

# QUICKSTART GUIDE

Power Plant & Model Test Bench (PP-MTB)  
Powerfactory

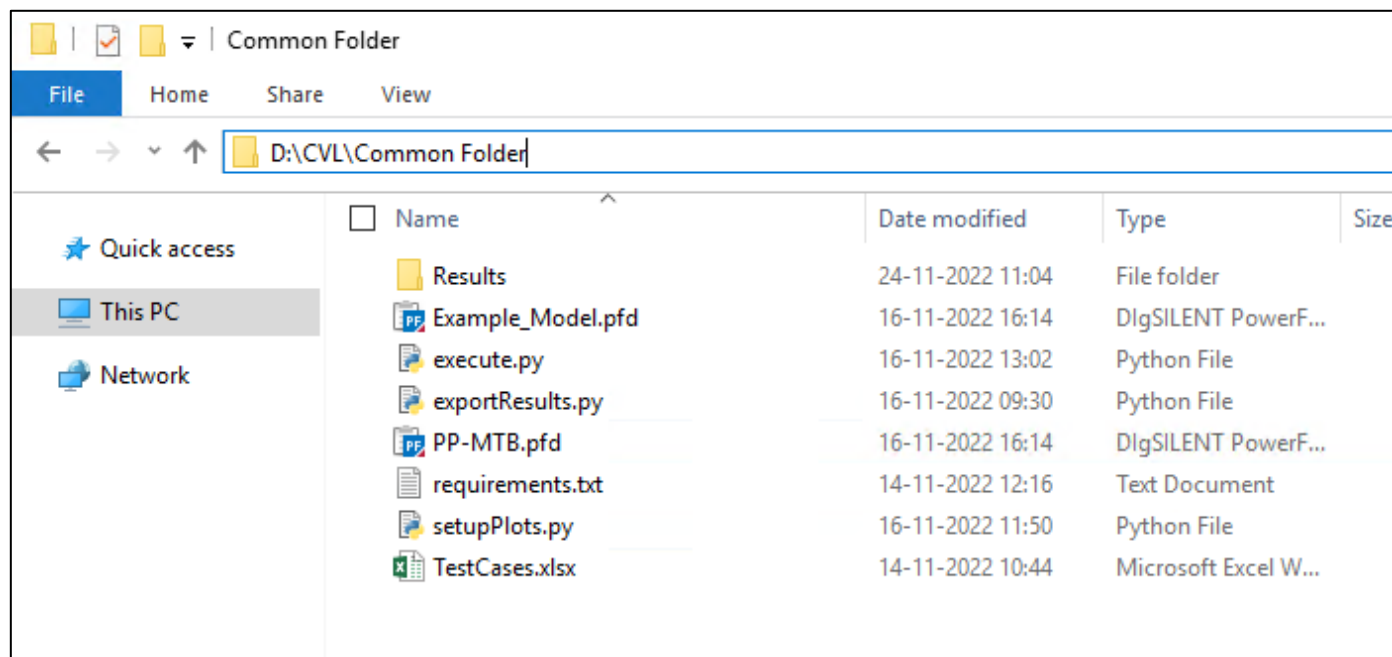
# 0. SYSTEM REQUIREMENTS

Developed on powerfactory 2022 SP3 with python 3.8.8. Dependencies are listed in requirements.txt and can be installed with the command:

```
python -m pip install -r requirements.txt
```

# 1. PREPARATION

1.1 Extract or copy all test bench files and your model into a common folder on your PC and make a subfolder for Results used later on



# 1. PREPARATION

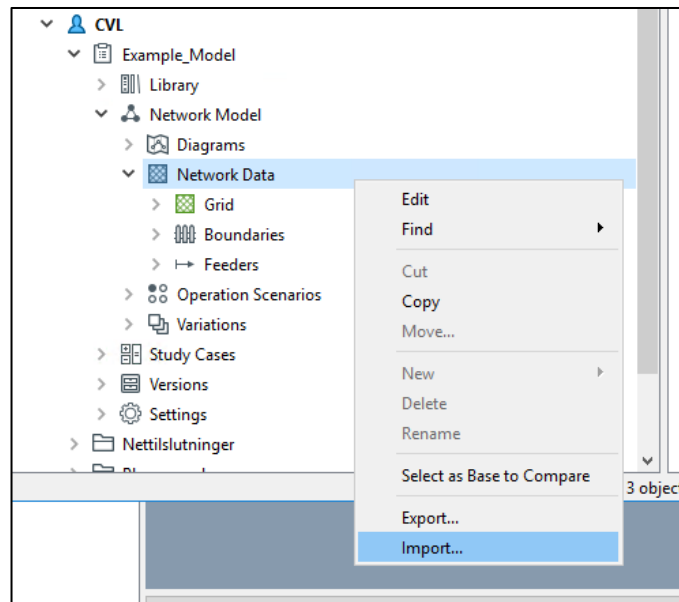
## 1.2 Fill in the model-specific information of the 'Input' sheet in the 'TestCases.xlsx' file

