Data formats

version 2.0 Feb 10, 2025

This document presents the format of plan4res datasets:

- IAMC format csv data
- plan4res csv input data
- plan4res NetCDF input data
- plan4res outputs

Details about the structure of plan4res and the workflow can be found in plan4resStructure.pdf (in your documentation directory or plan4res/documentation: plan4res documentation), in particular the location of the different files.

1 plan4res IAMC input data

The IAMC input data should be composed of one csv file whose name should be Dataset.xlsx. This file can be an output of the model GEneSYS-MOD. It will be used by the plan4res CREATE module to create plan4res csv input data files. It is also possible to adapt the configuration file of CREATE in order to use more than one IAMC file for retrieving the different variables (take some variables from one IAMC file, and other variables from another IAMC file).

These files have to follow the IAMC data format, described in the deliverable D4.2[1] (data exchange format and template) of open ENTRANCE, using the variables and regions defined in the open ENTRANCE nomenclature[2].

<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 1: example of IAMC input data used by script `CreateInputPlan4res.py` to create the plan4res files

2 Time series

- **A set of CSV files containing scenarized time serie**s: those files follow a 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 timeseries for One variable. Scenarios are identified by the corresponding column name in the header line (for example, past years, eg 1970, 1971, 1972…);

Table 2: Example of stochastic timeserie

Timestamp [UTC]	Base	PVminus10	Demandplus10
01/01/2050 00:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 01:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 02:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 03:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 04:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 05:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 06:00	3.31E-05	3.31E-05	3.64E-05
01/01/2050 07:00	4.91E-05	4.91E-05	5.40E-05
01/01/2050 08:00	6.51E-05	6.51E-05	7.16E-05
01/01/2050 09:00	8.11E-05	8.11E-05	8.92E-05

- **One (optional) unique CSV file containing all deterministic timeseries**: deterministic timeseries have to be present in a single csv file, whose name has to be given in the settingsCreateInputPlan4res_*xxx*.yml or in the settings_format_XXX.yml configuration file.

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

[2] openENTRANCE/openentrance: Definitions of common terms (variables, regions, etc.) for the openENTRANCE project

This file follows 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 correspond to each deterministic timeseries, the name of each deterministic timeseries being in the header.

Table 3: Deterministic time series

Timestamp [UTC]	TS_XX	TS_YY	TS_ZZ
01/01/2050 00:00	2539999.79	235940.523	1846508.33
01/01/2050 01:00	2330728.6	202206.466	1743163.67
01/01/2050 02:00	2400311.81	203914.562	1762293.74
01/01/2050 03:00	2373790.71	200641.82	1772480.46
01/01/2050 04:00	2242430.18	178585.198	1578006.06
01/01/2050 05:00	2139597.06	167443.384	1418207.69
01/01/2050 06:00	2524088.67	198991.4	1717258.86
01/01/2050 07:00	2466338.43	200655.869	1819359.65
01/01/2050 08:00	2339799.73	202049.705	1839468.73
01/01/2050 09:00	2360725.92	201430.798	1834015.73
01/01/2050 10:00	2311079.56	195961.933	1815442.03

3 Plan4res csv Input Data

This section describes the format of plan4res input data. A dataset is composed of 7 csv files representing the “fixed” data and of a number timeseries (described in section 2).

All the CSV input files are following.

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.

Important notice: within plan4res, the convention for the units is that all data and results are in MW, MWh, €, €/MW, €/MWh depending on the variable (see 5.3 Units)

The dataset is composed of:

- *ZP_ZonePartition.csv*: contains the description of the different regions;
- *ZV_ZoneValues.csv*: contains the data linked to the regions (in particular the demand);
- *IN_Interconnections.csv*: contains the description of the network;
- *TU_ThermalUnits.csv*: contains the description of the thermal power plants (including nuclear) and of any asset which can be modeled in plan4res as a thermal plant (even if not using any fossil fuel)
- *SS_SeasonalStorage.csv*: contains the description of hydropower with seasonal storages and other long-term seasonal storages;

- *STS_ShortTermStorage.csv*: contains the description of all the short-term storages, including eg. pumped hydro, batteries but it can also be used for demand-response mechanisms such as load-shifting;
- *RES_RenewableUnits.csv*: contains the description of PV, wind power and run-of-river technologies.

3.1 Geography and coupling constraints : files ZP_ZonePartition, ZV_ZoneValues, IN_Interconnections

This first group of data files is used to describe the regions in the dataset, the interconnections between the regions, and the constraints at region level, such as the power demand, but also the system services.

