#### This document outlines the primary components that make up the NYSolarCast Forecasting system. For questions, contact Jared Lee ([jaredlee@ucar.edu](mailto:jaredlee@ucar.edu)) or Tom Brummet ([brummet@ucar.edu](mailto:brummet@ucar.edu)).

#### **Misc. Notes**

* All of the below scripts contain thorough documentation within the scripts themselves
* Below references to ‘sys\_path’ are really referring to the ‘NYSC\_sys\_path.py’ file which is imported in most scripts as ‘sys\_path’
* Log output from the below scripts is written into the sys\_path.Log\_dir area. This log output is critical to assessing what the scripts do and if they ran properly

#### **Observation processing**

* **Script : run\_download\_nymeso\_data.py**
  + This script is responsible for downloading the latest observation data from the New York State Mesonet (NYSM).
  + Execution Frequency - 5 min
  + Output is produced in csv format under the location specified by the ‘output\_dir’ variable --
    - [sys\_path.DATA\_base\_dir]/phase3/NYMesonet\_real\_time/ascii
  + In order to get access to the data, one will need to be given access. Visit https://nysmesonet.org/about/data for information on how to request access to NYSM data.
* **Script : run\_nymeso\_basic\_and\_shading\_qc.py**
  + This script is responsible for performing basic quality control checks on the observations. A ‘shading’ check is also performed to remove data at times that have known issues related to shading effects.
  + Execution Frequency - 5 min
  + Information related to sites and times that have shading issues is required. Shading files can be found under the ‘shading/’ folder.
  + Output is produced in csv format under the location specified by the ‘nymeso\_qc\_dir’ variable --
    - [sys\_path.DATA\_base\_dir]/phase3/NYMesonet\_real\_time/qc
* **Script : run\_nymeso\_15min\_avg.py**
  + This script is responsible for creating 15-minute averages of the QC’d mesonet data. The code uses configuration information specified in ‘static/config/nymeso\_15min\_avg.json’ which tells the code how to create the averages.
  + Execution Frequency - 15 min
  + Output is produced in csv format under the ‘nymeso\_avg\_dir’ variable --
    - [sys\_path.DATA\_base\_dir]/phase3/NYMesonet\_real\_time/15min
* **Script : run\_nymeso\_csv\_to\_nc.py**
  + This script is responsible for converting the 15-min averaged CSV data to a NetCDF (NC) format. The NC formatted data is used as input to the statcast forecasting engine.
  + Execution Frequency - 15 min
  + Output is produced under the ‘nymeso\_nc\_dir’ variable --
    - [sys\_path.DATA\_base\_dir]/phase3/NYMesonet\_real\_time/15min\_nc

#### **Forecast Processing**

* **Script : run\_hrrr\_grib2nc.py**
  + This script is responsible for converting the HRRR data from grib format to NetCDF (NC)
  + Execution Frequency - Hourly
  + The location of the input HRRR files is specified by the ‘hrrr\_fcst\_base’ variable.
    - HRRR files should be in dated subdirs matching [YYYY-MM-DD\_HH]
  + The output NC files are written under the ‘output\_dir’ variable --
    - [sys\_path.Data\_base\_dir]/hrrr\_nc
* **Script : run\_WrfNetCDF2StatcastNetCDF.py**
  + This script is responsible for reformatting the WRF NC data into a new NC file that can be ingested into the StatCast engine.
  + Execution Frequency - Hourly
  + The location of the input WRF files is specified by the ‘wrf\_fcst\_base’ and ‘wrf\_day\_ahead\_fcst\_base’ variables
  + The output is written under the ‘output\_dir’ variable --
    - [sys\_path.Data\_base\_dir]/wrf\_fcst\_4\_statcast
* **Script : run\_NWP\_grid\_blender.py**
  + This script is responsible for blending the WRF and HRRR datasets. Static files (‘static/config/WRF\_HRRR\_blend.json’ & ‘static/config/WRF\_HRRR\_blend\_dayAhead.json’) control the weight that is given to each model at each lead time.
    - Currently, an even (50/50) blend is used
  + Execution Frequency - Hourly
  + Output is written under the ‘output\_dir’ variables
    - [sys\_path.Data\_base\_dir]/nwp\_blend
* **Script : run\_ghi\_fcst.py**
  + This script is responsible for running the StatCast forecast engine, ultimately calling the ‘apps/ghi\_fcst’ application. StatCast leverages machine learning to produce accurate GHI forecasts at the mesonet sites.
  + Execution Frequency - 15 min
  + Output is written under --
    - [sys\_path.Statcast\_fcst\_dir]/netcdf
* **Script : run\_NWP\_Statcast\_blender.py**
  + This script is responsible for spatially and temporally blending the point-based statcast GHI forecast into the grid-based NWP blended GHI forecast (output from ‘run\_NWP\_grid\_blender.py’).
    - StatCast models are trained and tuned for each lead time (15 minutes out to 6 hours)
  + Execution Frequency - 15 min
  + Output is written under the ‘output\_dir’ and ‘output\_hourly\_dir’ variables --
    - [sys\_path.Data\_base\_dir]/nwp\_statcast\_blend
    - [sys\_path.Data\_base\_dir]/nwp\_statcast\_blend\_hourly
* **Script : run\_power\_fcst.py**
  + This script is responsible for running the power conversion models to convert the GHI output (from ‘run\_NWP\_statcast\_blender.py’) into power.
    - GHI to Power models are trained and tuned for each specific farm to maximize performance.
    - A separate model was trained to produce the distributed power forecast
  + Execution Frequency - 15 min for individual farms, hourly for distributed power.
  + Output is written under --
    - [sys\_path.Statcast\_fcst\_dir]/pct\_power\_fcst\_month/netcdf
* **Script : run\_pct\_power\_rollup.py**
  + This script is responsible for aggregating the distributed percent power forecasts into a total power forecast based on regional zip codes.
  + Execution Frequency - Hourly
  + Output is written under the ‘output\_dir’ variable --
    - [sys\_path.Total\_fcst\_dir]
* **Script : run\_pct\_power\_region\_rollup.py**
  + This script is responsible for aggregating the distributed percent power forecasts into a total power forecast based on regions. Each ZIP code is assigned to a region, based on information in the ‘static/site\_list/zip\_match\_region.csv’ file.
  + Execution Frequency - Hourly
  + Output is written under the ‘output\_dir’ variable --
    - [sys\_path.Total\_fcst\_dir]\_region
* **Script: run\_pct\_power\_to\_total\_power.py**
  + This script is responsible for converting the percent power forecasts at the individual farms into a total power forecast for each farm.
    - The code requires farm capacity information, which is specified in the ‘static/site\_list/farm\_capacity\_list.csv’ file
  + Execution Frequency - 15 min
  + Output is written under the ‘output\_dir’ variable --
    - [sys\_path.Total\_fcst\_dir]

#### **System Monitoring**

* **Script : run\_auto\_file\_watcher.py**
  + This script is responsible ensuring that all expected files are created on time
  + This script will send emails if it notices a problem
* **Script : run\_auto\_file\_watcher\_day\_ahead.py**
  + This script is responsible for ensuring that all expected day-ahead files are created on time
  + This script will send emails if it notices a problem