TestCases.xlsx • Germt																	
Søg (Alt+Q)																	
Filer Hjem Indsæt Tegning Sidelayout Formler Data Gennemse Vis Hjælp Acrobat 360°																	
Kommentarer Del																	
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Input model-specific information here:	Pn (MW)	Sb (MVA)	Vn_PoC (kV)	SCR	X/R	Qmode_Q	Qmode_V	Qmode_PF	ProjectName	PSCAD compiler	PSCAD TimeStep (us)	TotalCases	Volley				
1																	
2	500	500	400	10	10	0	1		2 Fredericia	Intel(R) Visual Fortran Compiler XE 15.0.5.280	2	32	8				
3																	
4						Note:				See 'PSCAD Compilers' sheet for guidance			The number of simulations that are run in parallel				
5						Input the specific number specified by the model developer for different reactive power control modes in columns 'Qmode_Q', 'Qmode_V' and 'Qmode_PF'											
6						Qmode_Q: Reactive power reference control											
7						Qmode_V: Voltage control											
8						Qmode_PF: Power factor control											
9																	

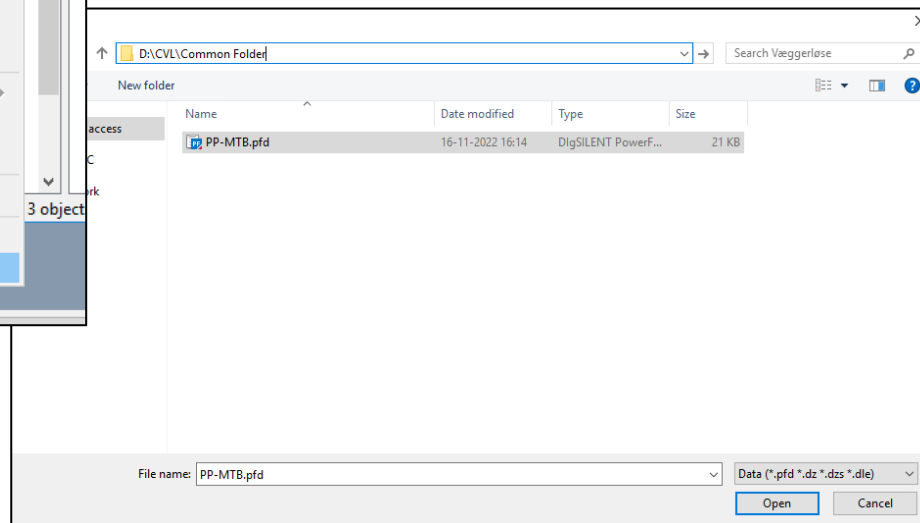
# 2. MODEL SETUP IN POWERFACTORY

2.1 Right-click the Network Data folder in your PowerFactory model and Import PP-MTB.pfd from the common folder.

2.2 Activate the PP-MTB grid

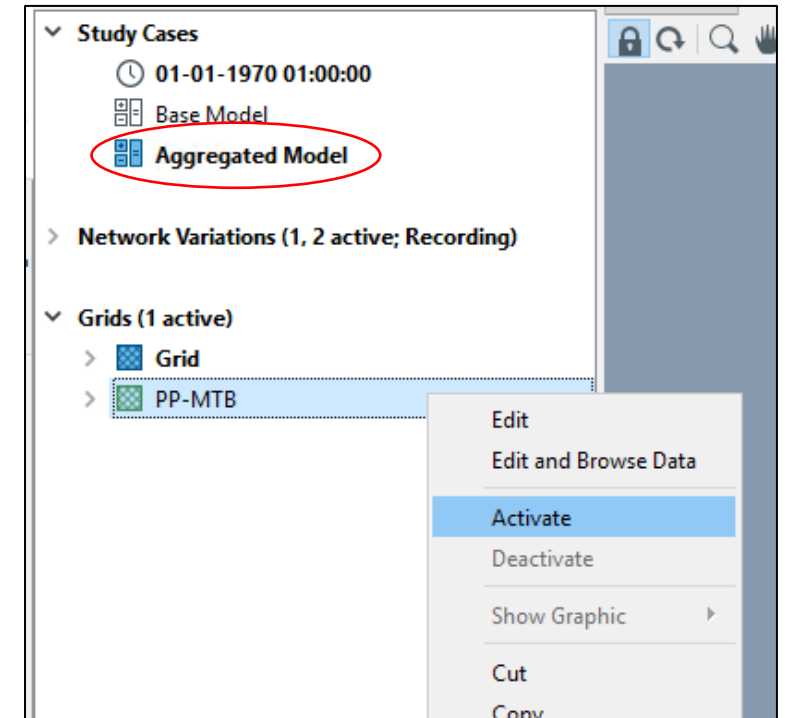


2.1



**NOTE:** The project must be inactive to import using this method

2.2



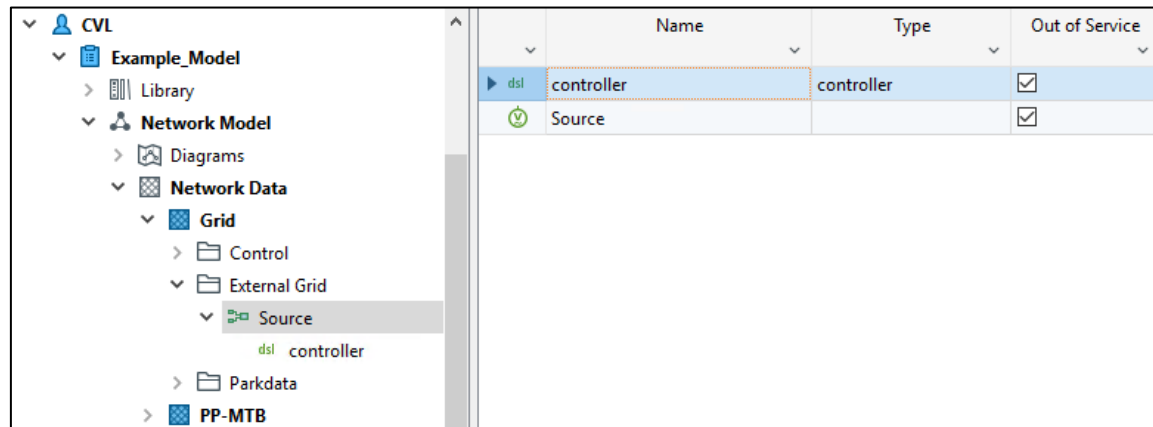
# 2. MODEL SETUP IN POWERFACTORY

## 2.3 Replace the external grid with PP-MTB

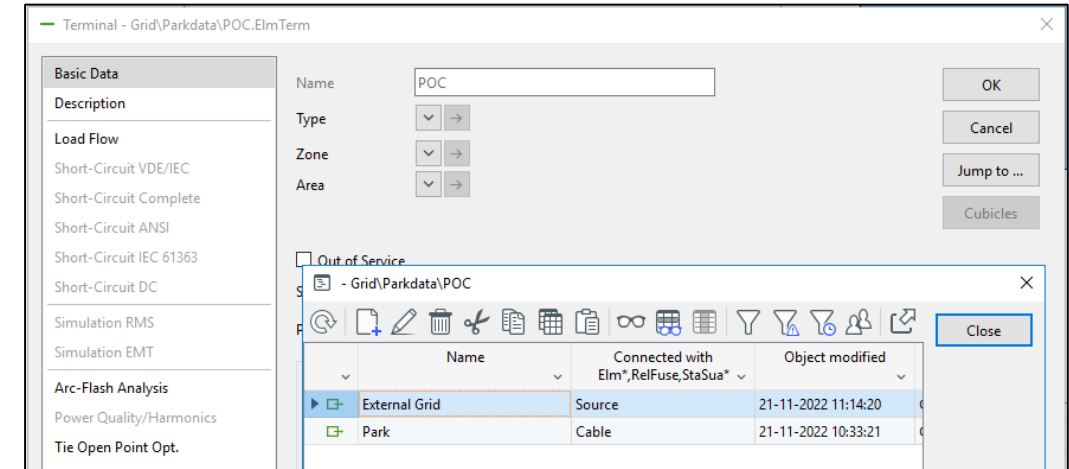
2.3.1 Deactivate the existing external grid, as well as any linked dynamic control used for the external grid, by leaving it Out of Service

