ESPy

Contents

Chapter 1

espy

Python API for ESP-r

Please note that these modules are primarily designed for my own use, and functions may change inputs or names without warning or concern for maintaining compatibility with any scripts you may have.

Install procedure

```
git clone https://github.com/johnallison0/espy.git
cd espy
python setup.py install
```

Structure

ESPy is broken up into modules. Many of these echo names of ESP-r modules (e.g. bps, res, clm), that contain functions to automate functionality of these modules. Other modules contain various support facilities, as well as functions for interacting with ESP-r models without using the ESP-r interface.

For full namespace documentation, refer to the ./doc directory.

Usage

If the installation procedure above was followed, then ESPy modules should be included in Python code in the same manner as any other installed modules. A minimal workflow for a typical simulation and results extraction task might be the following:

```
import espy.bps as bps
from espy.res import time_series

bps.run_preset('./cfg/model.cfg','annual')
time_series('./cfg/model.cfg','annual.res',[['all', 'Zone db T']],'res.csv')
```

You would then have dry bulb temperature for all zones, in file res.csv. Alternatively, you could work with a Data↩ Frame of the results:

```
import pandas as pd
from espy.res import time_series

res_df = time_series('./cfg/model.cfg','annual.res',[['all', 'Zone db T']])
res_df['Zone1_dbT'].to_csv('Zone1res.csv')
```

There are many other functions provided by ESPy. For full documentation, refer to the ./doc directory.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Namespace Documentation

4.1 espy.bps Namespace Reference

Classes

- class Bps

Functions

- def run_preset (cfg_file, preset)
- def run_sim (cfg_file, res_file, sim_start_d, sim_start_m, sim_end_d, sim_end_m, start_up_d, tsph, integrate)

4.1.1 Detailed Description

```
Functions to interact with bps.
```

4.1.2 Function Documentation

4.1.2.1 run_preset()

```
def espy.bps.run_preset (
            cfg_file,
            preset )
```

```
Run simulation with preset.
```

4.1.2.2 run_sim()

```
def espy.bps.run_sim (
            cfg_file,
            res_file,
            sim_start_d,
            sim_start_m,
            sim_end_d,
            sim_end_m,
            start_up_d,
            tsph,
            integrate )
```

```
Run basic simulation.
```

4.2 espy.clm Namespace Reference

Functions

- def get_avg_degree_days (weather_file, temp_base=15.5)

4.2.1 Detailed Description

```
Functions to interact with clm.
```

4.2.2 Function Documentation

4.2.2.1 get_avg_degree_days()

```
def espy.clm.get_avg_degree_days (
            weather_file,
            temp_base = 15.5 )
```

```
Returns the daily average degree days
```

4.3 espy.convert Namespace Reference

Functions

- def zone_to_predef_entity (geo_file, name, desc, category)
- def epw_to_espr (epw_file, espr_file="newclim")
- def weather_bin_to_ascii (bin_file, ascii_file="newclim.a")

4.3.1 Detailed Description

```
Functions to convert between various file formats.
```

4.3.2 Function Documentation

4.3.2.1 epw_to_espr()

```
def espy.convert.epw_to_espr (
             epw_file,
             espr_file = "newclim" )
```

```
Convert EPW file to ESP-r binary weather file.
```

4.3.2.2 weather_bin_to_ascii()

```
def espy.convert.weather_bin_to_ascii (
             bin_file,
             ascii_file = "newclim.a" )
```

```
Convert ESP-r binary weather file to ascii file.
```

4.3.2.3 zone_to_predef_entity()

```
def espy.convert.zone_to_predef_entity (
             geo_file,
             name,
             desc,
             category )
```

```
Convert a zone geometry file to a predefined entity entry.

Args:
    geo_file: ESP-r geometry file.

Returns:
    A text file that can be copied into an ESP-r predefined entities
    database.
```

4.4 espy.edit Namespace Reference

Functions

- def door_usage (geo_file, original, updated)
- def window_usage (geo_file, original, updated)
- def frame_usage (geo_file, original, updated)

4.4.1 Detailed Description

```
Functions that directory edit ESP-r files.
```

4.4.2 Function Documentation

4.4.2.1 door_usage()

```
def espy.edit.door_usage (
            geo_file,
            original,
            updated )
```

```
Directly edit door usage in geometry file.
```

