output grid

Used by: compute_surrogate.py, gapfill_surrogate.py

The control variables file contains three columns named VARIABLE, VALUE, and DESCRIPTION. Each variable is detailed below.

| Variable Name | Description |
|---|---|
| CLEANUP INTERMEDIATE TABLES | YES = automatically deletes intermediate tables created during surrogate generation; if set to NO, the utils.py script can be used later to clean up old tables |
| GENERATION CONTROL FILE | Location of the surrogate generation file |
| GRIDDESC | Location of the grid description file |
| OUTPUT DIRECTORY | Location where the output surrogate files should be written |
| OUTPUT_GRID_NAME | Name of the grid to generate surrogates for; grid name must appear in the GRIDDESC file |
| OUTPUT SRGDESC FILE | Location to write the output surrogate description file |
| OVERWRITE OUTPUT FILES | YES = overwrite existing output files; NO = existing output files will not be overwritten |
| SHAPEFILE CATALOG | Location of the shapefile catalog |
| SRID_FINAL | Database SRID corresponding to the output grid’s map projection |

| | |
|---|---|
| SURROGATE SPECIFICATION FILE | Location of the surrogate specification file |

## Shapefile load list (shapefile_load.csv)

Description: list of shapefile locations on disk to load into the database

Used by: load_shapefiles.py

The shapefile load list has two columns “name” and “dir”. When a shapefile is loaded, the database table will correspond to the “name” column, and the files will be read from the location specified by “dir”.

## Shapefile catalog (shapefile_catalog.csv)

Description: list of all shapefiles available (both data and weight) for use when generating surrogates

Used by: compute_surrogate.py, reproject.py

| Column Name | Description | Example |
|---|---|---|
| SHAPEFILE NAME | Name of the shapefile | acs2016_5yr_bg |
| GEOMTYPE | Geometry type of the shapefile | MultiPolygon |
| DENSITY_ATTRS | Attributes to calculate density versions | "pop2016,hu2016,util_gas,wood" |
| DESCRIPTION | Description of the shapefile (metadata only) | 2016 US Census American Community Survey 5-year product with population and housing related features |
| DATA SOURCE | Data source for the shapefile (metadata only) | |
| QUESTION | Outstanding questions about the shapefile (metadata only) | |

## Surrogate specification (surrogate_specification.csv)

Description: lists all defined surrogates including name, weights to use, and gap-filling fallbacks

Used by: compute_surrogate.py, gapfill_surrogate.py

| Column Name | Description | Example |
|---|---|---|
| REGION | Surrogate region | USA |
| SURROGATE | Surrogate name | Golf Courses |
| SURROGATE CODE | Surrogate code; integer | 850 |
| DATA SHAPEFILE | Name of the data shapefile | cb_2017_us_county_500k |

| | | |
|---|---|---|
| DATA ATTRIBUTE | Attribute of the data shapefile | geoid |
| WEIGHT SHAPEFILE | Name of the weight shapefile | acs2016_5yr_bg |
| WEIGHT ATTRIBUTE | Attribute of the weight shapefile | hu2016 |
| WEIGHT FUNCTION | Optional weight function | |
| FILTER FUNCTION | Optional filter function | "moves2014>1 and moves2014<6" |
| MERGE FUNCTION | Optional merge function | |
| SECONDARY SURROGATE | Secondary surrogate name | Housing |
| TERTIARY SURROGATE | Tertiary surrogate name | Population |
| QUARTERNARY SURROGATE | Quaternary surrogate name | NLCD Land |
| DETAILS | Details of the surrogate (metadata only) | Total Housing Units from Census 2016 |
| COMMENTS | Comments about the surrogate (metadata only) | |

Example records from a surrogate specification file are shown below.

```
REGION, SURROGATE, SURROGATE CODE,
 DATA SHAPEFILE, DATA ATTRIBUTE,
 WEIGHT SHAPEFILE, WEIGHT ATTRIBUTE, WEIGHT FUNCTION, FILTER FUNCTION,
 MERGE FUNCTION, SECONDARY SURROGATE, TERTIARY SURROGATE, QUARTERNARY SURROGATE,
 DETAILS, COMMENTS
USA, Population, 100,
 cb_2017_us_county_500k, geoid,
 acs2016_5yr_bg, pop2016, , ,
 , , , ,
 Total population from Census 2016 blocks,
```

For the Population surrogate specified above, there are no surrogates for gap-filling; the columns for secondary, tertiary, and quaternary surrogate are left blank. Population is often any ultimate fallback surrogate for emissions.

```
USA, Housing, 110,
 cb_2017_us_county_500k, geoid,
 acs2016_5yr_bg, hu2016, , ,
 , Population, , ,
 Total Housing Units from Census 2016,
```

For the Housing surrogate, it uses the same weight shapefile as the Population surrogate but specifies a different weight attribute, resulting in a different set of county-to-grid-cell allocation fractions.

```
USA, Total Road AADT, 239,
 hpms2017_v3_04052020, fips,
```

```
 hpms2017_v3_04052020, aadt, , "moves2014>1 and moves2014<6",
 , Total Road Miles, , ,
 HPMS 2013 AADT for all roads,
```

The Total Road AADT surrogate uses a filter function to restrict the shapes used from the weight shapefile.

```
USA, Golf Courses, 850,
 cb_2017_us_county_500k, GEOID,
 usa_golf_courses_2019_10, , , ,
 , Housing, Population, NLCD Land,
 Point locations of Golf Courses,
```

The Golf Courses surrogate doesn't specify a weight attribute, so the number of points is used for weighting. This surrogate also specifies a full set of fallback surrogates for gap-filling.