2.3.2 Connect the cubicle at POC, formerly used for the external grid, to the element "meas.ElmSind"

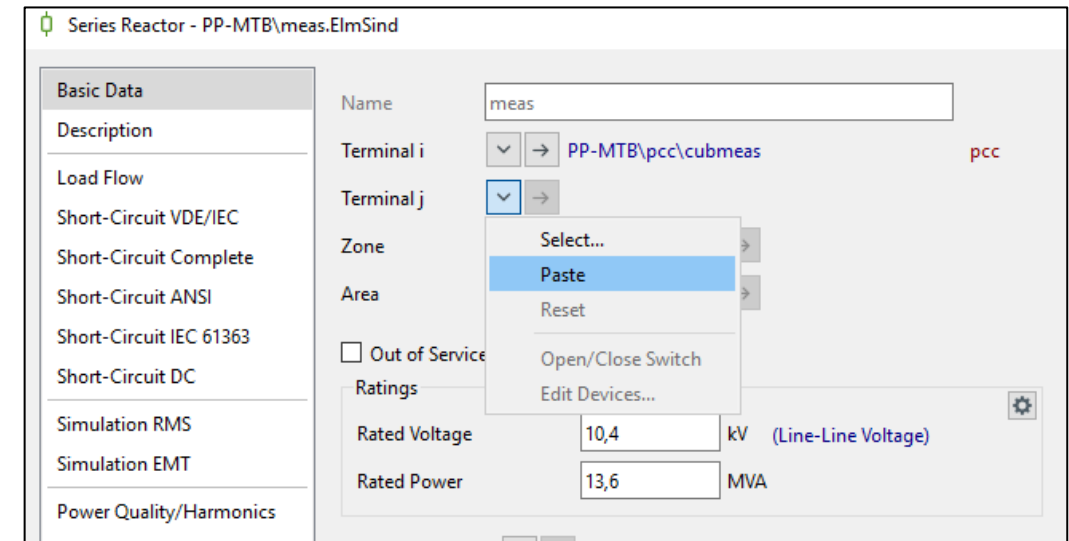
2.3.1: Select the elements and mark them as Out of Service



2.3.2 a): Copy the cubicle from the busbar used for the external grid (Ctrl+C)



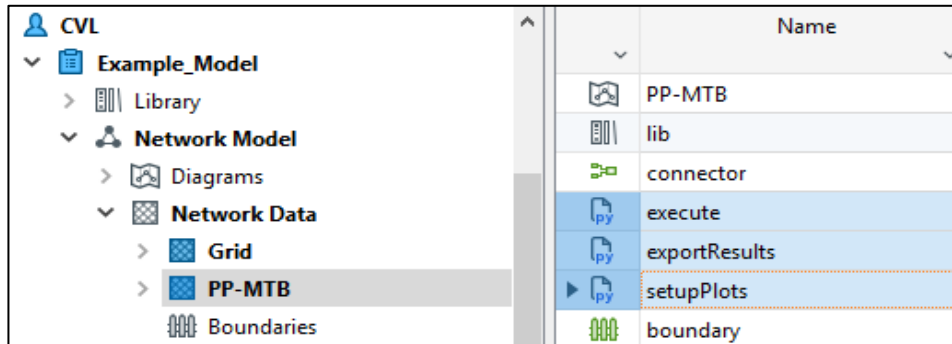
2.3.2 b): Paste the cubicle to the element "meas.ElmSind", Terminal j