3.1.1 File ZP_ZonePartition.csv

The file ZP_ZonePartition.csv describes the different regions that are used in the dataset. In the example below, the lower level (on the left) corresponds to countries.

The lower level is used to define **Nodes**. Each generation unit (defined in TU_ThermalUnits.csv, RES_RenewableUnits.csv, STS_ShortTermUnits.csv and SS_SeasonalStorage.csv) belongs to one **Node**. Interconnections (defined in IN_Interconnections.csv) are connecting nodes.

The higher level corresponds to continents (with a unique continent called 'MiddleEarth'). There could exist intermediate levels. The different levels are used to define "coupling constraints" (see *5.2 Coupling Constraints*), such as the power demand which is always linked to a Node. Other coupling constraints (such as system services) may apply to different levels. (See *5.1 Description of geography* for more details about regions)

Table 4: ZP_ZonePartition.csv

Countries	Continent
RoGonDor	MiddleEarth
DolAmroth	MiddleEarth
Harad	MiddleEarth

The different regions can be defined by the user in the settingsCreateInputPlan4res_*xxx*.yml configuration file (Regions section).

3.1.2 File ZV_ZoneValues.csv

This file contains the values of all coupling constraints (in particular the demand at each Node, which is mandatory). It may also optionally contain the parameters for defining the costs associated to imbalance (for each coupling constraint). These parameters can also be defined in the settingsCreateInputPlan4res_*xxx*.yml configuration file (Coupling constraints section), in particular if the user wishes to use the same parameters for all the regions. If the user wishes to use different values, they must appear in ZV_ZoneValues.csv.

Table 5: ZV_ZoneValues.csv

Type	Zone	Value	Profile_Timeserie
Cooking	DolAmroth	7178950.652	TS_COOKING_DolAmroth.csv
ElecHeating	DolAmroth	54635950.13	TS_HEAT_LOW_DolAmroth.csv
ElecVehicle	DolAmroth	13541851.26	TS_MOBILITY_PSNG_DolAmroth.csv

OtherExclHeatTranspCooking	DolAmroth	51369219.81	TS_LOAD_DolAmroth.csv
CostActivePowerDemand	DolAmroth	10000	
MaxActivePowerDemand	DolAmroth	1500000	

Table 5 shows the content of ZV_ZoneValues.csv for the region "DolAmroth". In this example, tere is only one type of coupling constraint per region, which is the power demand. The power demand is composed of 4 parts: power demand for cooking, for heating (ElecHeating), for transportation (ElecVehicle) and for other uses (OtherExclHeatTranspCooking). The unit of the values in the column 'value' is MWh or €/MWh (for the variables of Type Cost*). The row 'Cooking' gives in column 'value' the power demand for cooking in the region DolAmroth for the year, in MWh, as well as the name of the timeseries to use to compute the hourly power demand for cooking (here TS_COOKING_DolAmroth.csv). Whenever the timserie would be deterministic, the Profile_Timeserie column would contain the name of the column corresponding to this timeseries in the csv file containing all deterministic timeseries. Table 6 shows the first 6 hours of this timeseries, for the 3 scenarios 'Base', 'PVminus10' and 'Demandplus10'). The (here scenarized) hourly power demand for cooking is computed by multiplying the timeseries by the numerical value in ZV_ZoneValue.csv.

Table 6: TS_COOKING_DolAmroth.csv

Timestamp [UTC]	Base	PVminus10	Demandplus10
01/01/2050 00:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 01:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 02:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 03:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 04:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 05:00	1.60E-05	1.60E-05	1.76E-05
01/01/2050 06:00	3.31E-05	3.31E-05	3.64E-05

The total power demand (scenarized) timeseries of the region is computed by adding the 4 timeseries corresponding to the 4 parts of the power demand. The way of computing the power demand is defined in the settingsCreateInputPlan4res_*xxx*.yml configuration file. Figure 1 shows the total power demand in the region DolAmroth, as it is computed by plan4res out of the values in column value of ZV_ZoneValues.csv and the different timeseries in column Profile_Timeseries.

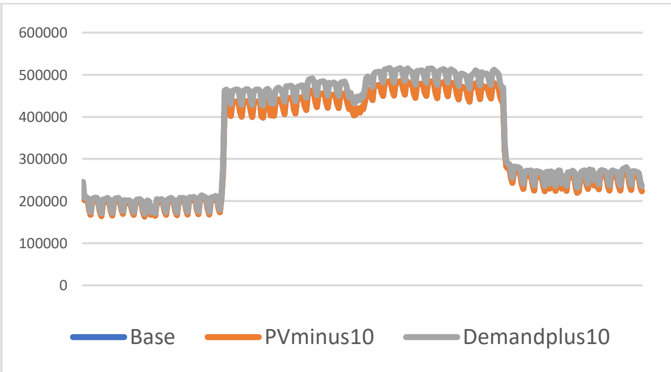

Figure 1: Power demand scenarios in region DolAmroth

3.1.3 File IN_Interconnections.csv

This file contains the description of the network, ie the set of **lines** connecting the different **nodes**. (remind that the nodes are described in the first column of ZP_ZonePartition.csv).