### Surrogate generation (surrogate_generation.csv)

Description: lists which surrogates should be created during a particular run of the Surrogate Tools DB

Used by: compute_surrogate.py, gapfill_surrogate.py

| Column | Description | Example |
|---|---|---|
| REGION | Surrogate region | USA |
| SURROGATE | Surrogate name | Population |
| SURROGATE CODE | Surrogate code | 100 |
| GENERATE | Indicates if surrogate should be generate/gap-filled (YES, NO) | YES |

## Script Reference

Common arguments:

- -h|--help: show usage information
- --config: path to the database config file; defaults to database_config.csv in main directory
- --schema: database schema; default is public

In the scripts' usage information, parameters in brackets [] are optional.

### utils.py

This utility script can query the database to see what shapefiles and grids are loaded. It can also show shapefile reprojections and density attributes that have been added.

Usage:

- python scripts/utils.py shapefiles [-h] [--config PATH] [--schema SCHEMA]
- python scripts/utils.py shapefiles-ibis [-h] [--config PATH] [--schema SCHEMA]
- python scripts/utils.py grids [-h] [--config PATH] [--schema SCHEMA]
- python scripts/utils.py cleanup-intermediate --code CODE --srid SRID --grid GRID [--config PATH] [--schema SCHEMA]
- python scripts/utils.py cleanup-intermediate [-h] [--config CONFIG] [--schema SCHEMA] --code CODE --srid SRID –grid GRID

Arguments:

- shapefiles: lists loaded shapefiles using raw SQL queries
- shapefiles-ibis: lists loaded shapefiles using the Ibis library
    - Output from both shapefiles commands includes shapefile name, geometry type, SRID, number of rows, reprojections, and calculated density attributes
- grids: lists defined grids including grid name, SRID, number of rows and columns, and number of grid cells
- cleanup-intermediate: removes intermediate database tables for the specified surrogate code, projection (SRID), and grid

### load_shapefiles.py

This script loads shapefiles into the database. It reads a CSV file indicating what shapefiles to load and where the files are located.

load_shapefiles.py runs the GDAL utility program ogr2ogr to load shapefile data into the PostgreSQL database. The ogr2ogr must be in the user's path.

Usage: python scripts/load_shapefiles.py [-h] [--config CONFIG] [--load-list LOAD_LIST] [--schema SCHEMA] [--dry-run]

Arguments:

- --load-list: path to the shapefile load list; defaults to shapefile_load.csv
- --dry-run: print ogr2ogr commands without executing them

### generate_grid.py

This script is used to create a modeling grid. It reads the GRIDDESC grid description file; this same file is used in SMOKE and CMAQ. The script can automatically handle Lambert-Conformal projections. For other projections, the database spatial reference will need to be set up directly.

Usage: python scripts/generate_grid.py [-h] --griddesc GRIDDESC --grid-name GRID_NAME [--config CONFIG] [--schema SCHEMA]

Arguments:

- --griddesc: path to the grid description (GRIDDESC) file
- --grid-name: name of the grid to generate as specified in the grid description file

### reproject.py

This script handles reprojecting shapefiles to the modeling grid projection and also calculates density attributes.

Usage: python scripts/reproject.py [-h] --srid SRID --catalog CATALOG [--sql-file SQL_FILE] [--tables [TABLES ...]] [--config CONFIG] [--schema SCHEMA]

Arguments:

- --srid: target spatial reference ID (SRID)
- --catalog: path to the shapefile catalog
- --sql-file: path to SQL file to register a new SRID
- --tables: list of tables to process; default is all tables in the shapefile catalog

### compute_surrogate.py

This script is used to compute surrogates. The script reads the surrogate generation file to determine which individual surrogates to create. The script references the surrogate specification file for details on how to calculate each surrogate.

Usage: compute_surrogate.py [-h] --control-file CONTROL_FILE [--config CONFIG] [--schema SCHEMA] [--codes CODES [CODES ...]] [--dry-run]

Arguments:

- --control-file: path to control variables file
- --codes: list of surrogate codes to compute; overrides surrogate generation file
- --dry-run: print job list without executing

#### *Intermediate database tables created by compute_surrogate.py*

The following section lists the intermediate database tables created by compute_surrogate.py. If the variable CLEANUP INTERMEDIATE TABLES is set to YES in the control variables file, these tables will be automatically deleted when the script completes successfully. Otherwise, the utils.py script can be run with the cleanup-intermediate command to remove the tables.

wp_cty_[surrogate code]_[srid] (example name: wp_cty_100_900921)

- Intersections between the weight shapefile and the counties

wp_cty_cell_[surrogate code]_[grid name] (example name: wp_cty_cell_100_us12k_516x444)

- Intersections from "wp_cty" with the gridcells

numer_[surrogate code]_[grid name] (example name: numer_100_us12k_516x444)

- Calculated numerator for the surrogate fractions

denom_[surrogate code]_[grid name] (example name: denom_100_us12k_516x444)

- Calculated denominator for the surrogate fractions

surg_[surrogate code]_[grid name] (example name: surg_100_us12k_516x444)

- Final surrogate fractions. This table is output to a text file as the surrogates data.

## gapfill_surrogate.py

This script handles the gap-filling process. The script checks each surrogate in the surrogate generation file for data gaps. The script supports gap-filling with secondary, tertiary, and quaternary surrogates.

Usage: python scripts/gapfill_surrogate.py [-h] --control-file CONTROL_FILE [--codes CODES [CODES ...]]

Arguments:

- --control-file: path to control variables file
- --codes: list of surrogate codes to gap-fill; overrides surrogate generation file