

## eSim2022 BTAP Workshop Exercise 2 Instructions

Chris Kirney

2022-06-16

Exercise 2:

In this exercise we will run `btap_batch` with the custom geometry file we modified in exercise 1. If you didn't complete exercise 1, or you had trouble with your file, you can use the 'btap\_batch/esim2022/exercise2.osm' file. This exercise assumes that you have already set-up everything you need to run `btap_batch` and have already tried a parametric analysis locally as per the README.md instructions here:

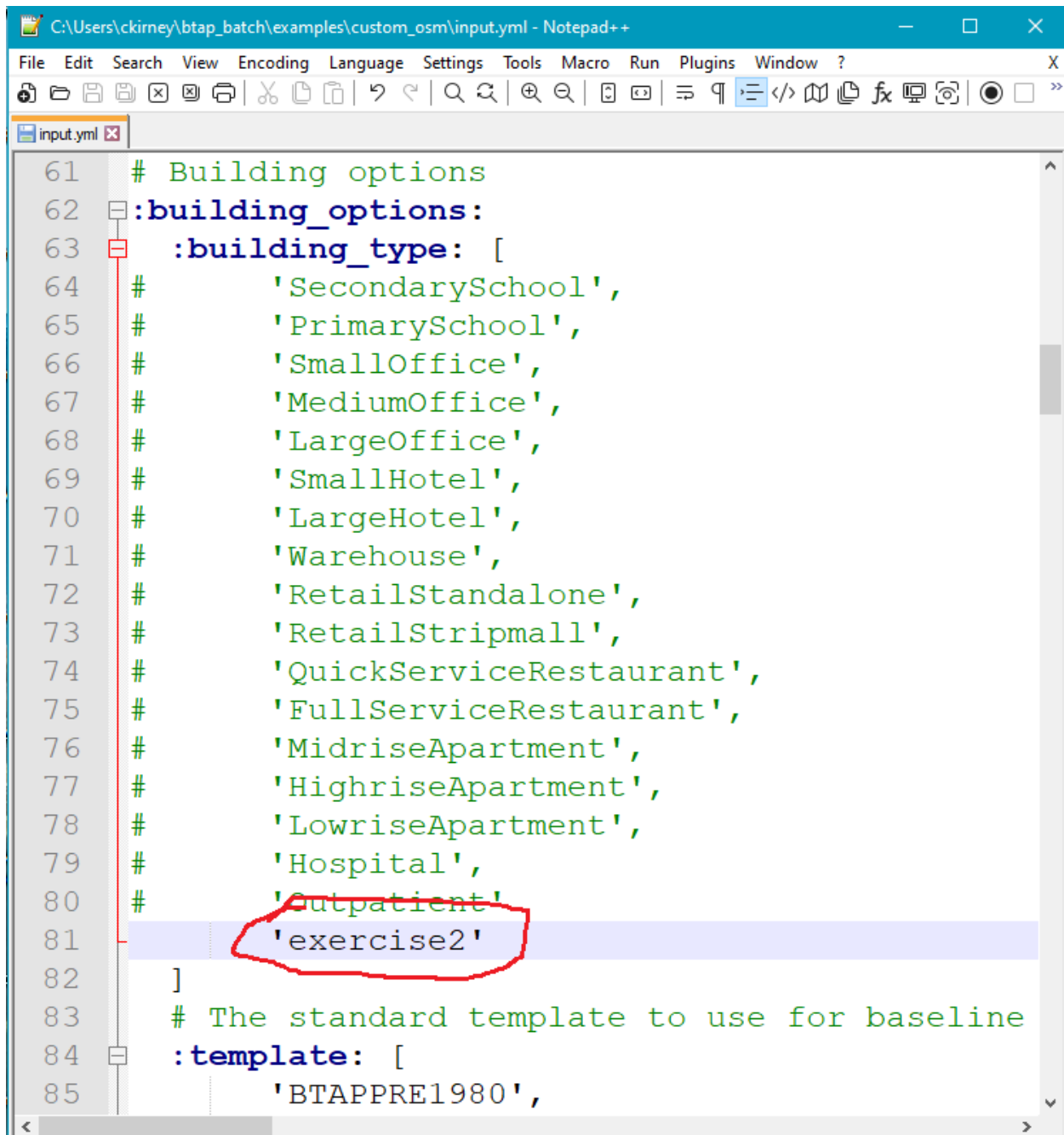
[https://github.com/canmet-energy/btap\\_batch/blob/esim\\_mod/README.md](https://github.com/canmet-energy/btap_batch/blob/esim_mod/README.md)

To continue your computer should have internet access.

