Data formats

version 1.0 May 7, 2024

This document presents the conventions to name the files, the format of datasets, and the main transformations.

All the input files are given as CSV files.

1. One row containing column labels
2. Serie of Rows containing the data (consistent with column labels)
3. Each row contains a name, a zone and various values of variables. Zone can be any level of geographical partition (see below).

Columns that are not used may be skipped.

Description of geography

Each dataset is linked to a geographical area that may be partitioned. Different partitions may be used for dealing with different levels of constraints or computations.

Those partitions are described in files or sheets named ZP_ZonePartition, that follow the same rules:

The first column 'Level1' lists the lowest level partition (= smallest), **also called nodes**. The partition 'Level2' is a higher level partition, meaning that the regions in 'Level2' are larger than the regions in 'Level1', and each region in 'Level2' is composed of a list of regions in 'Level1'. 'Level3' partitions again are bigger than 'Level2'. At each level we may have different partitions obtained by regrouping regions differently, as exemplified in Figure 1 below.

- Level1 = identifier of the first level zone (e.g., datazone) - string
- Level2 = identifier of the second level zone (e.g., cluster) - string
- Level3Part1 = identifier of the third level zone (e.g., region), in a first partition – string
- Level3Part2 = identifier of the third level zone (e.g., region), in a second partition – string
- Etc.

A row with values L1, L2, L31, L32 … means that zone L1 belongs to second level zone L2, and this zone belongs to third level zones L31 and L32, …; Each L1 belongs to a unique L2; Each L2 belongs to a unique L3 for the first partition, and to a unique L3 – that can be different- for the second partition, etc…

The Figure 1 below illustrates this. In blue: Level1; in green: level2, in red and orange: level3 (partition1 and partition2), and in black level4. Here there are 7 zones in level1, aggregated in 4 level 2 clusters: BE, FR1, FR2 and ES. Each level1 zone belongs to a unique level2 cluster. At level3 there are 2 groupings: Level3part1 comprises the 4 clusters as North=BE+FR1, South=FR2+ES while Level3part2 comprises the same 4 clusters as BE=BE, FR=FR1+FR2 and ES=ES.

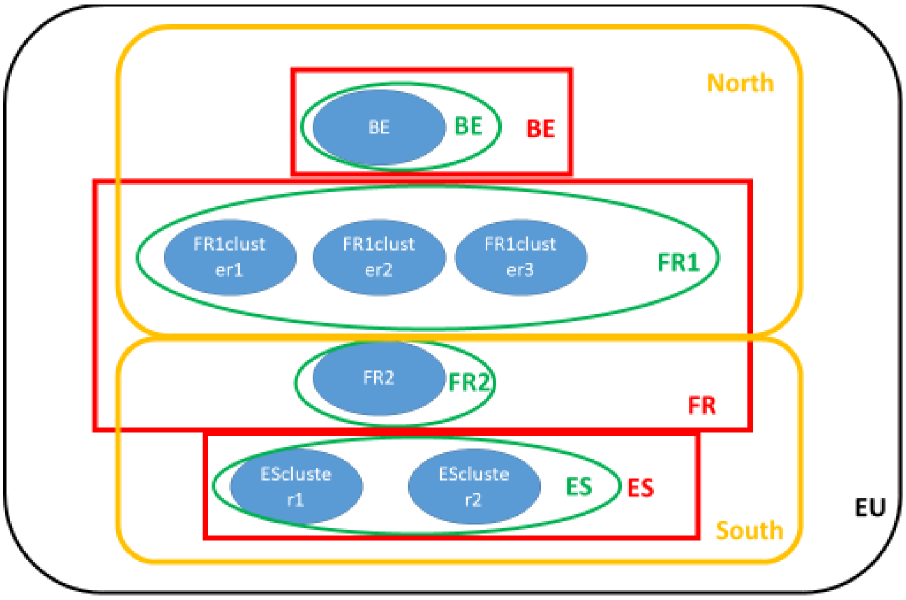

Figure 1: partitions

With this example, ZP_ZonePartition.csv would be:

Level1	Level2	Level 3 (grouping 1)	Level 3 (grouping 2)
BE	BE	BE	North
FR1cluster1	FR1	FR	North
FR1cluster2	FR1	FR	North
FR1cluster3	FR1	FR	North
FR2	FR2	FR	South
EScluster1	ES	ES	South
EScluster2	ES	ES	South

Table 1: example of ZP_ZonePartition file

Coupling Constraints

The constraints are described using YAML files, namely settingsCreateInputPlan4res.yml[1] found `p4r-env/scripts/python/plan4res-scripts` in after the installation procedure (see the installation guide). They are also available at https://github.com/openENTRANCE/plan4res-scripts.

The coupling constraints, ie the constraints linking different assets together, can be of the following categories:

- ActivePowerDemand: this is related to the equilibrium constraint Active power = Active Power demand at each node of the network. **The nodes are the lowest level of partitions described in ZP_ZonePartition** (so corresponding to Level1). The Active Power Demand is computed as the sum of different components that are listed in the 'SumOf' row of

[1] Two different file settings exist `plan4res/p4r-env/scripts/python/plan4res-scripts` : settingsCreateInputPlan4res_invest.yml, dedicated to the Capacity Extension Model (CEM), and settingsCreateInputPlan4res_simul.yml dedicated to the simulation (SIM). The only difference between the two files if that the first one contains `invest: yes` and the latter `invest: no`.

settingsCreateInputPlan4res.yml[2] corresponding line (this means that the power demand is computed as the sum of the electric heating demand, the electric cooling demand). In this example, the .yml file would look like:

```
CouplingConstraints: # MaxPower and variable cost to use for creating slack units (units for non served)
    # Coupling constraints which are not listed here will not be created in dataset
    # Coupling contraints can be : Demand, Primary, Secondary, Inertia, CO2
    # For the case of CO2, MaxPower and Cost are not needed, Budget can be defined (tons/CO2)
    # Partition defines the level at which the coupling constraint applies
    ActivePowerDemand : # mandatory - demand constraint
        Partition: 'Countries'
        MaxPower : 1500000
        Cost : 10000
        #SumOf: ['ElecHeating','ElecVehicle','OtherExclHeatTransp']
        SumOf: ['Total']
```