4.4.2.2 frame_usage()

```
def espy.edit.frame_usage (
            geo_file,
            original,
            updated )
```

```
Directly edit frame usage in geometry file.
```

4.4.2.3 window_usage()

```
def espy.edit.window_usage (
            geo_file,
            original,
            updated )
```

```
Directly edit window usage in geometry file.
```

4.5 espy.get Namespace Reference

Classes

- class Component

Functions

- def zone_selection (cfg_file, zone_input)
- def surface_selection (geo_file, surf_input)
- def vtk_view (actors, edge_actors, outlines)
- def generate_vtk_actors (surf_obj, outer_colour, show_edges=False, show_outline=True)
- def calculate_normal (p)
- def area (poly)
- def config (filepath)
- def geometry (filepath)
- def constructions (con_file, geo_file)
- def controls (filepath)
- def pos_from_vert_num_list (vertices_zone, edges)
- def weather (file_path)
- def weather_v2 (file_path)

4.5.1 Detailed Description

```
Functions for importing and reading ESP-r files
```

4.5.2 Function Documentation

4.5.2.1 area()

```
def espy.get.area (
            poly )
```

```
area of polygon poly
Source: https://stackoverflow.com/a/12643315
Source 2: http://geomalgorithms.com/a01-_area.html#3D%20Polygons
TODO(j.allison): this function should probably live in a different module
```

4.5.2.2 calculate_normal()

```
def espy.get.calculate_normal (
            p )
```

```
Newell's method for calculating the normal of an arbitrary 3D polygon.
```

4.5.2.3 config()

```
def espy.get.config (
            filepath )
```

Reads in an ESP-r configuration file.

4.5.2.4 constructions()

```
def espy.get.constructions (
            con_file,
            geo_file )
```

Get data from construction file.

4.5.2.5 controls()

```
def espy.get.controls (
            filepath )
```

Import model controls.

4.5.2.6 generate_vtk_actors()

```
def espy.get.generate_vtk_actors (
            surf_obj,
            outer_colour,
            show_edges = False,
            show_outline = True )
```

Generates 3 VTK actors.

Returns 3 VTK actors, which represents an object (geometry & properties) in a rendered scene

```
Args:
    surf_obj (vtkObject): vtk Object that defines the surface
    outer_colour (list): Colour and opacity of surface i.e. ["#f5f2d0", 1]

Returns:
    surface_actor (vtkOpenGLActor): 2D component surface projected on 3D plane
    edge_actor (vtkOpenGLActor): Mesh of surface
    outline_actor (vtkOpenGLActor): Boundary outline of surface
```

4.5.2.7 geometry()

```
def espy.get.geometry (
            filepath )
```

Reads in an ESP-r geometry file.

Returns the name and description of the zone.

Returns the last modified date.

Returns a list of the vertices, where each element is a list of floats
specifying the x, y, z coordinate in space.

Returns a list of the surface edges, where each element is a list of ints
specifying the vertex numbers that make up the surface.
Note that these are referenced as 1-indexed.

Returns a list of the surface attributes, where each element is:
['surf name', 'surf position', 'child of (surface name)',
'useage1', 'useage2', 'construction name', 'optical name',
'boundary condition', 'dat1', 'dat2']

4.5.2.8 pos_from_vert_num_list()

```
def espy.get.pos_from_vert_num_list (
            vertices_zone,
            edges )
```

Get x,y,z position of vertices that comprise a surface
from the zone vertices and their indices as defined in
the edges list

4.5.2.9 surface_selection()

```
def espy.get.surface_selection (
            geo_file,
            surf_input )
```

Maps requested surface selection to ESP-r menu selection.

4.5.2.10 vtk_view()

```
def espy.get.vtk_view (
            actors,
            edge_actors,
            outlines )
```

VTK visualisation setup and render

4.5.2.11 weather()

```
def espy.get.weather (
            file_path )
```

Read ESP-r ascii weather file.

```
col 1: Diffuse solar on the horizontal (W/m^2)
col 2: External dry bulb temperature   (Tenths °C)
col 3: Direct normal solar intensity   (W/m^2)
col 4: Prevailing wind speed           (Tenths m/s)
col 5: Wind direction                  (clockwise ° from north)
col 6: Relative humidity               (%)
```

4.5.2.12 weather_v2()

```
def espy.get.weather_v2 (
            file_path )
```

Read ESP-r ascii weather file.