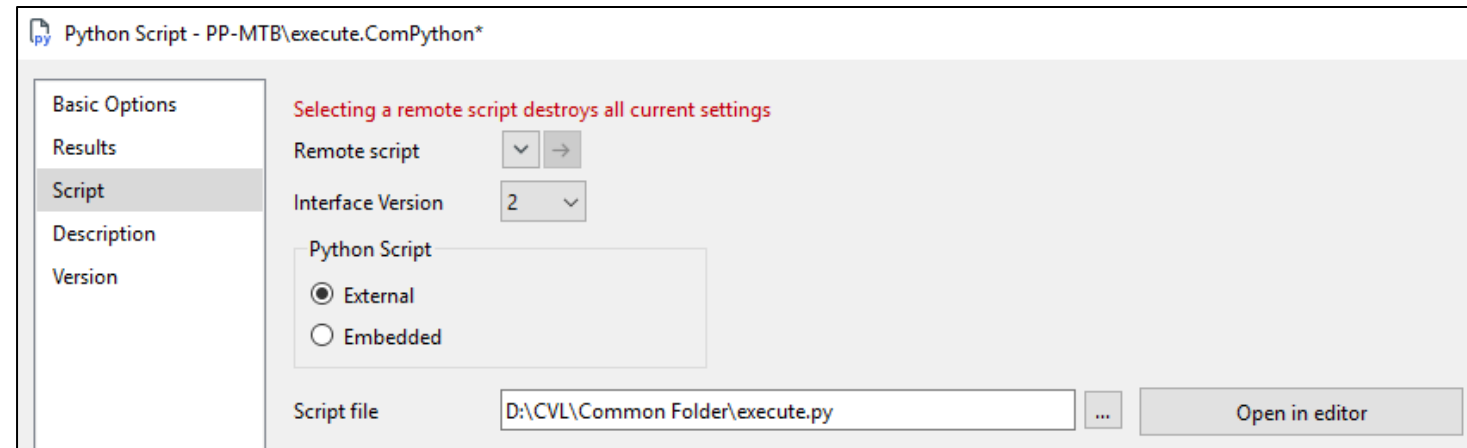
## 2. MODEL SETUP IN POWERFACTORY

### 2.4 Define a path in the scripts to point to the actual python scripts in the common folder with the matching name (e.g. execute.ComPython = execute.py)

When referring to the scripts, it is the following three under the PP-MTB folder:



The complete filepath must be defined in the field 'Script file' – Click the '...' button and navigate to execute.py you copied or extracted to a common folder earlier and repeat for the other two scripts as well



# 2. MODEL SETUP IN POWERFACTORY

## 2.5 Setup the execute.ComPython script Basic Options

Python Script - PP-MTB\execute.ComPython

Basic Options

Name: execute

Input parameters:

	Type	Name	Value	Unit	Description
1	string	file	D:\CVL\Common Folder\TestCases.xlsx		Matrix with balances and events
2	int	backup	1		1 = Backup project before setup
3	int	setup	1		1 = Setup cases
4	int	run	1		1 = Run simulations
5	int	plot	1		1 = Create/update plots
6	int	export	1		1 = Export results in figures and csv; otherwis...
7	int	eventPlot	1		1 = Plot eventplot (PQuf) for all cases, other...
8	int	faultPlot	1		1 = Plot faultplot (Idq) for all cases, otherwis...
9	int	phasePlot	1		1 = Plot phasePlot (Ulab) for all cases, othe...
10	int	nameByRank	1		1 = Use projectname and rank for export
11	int	variableStep	1		1 = Variable step size
12	int	asymSim	1		1 = Force asymmetrical simulations
13	int	autoFindGen	1		1 = Assign all generators to "sG" for dispatch
14	string	fpath	D:\CVL\Common Folder\Results		Folder for result files

External objects:

Name	Object	Description
1 Pctrl	Pctrl	Plant P setpoint inputblock
2 Qctrl	Qctrl	Plant Q setpoint inputblock
3 QUctrl	QUctrl	Plant Q(u) setpoint inputblock
4 QPFctrl	QPFctrl	Plant powerfactor setpoint inputblock
5 QmodeVar	Qmode_Variation	Q mode variation
6 QUmodeVar	QUmode_Variation	Q(u) mode variation
7 QPFmodeVar	QPFmode_Variation	Powerfactor mode variation

Annotations:

- Full filepath of the TestCases.xlsx
- Full filepath of Results folder defined earlier
- Reference both the P and Q control blocks and each Q control mode (If only one block controls all modes, reference the same block multiple times)
- You must create a variation object for each Q control mode, with the power plant control logic in that specific Q control mode and reference them here

Python Script - PP-MTB\execute.ComPython

Basic Options

Name: execute

Input parameters:

	Type	Name	Value	Unit	Description
9	int	phasePlot	1		1 = Plot phasePlot (Ulab) for all cases, otherwis...
10	int	nameByRank	1		1 = Use projectname and rank for export
11	int	variableStep	1		1 = Variable step size
12	int	asymSim	1		1 = Force asymmetrical simulations
13	int	autoFindGen	1		1 = Assign all generators to "sG" for dispatch
14	string	fpath	D:\CVL\Common Folder\Results		Folder for result files
15	string	PspSignal	P_Setpoint_Signal		Plant P setpoint signal/parameter name
16	double	PspScale	1		Scaling of the active power setpoint
17	string	QspSignal	Q_Setpoint_Signal		Plant Q setpoint signal/parameter name
18	double	QspScale	3		Scaling of the reactive power setpoint
19	string	QUspSignal	QU_Setpoint_Signal		Plant Q(u) setpoint signal/parameter name
20	double	QUspScale	1		Scaling of the voltage setpoint
21	string	QPFspSignal	QPF_Setpoint_Signal		Plant powerfactor setpoint signal/parameter na...
22	double	QPFspScale	1		Scaling of the powerfactor setpoint

Annotations:

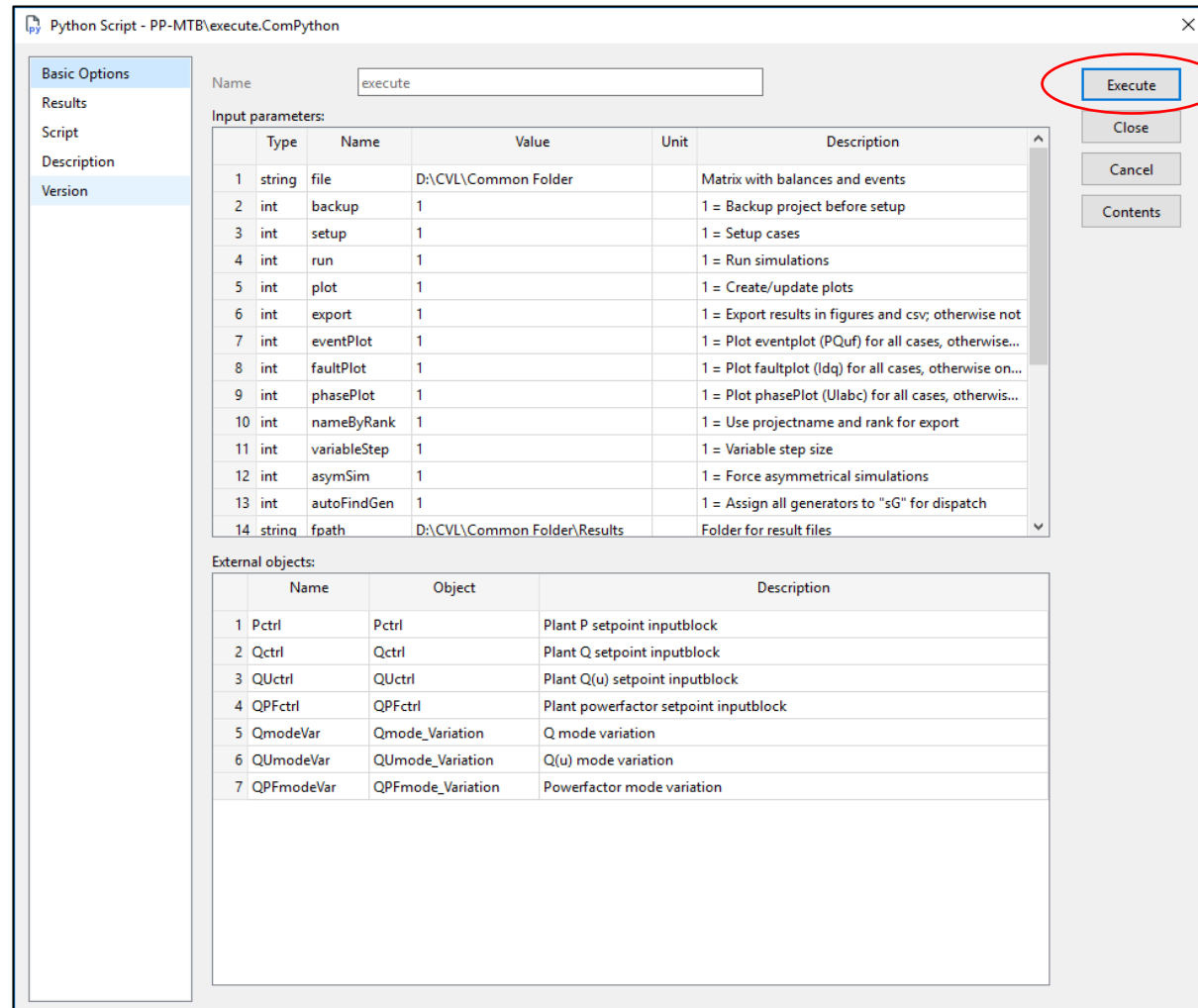
- Name of setpoint signal for P, Q, QU and QPF
- Scaling of P-, Q-, QU- and QPF-setpoint