1. Copy the file the completed in exercise 1, or the 'btap\_batch/esim2022/exercise2.osm' file, to the 'btap\_batch/examples/custom\_osm/osm\_folder/' folder.
2. Use a text editor (e.g. Notepad, Notepad++, Sublime Text, etc.) to open the 'btap\_batch/examples/custom\_osm/input.yml' file.
3. Look for 'test1' in the ':building\_type:' array (see below):

```
C:\Users\ckirney\btap_batch\examples\custom_osm\input.yml - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
input.yml x
61 # Building options
62 :building_options:
63 :building_type: [
64 # 'SecondarySchool',
65 # 'PrimarySchool',
66 # 'SmallOffice',
67 # 'MediumOffice',
68 # 'LargeOffice',
69 # 'SmallHotel',
70 # 'LargeHotel',
71 # 'Warehouse',
72 # 'RetailStandalone',
73 # 'RetailStripmall',
74 # 'QuickServiceRestaurant',
75 # 'FullServiceRestaurant',
76 # 'MidriseApartment',
77 # 'HighriseApartment',
78 # 'LowriseApartment',
79 # 'Hospital',
80 # 'Outpatient'
81 'test1'
82 ]
83 # The standard template to use for baseline
84 :template: [
85 'BTAPPRE1980',
```

4. Replace 'test1' text with 'exercise2' or whatever is the name of the file (without the '.osm' extension) you put in the 'osm\_folder' (see below):



```
61 # Building options
62 :building_options:
63   :building_type: [
64     # 'SecondarySchool',
65     # 'PrimarySchool',
66     # 'SmallOffice',
67     # 'MediumOffice',
68     # 'LargeOffice',
69     # 'SmallHotel',
70     # 'LargeHotel',
71     # 'Warehouse',
72     # 'RetailStandalone',
73     # 'RetailStripmall',
74     # 'QuickServiceRestaurant',
75     # 'FullServiceRestaurant',
76     # 'MidriseApartment',
77     # 'HighriseApartment',
78     # 'LowriseApartment',
79     # 'Hospital',
80     # 'Outpatient',
81     'exercise2'
82   ]
83 # The standard template to use for baseline
84 :template: [
85   'BTAPPRE1980',
```

5. We will also use a new weather location. Use the link below to go to where the weather files are stored on the 'openstudio-standards' repository:

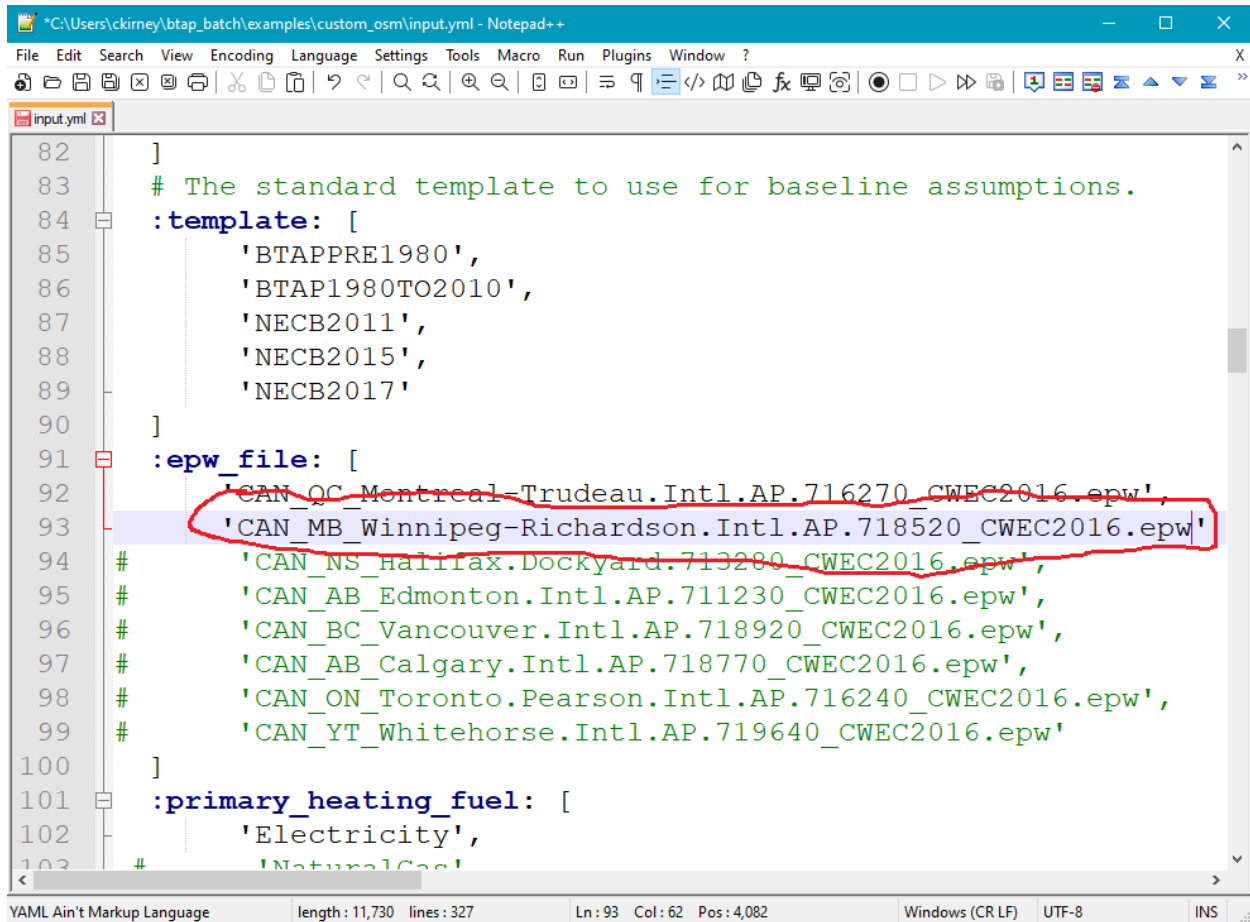
<https://github.com/NREL/openstudio-standards/tree/nrcan/data/weather>

6. You can use whichever weather location there that you want as long as it has a '.epw', '.stat', and '.ddy' file associated with it. For this exercise we'll use Winnipeg (see below):

The screenshot shows a GitHub repository page for 'openstudio-standards/data/weather'. The table lists various weather files and their commit history. The file 'CAN\_MB\_Winnipeg-Richardson.Intl.AP.718520\_...' is highlighted with a red circle.

File Name	Commit Message	Commit Hash	Time Ago
CAN_MB_Brandon.Muni.AP.711400_CWEC2016.s...	added rsmens epw locations	(#743)	3 years ago
CAN_MB_Portage-Southport.AP.718510_CWEC2...	added rsmens epw locations	(#743)	3 years ago
CAN_MB_Portage-Southport.AP.718510_CWEC2...	added rsmens epw locations	(#743)	3 years ago
CAN_MB_Portage-Southport.AP.718510_CWEC2...	added rsmens epw locations	(#743)	3 years ago
CAN_MB_The.Pas.AP.718670_CWEC2016.d...	Nrcan 280	(#713)	3 years ago
CAN_MB_The.Pas.AP.718670_CWEC2016.epw	Removed old weather files and replaced them with new 2016 files based...		5 years ago
CAN_MB_The.Pas.AP.718670_CWEC2016.stat	Removed old weather files and replaced them with new 2016 files based...		5 years ago
CAN_MB_Winnipeg-Richardson.Intl.AP.718520_...	Update ddy files to E+ 9.0 RunPeriod object		4 years ago
CAN_MB_Winnipeg-Richardson.Intl.AP.718520_...	added rsmens epw locations	(#743)	3 years ago
CAN_MB_Winnipeg-Richardson.Intl.AP.718520_...	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Bathurst.Rgnl.AP.715740_CWEC2016.d...	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Bathurst.Rgnl.AP.715740_CWEC2016.e...	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Bathurst.Rgnl.AP.715740_CWEC2016.stat	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Fredericton.Intl.AP.717000_CWEC2016....	Update ddy files to E+ 9.0 RunPeriod object		4 years ago
CAN_NB_Fredericton.Intl.AP.717000_CWEC2016....	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Fredericton.Intl.AP.717000_CWEC2016....	added rsmens epw locations	(#743)	3 years ago
CAN_NB_Miramichi.AP.717440_CWEC2016.d...	Nrcan 280	(#713)	3 years ago
CAN_NB_Miramichi.AP.717440_CWEC2016.epw	Removed old weather files and replaced them with new 2016 files based...		5 years ago

