|  |  |
| --- | --- |
| **Future Meteorological Year Quick Start Guide** |  |

**August 23, 2019**

**Prepared by:**

Paul Kintner

Ecotope, Inc.

Contents

[FMY General Overview 3](#_Toc17710613)

[FMY Usage 4](#_Toc17710614)

[Interface 4](#_Toc17710615)

[Input Settings 4](#_Toc17710616)

[Output Settings 5](#_Toc17710617)

[Experimental Settings 6](#_Toc17710618)

[FMY Troubleshooting 8](#_Toc17710619)

[Warnings 8](#_Toc17710620)

[Fatal Errors 8](#_Toc17710621)

# FMY General Overview

### Motivation

The long running scientific consensus is that man-made emissions are the predominate cause of climate change. The warming trend has largely been ignored in the buildings industry, which typically models with weather data uses the typical meteorological year 3 (TMY3). TMY3 is hourly weather taken from weather stations around the United States, representing a typical year over some period. The TMY3 datasets were originally published with data through 2005. The Northwest Power and Conservation Council (NPCC) has recognized that TMY3 files are outdated and wishes to incorporate predictions of future climate conditions in to building energy use simulations.

Here we describe plans to update simulation weather file inputs to reflect those future climates. This will result in weather files with differing input variables like temperature and relative humidity which will lead to different predictions of building energy use. The building simulations are conducted in Simplified Energy Enthalpy Model (SEEM), written by Ecotope, which use TMY3 weather data. Data to update those weather files will come from the [Multivariate Adaptive Constructed Analogs Datasets](https://climate.northwestknowledge.net/MACA/MACAproducts.php) (MACA).

The Future Meteorological Years (FMY) script was designed to be more flexible than other similar future weather generation programs. Where WeatherShift and CCWeatherGen give output for a collection of GCMs, FMY will return hourly weather any for all GCMs available through MACA. The user is free to aggregate GCM output or use a single GCM for hourly weather. Weather-Shift is also available only in 30 US cities, where as FMY will use any TMY2 or TMY3 file as input. FMY is limited in the number of variables adjusted, other programs cover what FMY covers and adjust at least atmospheric pressure and wind speed.

# FMY Usage

### Interface

The FMY script, written in Python 3.7, is designed to read TMY2 or TMY3 files then write to the same file structure with new weather variables. The script is run using main.py, the easiest way to run it would be in Sypder (as part of Anaconda) or another Python Development Environment.

Using the FMY script is simple with Anaconda installed. The basic packages in Anaconda, or another Python editor should be enough to get FMY running. The necessary packages should be installed automatically during the first run of FMY.py if not. The packages needed are:

* pip
* numpy
* pandas
* matplotlib
* seaborn
* json
* datetime
* time
* metpy
* netCDF3

To run the program needs available TMY2 or TMY3 files. The input directory to the TMY2 or TMY3 weather files is defined in weatherpath. The assumption in cfg.py is that the file names for TMY2 follow the format state abbreviation, city, 3, for example WASeattle3 or IDBoise3. And for TMY3 the file name is the city code number, for example 727930 or 726810, for Seattle and Boise respectively. The extensions are handled by the software, TMY2 is .tm2 and TMY3 is .csv.

The program will first check locally for GCM data in datapath, since the download data from the MACA servers may take an excessive amount of time if running the program repeatedly and if the user is asking for a large amount of data from the MACA servers they make get booted. So the program can be set to download data for the specified cities and variables locally, using the Boolean flag download\_data. This condenses the data to just a city and variable, on the order of 10s of MB, instead of the entire CONUS. The data for all the cities in this project is about 1 GB. If there is no local data or the local dataset is incomplete for the desired run the script will use OPenDAP to access the MACA server and download a large region of GCM in the Northwest US and then save just the data for the city if download\_data is set to true.

### Input Settings

All the basic settings for creating FMY files are available in main.py, but running it will take some basic set up of directories for one’s person computer. The directories for data and graphical outputs need not necessarily exist on the computer, if they do not, they will be created. The directories are as follows:

* workingdir – The working directory, this is where the .py files are held locally.
* weatherpath – The location of TMY files on the user’s computer.