Figure 2: an example of the .yml file regarding the ActivePowerDemand constraint

Each component is computed as the product of an energy and a timeseries. By default, ActivePowerDemand = Final Energy|Electricity * TimeSeries (only one component, as seen in Figure 2). The TimeSeries is a profil applied to the total energy value, so the average on all scenarios of the sum on one year of the TimeSeries values is equal to 1. ActivePowerDemand may also be the sum of:

- ElecHeating = Final Energy|Electricity|Heat * ElecHeating Timeseries
- AirCondition= Final Energy|Electricity|Cooling * AirCondition Timeseries
- ElecVehicle= Final Energy|Electricity|Transportation * ElecVehicle Timeseries
- OtherExclHeatTransp= Final Energy|Electricity|Other (excl. Heat, Cooling, Transport) * OtherExclHeatTransp Timeseries

- <u>PrimaryDemand</u>: to represent the primary reserve requirement in each zone.

  Equivalent to Network|Electricity|Reserve|Requirement|Frequency Containment

- <u>SecondaryDemand</u>: to represent the primary reserve requirement in each zone.

  Equivalent to Network|Electricity|Reserve|Requirement| Automatic Frequency Restoration

- <u>InertiaDemand</u>: to represent the inertia requirement in each zone.

  Equivalent to Network|Electricity|Reserve|Requirement|Inertia

- <u>PollutantBudget</u>: to represent the max quantity of polluting emissions of a specific pollutant.

As for *ActivePowerDemand*, *PrimaryDemand*, *SecondaryDemand*, *InertiaDemand* and *PollutantBudget* may be the product how a value and a time series. If only one value is used, it is duplicated for all time steps.

Each coupling constraint applies to a partition (= one of the zones described in *ZP_ZonePartition,* see sections *Description of geography* and *1.2*), referenced by the constraint's *Partition* field in the .yml files. Different categories of coupling constraints may apply to different partitions or groupings, as described above (eg. *ActivePowerDemand* may apply to Level1 and *InertiaDemand* to a group in Level3).

[2] Two different file settings exist `plan4res/p4r-env/scripts/python/plan4res-scripts` : settingsCreateInputPlan4res_invest.yml, dedicated to the Capacity Extension Model (CEM), and settingsCreateInputPlan4res_simul.yml dedicated to the simulation (SIM). The only difference between the two files if that the first one contains `invest: yes` and the latter `invest: no`.

Input Data

This section describes the format of data used to feed the models used in Capacity Expansion Model, Seasonal Storage Valuation Model and European Unit Commitment model.

The dataset follows the common format described in (Krey, 2019)[3] (see also [openentrance/definitions at main · openENTRANCE/openentrance · GitHub]), and is organized as follows:

- A set of csv files representing the 'fixed' data:
  - *ZP_ZonePartition*: contains the description of the different geographical partitions (follows common format for partitions).
  - *ZV_ZoneValues*: contains data linked to zones (follows common format for data depending on zones).
  - *IN_Interconnections*: contains the description of the network (follows common format on interconnections) .
  - *TU_ThermalUnits*: contains the description of the thermal power plants (follows common format for data depending on zones).
  - *SS_SeasonalStorage*: contains the description of hydropower with seasonal storages and other long-term storages (follows common format for data depending on zones).
  - *STS_ShortTermStorage*: contains the description of other storages: pumped storage hydropower, batteries but also demand-response (it follows common format for data depending on zones).
  - *RES_RenewableUnits*: contains the description of PV, wind power and run-of-river (follows common format for data depending on zones).
- A set of CSV files containing scenarized time series: those files follow the common format for time series: the first row is the header, the first column contains the UCT timestamp (in any format readable by Python pandas) and the following columns are different scenarios of the current timeseries. One CSV file contains only one scenarized timeseries. Scenarios are identified by the corresponding column name in the header line (for example, past years, eg 1970, 1971, 1972…); deterministic timeseries can either be aggregated in a single csv file or be given each in a different csv file.

1.1 Foreword on the units of values

The units of values are not explicitly mentioned in the input files of plan4res *per se*. In the context of the OpenMod4Africa toolbox, they are defined in inputs which follow the IAMC format and which script `p4r-env/scripts/python/plan4res-scripts/CreateInputPlan4res.py` then uses to create the plan4res dataset:

<table>
<tr><th>Model</th><th>Scenario</th><th>Region</th><th>Variable</th><th>Unit</th><th>2018</th><th>2025</th><th>2030</th><th>2035</th><th>2040</th><th>2045</th><th>2050</th></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capacity|Electricity|Hydro|Run of River</td><td>GW</td><td>7,6627</td><td>7,6627</td><td>7,6627</td><td>7,6627</td><td>7,6627</td><td>7,6627</td><td>7,6627</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capacity|Electricity|Nuclear</td><td>GW</td><td></td><td></td><td></td><td></td><td></td><td>26,208</td><td>59,567</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capacity|Electricity|Oil|w/o CCS</td><td>GW</td><td>14,456</td><td>10,69</td><td>10,41</td><td>0,5857</td><td>0,1376</td><td>0,0039</td><td>0,002</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capacity|Electricity|Solar|PV|Utility</td><td>GW</td><td>20,384</td><td>57,374</td><td>72,403</td><td>84,514</td><td>96,33</td><td>128,44</td><td>128,44</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capacity|Electricity|Wind|Onshore</td><td>GW</td><td>10,92</td><td>10,204</td><td>7,8154</td><td>4,5544</td><td>12,649</td><td>20,203</td><td>20,203</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capital Cost|Electricity|Battery|Lithium-Ion</td><td>MEUR/GW</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capital Cost|Electricity|Battery|Redox Flow</td><td>MEUR/GW</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td></tr>
<tr><td>GENeSYS-MOD.jl</td><td>Gondor_globalLimit_364</td><td>Harad</td><td>Capital Cost|Electricity|Biomass|w/ CCS</td><td>MEUR/GW</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td><td>0,01</td></tr>
</table>

Table 2: example of IAMC input data used by script `CreateInputPlan4res.py` to create the plan4res files

[3] Available at https://doi.org/10.5281/zenodo.5521098

1.2 File ZP_ZonePartition.csv

Countries	Continent
DolAmroth	World
Gondor	World
Harad	World
Mordor	World
Rohan	World

Figure 3: 'ZP_ZonePartition.csv'

The file ZP_ZonePartition.csv describes the different zones, partitions and groupings that are used for dealing with different coupling constraints (see also section *Coupling Constraints*).