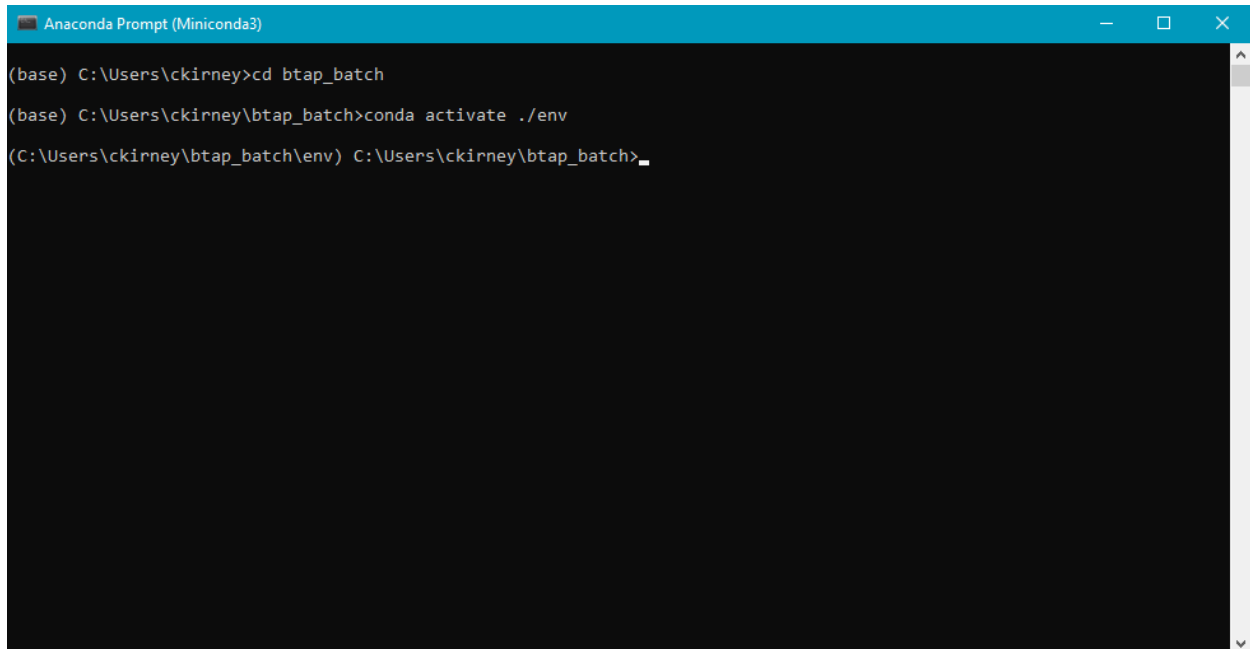
7. In the 'btap\_batch/examples/custom\_osm/input.yml' file add a line under 'CAN\_QC\_Montreal-Trudeau.Intl.AP.716270\_CWEC2016.epw' in the 'epw\_file:' array. Put 'CAN\_MB\_Winnipeg-Richardson.Intl.AP.718520\_CWEC2016.epw' in the space you just created (see below):



```
82 ]
83 # The standard template to use for baseline assumptions.
84 :template: [
85   'BTAPPRE1980',
86   'BTAP1980TO2010',
87   'NECB2011',
88   'NECB2015',
89   'NECB2017'
90 ]
91 :epw_file: [
92   'CAN_QC_Montreal-Trudeau.Intl.AP.716270_CWEC2016.epw',
93   'CAN_MB_Winnipeg-Richardson.Intl.AP.718520_CWEC2016.epw',
94   # 'CAN_NS_Halifax.Dockyard.713280_CWEC2016.epw',
95   # 'CAN_AB_Edmonton.Intl.AP.711230_CWEC2016.epw',
96   # 'CAN_BC_Vancouver.Intl.AP.718920_CWEC2016.epw',
97   # 'CAN_AB_Calgary.Intl.AP.718770_CWEC2016.epw',
98   # 'CAN_ON_Toronto.Pearson.Intl.AP.716240_CWEC2016.epw',
99   # 'CAN_YT_Whitehorse.Intl.AP.719640_CWEC2016.epw'
100 ]
101 :primary_heating_fuel: [
102   'Electricity',
103   # 'NaturalGas'
```