Table 1. FMY Script Settings in main.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable List Name in main.py:** | stations | models | variables | scenarios |
| **List Number** | **Location** | **GCM** | **Weather Variable in MACA** | **Scenario** |
| 0 | Seattle (WA) | bcc-csm1-1 | Max Temperature | Historical |
| 1 | Corvallis (OR) | bcc-csm1-1-m | Min Temperature | RCP4.5 |
| 2 | Boise (ID) | BNU-ESM | Max Relative Humidity | RCP8.5 |
| 3 | Redmond(OR) | CanESM2 | Min Relative Humidity |  |
| 4 | Elko (NV) | CCSM4 | Precipitation (Not FMY Supported) |  |
| 5 | Burley (ID) | CNRM-CM5 | Surface Downwelling Shortwave Flux in Air |  |
| 6 | Soda\_Springs (ID) | CSIRO-Mk3-6-0 | Eastward Wind Component (Not FMY Supported) |  |
| 7 | Havre (MT) | GFDL-ESM2G | Northward Wind Component (Not FMY Supported) |  |
| 8 | Miles City (MT) | GFDL-ESM2M | Specific Humidity |  |
| 9 | Portland (OR) | HadGEM2-CC365 |  |  |
| 10 | Spokane (WA) | HadGEM2-ES365 |  |  |
| 11 | Kalispell (MT) | inmcm4 |  |  |
| 12 |  | IPSL-CM5A-MR |  |  |
| 13 |  | IPSL-CM5A-LR |  |  |
| 14 |  | IPSL-CM5B-LR |  |  |
| 15 |  | MIROC5 |  |  |
| 16 |  | MIROC-ESM |  |  |
| 17 |  | MIROC-ESM-CHEM |  |  |
| 18 |  | MRI-CGCM3 |  |  |
| 19 |  | NorESM1-M |  |  |

* datapath – This is the directory on the computer that GCM files for the cities will be downloaded into and searched for later. Note: here a user should specify if the weather files to be loaded are in TMY2 or TMY3 format by setting load\_tmy23 to either 2 or 3.
* graphpath – The directory that graphs will be printed to, in the event they are asked for.
* outputpath – The output path for the FMY files, or just generic csv output.

At this point main.py is ready to run. However, a user might be more interested in looking at a specific few cities for a couple GCMs. This is easily set within main.py, the script can access any model, scenario, and variable combination hosted on MACA for any TMY2 or TMY3 city in the continuous US. Although precipitation and wind are not converted to FMY. Firstly, the stations list of integers will specify which city to look at from the list below, any combination of cities can be added to the list, just include the corresponding number for that city. The script will only run for the nine TMY3 cities listed in the NPCC project proposal, they are given in Table 1. But setting up new cities is a simple task. The filename for the city would have to be added to the end of the CITY list in cfg.py and then the next incremental number in the stations list will refer to that city.

Second, the GCMs available from MACA are listed in Table 1. Any combination of models can be added to the models list. But note bcc-csm1-1 (0) does not have daily data available for 12/31/2099 for the RCP8.5 scenario only. And CCSM4 (4) and NorESM1-M (19) do not have relative humidity available at daily timescales.

The representative climate pathway (RCP) scenario the GCM was run with is set under scenario, where 1 is RCP4.5, 2 is RCP8.5. Scenario number 0 is historical part of the GCM and is set to automatically load.

The time frame is set in tmy3\_years, the historical period, and future\_years, the future period. These can be changed, but it is recommended they stay set to 1976 – 2005 and 2020 – 2049 respectively. The future period for the GCMs is loaded between 2006 and 2099. Here we look at the 2030’s, 2020 – 2049, but if it is desired to look farther ahead the time period can be set here. It is standard practice to use a 30-year period to judge an average climate.

Any or all the weather variables can be run but must be specified in the variables list. The variables provided by MACA and their references are listed in Table 1. You cannot run a max or min variable without running the other. If running one temperature variable (0/1) the other variable (1/0) must be run as well. This allows the user to create an FMY file that only has the temperature adjusted for example, or all the supported variables. Note that running with variable 5, the total solar radiation, in the variables list will change all the solar variables, the total, direct, and diffuse, in the FMY.

Running variables with precipitation (4) or a wind component (6/7) is not supported for FMY files, but the program will download the relevant time series for these variables at the specific stations, scenarios, and GCM’s defined by the user.