The mapping between the coupling constraints and the partitions is defined in settingsCreateInputPlan4res_*xxx*.yml files: in our example, *ActivePowerDemand* is using *Countries* of *ZP_ZonePartition*, as exemplified in Figure 2. This means that level1 (*Countries*) is the level used to generate the nodes for *ActivePowerDemand* in the model. In the example we have 5 nodes; we could define another coupling constraint, for example *InertiaDemand* on level2 (*Continent*), which in the example is “World”. This means that there is only one *InertiaDemand* constraint: the model will receive the constraint *‘Total inertia (i.e sum of the inertia in all level1 zone in partition P2) = inertia demand for partition P2’* for each partition *P2* in level2. In our example, the corresponding part of the settingsCreateInputPlan4res_*xxx*.yml files would be:

```
InertiaDemand : # optional - inertia constraint
    Partition: 'Continent'
    MaxPower : 10000
    Cost : 15000
    SumOf: ['Inertia']
```

Figure 4: an example of the .yml file regarding the IntertiaDemand constraint

1.3 File ZV_ZoneValues.csv

	A	B	C	D	E
1	Type	Zone	value	Profile_Timeserie	
2	Total	DolAmroth	63594926	Profile-EasternEurope.csv	
3	Total	Gondor	337117595	Profile-Germany.csv	
4	Total	Harad	204208526	Profile-Iberia.csv	
5	Total	Mordor	89650615.7	Profile-France.csv	
6	Total	Rohan	300524132	Profile-Italy.csv	
7	MaxActivePowerDemand	DolAmroth	1500000		
8	MaxActivePowerDemand	Gondor	1500000		
9	MaxActivePowerDemand	Harad	1500000		
10	MaxActivePowerDemand	Mordor	1500000		
11	MaxActivePowerDemand	Rohan	1500000		
12	CostActivePowerDemand	DolAmroth	10000		
13	CostActivePowerDemand	Gondor	10000		
14	CostActivePowerDemand	Harad	10000		
15	CostActivePowerDemand	Mordor	10000		
16	CostActivePowerDemand	Rohan	10000		

Figure 5: ZV_ZoneValue.csv

This file contains the values of all coupling constraints and may also optionally contain the costs associated (imbalance costs).

As mentioned in section *Coupling Constraints*, the Timeseries .csv files referenced by column *Profile_Timeserie* are (optionally scenarized) profiles applied to the corresponding value in column *value*:

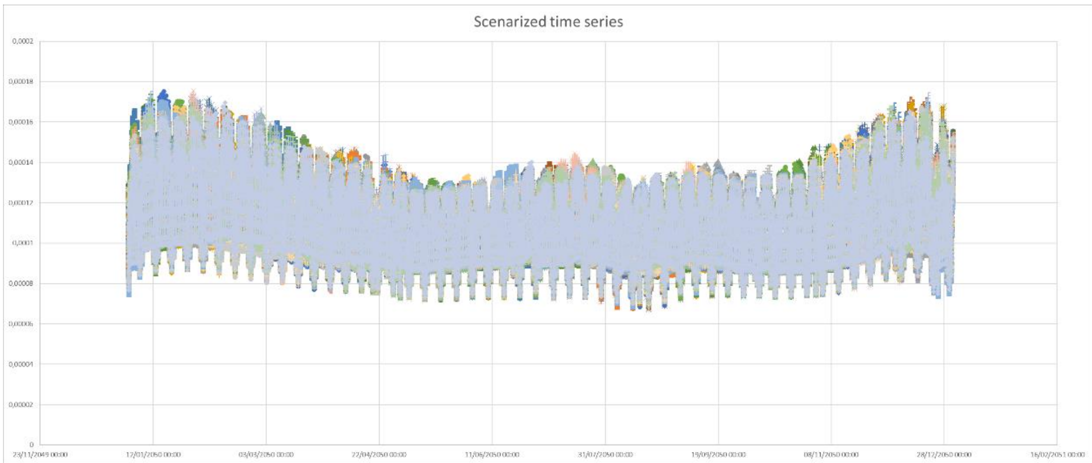

Figure 6: example of a scenarized profile contained in file Profile-Iberia.csv

1.4 File IN_Interconnections.csv

Name	StartLine	EndLine	MaxPowerFl	MinPowerFlow
DolAmroth>(	DolAmroth	Gondor	2000	-2200
DolAmroth>I	DolAmroth	Harad	71	-100
Gondor>Har.	Gondor	Harad	866	-1160
Gondor>Moı	Gondor	Mordor	100	-225
Gondor>Roh	Gondor	Rohan	1675	-1800

Figure 7: IN_Interconnections.csv

This file describes the characteristics of the lines that are linking the nodes of the network, where the nodes the partition linked to the *ActivePowerDemand* constraints in the .yml setting files, and defined in file *ZP_ZonePartition.csv* (see section *File ZP_ZonePartition*.csv for some examples):

- Name: name of the line (used for processing results)
- StartLine and EndLine must be nodes defined in the partition linked to the ActivePowerDemand.
- MaxPowerFlow and MinPowerFlow are the bounds on the flows for this line (one way or the other); note that *MinPowerFlow* is a negative value: it defines the maximal flow from End to Start, whereas *MaxPowerFlow* defines the maximal flow from Start to End.