YAML Ain't Markup Language    length: 11,730    lines: 327    Ln: 93    Col: 62    Pos: 4,082    Windows (CR LF)    UTF-8    INS

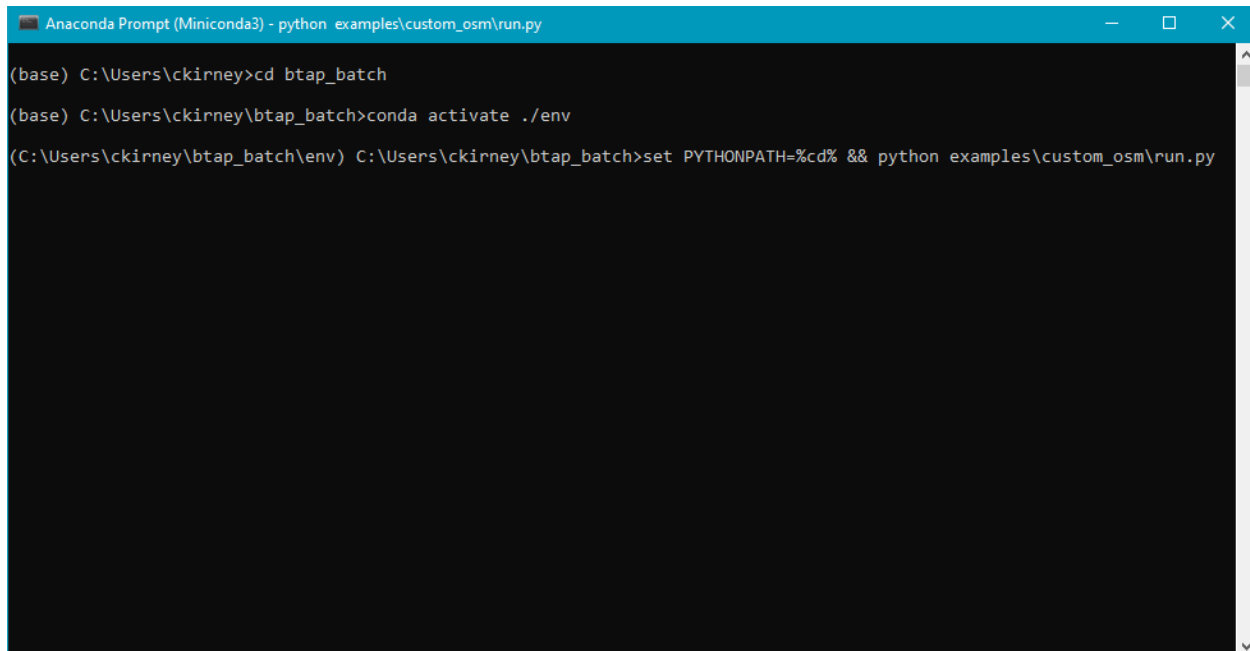
8. Save and close the 'btap\_batch/examples/custom\_osm/input.yml'.
9. Make sure Docker Desktop is running and open the 'Anaconda Prompt (Miniconda3)'.
10. In the 'Anaconda Prompt (Miniconda3)' window navigate to 'btap\_batch'.
11. Assuming you created the Anaconda environment as per the btap\_batch README.md, activate the Anaconda environment by typing 'conda activate ./env' (see below):



```
Anaconda Prompt (Miniconda3)

(base) C:\Users\ckirney>cd btap_batch
(base) C:\Users\ckirney\btap_batch>conda activate ./env
(C:\Users\ckirney\btap_batch\env) C:\Users\ckirney\btap_batch>_
```

12. We will now start our btap\_batch run by typing 'set PYTHONPATH=%cd% && python examples\custom\_osm\run.py' (see below):



```
Anaconda Prompt (Miniconda3) - python examples\custom_osm\run.py

(base) C:\Users\ckirney>cd btap_batch
(base) C:\Users\ckirney\btap_batch>conda activate ./env
(C:\Users\ckirney\btap_batch\env) C:\Users\ckirney\btap_batch>set PYTHONPATH=%cd% && python examples\custom_osm\run.py
```