Table 7: IN_Interconnectionscsv

Name	StartLine	EndLine	MaxPowerFlow	MinPowerFlow
DolAmroth>Harad	DolAmroth	Harad	71	-100
RoGonDor>DolAmroth	RoGonDor	DolAmroth	2200	-2000
RoGonDor>Harad	RoGonDor	Harad	866	-1160

In_Interconnections.csv may contain the following columns (optional columns are highlighted): (*Note that all numerical values may be replaced by the name of a deterministic Timeserie*):

- **Name**: name of the line (used for processing results)
- **StartLine** and **EndLine** must be nodes defined in the first column of ZP_ZonePartition.csv.
- **MaxPowerFlow** and **MinPowerFlow** are the bounds in MW on the flows for this line (one way or the other); *MaxPowerFlow* is the maximal flow from Start to End, while *MinPowerFlow* (which can be negative) is the minimum flow from Start to End. *(-1)*MinPowerFlow* is also the maximum flow between End and Start while *(-1)*MaxPowerFlow* is the minimum flow between Start and End.
- **Impedance** (optional, default 0): Impedance of the line
- **Cost** (optional, default 0): Cost of the line in €/MWh

3.2 Description of generation assets : files TU_ThermalUnits, SS_SeasonalStorage, STS_ShortTermStorage and RES_RenewableUnits

These data files allow to describe the different kinds of generation units. It also allows to describe some load control mechanisms, such as load shifting, or management of electric vehicle charging.

3.2.1 File TU_ThermalUnits.csv

This files gives the characteristics of all thermal power plants (and of all the power plants which are modelled as thermal power plants in plan4res, such as Geothermal plants).

Table 8: TU_ThermalUnits.csv

Zone	Name	NumberUnits	MaxPower	VariableCost	CO2Rate
RoGonDor	Biomass|w/o CCS	1	19573.5	3.816	0
DolAmroth	Biomass|w/o CCS	1	367.12	3.816	0
Harad	Biomass|w/o CCS	1	469.141614	3.816	0
RoGonDor	Coal|Hard coal|w/o CCS	1	16445.9549	3	0.7512
DolAmroth	Coal|Hard coal|w/o CCS	1	11.923392	3	0.7512
Harad	Coal|Hard coal|w/o CCS	1	1783.37195	3	0.7512
RoGonDor	Coal|Lignite|w/o CCS	1	5339.78409	3	0.94971429
RoGonDor	Gas|CCGT|w/o CCS	1	83694.0023	81.6648775	0.34634483
DolAmroth	Gas|CCGT|w/o CCS	1	2000	82.0429943	0.34634483
Harad	Gas|CCGT|w/o CCS	1	15000	81.517916	0.34634483
RoGonDor	Gas|OCGT|w/o CCS	1	22000	113.409441	0.52863158
Harad	Gas|OCGT|w/o CCS	1	25000	113.205328	0.52863158
RoGonDor	Nuclear	1	50783.1831	8	0

DolAmroth	Nuclear	1	5000	8	0
Harad	Nuclear	1	40000	8	0
RoGonDor	Oil|w/o CCS	1	8652.37627	0.036	0.70105263
Harad	Oil|w/o CCS	1	3.94960304	0.036	0.70105263

It contains the following data (*Note that some of the numerical values may be replaced by the name of a deterministic Timeserie.* In that case the timeseries must be present in the Deterministic timeseries CSV file (see):

