

Monthly TCC user documentation

Introduction

The following documentation gives simple instructions for users who are going to use script named GR-RO_monthly.py in path I:\Konstantinos Sidiropoulos\Documentation\TCC.

Script loads a folder that corresponds to a specific month of the year (**monthly folder**). Folder contains subfolders that correspond to specific days of this particular month (**daily folders**).

Each daily folder contains four subfolders that correspond to the type of Nodes (Greek/Romanian) and the capacity direction (Import/Export). Each subfolder contains 24 CGMs.

Script automate a process where for each CGM a row will be saved in a dataframe and will contain columns that provide date, time, capacity direction and TCC information.

A final dataframe is created and saved in .xlsx form and contains for a period of a month all the TCCs of each hourly specified CGM.

The code has been tested and works for CGMs in UCTE format. It was used in computer 10.91.100.15 with the following installations:

- Python version 3.12.0
- Pypowsybl library: User can simply install released versions of pypowsybl from [PyPI](#) using pip:

```
pip install pypowsybl
```

- Visual Studio Code: VSCodeUserSetup-x64-1.91.0.exe

Note: File_type was always 2D = Two Days Ahead (D-2)

All UCTE's had name structure:

ucte_filename = {Date}_{hour}_{File_type}{number}_{country_code}{number}.{format}

1. How to use

User has to enter after **running** the script the following information to console:

- **Enter the path to your base folder:** Enters the path where his monthly folders are.
- **Enter the path where TCC should be saved:** Enters the path where his TCC are going to be saved.
- **Enter the Year_Month:** Enters the monthly folder name.
- **Enter specific dates :** Enters the dates names of his folders inside the monthly folder.
- **Enter types:** Enters the types of his folders inside a date folder.

NOTE: Folders where the UCTE files are have the following structure (line 133): destination_folder = f'{base_folder}\\{Date}\\CGM\\{Type}'.

NOTE: UCTE filenames have the following structure (line 140): base_ucte_filename = f'{Date}_{{:04d}}_2D{D}_UX{U}.uct'. File type, country code and format are constant variables.

2. AC LoadFlow User execution and results

User has to define the parameters he needs for the AC LoadFlow execution.

Monthly TCC user documentation

Replaces **True/False** instead of **None** values for all the parameters he needs except: **balance_type**, **countries_to_balance**, **connected_component_mode**, **dc_power_factor**, **provider_parameters**.

All parameters description for AC LoadFlow execution are available in:

<https://powsybl.readthedocs.io/projects/pypowsybl/en/stable/reference/loadflow/parameters.html#pypowsybl.loadflow.Parameters>

```
1. p = lf.Parameters(  
2. distributed_slack=False ,  
3. transformer_voltage_control_on=False ,  
4. phase_shifter_regulation_on=True ,  
5. shunt_compensator_voltage_control_on=True ,  
6. voltage_init_mode=None,  
7. use_reactive_limits=None,  
8. twt_split_shunt_admittance=None,  
9. read_slack_bus=None,  
10. write_slack_bus=None,  
11. balance_type=None,  
12. dc_use_transformer_ratio=None,  
13. countries_to_balance=None,  
14. connected_component_mode=None,  
15. dc_power_factor=None,  
16. provider_parameters={  
17. 'maxOuterLoopIterations' : str(30) ,  
18. 'lowImpedanceBranchMode' : 'REPLACE_BY_MIN_IMPEDANCE_LINE' ,  
19. })
```

Figure 1: Parameters

Provider_parameters contain a list of parameters linked to the loadflow provider. All provider parameters available in the excel named '**provider_parameters.xlsx**' in I:\Konstantinos Sidiropoulos\Documentation\TCC. User can add the follow orders to get anytime the provider parameters names and description:

```
1. It =lf.get_provider_parameters() # OPTIONAL --> GET A LIST OF AVAILABLE provider PARAMETERS TO  
EXECUTE AC LOADFLOW  
2. It.to_excel('provider_parameters.xlsx')
```

Figure 2:Provider parameters

A final excel with the TCC of each CGM will be created named '**{Year_Month}_TCCS.xlsx**' (line 157). User can change the name if he wants to.

Data structure of final excel available in **202402_TCCs._Demonstration.xlsx**.

3. Merge.py

This script is useful only if user wants to have TCC over a monthly period.

User has to:

- Fill in **folder_path** variable the path to the monthly TCC excels (they have to be in the same folder).

Monthly TCC user documentation

A final excel named '**combine_monthly_TCCs.xlsx**' will be created that will contain TCC from different months.

```
1. import pandas as pd
2. import os
3.
4. #Script for user to combine monthly TCC calculations
5. folder_path = r'C:\Path\to\Monthly\TCC\excels'
6. df_list = []
7. for filename in os.listdir(folder_path):
8.     if filename.endswith('.xlsx') or filename.endswith('.xls'):
9.         file_path = os.path.join(folder_path, filename)
10.        # Read each Excel file and append to the list
11.        df = pd.read_excel(file_path)
12.        df_list.append(df)
13.
14. combine = pd.concat(df_list , ignore_index=True)
15. output_file = os.path.join(folder_path, 'combine_monthly_TCCs.xlsx')
16. combine.to_excel(output_file, index=False)
```

Figure 3: Merge script