1.5 File TU_ThermalUnits.csv

Zone	Name	NumberUnit:	MaxPower	VariableCost	Capacity	CO2Rate
DolAmroth	Biomass|w/c	1	557.44	36.9957988	557.44	0
Harad	Biomass|w/c	1	1924	36.9557811	1924	0
Mordor	Biomass|w/c	1	478.247304	32.9898836	478.247304	0
Rohan	Biomass|w/c	1	757.95	33.3486752	757.95	0
Gondor	Biomass|w/c	1	7837.5	36.5680302	7837.5	0
DolAmroth	Coal|Hard cc	1	11.923392	13.0525392	11.923392	0.7512
Harad	Coal|Hard cc	1	25364.3377	13.0525392	25364.3377	0.7512
Mordor	Coal|Hard cc	1	5010.114	13.0525392	5010.114	0.7512
Rohan	Coal|Hard cc	1	1497.97541	13.0525392	1497.97541	0.7512
Gondor	Coal|Hard cc	1	5908.1	13.0525392	5908.1	0.7512
Mordor	Coal|Lignite|	1	11714.5921	9.72193608	11714.5921	0.94971429
Gondor	Coal|Lignite|	1	17348.4217	9.72193608	17348.4217	0.94971429
Harad	Gas|CCGT|w	1	6020.30214	108.741681	6020.30214	0.34634483
Mordor	Gas|CCGT|w	1	5004.05802	107.131031	5004.05802	0.34634483
Rohan	Gas|CCGT|w	1	32745.0664	107.276745	32745.0664	0.34634483
Gondor	Gas|CCGT|w	1	4970.34932	108.584206	4970.34932	0.34634483
Rohan	Gas|OCGT|w	1	5266.11133	151.147959	5266.11133	0.52863158
Mordor	Nuclear	1	1223.72352	6.8088464	1223.72352	0

Figure 8: TU_ThermalUnits.csv

This sheet gives the characteristics of all thermal power plants.

It contains the following data:

- Name
- Zone: **it must be the same level as the partition used to define the *ActivePowerDemand*** (= node)
- NumberUnits: number of units of the same type at the same location
- MaxPower

- MinPower (optional, 0 by default)[4]
- Pauxiliary: Power taken from the system when off. Optional (0 by default)
- VariableCost: proportional cost (Optional, 0 by default)
- Quadterm: quadratic cost (optional, 0 by default)
- StartUpCost (optional, 0 by default)
- MinUpTime: minimum duration when the plant is on (optional, 1 hour by default)
- MinDownTime: minimum duration when the plant is off (optional, 1 by default)
- Inertia: max inertia that can be provided by a unit in MWs/MWA; (optional, 0 by default)
- PrimaryRho: this parameter, multiplied by Maxpower, gives the maximum share of the active power that can be used as primary reserve (optional, 0 by default)
- SecondaryRho: this parameter, multiplied by MaxPower, gives the maximum share of the active power that can be used as secondary reserve (optional, 0 by default)
- DeltaRampDown: maximum gradient when the power is decreased from one time step to the other (optional, MaxPower by default)
- DeltaRampUp: maximum gradient when the power is increased from one time step to the other (optional, MaxPower by default)
- CO2: emission rate in tons CO2 per MWh (optional, default 0)

1.6 File SS_SeasonalStorage.csv

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Name	Zone	HydroSystem	NumberUnit:	MaxPower	MinPower	MaxVolume	MinVolume	Inflows	InflowsProfil	InitialVolume	TurbineEffici	PumpingEfficiency
2	Hydro|Reser	DolAmroth	0	1	15164.2333	0	30328466.7	0	39887350.8	EDF__Inflow	18197080	1	0
3	Hydro|Reser	Harad	0	1	9578.4	0	19156800	0	27689238.7	EDF__Inflow	9578400	1	0
4	Hydro|Reser	Mordor	0	1	2606.19	0	5212380	0	7533974.05	EDF__Inflow	2606190	1	0
5	Hydro|Reser	Rohan	0	1	21324.3044	0	42648608.9	0	61644299.2	EDF__Inflow	34118887.1	1	0
6	Hydro|Reser	Gondor	0	1	30545.9798	0	61091959.6	0	84129077.8	EDF__Inflow	42764371.7	1	0

Figure 9: SS_SeasonalStorage.csv

This sheet gives the characteristics of all seasonal storages.

It contains the following:

- Name
- Zone: **it must be the same level as the partition used to define the *ActivePowerDemand*** (= node)
- NumberUnits: number of units of the same type at the same location
- MaxPower
- MinPower[5] (optional, 0 by default).
- DeltaRampDown: maximum gradient when the power is decreased from one time step to the other (optional, MaxPower by default).
- DeltaRampUp: maximum gradient when the power is increased from one time step to the other (optional, MaxPower by default).
- MaxVolume
- MinVolume (optional, 0 by default)
- TurbineEfficiency: (optional, 1 by default). This value, multiplied by the flow, gives the generated power.

[4] Note that data consitency is ensured: of MinPower > MaxPower at some point, then MinPower = MaxPower. For example, this is useful if MaxPower = 0 during a period to model a maintenance or an outage. For thermal units, MinPower constraint applies only if the unit is running.

[5] Note that for hydro storages, MinPower forces the plant to run: it aims at representing hydro operational constraints such as minimal river flows. It is different from the MinPower constraint applied to thermal units.

- PumpingEfficiency: (optional, 1 by default). This value, multiplied by the flow, gives the generated power.
- Inflows: (optional, 0 by default). Inflows to the upstream reservoir (energy per year).
- Inflows profile: (optional): time series profile for Inflows. Multiplied by the Inflows in energy, gives the inflows time series.
- InitialVolume: (optional, 0 by default). Initial Volume of the upstream reservoir
- Inertia: max inertia that can be provided by a unit (optional, 0 by default)
- PrimaryRho: (%) this parameter, multiplied by MaxPower, gives the maximum primary reserve that can be provided by a unit (optional, 0 by default).
- SecondaryRho: (%) this parameter, multiplied by MaxPower, gives the maximum secondary reserve that can be provided by a unit (optional, 0 by default).
- WaterValues: optional; used to specify water values as input data from a file, if they not computed by the current run (for example, coming from a previous SSV run). Contains the name of the file/sheet where the water values are stored. When using the simulation mode with one Bellman values (BV) file for all units, this file is only given in the first line; when using the simulation mode with 1 BV per unit, this file is given at each line; in optimization mode (= when running the SSV), this is not required (output of the model).

