# LightingBaselineAgent – A Lighting Baseline Volttron Agent

## Purpose

This agent serves a wrapper for the lighting\_baseline module for integration with the Volttron system. It defines an agent class (LightingBaselineAgent) that acts as the single point of interaction between the lighting\_baseline module and the Volttron message bus.

This program defines two types of calculations that can be requested: summary and comparison. It provides all Volttron-related logic for processing requests for these calculations and posting responses.

The standard use case is to create one LightingBaselineAgent that will listen to all requests for lighting baseline analysis. When a message is received on one of the two analysis request topics the agent will perform the requested analysis and post the response to the corresponding response topic.

As a wrapper this document only contains some information about the wrapped lighting\_baseline module. This document should be enough to work with the agents but for more details about the analysis being performed consult the documentation for the lighting\_baseline module itself.

## System Requirements and Dependencies

Requires Python 2 (tested on 2.7.5)

Requires the Volttron environment: https://github.com/VOLTTRON/volttron

Requires the lighting\_baseline module: Currently this is located at <https://bitbucket.org/berkeleylab/eetd-tn-lighting/src> under the Modules folder but it is planned to be hosted in PyPi under the name lighting\_baseline.

The agent code itself introduces no additional dependencies outside of those required to run the Volttron system or the lighting\_baseline module.

## Additional Considerations and Concepts

This program acts as a wrapper around the lighting\_baseline module and defines an interface for making requests for lighting baseline analysis. It provides all Volttron-related logic for processing requests for these calculations and posting responses.

**Timeseries**: Most of the data that this application uses is in the form of a timeseries. A timeseries is a list of (timestamp, value) pairs.

## Installation Instructions

The lighting\_baseline module must be visible to the wrapper. Volttron must be installed and running. See <https://github.com/VOLTTRON/volttron/wiki> for instructions on configuring/running Volttron and general agent usage.

If this package is installed with the rest of the Volttron package it should automatically download and install any dependencies.

If the module needs to be installed manually first download and extract the package. Open a terminal and navigate to the root of the extracted archive. There should be a file called setup.py. Run “python setup.py install”. As long as there is a connection to the internet python should install the package and all dependencies.

## Execution Instructions

See <https://github.com/VOLTTRON/volttron/wiki> for instructions running Volttron agents.

Topics this agent subscribes to:

* “lighting\_baseline/summary\_request”
* “lighting\_baseline/comparison\_request”

These correspond to the summary and comparison requests mentioned above.

Summary\_request:

* Header fields:
  + Required:
    - **requestorID**: The ID of the requesting agent. Needed to craft the response topic
  + Optional:
    - **From**: Will be used to set the “To” field in the response headers if present
* Message fields:
  + Required:
    - **load\_data**: A timeseries of the load values to be evaluated
  + Optional:
    - **timezone**: The timezone of the values in load\_data. If none is present then the OS timezone will be used.
    - **sq\_ft**: The square-footage of the space being evaluated.
    - **workdays**: A string representing which of the days of the week are to be considered workdays. Days are encoded as 0-6 where 0 is Sunday. Default is a workweek of Mon-Fri which is represented as “12345”
    - **workday\_start**: A string representing the start of a 12-hour workday. Default is “6:30”. String format should be “%H:%M”
* Response Topic: “lighting\_baseline/summary\_response/{requestor\_ID}” where requestor\_ID is the value in the initial request header’s “requestorID” field. For example, if the requestorID value was “CustomBuildingManagementAgent” the response topic would be “lighting\_baseline/summary\_response/CustomBuildingManagementAgent”
* Response Headers:
  + **To**: The value in the “From” header field from the initial request if it exists, “Unknown” otherwise.
  + **From**: This will always be “lightingAnalysisAgent”
* Response Message: A dictionary containing the results of the lighting\_baseline module calculations. Current results are listed as “summary\_stats” and they are: "nonworkdaynightmean", "holidaynightmean", "nonworkdaydaymean", "nonholidayweekmean", "workdaynightmean", "workdaydaymean", and "holidaydaymean". Consult the lighting\_baseline module documentation for additional information on these results.