**NOTE:** Any logic previously connected to the setpoint signals must also be Out of Service



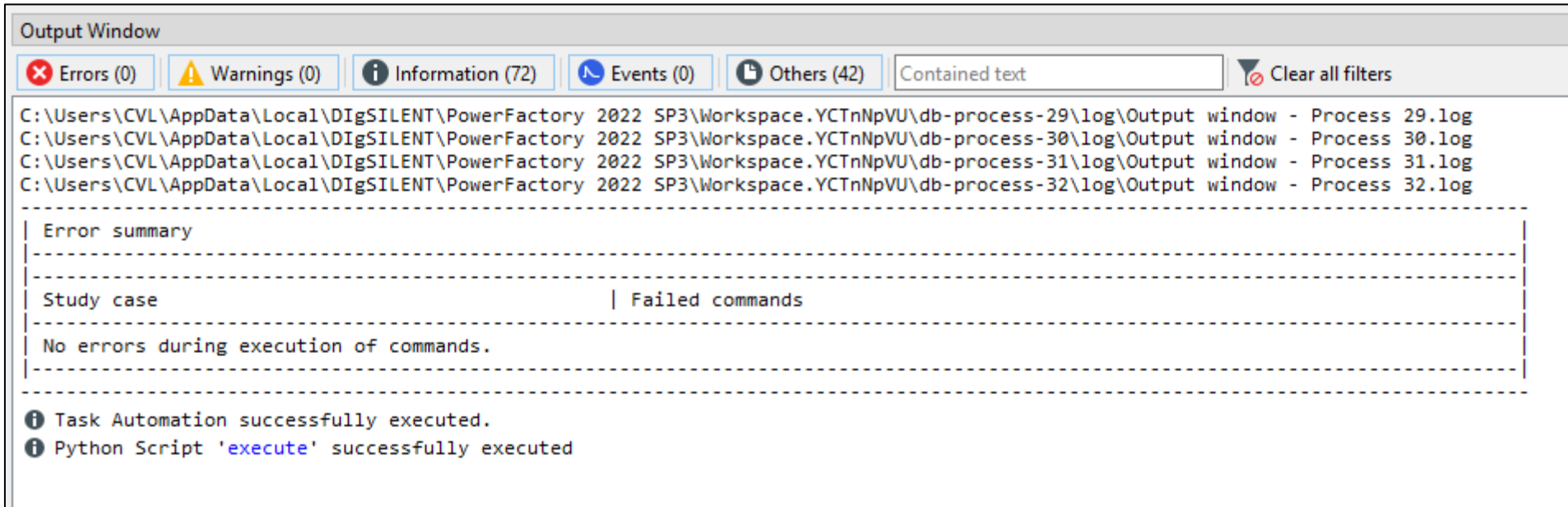
# 3. SCRIPT EXECUTION

3.1 Pressing Execute, the script does both simulation and plotting of results automatically.



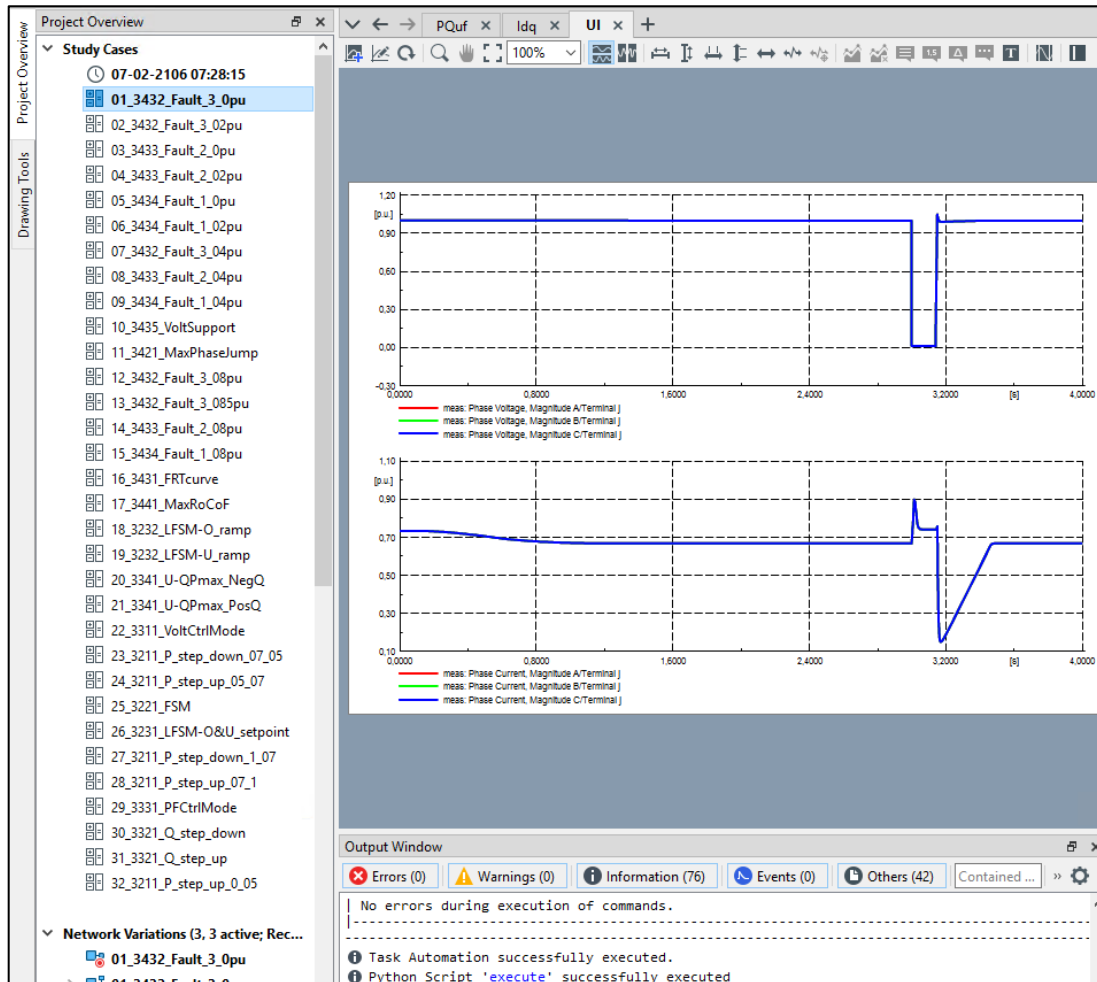
# 4. SCRIPT EXECUTION FINISHED

4.1 When the script is finished the output window should look like the snapshot below, with the text "Python Script 'execute' succesfully executed" at the bottom



# 4. SCRIPT EXECUTION FINISHED

## 4.2 Viewing results and outputs from the simulations



The screenshot shows the 'Results' folder with a list of files generated from the simulations. The files are organized by study case ID and type (CSV, PNG, or UI).

Name	Date modified	Type	Size
Example_Model_1.csv	24-11-2022 13:05	Microsoft Excel...	791 KB
Example_Model_1_Idq.png	24-11-2022 13:05	PNG File	62 KB
Example_Model_1_PQuf.png	24-11-2022 13:05	PNG File	58 KB
Example_Model_1_UI.png	24-11-2022 13:05	PNG File	56 KB
Example_Model_2.csv	24-11-2022 13:05	Microsoft Excel...	791 KB
Example_Model_2_Idq.png	24-11-2022 13:05	PNG File	63 KB
Example_Model_2_PQuf.png	24-11-2022 13:05	PNG File	59 KB
Example_Model_2_UI.png	24-11-2022 13:05	PNG File	57 KB
Example_Model_3.csv	24-11-2022 13:05	Microsoft Excel...	792 KB
Example_Model_3_Idq.png	24-11-2022 13:05	PNG File	64 KB
Example_Model_3_PQuf.png	24-11-2022 13:05	PNG File	59 KB
Example_Model_3_UI.png	24-11-2022 13:05	PNG File	59 KB
Example_Model_4.csv	24-11-2022 13:05	Microsoft Excel...	792 KB
Example_Model_4_Idq.png	24-11-2022 13:05	PNG File	64 KB

All results from the simulations are exported to the Results folder as seen above, each with a unique ID to match the study case, for a total of 32 if all cases are run.

All simulations can also be reviewed in PowerFactory through the 32 study cases created by the test bench.