1.7 File STS_ShortTermStorage.csv

This file is used for:

- Pumped hydro storage
- Batteries
- Other flexibilities such as residential flexibilities or electric vehicles (EV), which are modelled as short-term storages.

A	B	C	D	E	F	G	H	I
Name	Zone	NumberUnits	MaxPower	MaxVolume	TurbineEfficiency	PumpingEfficiency	MinPower	MinVolume
Hydro|Pump	DolAmroth	1	3302	330200	0.866	0.866	-3302	0
Hydro|Pump	Harad	1	5920.72	592072	0.866	0.866	-5920.72	0
Hydro|Pump	Mordor	1	1540.17	154017	0.866	0.866	-1540.17	0
Hydro|Pump	Rohan	1	4196.16	419616	0.866	0.866	-4196.16	0
Hydro|Pump	Gondor	1	6414.21	641421	0.866	0.866	-6414.21	0

Figure 10: STS_ShortTermStorage.csv

It contains the following data:

- Name
- Zone: **it must be the same level as the partition used to define the *ActivePowerDemand*** (= node)
- NumberUnits: number of units of the same type at the same location
- MaxPower; can be a (deterministic) timeseries, in this case: name of the time serie
- MinPower (optional, 0 by default); can be a (deterministic) timeseries, in this case: name of the time serie
- Cost: proportional cost (optional, 0 by default).
- MaxVolume: can be a (deterministic) timeseries, in this case: name of the time serie
- MinVolume (optional, 0 by default): can be a (deterministic) timeseries, in this case: name of the time series.
- TurbineEfficiency: (optional, 1 by default). This value, multiplied by the flow, gives the generated power.

- PumpingEfficiency: (optional, 1 by default). This value, multiplied by the flow, gives the generated power.
- Inflows: (optional, 0 by default). Inflows to the upstream reservoir.
- InitialVolume: (optional, 0 by default)
- VolumeLevelTarget: (optional) used to force the optimization to reach this volume at the end of each stage. If there is a VolumeLevelTarget, the minimum volume constraint is replaced by this value at the first and last time-steps of each stage.
- MaxPrimaryPower: maximum primary reserve that can be provided by a unit (optional, 0 by default).
- MaxSecondaryPower: maximum secondary reserve that can be provided by a unit (optional, 0 by default).

1.8 File RES_RenewableUnits.csv

Name	Zone	NumberUnits	MaxPower	MinPower	MaxPowerProfile	Energy	Gamma	Capacity
Solar|PV|Uti	DolAmroth	1	19180.0149	0	EDF__PV-LoadFactor-PresentClimate-DE__13082019__13082019__v1.csv	31726242.9	1	19180.0149
Solar|PV|Uti	Harad	1	72402.5793	0	EDF__PV-LoadFactor-PresentClimate-CH__13082019__13082019__v1.csv	138044713	1	72402.5793
Solar|PV|Uti	Mordor	1	44630.2396	0	EDF__PV-LoadFactor-PresentClimate-FR_C8__13082019__13082019__v1.csv	101310192	1	44630.2396
Solar|PV|Uti	Rohan	1	148071.644	0	EDF__PV-LoadFactor-PresentClimate-ES__13082019__13082019__v1.csv	213346546	1	148071.644
Solar|PV|Uti	Gondor	1	117719.929	0	EDF__PV-LoadFactor-PresentClimate-IT__13082019__13082019__v1.csv	192919306	1	117719.929
Wind|Onshc	DolAmroth	1	6203.20125	0	EDF__WindOnshore-LoadFactor-PresentClimate-DE__13082019__13082019__v1.	13066984.3	1	6203.20125
Wind|Onshc	Harad	1	7815.43343	0	EDF__WindOnshore-LoadFactor-PresentClimate-CH__13082019__13082019__v1.	9038343.75	1	7815.43343
Wind|Onshc	Mordor	1	26167.8907	0	EDF__WindOnshore-LoadFactor-PresentClimate-FR__13082019__13082019__v1.	48311039.6	1	26167.8907
Wind|Onshc	Rohan	1	15895.4951	0	EDF__WindOnshore-LoadFactor-PresentClimate-ES__13082019__13082019__v1.	27962349.5	1	15895.4951
Wind|Onshc	Gondor	1	20532.3992	0	EDF__WindOnshore-LoadFactor-PresentClimate-IT__13082019__13082019__v1.c	43716595	1	20532.3992
Wind|Offshc	Rohan	1	39930.7113	0	EDF__WindOffshore-LoadFactor-PresentClimate-UK__25112019__25112019__v1	144300078	1	39930.7113
Wind|Offshc	Gondor	1	60211.2438	0	EDF__WindOffshore-LoadFactor-PresentClimate-UK__25112019__25112019__v1	227915199	1	60211.2438
Hydro|Run c	DolAmroth	1	22602012.5	0	EDF__RunOfRiver-HourlyCoefficient-PresentClimate-DE__18092019__18092019_	22602012.5	1	6254.56
Hydro|Run c	Harad	1	19832488.4	0	EDF__RunOfRiver-HourlyCoefficient-PresentClimate-CH__18092019__18092019_	19832488.4	1	7662.72
Hydro|Run c	Mordor	1	1836924.65	0	EDF__RunOfRiver-HourlyCoefficient-PresentClimate-FR__18092019__18092019_	1836924.65	1	919.200665
Hydro|Run c	Rohan	1	32609363.2	0	EDF__RunOfRiver-HourlyCoefficient-PresentClimate-ES__18092019__18092019_	32609363.2	1	9602.25
Hydro|Run c	Gondor	1	32192470.1	0	EDF__RunOfRiver-HourlyCoefficient-PresentClimate-IT__18092019__18092019_	32192470.1	1	7848.28

Figure 11: RES_renewableUnits.csv

This sheet gives the characteristics of all renewable units: windpower, PV power and run-of-river.