Comparison\_request:

* Header fields:
  + Required:
    - **requestorID**: The ID of the requesting agent. Needed to craft the response topic
  + Optional:
    - **From**: Will be used to set the “To” field in the response headers if present
* Message fields:
  + Required:
    - **baseline\_load\_data**: A timeseries of the load values representing the baseline to which the comparison will be made. Any results will be with respect to this baseline data. I.E. a positive value indicates that the comparison value is greater than the baseline and a negative value indicates that the comparison period is less than the baseline.
    - **comparison\_load\_data**: A timeseries of the load values representing the period to be compared to the baseline.
  + Optional:
    - **timezone**: The timezone of the values in load\_data. Must be a string that is parsable by pytz.timezone (http://pytz.sourceforge.net/). If none is present then the OS timezone will be used.
    - **sq\_ft**: The square-footage of the space being evaluated.
    - **workdays**: Which of the days of the week are to be considered workdays. Default is a workweek of Mon-Fri which is represented as “12345”
    - **workday\_start** (default “6:30): The start of a 12-hour workday.
* Response Topic: “lighting\_baseline/comparison\_response/{requestor\_ID}” where requestor\_ID is the value in the initial request’s “requestorID” header field. For example, if the requestorID value was “CustomBuildingManagementAgent” the response topic would be “lighting\_baseline/summary\_response/CustomBuildingManagementAgent”
* Response Headers:
  + **To**: The value in the “From” header field from the initial request if it exists, “Unknown” otherwise.
  + **From**: This will always be “lightingAnalysisAgent”
* Response Message: A dictionary containing the results of the lighting\_baseline module calculations. Current results are listed as “comparison\_stats” and they are: "nonworkdaynightmean", "holidaynightmean", "nonworkdaydaymean", "nonholidayweekmean", "workdaynightmean", "workdaydaymean", and "holidaydaymean". These numbers are the change from the baseline state. For example, if the workdaydaymean is negative that means that there was less usage in the comparison period during workday days. If it were positive that means that there was less usage in the baseline period during workday days. Consult the lighting\_baseline module documentation for additional information on these results.

### Input Formats

**Load data timeseries**

A timeseries of load values. load\_data, baseline\_load\_data, and comparison\_load data are all examples of fields that are load data timeseries. The timeseries is a list of (timestamp, value) pairs where value is the load value and timestamp may be in any of the following formats:

1. Seconds since 1970-01-01 00:00:00
2. Milliseconds since 1970-01-01 00:00:00
3. Year-month-day hours:minutes:seconds, e.g. 2014-08-28 16:30:00

The program automatically determines what time format is being used: if the first timestamp in the file includes a “:” then the program assumes format 3 is being used. If it does not, then numbers exceeding 3 x 109 are assumed to be in milliseconds, otherwise they are assumed to be in seconds. Timestamps do not need to be at any particular frequency and do not need to fall on round numbers of minutes or hours. They are assumed to be regularly spaced in time; if they are not, no error is generated but the program’s results may be incorrect.

**timezone**: A string that can be parsed by pytz.timezone (<http://pytz.sourceforge.net/>).

**sq\_ft**: Convertible to a float or None.

**workdays**: String of at most 7 characters. Days are encoded as 0-6 where 0 is Sunday. Default is a workweek of Mon-Fri which is represented as “12345”

**workday\_start**: A string of the form hour:minute. In strftime notation “%H:%M”

## Output

The agent responds to any output by posting a message to the output topic that corresponds to the topic the request was made on. For a summary request the response will be posted to “lighting\_baseline/summary\_response/{requestor\_ID}” while for a comparison request that will be “lighting\_baseline/comparison\_response/{requestor\_ID}”

If there was a problem with the request the agent will respond with a message dictionary that has one entry called “error” where the contents are a description of the error.

If there was no problem with agent will post the results of the analysis. Currently the available results are: "nonworkdaynightmean", "holidaynightmean", "nonworkdaydaymean", "nonholidayweekmean", "workdaynightmean", "workdaydaymean", and "holidaydaymean”. For information about what exactly these numbers mean please consult the lighting\_baseline module documentation.

### Output Formats

The values in the result fields will either be convertible to a float or “Nan”.