4.5.2.13 zone_selection()

```
def espy.get.zone_selection (
            cfg_file,
            zone_input )
```

Maps requested zone selection to ESP-r menu selection.

4.6 espy.prj Namespace Reference

Functions

- def edit_material_prop (cfg_file, change_list)
- def edit_layer_thickness (cfg_file, change_list)
- def gen_qa_report (cfg_file, filename)
- def rebuild_con_files (cfg_file)
- def add_door (cfg_file, door_name, zone_surf1, zone_surf2, x_off, size)
- def add_window (cfg_file, zone, surf, location, size, sill=None, reveal=None)
- def add_zone (cfg_file, name, vertices, description=None, z_base=0, z_top=2.7, rot_angle=0)

4.6.1 Detailed Description

Functions to interact with prj.

4.6.2 Function Documentation

4.6.2.1 add_door()

```
def espy.prj.add_door (
            cfg_file,
            door_name,
            zone_surf1,
            zone_surf2,
            x_off,
            size )
```

Adds door between two zones.

4.6.2.2 add_window()

```
def espy.prj.add_window (
            cfg_file,
            zone,
            surf,
            location,
            size,
            sill = None,
            reveal = None )
```

Adds window to a surface in a zone.

4.6.2.3 add_zone()

```
def espy.prj.add_zone (
            cfg_file,
            name,
            vertices,
            description = None,
            z_base = 0,
            z_top = 2.7,
            rot_angle = 0 )
```

Adds new zone to model.

4.6.2.4 edit_layer_thickness()

```
def espy.prj.edit_layer_thickness (
            cfg_file,
            change_list )
```

Edit layer thickness of multi-layered construction.
This function will build the command list to edit the layer thickness in
the MLC db via prj.

4.6.2.5 edit_material_prop()

```
def espy.prj.edit_material_prop (
            cfg_file,
            change_list )
```

Edit material properties.
This function will build the command list to edit material properties in
the materials db via prj.

4.6.2.6 gen_qa_report()

```
def espy.prj.gen_qa_report (
            cfg_file,
            filename )
```

Generate model QA report.

4.6.2.7 rebuild_con_files()

```
def espy.prj.rebuild_con_files (
            cfg_file )
```

Updates the zone construction files.

4.7 espy.res Namespace Reference

Functions

- def calc_airtightness (res_file, mfr_file, volume, zones)
- def air_supply (res_file, mfr_file, zones)
- def time_series (cfg_file, res_file, param_list, out_file=None, time_fmt='DateTime')
- def abovebelow (cfg_file, res_file, is_below=False, out_file=None, query_point=25)
- def energy_balance (cfg_file, res_file, out_file=None, group=None)
- def get_pv (res_file, elr_file, out_file=None)

4.7.1 Detailed Description

```
Module to automate retrieval of data from res.
```

4.7.2 Function Documentation

4.7.2.1 abovebelow()

```
def espy.res.abovebelow (
            cfg_file,
            res_file,
            is_below = False,
            out_file = None,
            query_point = 25 )
```

```
Get hours above or below a value.
```

4.7.2.2 air_supply()

```
def espy.res.air_supply (
            res_file,
            mfr_file,
            zones )
```

```
Retreive air supply from ambient to zones.

Args:
    res_file: ESP-r results database.
    mfr_file: ESP-r mass flow results database.
    zones: List of strings with zones to include e.g.
        zones = ["a", "b"] to get air flow from those air flow nodes

Returns:
    df: Pandas dataframe with volume flow rate to/from ambient per zone.
```

4.7.2.3 calc_airtightness()

```
def espy.res.calc_airtightness (
          res_file,
          mfr_file,
          volume,
          zones )
```

Calculate building airtightness at 50 Pa.

```
Args:
    res_file: ESP-r results database.
    mfr_file: ESP-r mass flow results database.
    volume: Heated volume of building (m^3).
    zones: List of strings with zones to include e.g.
        zones = ["a", "b"] to get air flow from those air flow nodes

Returns:
    n_50: Air change rate (1/h)
    q_50: Air permeability (m^3/(h.m^2))
    w_50: Specific leakage rate (m^3/(h.m^2))
```

4.7.2.4 energy_balance()

```
def espy.res.energy_balance (
          cfg_file,
          res_file,
          out_file = None,
          group = None )
```

Get zone energy balance.