It contains the following data:

- Name
- Zone: **it must be the same level as the partition used to define the *ActivePowerDemand*** (= node)
- NumberUnits: number of units of the same type at the same location
- MaxPower (capacity for PV and WindPower; yearly average energy for run-of-river) – This is due to the fact that the available load factor timeseries for Wind and PV Power and for run-of-river were computed with different methods. The maximum potential production for Wind or PV is equal to the capacity multiplied by the load factor timeseries while the maximum potential production for a run-of-river is equal to the average yearly energy multiplied by the timeseries.)
- MinPower (optional, 0 by default)
- Inertia: max inertia that can be provided by a unit; (optional, 0 by default)
- Gamma (optional, 1 by default): this parameter is used by the model to determine the maximum available primary and secondary reserve. It is used to take into account the fact that some renewable units, due to the uncertainty in their maximum capacity, may not be able to provide reserves at full capacity.
- Capacity: only when this data was created by the plan4res creation script; Capacity of the technology in MW. Equal to MaxPower when NumberUnits=1 except for units whose maximum power is computed by multiplying an energy with the load factor time serie.
- Energy: only when this data was created by the plan4res creation script. Secondary Energy (MWh) generated by the technology in the scenario used for generation.

1.9 Additional columns for Investments (CEM)

To run the Capacity Expansion Model (CEM), 4 columns need to be added to the sheets corresponding to assets in which one wants to invest (see below):

- MaxAddedCapacity: this is the maximum capacity that may be added.
- MaxRetCapacity: this is the maximum capacity that may be taken out.
- InvestmentCost: this is the cost for investing into 1 unit of the given capacity in the given zone. Note that these are yearly costs, computed as Capital Cost / LifeTime (in years) + Fixed Cost.
- DecommissionCost: this is the cost for taking out 1 unit of the given capacity in the given zone.

Costs are not discounted.

These columns may be added to the following files:

- IN_Interconnections.csv
- TU_ThermalUnits.csv
- STS_ShortTermStorage.csv
- RES_RenewableUnits.csv

Note that investment in new seasonal storages is not available at present.

2 Plan4res outputs: results of SMS++

The results of SMS++ are detailed in the following subsections. SMS++ is the modelling and optimization library that is integrated into plan4res to formulate the underlying optimization problems and solve them using a solver such as CPLEX or SCIP for example. Plan4res also uses the Stochastic Optimization library (StOpt) for solving stochastic optimization problems.

See the installation guide for more information.

2.1 Results of SSV

The SSV uses SMS++ and StOpt to run a Stochastic Dynamic Dual Programming (SDDP) algorithm to compute Bellman values for all the seasonal storages.

Note: Bellman values represent the *cost-to-go functions* = the expected economic value associated to the various levels of the seasonal storages at each time stages. They are usually represented as sets of hyperplanes called *cuts*.

The results are:

- When the convergence criteria of the SDDP algorithm is met[6]: *
    - BellmanValuesOUT.csv: redundant cuts have been pruned out.
    - BellmanValuesAllOUT.csv: contains all the cuts found by the algorithm.

[6] See the plan4res run guide for more information.

- In any case, ie even without convergence: cuts.txt, which contains all the cuts already found by the SDDP algorithm. This is useful to help reach convergence since it is possible to launch the SSV again using the cuts from a previous run stored in cuts.txt as a hot start.

2.2 Results of SIM

The simulation produces the following results for each scenario:

- ActiveDemandOUT.csv: 1 column per zone with the demand for the zone.
- ActivePowerOUT.csv: generation schedules. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part).
- PrimaryOUT.csv: primary reserve schedules. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part)..
- SecondaryOUT.csv: secondary reserve schedules. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part)
- InertiaOUT.csv : inertia provided by each unit. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part).
- VolumeOUT.csv: volumes of each storage (seasonal storages and short-term storages).
- FlowsOUT.csv: flows between zones. 1 column per transmission line.
- MarginalCostActivePowerDemandOUT.csv: marginal costs of the demand constraint. 1 column per zone (in this case, the nodes on which the *ActivePowerDemand* constraints apply).
- MarginalCostPrimaryOUT.csv: marginal costs of the primary reserve constraint. 1 column per zone (at the partition level of the *PrimaryDemand* constraints).
- MarginalCostSecondaryOUT.csv: marginal costs of the secondary reserve constraint. 1 column per zone (at the partition level of the *SecondaryDemand* constraints).
- MarginalCostInertiaOUT.csv: marginal costs of the Inertia constraint (at the partition level of the *InertiaDemand* constraints). 1 column per zone.
- MarginalCostFlowsOUT.csv: marginal costs of the lines. 1 column per line.
- MaxPowerOUT.csv: maximum available power. 1 column per technology.
- DemandOUT.csv: Demand. 1 column per zone.

All files share the same format: a header containing the names of the series, an index with the time steps.

2.3 Plan4res outputs: results of PostTreatPlan4res.py

2.3.1 Installed Capacity

InstalledCapacity.csv and AggrInstalledCapacity.csv: installed capacities (aggregated installed capacity, with aggregations such as defined in `settingsPostTreatPlan4res.py`) in the different available technologies per zone. 1 column per technology, 1 row per zone.