- **Name**: name of the technology, or name of the plant. The list of names must be present in the settingsCreateInputPlan4res_*xxx*.yml configuration file (technos section, thermal), as this list is used in particular for post-treatments of results. Whenever the user wishes to add a new technology or plant, it must be added to the list in the settingsCreateInputPlan4res_*xxx*.yml configuration file.
- **Zone**: the values in this column must be included in the list of **nodes**, which is given in the first column of ZP_ZonePartition.csv, and corresponds to the lowest granularity of regions.
- **NumberUnits**: number of units of the given characteristics.
- **MaxPower**: maximum power of a single unit in MW. When NumberUnits=1, MaxPower is equal to Capacity.
- **Profile_Timeserie** (optional): deterministic or stochastic timeseries which will be applied to MaxPower. This allows accounting for eg. Maintenances and outages.
- **MinPower** (optional, 0 by default)[3] : minimum power of a single unit in MW. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **Pauxiliary** (optional, 0 by default): Power (MW) taken from the system when off for each unit. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **VariableCost** (optional, 0 by default): proportional cost in €/MWh. Can be a deterministic timeseries.
- **FixedCost** (optional, 0 by default): fixed cost in €. Can be a deterministic timeseries.
- **Quadterm** (optional, 0 by default): quadratic cost. Can be a deterministic timeseries.
- **StartUpCost** (optional, 0 by default): cost (€) for starting the unit after a shut down. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **MinUpTime** (optional, 1 hour by default): minimum duration when the plant is on (number of hours). *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **MinDownTime** (optional, 1 hour by default): minimum duration when the plant is off (number of hours). *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **Inertia** (optional, 0 by default): maximum inertia that can be provided by a unit in MWs/MWA. Can be a deterministic timeseries.
- **PrimaryRho** (optional, 0 by default): this parameter, multiplied by Maxpower, gives the maximum share of the active power that can be used as primary reserve. Can be a deterministic timeseries.
- **SecondaryRho** (optional, 0 by default): this parameter, multiplied by MaxPower, gives the maximum share of the active power that can be used as secondary reserve (optional, 0 by default). Can be a deterministic timeseries.

[3] Note that data consistency is ensured: if 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 started.

- **DeltaRampDown** (optional, MaxPower by default): maximum gradient when the power is decreased from one time step to the other, MW per hour. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **DeltaRampUp** (optional, Maxpower by default): maximum gradient when the power is increased from one time step to the other, MW per hour. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **CO2** (optional, 0 by default): emission rate in tons CO2 per MWh
- **Any other pollutant** which is defined in the coupling constraints section of the settingsCreateInputPlan4res_*xxx*.yml configuration file.

3.3 File SS_SeasonalStorage.csv

This file gives the characteristics of all seasonal storages. Only aggregated seasonal storages (ie one with one reservoir) can be described in this file (although SMS++ is capable of handling complex hydrovalleys).

Table 9: SS_SeasonalStorage.csv

Name	Zone	Hydro System	Number Units	Max Power	Min Power	Max Volume	Min Volume	Inflows	Inflows Profile	Initial Volume	Turbine Efficiency	Pumping Efficiency
Hydro|Reservoir	RoGonDor	0	1	25814.05	0	51628100	0	7.5E+07	Inflows_RoGonDor.csv	34418733.3	1	0
Hydro|Reservoir	DolAmroth	0	1	5524.2394	0	11048478.9	0	1.6E+07	Inflow_DolAmroth.csv	6629087.33	1	0
Hydro|Reservoir	Harad	0	1	6000	0	12000000	0	2.8E+07	Inflow_Harad.csv	6000000	1	0

It contains the following columns (*Note that all numerical values may be replaced by the name of a deterministic Timeserie*):

- **Name**: name of the technology, or name of the plant. The list of names must be present in the settingsCreateInputPlan4res_*xxx*.yml configuration file (technos section, reservoir), as this list is used in particular for post-treatments of results. Whenever the user wishes to add a new technology or plant, it must be added to the list in the settingsCreateInputPlan4res_*xxx*.yml configuration file.
- **Zone**: the values in this column must be included in the list of **nodes**, which is given in the first column of ZP_ZonePartition.csv, and corresponds to the lowest granularity of regions.
- **NumberUnits**: number of units of the same type at the same location.
- **MaxPower** : maximum power of a single unit in MW. When NumberUnits=1, MaxPower is equal to Capacity. Can be a deterministic timeserie.
- **MinPower**[4] (optional, 0 by default) [5] : minimum power of a single unit in MW. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **DeltaRampDown** (optional, MaxPower by default): maximum gradient when the power is decreased from one time step to the other, MW per hour. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*

[4] As in ThermalUnits, consistency between MinPower and MaxPower is ensured such that Minpower can never be greater than MaxPower.
[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.

