

# 1 Using get\_scenarios

To get started you would include the package as so:

```
import prescient_gosm.get_scenarios as get_scenarios
```

Whenever you want to create scenarios you would call the package's 'create\_scenarios' function. This function takes two dictionaries corresponding to the options for the scenario generation process as arguments. This is described more in the next section but for now know that Pop\_opts and SC\_opts are dictionaries.

```
get_scenarios.generate(Pop_opts, SC_opts)
```

This will generate the scenarios and place the relevant files in the output directory that was specified in the Pop\_opts dictionary. If you need the relative path to the output directory you can just assign the function to a value which will thus contain the path name.

```
path_to_output = get_scenarios.generate(Pop_opts, SC_opts)
```

## 2 Passing in Options

We will now describe the dictionaries you are supposed to pass in for the get\_scenarios functions. The dictionaries contain a key for the option you would like to alter in string format and a corresponding value. If the value is not numerical or boolean it must be in string format. You will pass in two of these dictionaries.

1. The first dictionary will correspond to the options passed into the populator options file. To learn more about every option available for the populator options refer to the gosm documentation. However, we will list the required options. Not including these options will result in an error, since these options are required.
  - **start-date & end-date:** These dates determine the period for which scenarios are created. Both must be in the format of YYYY-MM-DD-HH:MM.
  - **output-directory:** The path that stores the output.
  - **sources-file:** The sourcelist file.

```
Pop_opts = {'start_date': '2014-10-03-0:00', 'end_date': '2014-10-03-15:00', 'load_scaling_factor': 0.045,  
            'output_directory': 'multiple_sources_gosm_test/CAISO/caiso_output',  
            'sources_file': 'multiple_sources_gosm_test/CAISO/caiso_sourcelist.txt', 'allow_multiprocessing': 0,  
            'traceback': True}
```

Figure 1: Example of dictionary for populator options.

2. The second dictionary will correspond to options passed into the actual scenario generation process. Again, to learn about every option refer to the documentation. The required options for this dictionary are:

- **use-markov-chains:** Determines if the Markov Chain method is used for generating the scenarios. If this option is set, Markov Chains will be used. (This will be the default method in GOSM 3.0)
- **copula-random-walk:** Determines if copulas are used for the random walk. Can only be set, if use-markov-chains is also set. For multiple sources, this is a must.
- **planning-period-length:** The length for the planning or scenario period as an integer directly following by a capital letter, where "H" stands for hours and "T" stands for minutes.
- **scenario-template-file:** The scenario template file used for creating the scenarios. For more information look into the User Manual for GOSM 2.0.
- **tree-template-file:** The tree template file used for creating the scenarios. For more information look into the User Manual for GOSM 2.0.
- **reference-model-file:** The reference model which is used for creating the scenarios. For more information look into the User Manual for GOSM 2.0.

```
SC_opts = {'use_markov_chains': True, 'copula_random_walk': True, 'lead_time': '5H',  
          'scenario_template_file': 'multiple_sources_gosm_test/CAISO/caiso_simple_nostorage_skeleton.dat',  
          'tree_template_file': 'multiple_sources_gosm_test/TreeTemplate.dat',  
          'reference_model_file': '../models/knueven/ReferenceModel.py', 'number_scenarios': 1}
```

Figure 2: Example of dictionary for scenario creator options.