	Geothermal	Hydro|Run c	Nuclear	Coal|w/o CC	Wind|Offsh	Wind|Onsh	Solar|PV	Biomass|w/	Biomass|w/	Hydrogen|O	Gas|CCGT|w	Gas|CCGT|w	Oil|w/o CCS	Reservoir	Pumped Storage
France	0	10325	47037.0434	0	50840.0435	82643.272	148447.229	0	660	0	0	1879.16385	68.50722	13422.5	4512
Germany	0	7134.8	0	9586.78646	83829.4524	114867.893	242816.939	16812.0883	6327	4527.18441	0	1977.76184	83.8	9275.24	5831.1
Italy	0	7368	0	1705.22558	0	47754.235	130262.048	0	691.650085	7280.68425	0	2415.9806	1.947536	9578.4	14922.2681
Switzerland	0	3854	0	0	0	4559.2662	25284.0346	0	102.619516	867.435571	0	0	0	10390	0
EasternEurop	0	8498.48961	19012.6324	4298.16565	0	77796.9701	135757.304	7964.83126	1380.00558	17989.9229	16711.6989	1938.1211	79.036044	12637.665	11923.6886
Benelux	0	168.35881	0	930.53843	27571.9364	23477.9272	81047.1945	6969.43405	762.342245	4340.58573	4442.92019	4966.5955	0	256.316667	3281.02129
Iberia	0	19635.9413	0	0	0	194148.393	250872.819	4621.81626	459.347161	0	0	2679.72441	0	27177.9465	5977.99842
Britain	0	722.093437	0	1551.42732	176460.491	100047.174	132642.325	7489.93595	1634.17446	0	0	4493.21799	19.362935	1768	4869.42198
Balkans	1170	15132.0372	8521.38833	9601.51157	23195.6116	34057.5587	139547.493	0	177.837147	0	0	2572.64982	198	23939.6956	3107.58612
Baltics	0	1668.42454	0	0	6576.11002	1346.91798	12531.0172	320.721047	186.756955	0	0	621.434573	0	2240.16	900
Scandinavia	0	19770.3978	43.808304	0	11570.9374	95189.9336	70921.2225	1969.84435	1057.51349	0	0	79.615644	270.154414	42081	1730

Figure 12: InstalledCapacity.csv

2.3.2 Generation

Generation.csv and AggrGeneration.csv: total generation (aggregated generation, with aggregations such as defined in `settingsPostTreatPlan4res.py`) from the different available technologies per zone. 1 column per technology, 1 row per zone.

	Hydro	Geothermal	WindPower	PV	Biomass	Nuclear	Hydrogen	Coal	Gaz	Battery	Oil	Non Served
France	120568847	0	287617484	150482515	0	287384845	0	0	0	-1648018.88	0	0
Germany	47197794.3	0	466410144	138020686	206915191	0	5320505.4	0	217304.966	-16356865.6	0	0
Italy	73743874.7	0	34957015	133427172	8102071.33	0	91583830.4	0	32279.2293	-16598953.3	0	0
Switzerland	50951022.6	0	1477770.7	21123258.1	714789.719	0	2148116.69	0	0	0	0	0
EasternEurop	84145229.2	0	217061316	91833905.7	243336.745	216962878	76422.7969	0	0	-267710.937	0	0
Benelux	189920.775	0	141644805	55295622.3	13518823	0	64173543.4	0	13955643.8	-2140883.88	0	0
Iberia	180805663	0	198896982	143119773	43827319.7	0	0	0	20815442.6	-204619.364	0	712620.033
Britain	2740891.89	0	549537627	62271837.7	16549265.3	0	0	0	14795329.1	-6222047.55	0	0
Balkans	134984520	77272563.4	83200528.5	81900461.8	28365.2631	54020159.3	0	130371.037	714882.155	-3359441.48	2311.37379	0
Baltics	12107630.1	0	17486942.2	6238427.46	3387217	0	0	0	5528985.5	0	0	5542697.36
Scandinavia	248290300	0	230230093	39702643.5	7150297.67	1649279.83	0	0	138974.023	-748834.085	34420.6142	325752.246

Figure 13: AggrGeneration.csv

Generation-ZONE.csv and ***AggrGeneration-ZONE.csv***: average generation (aggregated generation, with aggregations such as defined in `settingsPostTreatPlan4res.py`) from the different available technologies per zone. 1 file per zone, 1 column per technology, 1 row per time step.

	Hydro	Geothermal	WindPower	PV	Biomass	Nuclear	Hydrogen	Coal	Gaz	Battery	Oil	Non Served
2050-07-02 0	24195.4862	0	24951.7027	0	0	29436.9746	0	0	0	0	0	0
2050-07-02 0	24211.1757	0	24268.5717	0	0	23382.5813	0	0	0	0	0	0
2050-07-02 0	23917.7569	0	23099.7627	0	0	17123.7441	0	0	0	0	0	0
2050-07-02 0	23600.885	0	22407.0559	0	0	15345.5185	0	0	0	0	0	0
2050-07-02 0	23544.0451	0	21705.4391	0	0	14254.2005	0	0	0	-1.12768727	0	0
2050-07-02 0	23171.4812	0	20999.688	481.450471	0	11387.2616	0	0	0	-114.980598	0	0
2050-07-02 0	22158.059	0	20037.9383	5789.44191	0	6963.85983	0	0	0	-216.559966	0	0
2050-07-02 0	16354.4361	0	18850.0071	16734.4159	0	1038.06033	0	0	0	-1330.34312	0	0
2050-07-02 0	10361.1615	0	17358.2596	31534.7191	0	445.81691	0	0	0	-2785.74124	0	0
2050-07-02 0	5225.88184	0	14859.6548	46283.144	0	214.116341	0	0	0	-6615.77903	0	0
2050-07-02 1	3804.58107	0	13583.7764	49298.4458	0	128.142529	0	0	0	-6967.88798	0	0
2050-07-02 1	3382.15089	0	9909.77199	55267.6422	0	100.977214	0	0	0	-7598.57274	0	0
2050-07-02 1	1845.99949	0	7674.4139	61184.5746	0	100.977214	0	0	0	-9811.47378	0	0
2050-07-02 1	3245.04012	0	5391.96823	58391.0602	0	100.977214	0	0	0	-9924.06661	0	0

Figure 14: AggrGeneration-France.csv

Generation-ZONE-$i.csv: generation from the different available technologies in the zone for scenario $i. 1 file per zone and scenario, 1 column per technology, 1 row per time step.

