

# Project Card Creation Quickstart

In this notebook we will run through:

1. creating a highway project card from a cube log file
2. creating a transit project card from two cube line files

These are set up steps, no input changes

```
In [1]: import os
import sys

from lasso import Project
from lasso import CubeTransit
from network_wrangler import WranglerLogger
```

```
In [2]: %load_ext autoreload
%autoreload 2
```

```
In [3]: import logging
logger = logging.getLogger("WranglerLogger")
logger.handlers[0].stream = sys.stdout
# if you don't want to see so much detail, set to logging.INFO or DEBUG
logger.setLevel(logging.DEBUG)
```

## Set the Directory here

```
In [4]: #set examples directory
EX_DIR = os.path.join(os.path.dirname(os.getcwd()), "Test1")
EX_DIR
```

```
Out[4]: 'C:\\Lasso\\Test1'
```

## Roadway Project Card

Roadway project cards are built by reading in a base network in the standard network format and a then processing cube log files.

The log file information is stored in a dataframe called `roadway_changes` - I don't understand what dataframe means here

The project card data is created when comparing log file to the base network and is stored in the variable `card_data` - I don't understand what variable means here

```
In [5]: #LOAD NETWORKS
#this step can take a while to load networks
test_roadway_project = Project.create_project(
    base_roadway_dir=os.path.join(EX_DIR, "base_network"), #NETWORK FILE in json format only
    roadway_log_file=os.path.join(EX_DIR, "logs", "Test1_Bridge.log"),
)
```

```
2020-06-03 10:14:36, INFO: No base transit network.
2020-06-03 10:14:36, INFO: No transit changes given or processed.
2020-06-03 10:14:36, INFO: Reading logfile: C:\Lasso\Test1\logs\Test1_Bridge.log
2020-06-03 10:14:36, INFO: Processed 1 Node lines and 6 Link lines
2020-06-03 10:14:36, INFO: Reading from following files:
-C:\Lasso\Test1\base_network\link.json
-C:\Lasso\Test1\base_network\node.geojson
-C:\Lasso\Test1\base_network\shape.geojson.
2020-06-03 10:18:47, INFO: Read 1134074 links from C:\Lasso\Test1\base_network\link.json
2020-06-03 10:18:47, INFO: Read 354346 nodes from C:\Lasso\Test1\base_network\node.geojson
2020-06-03 10:18:47, INFO: Read 1134074 shapes from C:\Lasso\Test1\base_network\shape.geojson
2020-06-03 10:28:28, INFO: Lasso base directory set as: C:\Lasso
2020-06-03 10:28:28, INFO: Creating calculated roadway variables.
2020-06-03 10:28:28, INFO: Calculating Area Type from Spatial Data and adding a roadway network variable: area_type
2020-06-03 10:28:37, DEBUG: Reading Area Type Shapefile C:\Lasso\metcouncil_data\area_type\ThriveMSP2040CommunityDesignation.shp
```

```
-----
CRSError                                Traceback (most recent call last)
<ipython-input-5-1f08ded25676> in <module>
      3 test_roadway_project = Project.create_project(
      4     base_roadway_dir=os.path.join(EX_DIR, "base_network"), #NETWORK
FILE in json format only
----> 5     roadway_log_file=os.path.join(EX_DIR, "logs", "Test1_Bridge.log")
      6 )

c:\lasso\lasso\project.py in create_project(roadway_log_file, roadway_shp_file,
roadway_csv_file, base_roadway_dir, base_transit_source, build_transit_source,
roadway_changes, transit_changes, base_roadway_network, base_transit_network,
build_transit_network, project_name, parameters)
    229         True,
    230     )
--> 231     base_roadway_network.create_calculated_variables()
    232     base_roadway_network.calculate_distance(overwrite = True)
    233     base_roadway_network.fill_na()

c:\lasso\lasso\roadway.py in create_calculated_variables(self)
    165     """
    166     WranglerLogger.info("Creating calculated roadway variables.")
--> 167     self.calculate_area_type()
    168     self.calculate_county()
    169     self.calculate_centroidconnect()
```

```

c:\lasso\lasso\roadway.py in calculate_area_type(self, area_type_shape, area_type_shape_variable, network_variable, area_type_codes_dict, overwrite)
    359         WranglerLogger.debug("Reading Area Type Shapefile {}".format(area_type_shape))
    360         area_type_gdf = gpd.read_file(area_type_shape)
--> 361         area_type_gdf = area_type_gdf.to_crs(epsg=RoadwayNetwork.CRS)
    362
    363         joined_gdf = gpd.sjoin(

