

# CAMS Methane Emissions Estimation Tool

## Quickstart

Install the miniconda version of python from <https://docs.conda.io/en/latest/miniconda.html>. A python 3 is required.

Download the source code from [http://dept.ceer.utexas.edu/ceer/meet/files/CAMS-MEET-1.1\\_release%20MEET.zip](http://dept.ceer.utexas.edu/ceer/meet/files/CAMS-MEET-1.1_release%20MEET.zip) (or find link in <http://dept.ceer.utexas.edu/ceer/meet/>). Expand the archive. You should see a directory GTI-CAMS

From the start menu, start Anaconda (64-bit) -> Anaconda Prompt (Miniconda 3). This will start a command prompt window with python in the path.

Please follows the sequence of command to set up environment adequate for MEET1.

```
conda create --name meet python=3.7
```

```
conda activate meet
```

First command, conda create, is needed to run only once, and it will installs Python 3.7. Second command is needed to be issued to run MEET, so that python 3.7 is used. With these, the command prompt looks like (meet) PS C:\Users\Yosuke>

In this command prompt, type the following commands

```
cd GTI-CAMS/DES1
```

```
pip install -r ../requirements.txt
```

Download database file, thermodynamics\_database\_v2\_20200303.db from [https://dept.ceer.utexas.edu/ceer/ect/files/thermodynamics\\_database\\_v2\\_20200303.zip](https://dept.ceer.utexas.edu/ceer/ect/files/thermodynamics_database_v2_20200303.zip), and place it in GTI-CAMS/DES1/input directory.

```
python SiteMain.py to run the default case.
```

:information\_source: As of January 2022, MEET 1.x is known not to work on python 3.7, and fails with newer version. Anaconda installer has specific python version as default (as of 2023-05-15 it is python 3.10), but it allows any version of python including 3.7.

## Command Line Arguments

To run customized simulation, run SiteMain.py with additional arguments. See example 1 below.

Occasionally you want to revisit this run with different time resolution (we may call this analysis "drilldown"). Use SerializerMain.py for that purpose. See example 2 below.

In either cases, append these additional arguments after the call to SiteMain.py to control the behavior of the MEET model.

**Example 1:** I want to run the model with an input called "MyData.xlsx" without automatically generated graphs. I want to run 100 iterations using multiprocessing with 4 processors.

```
python SiteMain.py -f MyData.xlsx -g -m 100 -p 4
```

**Example 2:** I got output from the run above in a directory output/MySym\_200908\_105245. I want to take a closer look at MC instance #26, day3, in minute resolution.

```
python SerializerMain.py -f MyData.xlsx -t 200908_105245 -r 26 -s 3 -e 3 -d MINUTES
```

Argument	Description	Default	Type
-c	Allows the selection of an alternate ScenarioInstanceConfig.json file, which is required for Drilldown processing	None	
-d	Serializer granularity Drilldown units; can be SECONDS, MINUTES, HOURS, DAYS, MONTHS, and YEARS, overwriting specified values in "Plot Interval" in Global Simulation Parameters" sheet	DAYS	
-e	End of serialization time, in original sim units; e.g., (day) 2.8	None	Float
-f	Specifies the input spreadsheet to be used for the simulation	Master_Equipment_List.xlsx	filename with .xlsx extension
-g	Supresses automatically generated graphs. This will save processing time, particularly for very large simulations. To suppress graphs enter -g after SiteMain.py	active	
-m	Number of Monte Carlo runs; overrides "Number of Monte Carlo Iterations in "Global_Simulation_Parameters" in spreadsheet	None	Int
-o	Overrides Output file directory specified in "Output Directory" in Global_Simulation_Parameters sheet	None	

-p Argument	Number of processors for multiprocessing Description	1 Default	Int Type
-r	Selects a particular scenario for a drilldown Run: 0, 1, 2, etc.; used in conjunction with the timestamp contained in runCache.json output file	0	Int
-s	Start of serialization time, in original sim units; e.g., (day) 2.5	None	Float
-t	Simulation Timestamp; overrides date/time value stored in runCache.json; this value is used to locate the ScenarioInstanceConfig.json file required for Drilldown	None	
-x	Length of serialzier run, days	30	Int

## Validating model runs

Output files will be placed in the `GTI-CAMS/DES1/outputs` directory. Each simulation run will generate a timestamped folder with the simulation results. The timestamp will be of the format `YYMMDD_HHMMSS`.

Within the timestamped directory, there will be multiple folders with numerical names. Each numbered folder represents a single Monte Carlo iteration. These contain many files, some of which are primarily for internal system use. A number of automatically generated plots are saved as `.html` files. These can be viewed with any web browser. Numerical outputs of a subset of gas species are saved as `gas_composition.csv`. This can be viewed and edited with any program that can open `.csv` file extensions (e.g, Microsoft Excel, etc.). All emitters in the simulation will have an entry of zero at time `t=0`. Note that in order to minimize file size, emitters for which emissions were zero for a given time step are not logged after time `t=0`.