13. The analysis should start after a minute or two and you should see something like the following:

```
Anaconda Prompt (Miniconda3) - python examples\custom_osm\run.py

(base) C:\Users\ckirney>cd btap_batch

(base) C:\Users\ckirney\btap_batch>conda activate ./env

(C:\Users\ckirney\btap_batch\env) C:\Users\ckirney\btap_batch>set PYTHONPATH=%cd% && python examples\custom_osm\run.py
Log file created: C:\Users\ckirney\btap_batch\examples\custom_osm\19f06fcd-0072-4011-8d1a-0aec915b580f.log
Compute Environment:local
Analysis Type:parametric
Building image:btap_private_cli
OS Version:3.2.1
BTAP_COSTING Branch:
OS STANDARDS Branch:nrcan_prod
Docker folder being use to build image:C:\Users\ckirney\btap_batch\src\Dockerfiles\btap_private_cli
Using existing image.
Docker version 20.10.16, build aa7e414
analysis_id is: 19f06fcd-0072-4011-8d1a-0aec915b580f
analysis_name is: custom_osm_example_ref
analysis type is: reference
Deleting previous runs from: C:\Users\ckirney\btap_batch\examples\custom_osm\custom_osm_example_ref
Creating new folders for analysis
running reference stage
Using 46 threads.
Failed:0: Progress Bar: 0%| | 0/10 [00:00<?, ?it/s]
```