```

```

c:\programdata\miniconda3\envs\lasso_env\lib\site-packages\geopandas\geodataframe.py in to_crs(self, crs, epsg, inplace)
    562         else:
    563             df = self.copy()
--> 564             geom = df.geometry.to_crs(crs=crs, epsg=epsg)
    565             df.geometry = geom
    566             df.crs = geom.crs

```

```

c:\programdata\miniconda3\envs\lasso_env\lib\site-packages\geopandas\geoseries.py in to_crs(self, crs, epsg)
    426         crs = CRS.from_user_input(crs)
    427         elif epsg is not None:
--> 428             crs = CRS.from_epsg(epsg)
    429         else:
    430             raise ValueError("Must pass either crs or epsg.")

```

```

c:\programdata\miniconda3\envs\lasso_env\lib\site-packages\pyproj\crs\crs.py in from_epsg(code)
    329         CRS
    330         """
--> 331         return CRS(_prepare_from_epsg(code))
    332
    333     @staticmethod

```

```

c:\programdata\miniconda3\envs\lasso_env\lib\site-packages\pyproj\crs\crs.py in __init__(self, projparams, **kwargs)
    294         projstring = _prepare_from_string(" ".join((projstring, projkwargs)))
    295
--> 296         super().__init__(projstring)
    297
    298     @staticmethod

```

```

pyproj/_crs.pyx in pyproj._crs._CRS.__init__()

```

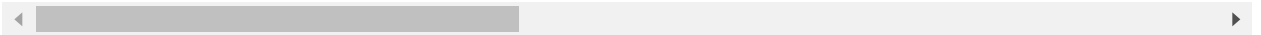
**CRSError:** Invalid projection: epsg:+proj=longlat +ellps=WGS84 +datum=WGS84 +no\_defs +type=crs: (Internal Proj Error: proj\_create: crs not found)

```
In [6]: test_roadway_project.roadway_changes[0:10]
```

```
Out[6]:
```

	OBJECT	OPERATION	GROUP	A	B	model_link_id	county	trn_priority	area_type
0	L	C	0	3230	52771	224	5	0	4
1	L	C	0	3261	3262	280	5	0	4
2	L	C	0	3261	68075	282	5	0	4
3	L	A	0	3230	3262	999998	5	0	4
4	L	A	0	3262	3230	999997	5	0	4
5	L	D	0	3261	131209	281	5	0	4
6	L	D	0	178775	42542	477533	5	0	4
7	N	A	0	NaN	NaN	NaN	NaN	NaN	NaN

8 rows × 35 columns



```
In [7]: test_roadway_project.roadway_changes.columns
```

```
Out[7]: Index(['OBJECT', 'OPERATION', 'GROUP', 'A', 'B', 'model_link_id', 'county',
              'trn_priority', 'area_type', 'HOV_access', 'aadt', 'count_AM',
              'count_MD', 'count_PM', 'count_NT', 'count_daily', 'assign_group',
              'lanes', 'centroidconnect', 'roadway_class', 'bike_facility',
              'drive_access', 'walk_access', 'bike_access', 'transit_access',
              'distance', 'ttime_assert', 'model_node_id', 'X', 'Y', 'osm_node_id',
              'drive_node', 'walk_node', 'bike_node', 'transit_node'],
              dtype='object')
```

In [8]: test\_roadway\_project.card\_data

```
Out[8]: {'project': 'USER TO define',
'changes': [{'category': 'Roadway Deletion',
'links': {'model_link_id': [281, 477533]}},
{'category': 'New Roadway',
'links': [{'A': 3230,
'B': 3262,
'model_link_id': 999998,
'county': 5,
'trn_priority': 0,
'area_type': 4,
'count_AM': 0,
'count_MD': 0,
'count_PM': 0,
'count_NT': 0,
'count_daily': 0,
'assign_group': 5,
'lanes': 1,
'centroidconnect': 0,
'roadway_class': 40,
'bike_facility': 3,
'drive_access': 1,
'walk_access': 1,
'bike_access': 1,
'distance': 0.10462,
'ttime_assert': 0.0,
'OPERATION_final': 'A'}},
{'A': 3262,
'B': 3230,
'model_link_id': 999997,
'county': 5,
'trn_priority': 0,
'area_type': 4,
'count_AM': 0,
'count_MD': 0,
'count_PM': 0,
'count_NT': 0,
'count_daily': 0,
'assign_group': 5,
'lanes': 1,
'centroidconnect': 0,
'roadway_class': 40,
'bike_facility': 3,
'drive_access': 1,
'walk_access': 1,
'bike_access': 1,
'distance': 0.10462,
'ttime_assert': 0.0,
'OPERATION_final': 'A'}]},
'nodes': [{'model_node_id': 354388,
'osm_node_id': '',
'drive_node': 1,
'walk_node': 0,
'bike_node': 0}],
'facility': {'link': {'model_link_id': [280, 282]}},
'properties': [{'property': 'assign_group', 'existing': 7, 'set': 5},
```

```
{'property': 'bike_facility', 'existing': 0, 'set': 3},
{'property': 'walk_access', 'existing': 0, 'set': 1},
{'property': 'bike_access', 'existing': 0, 'set': 1}],
'category': 'Roadway Attribute Change'},
{'facility': {'link': {'model_link_id': [224]}}},
'properties': [{'property': 'assign_group', 'existing': 7, 'set': 5},
{'property': 'drive_access', 'existing': 0, 'set': 1},
{'property': 'distance', 'existing': 0.0503218951, 'set': 0.10462}],
'category': 'Roadway Attribute Change']}]}
```

```
In [9]: test_roadway_project.write_project_card(
        "roadway_test_rachel.yml"
    )
```