4.7.2.5 get_pv()

```
def espy.res.get_pv (
          res_file,
          elr_file,
          out_file = None )
```

Get PV output.

4.7.2.6 time_series()

```
def espy.res.time_series (
            cfg_file,
            res_file,
            param_list,
            out_file = None,
            time_fmt = 'DateTime' )
```

```
Extract results from results database to CSV.

Args:
    cfg_file: ESP-r configuration file.
    res_file: ESP-r results database.
    param_list: List of parameters to extract.
        Examples –
        param_list = [['all', 'Zone db T']]
        param_list = [['id:reception', 'Zone db T']]
        param_list = [[['id:roof_space', 'id:reception'], 'Zone db T']]
        param_list = [[['a', 'b'], 'Zone db T'], [['id:reception', 'b'], 'Wind direction']]
    out_file (optional): Name of exported CSV file.
    time_fmt (optional): Format of datetime in exported CSV. Julian or DateTime, default DateTime.

Returns:
    res: DataFrame containing results.
```

4.8 espy.utils Namespace Reference

Functions

- def **header** (str_in, lvl=0)
- def split_to_float (string)
- def space_data_to_list (item, convert="int")
- def sed (pattern, replace, source, dest=None, count=0)

4.8.1 Detailed Description

```
Helper utilities
```

4.8.2 Function Documentation

4.8.2.1   sed()

```
def espy.utils.sed (
              pattern,
              replace,
              source,
              dest = None,
              count = 0 )
```

Reads a source file and writes the destination file.

In each line, replaces pattern with replace.

```
Args:
    pattern (str): pattern to match (can be re.pattern)
    replace (str): replacement str
    source  (str): input filename
    count (int): number of occurrences to replace
    dest (str):   destination filename, if not given, source will be over written.
```

4.8.2.2   space_data_to_list()

```
def espy.utils.space_data_to_list (
              item,
              convert = "int" )
```

Transform space separated data into specified type list

4.8.2.3   split_to_float()

```
def espy.utils.split_to_float (
              string )
```

Transform CSV string into list of floats.

4.9   espy.write Namespace Reference

Functions

- def construction (fout, constr_name, constr_data, air_gap_data, mat_names)
- def img_to_md (fout, img_file, caption)

4.9.1   Detailed Description

Write out various files.

4.9.2 Function Documentation

4.9.2.1 construction()

```
def espy.write.construction (
            fout,
            constr_name,
            constr_data,
            air_gap_data,
            mat_names )
```

Write out construction data in markdown format.

```
Args:
    constr_name: str
        Construction name.

    constr_data: list
        List of construction data layers and thermophysical properties.

    air_gap_data: list
        List of air gap locations and properties.

    mat_names: list
        List of str of length N with name of each material layer.

Returns:
    out_file: str
        Filename of open out file.
```

4.9.2.2 img_to_md()

```
def espy.write.img_to_md (
            fout,
            img_file,
            caption )
```

Generate markdown format image text.

Chapter 5

Class Documentation

5.1 espy.bps.Bps Class Reference

Public Member Functions

- def **__init__** (self)
- def **__del__** (self)

Static Public Attributes

- int **counter** = 0

5.1.1 Detailed Description

```
Instance of BPS.
```

The documentation for this class was generated from the following file:

- espy/bps.py

5.2 espy.get.Component Class Reference

Public Member Functions

- def **__init__** (self, property_list, vertices_surf)
- def generate_vtk_surface (self)
- def set_outer_colour (self)

Public Attributes

- **name**
- **position**
- **child**
- **usage**
- **construction**
- **optical_type**
- **boundary**
- **vertices_surf**

5.2.1 Detailed Description

```
Class defining zone component.
```

5.2.2 Member Function Documentation

5.2.2.1 generate_vtk_surface()

```
def espy.get.Component.generate_vtk_surface (
            self )
```

```
Generate building component surface as a VTK objects
```

5.2.2.2 set_outer_colour()

```
def espy.get.Component.set_outer_colour (
            self )
```

```
Set default colour of otherside surface based on boundary conditions.
```

The documentation for this class was generated from the following file:

- espy/get.py

Index