Slack-ZONE.csv: electricity not served per zone. 1 file per zone. 1 column per scenario, 1 row per time step.

France-0	France-1	France-2	France-3	France-4	France-5	France-6	France-7	France-8	France-9	F
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	

Figure 15: Slack-France.csv

nbHoursSlack.csv: number of hours with non-served electricity per zone and scenario.

2.3.3 Costs

MeanVariableCost.csv: average variable costs. 1 column per zone, 1 row per technology.

	Iberia	France
Geothermal	18937.575	22276.8
Hydro|Run o	0	0
Nuclear	269478485	5306254349
Coal|Hard cc	478527714	73416112
Coal|Hard cc	0	0
Coal|Lignite|	27731252.1	0

Figure 16: MeanVariableCost.csv

VariableCostPerScenario.csv: variable costs. 1 column per scenario and zone, 1 row per techno.

	Iberia-0	Iberia-1	France-0	France-1
Geothermal	19499.85	18375.3	22276.8	22276.8
Hydro|Run o	0	0	0	0
Nuclear	271282977	267673993	5391918134	5220590564
Coal|Hard cc	488554056	468501371	73503237.8	73328986.1
Coal|Hard cc	0	0	0	0
Coal|Lignite|	27957680.4	27504823.7	0	0
Wind|Offshc	0	0	0	0

Figure 17: VariableCostPerScenario.csv

2.3.4 Marginal Costs

meanScenCmar.csv: average over all scenarios of the marginal costs. 1 column par zone, 1 row per time step.

	Iberia	France
02/07/2030 00:00	31.2407507	12.4154082
02/07/2030 01:00	31.2406682	12.4153796
02/07/2030 02:00	31.2406682	12.4153796
02/07/2030 03:00	31.2406682	12.4153256
02/07/2030 04:00	31.2406682	12.4153256
02/07/2030 05:00	31.2406682	12.4153256
02/07/2030 06:00	31.2406682	12.4152986
02/07/2030 07:00	23.2569	12.4153526
02/07/2030 08:00	20.1404809	12.4153256

Figure 18: meanScenCmar.csv

meanTimeCmar.csv: average over all timesteps of the marginal costs. 1 column per zone, 1 row per scenario.

	France	Germany	Italy	Switzerland	EasternEuro|	Benelux	Iberia	Britain	Balkans	Baltics	Scandinavia
0	10.0017916	12.627596	42.490835	31.7141263	7.66411584	40.0294435	42.6613128	12.8529953	4.55587547	2060.97874	12.2201652
1	9.69227039	8.52444049	43.0467851	38.7308398	6.53666704	36.7040107	223.171869	10.2837684	4.6981039	2033.43619	4.74404652
2	10.3288737	9.37353669	42.117134	31.952233	7.53592615	41.0401734	42.5633024	18.501651	4.45614196	2698.81407	112.051093
3	9.55761209	15.9528174	44.105486	939.943149	7.14025279	47.4428936	194.728893	22.6292927	17.1144993	3377.18889	322.078168
4	9.82969121	12.1518885	42.7560408	369.574226	7.49095538	41.2517987	82.2259141	10.5424854	5.21314862	2594.29045	148.974584
5	10.1909432	15.7723012	41.8901917	464.109506	8.20773537	46.6836205	89.8033024	28.683413	7.81741969	3017.58513	148.236302
6	8.86570825	7.29514207	41.9248344	31.631455	6.85554654	35.4527305	38.7852692	8.29644509	4.16811995	2696.23831	7.39171049
7	9.53233609	7.45645671	38.6234038	46.8661374	6.43878895	34.1042506	66.9689116	12.6328536	3.64768332	982.615293	23.1759666
8	9.50194492	5.67742224	36.1771126	610.968409	5.74191136	25.2614037	112.640293	6.5424717	3.1135563	1088.85861	3.93019885
9	10.2479143	11.0077585	43.095968	157.16267	7.73642026	39.8542586	95.9500739	19.7408548	7.92131013	1941.32043	36.1178868
10	9.22268466	9.28545367	39.5594783	128.144848	7.30263483	36.8149199	160.460943	14.01241	4.59688779	1690.70131	2.45393333

Figure 19: meanTimeCmar.csv

sortedCurveCmar.csv: average of the marginal costs histogram. 1 column par zone, 1 row per sorted time step index.

	Iberia	France
0	10000	10000
1	10000	10000
2	10000	10000
3	10000	10000
4	10000	10000
5	10000	10000
6	10000	10000
7	10000	5039.46006
8	10000	5039.45993
9	10000	5032.01263
10	10000	5032.01263

Figure 20: MonotoneCmar.csv

MarginalCostActivePowerDemand-ZONE.csv: marginal costs of the zone (in this case, the nodes on which the *ActivePowerDemand* constraints apply). 1 column par scenario, 1 row per time step.

HistCmar-ZONE.csv : histogram of the marginal costs of the zone. 1 column par scenario, 1 row per sorted time step index.

2.3.5 Flows

MeanImportExport-ZONE.csv: average import and exports to/from the zone. 1 column for Import, 1 column for export, 1 row per time step.

	Export	Import
02/07/2030 00:00	5000	0
02/07/2030 01:00	5000	0
02/07/2030 02:00	5000	0
02/07/2030 03:00	5000	0
02/07/2030 04:00	5000	0
02/07/2030 05:00	5000	0

Figure 21: meanImportExport-ZONE.csv

ImportExport-ZONE-$i.csv: import and exports to/from the zone for scenario $i. 1 column for import, 1 column for export, 1 row per time step.

MeanImportExport.csv: average flows. 1 column for import, 1 column for line, 1 row per time step.

	Iberia>Franc
02/07/2030 00:00	-5000
02/07/2030 01:00	-5000
02/07/2030 02:00	-5000
02/07/2030 03:00	-5000
02/07/2030 04:00	-5000

Figure 22: MeanImportExport.csv