### Output Settings

The FMY script will write output files in either a CSV file, TMY2, and/or TMY3 formats. This is specified with strings in the variable outformats. If writing to a TMY file the script will read the initial TMY file and rewrite the TMY file with the new FMY variables that were specified by the variables list. There is no file conversion between TMY2 and TMY3, so the script will only write to the TMY file that is specified as input. Note that the CSV file is a smaller file which contains just the changed variables before and after the FMY transformations, and the TMY3 format will write to a CSV file contains all the TMY3 information.

The variable hourly\_plots is a list of variables that print out in plots for each month if suppress\_all\_plots is set to False; the plots are written out to graphpath. If suppress\_all\_plots is set to True it prints hourly plots of FMY and TMY weather variables for every month, and monthly averages of variables for the GCM and TMY datasets. Printing all these plots will slow the program down.

### Experimental Settings

These settings are available to be played with by the user but are not recommended to be changed.

The which\_current\_climate changes what the script uses as the baseline climate for adjustment. Current settings are for ‘gcm’/’GCM’ or ‘tmy’/’TMY’. ‘gcm’ is recommended as this uses a thirty year mean for the baseline and does not bias the data the way the TMY data does. The TMY data does not represent a city’s current climate and is representative of only one typical year, making the data more susceptible to outliers, which would shift the results into extremes.

The interpolate\_to\_station input, set to ‘True’ or ‘False’, will interpolate the GCM 4-km grid cell to the station. It is recommended to be left on, as this is the accepted practice in the literature.

The bias\_correction\_method input, is used to either not correct the data (0), use an equidistance cumulative distribution function (CDF) matching (EDCDFm) (1), which is the method used in MACA. Which is different from the typical quantile-based mapping method (CDFm) (2) as it allows for changes in the variance and skew, as well as the mean, in a variable’s distribution.

#### Method Settings

The method input changes what method is used, the NPCC (1), Belcher (2) or an adjusted version of NPCC (3),

The predominate difference between the NPCC method and the adjusted version, is that the NPCC method uses the max and min daily temperatures and relative humidity of each month to adjust these variables to FMY, instead of using the monthly average of the max/min temperatures and relative humidity. Because the NPCC method works on using the max/min of the whole month daily data is needed. MACA provides this data but it also increases run time as there is an order of magnitude more data to download from MACA. Both the NPCC methods only change the adjustment to the dry bulb temperatures and relative humidity, the method follows the Belcher method of morphing for all the other variables.

For details on the Belcher method of morphing, is the default method in the script and the accepted method in the literature. For details on the morphing method refer to the FMY report.

The NPCC method starts by looking at the monthly max and min over the future time period, for each variable, , then for each month chooses the month that is closest to the mean of all the future months. The choice is made by finding the model that minimizes:

Where is the mean for month, m, of all years, y, for a variable, . This is the normalized vector distance to the mean for all the variables. Once the future month closest to the average is found the temperature is found by solving for constants A and B below:

And then using the same equation to move to hourly data:

A similar transformation for the relative humidity with constants, C and D, i.e.

# FMY Troubleshooting

### Warnings

If variables are out of bounds physical boundaries, i.e. relative humidity > 100%, the program warns the user, but the script goes on. In the TMY2 format, there are bounds imposed on all the variables which keeps the relative humidity <= 100%. However, the physical bounds on the dew point temperature are imposed by the dry bulb and is thus missed by the limitations of the TMY2 boundaries. The same set of limitations occurs with the diffuse horizontal radiation and the global horizontal radiation. If these warning arises it is likely that the unphysical situation occurs in the FMY output. For the most part this seems to arise because of an issue with the input TMY data. Note that for Redmond (OR) the TMY3 observational data has the diffuse horizontal solar greater than global horizontal solar for 23 hours out of the year.

### Fatal Errors

If looking at temperature and relative humidity both the max and min variables must be included in the variable list or else the program will throw an error if the user selects only one of these variables for morphing.

A current climate for the baseline must be specified as ‘tmy’/’TMY’ or ‘gcm’/’GCM’. This is in experimental section and is recommended to be left untouched, but if a user is interested, they can use either one. This is true of the method input as well.

Handling packages can be difficult. If there are errors arising from packages not being installed correctly the easiest course of action may be to close the terminal or session using the packages and then to uninstall and reinstall the packages using pip, alternatively the FMY script should reinstall the necessary packages. In a terminal with python in the PATH (use the Anaconda prompt if unsure) for the python package ‘pandas’ this would look like:

>>python -m pip uninstall pandas

>>python -m pip install pandas