2020-06-02 15:05:27, INFO: Wrote project card to: roadway\_test\_rachel.yml

## Transit Project Card

Transit project cards are built by taking the differences between two cube transit line files.

```
In [11]: test_transit_project = Project.create_project(
        base_transit_source=os.path.join(EX_DIR, "cube", "transit.LIN"),
        build_transit_source=os.path.join(EX_DIR, "cube", "single_transit_route_attr
    )

test_transit_project.write_project_card(
    "transit_test_rachel.yml"
)
```

HI THERE

2020-06-02 15:15:32, DEBUG: Creating a new Cube Transit instance

2020-06-02 15:15:32, INFO: Lasso base directory set as: C:\lasso

reading: C:\lasso\examples\cube\transit.LIN

2020-06-02 15:15:32, DEBUG: reading transit source: C:\lasso\examples\cube\transit.LIN

2020-06-02 15:15:33, DEBUG: finished parsing cube line file

2020-06-02 15:15:34, DEBUG: Added lines to CubeTransit:

2020-06-02 15:15:34, DEBUG: Base network has 508 lines

2020-06-02 15:15:34, DEBUG: build

HI THERE

2020-06-02 15:15:34, DEBUG: Creating a new Cube Transit instance

2020-06-02 15:15:34, INFO: Lasso base directory set as: C:\lasso

reading: C:\lasso\examples\cube\single\_transit\_route\_attribute\_change\transit.LIN

2020-06-02 15:15:34, DEBUG: reading transit source: C:\lasso\examples\cube\single\_transit\_route\_attribute\_change\transit.LIN

2020-06-02 15:15:36, DEBUG: finished parsing cube line file

2020-06-02 15:15:36, DEBUG: Added lines to CubeTransit:

## Roadway Project Card - difference from csv files or shapefiles

For functionality of taking difference from csv files, or shapefiles

```
In [11]: test_roadway_project = Project.create_project(  
        roadway_csv_file=os.path.join(EX_DIR, "cube", "example_csv_roadway_change.c  
        base_roadway_dir=os.path.join(EX_DIR, "stpaul")  
    )  
  
    test_roadway_project.write_project_card(  
        os.path.join(  
            SCRATCH_DIR,  
            "t_" + "example_csv_roadway_change" + ".yaml",  
        )  
    )  
)
```

2020-05-13 14:43:40, INFO: No base transit network.

2020-05-13 14:43:40, INFO: No transit changes given or processed.

2020-05-13 14:43:40, INFO: Reading from following files:

-C:\lasso\examples\stpaul\link.json

-C:\lasso\examples\stpaul\node.geojson

-C:\lasso\examples\stpaul\shape.geojson.

2020-05-13 14:43:48, INFO: Read 66253 links from C:\lasso\examples\stpaul\lin  
k.json

2020-05-13 14:43:48, INFO: Read 17159 nodes from C:\lasso\examples\stpaul\nod  
e.geojson

2020-05-13 14:43:48, INFO: Read 66253 shapes from C:\lasso\examples\stpaul\sh  
ape.geojson

```
In [ ]: test_roadway_project = Project.create_project(  
        roadway_shp_file=os.path.join(EX_DIR, "cube", "example_shapefile_roadway_ch  
        base_roadway_dir=os.path.join(EX_DIR, "stpaul")  
    )  
  
    test_roadway_project.write_project_card(  
        os.path.join(  
            SCRATCH_DIR,  
            "t_" + "example_shapefile_roadway_change" + ".yaml",  
        )  
    )  
)
```