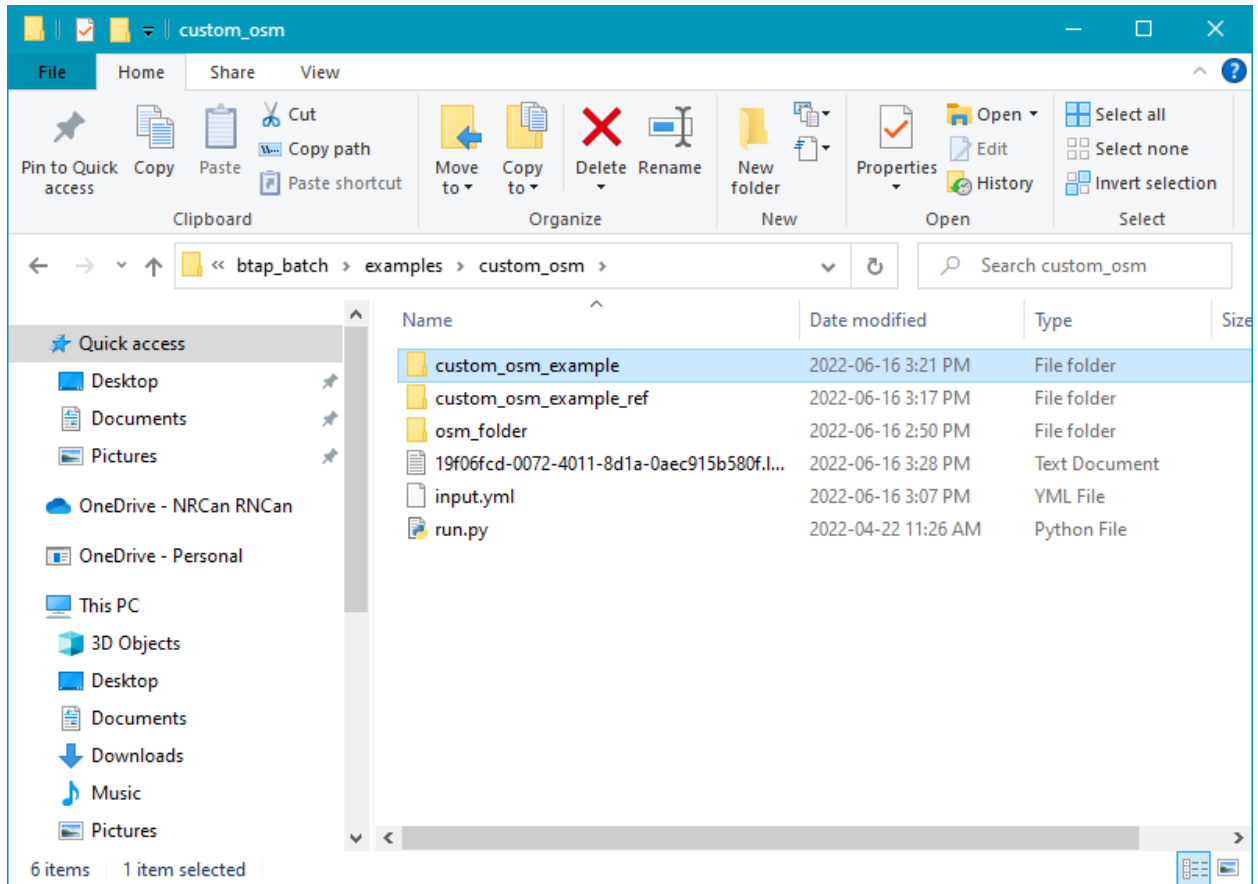
14. Wait until the analysis is complete. Assuming all went well, you should see something like the following:

```
Anaconda Prompt (Miniconda3)

10 Simulations completed. No. of failures = 0 Total Time: 0:03:28
Shutdown..
Downloading in.osm files: 100%| | 10/10 [00:00<00:00, 222.22it/s]
Downloading eplustbl.htm files: 100%| | 10/10 [00:00<00:00, 370.35it/s]
Downloading hourly.csv files: 100%| | 10/10 [00:00<00:00, 96.14it/s]
Saved Excel Output: C:\Users\ckirney\btap_batch\examples\custom_osm\custom_osm_example_ref\19f06fcd-0072-4011-8d1a-0aec915b580f\results\output.xlsx
Unexpected botocore.exceptions.ClientError error: An error occurred (InvalidClientTokenId) when calling the GetCallerIdentity operation: The security token included in the request is invalid
Run locally. No need to copy sum_hourly_res.csv to s3
Docker version 20.10.16, build aa7e414
analysis_id is: 19f06fcd-0072-4011-8d1a-0aec915b580f
analysis_name is: custom_osm_example
analysis type is: parametric
Deleting previous runs from: C:\Users\ckirney\btap_batch\examples\custom_osm\custom_osm_example
Creating new folders for analysis
Using 46 threads.
Failed:0: Progress Bar: 100%| | 20/20 [05:52<00:00, 17.64s/it]
20 Simulations completed. No. of failures = 0 Total Time: 0:05:54
Shutdown..
Downloading in.osm files: 100%| | 20/20 [00:00<00:00, 273.99it/s]
Downloading eplustbl.htm files: 100%| | 20/20 [00:00<00:00, 416.65it/s]
Downloading hourly.csv files: 100%| | 20/20 [00:00<00:00, 215.05it/s]
Saved Excel Output: C:\Users\ckirney\btap_batch\examples\custom_osm\custom_osm_example\19f06fcd-0072-4011-8d1a-0aec915b580f\results\output.xlsx
Unexpected botocore.exceptions.ClientError error: An error occurred (InvalidClientTokenId) when calling the GetCallerIdentity operation: The security token included in the request is invalid
Run locally. No need to copy sum_hourly_res.csv to s3

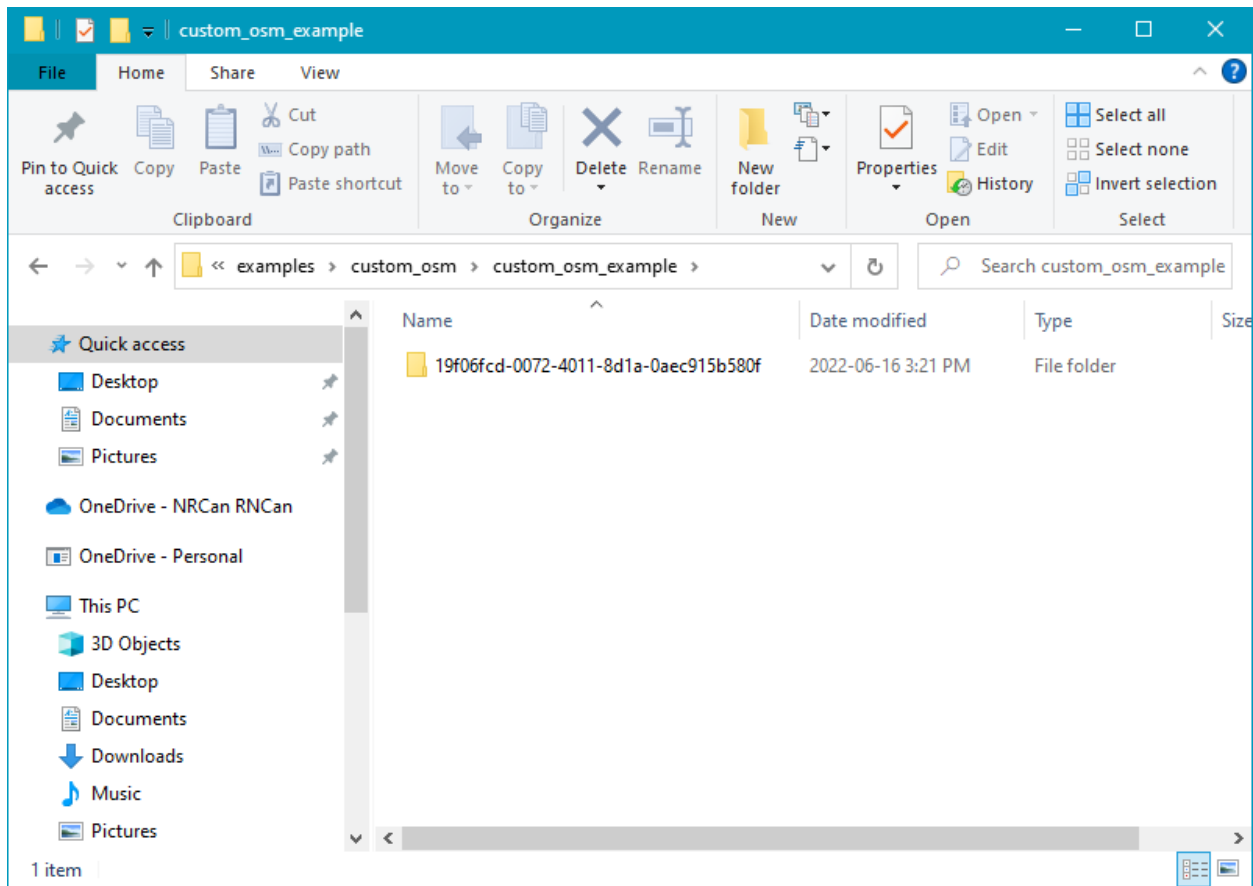
(C:\Users\ckirney\btap_batch\env) C:\Users\ckirney\btap_batch>
```

15. Congratulations! Your simulations are complete (and hopefully successful). Close the 'Anaconda Prompt (Miniconda3)' and open the Windows File Explorer and navigate to your btap\_batch folder and then into 'btap\_batch/examples/custom\_osm'. There should be a 'custom\_osm\_example' folder and a 'custom\_osm\_example\_ref' folder (see below):



16. The 'custom\_osm\_example\_ref' folder contains the results of a reference analysis. This is essentially the same as the analysis we did except that all building options (other than building type, template, weather, and primary heating fuel) are defaulted. Open the 'custom\_osm\_example' folder and you should see another folder with a long string of numbers and characters (see below, note that your folder should have a different string of numbers and characters than what is in the screenshot):





17. This is the folder that contains the analysis you ran. The long string of numbers and characters (called a hash from now on) is what btap\_batch has named your analysis. Open that folder and you should find an 'input' folder, an 'output' folder, and a 'results' folder.

The 'input' folder should contain a number of folders with hash names. Each of these folders represents a specific simulation and contain the OpenStudio file used in the simulation and a 'run\_optinos.yml' file that contain the specific btap\_batch characteristics of that particular simulation.

As with 'input', the output folder contains several folders with hash names. As with 'input', each folder represents a simulation. In this case the folders contain the outputs from the simulation including the OpenStudio and EnergyPlus results and some output files created by BTAP. Two in particular are 'btap\_data.json' and 'qaqc.json'. Both files contain summarized results for the simulation. They both contain similar information. However, 'qaqc.json' has more information but is presented in a way that is more difficult to input into data analysis software or scripts.

The results folder contain 'database', 'eplustbl.htm', 'failures', 'hourly.csv', and 'in.osm' folders and an output.xlsx file.

The 'database' folder contains '.csv' files named with the hash of the simulation they refer too. Each '.csv' contains summarized results for that simulation. The information in all '.csv' files are collected together and saved in the 'output.xlsx' file.

The 'eplustbl.htm' file contains the EnergyPlus results '.htm' report for each simulation (they are named after their corresponding simulation).

The 'failures' folder contains any error outputs from failed simulations. It should be empty.

If you chose to produce hourly results for some output variables in the 'input.yml', those results are stored in the 'hourly.csv' folder. In this exercise we did, so the folder should contain many '.csv' files named after their corresponding simulation and containing the simulation's hourly results.

The 'in.osm' folder contains the OpenStudio models run for the analysis. Each '.osm' file is named after their corresponding simulation.

The output.xlsx file contains the summarized inputs and outputs for the analysis.