- **DeltaRampUp** (optional, Maxpower by default): maximum gradient when the power is increased from one time step to the other, MW per hour. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **MaxVolume** : maximum volume of the reservoir (MWh). Can be a deterministic timeseries.
- **MinVolume**[6] (optional, 0 by default) : minimum volume of the reservoir (MWh). Can be a deterministic timeseries.
- **TurbineEfficiency**: (optional, 1 by default). This value, multiplied by the flow, gives the generated power.
- **PumpingEfficiency**: (optional, 1 by default, but should be lower than TurbineEfficiency in practice). This value, multiplied by the flow, gives the generated power.
- **Inflows**: (optional, 0 by default; mandatory if column InflowsProfile is present). Inflows to the upstream reservoir (energy per year in MWh).
- **InflowsProfile**: (optional): time series profile for Inflows. Multiplied by the Inflows in energy, gives the inflows time series. These timeseries may be stochastic (in that case the value has to be XXX.csv), or deterministic.
- **InitialVolume**: (optional, 0 by default). Initial Volume of the upstream reservoir (MWh)
- **Inertia** (optional, 0 by default): maximum inertia that can be provided by a unit in MWs/MWA. Can be a deterministic timeseries.
- **PrimaryRho**: (%) this parameter, multiplied by MaxPower, gives the maximum primary reserve that can be provided by a unit (optional, 0 by default). Can be a deterministic timeserie.
- **SecondaryRho**: (%) this parameter, multiplied by MaxPower, gives the maximum secondary reserve that can be provided by a unit (optional, 0 by default). Can be a deterministic timeserie.
- **WaterValues**: optional; used to specify water values as input data from a file, if they are/were 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 (ie a polyhedral function), this file is only given in the first line; when using the simulation mode with 1 BV per unit (ie one function per reservoir), this file is given at each line; in optimization mode (= when running the SSV), this is not used as computed by the model. This column is always optional as the BV file (in case it is a polyhedral function) can be passed to SMS++ when calling the solver.

3.4 File STS_ShortTermStorage.csv

This file is used for:

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

[6] Consistency between MinVolume and MaxVolume is ensured, such that MinVolume can never be greater than MaxVolume.

Table 10: STS_ShortTermStorage.csv

Name	Zone	NumberUnits	MaxPower	MinPower	MaxVolume	MinVolume	TurbineEfficiency	PumpingEfficiency
Hydro|Pumped Storage	RoGonDor	1	12150.54	-12150.54	1215054	0	0.866	0.866
Hydro|Pumped Storage	DolAmroth	1	3302	-3302	330200	0	0.866	0.866
Hydro|Pumped Storage	Harad	1	5920.72	-5920.72	592072	0	0.866	0.866
Battery|Lithium-Ion	RoGonDor	1	142775.466	-14275.47	571101.864	0	0.95	0.95
Battery|Lithium-Ion	DolAmroth	1	3915.34909	-391.3491	15661.3964	0	0.95	0.95
Battery|Lithium-Ion	Harad	1	27246.1193	-27246.119	108984.477	0	0.95	0.95

This file contains the following data:

- **Name**: name of the technology, or name of the plant. The list of names must be present in the settingsCreateInputPlan4res_*xxx*.yml configuration file, as this list is used in particular for post-treatments of results. Whenever the user wishes to add a new technology or plant, it must be added to the list in the settingsCreateInputPlan4res_*xxx*.yml configuration file.
- **Zone**: the values in this column must be included in the list of **nodes**, which is given in the first column of ZP_ZonePartition.csv, and corresponds to the lowest granularity of regions.
- **NumberUnits**: number of units of the same type at the same location.
- **MaxPower**: maximum power of a single unit in MW. When NumberUnits=1, MaxPower is equal to Capacity.
- **MaxPowerProfile** (optional): deterministic timeseries, which will be multiplied by MaxPower to obtain the MaxPower timeseries in MW.
- **MinPower** (optional, 0 by default) [7] [8]: minimum power of a single unit in MW. Can be a deterministic timeserie. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **MaxVolume**[9]: maximum volume of the reservoir (MWh)..
- **MaxStorageProfile** (optional): deterministic timeseries, which will be multiplied by MaxVolume to obtain the MaxVolume timeserie in MWh.
- **MinVolume** (optional, 0 by default): minimum volume of the reservoir (MWh). Can be a deterministic timeserie.
- **TurbineEfficiency** (optional, 1 by default): This value, multiplied by the flow, gives the generated power. Can be a deterministic timeserie.
- **PumpingEfficiency** (optional, 1 by default, but should be lower than TurbineEfficiency in practice): This value, multiplied by the flow, gives the generated power. Can be a deterministic timeserie.
- **Inflows** (optional, 0 by default):
- **InitialPower** (optional, 0 by default): Initial Power of the unit (MW)
- **InitialStorage** (optional, 0 by default): Initial Volume of the upstream reservoir (MWh)
- **Inertia** (optional, 0 by default): maximum inertia that can be provided by a unit in MWs/MWA. Can be a deterministic timeserie.
- **MaxPrimaryPower** (optional, 0 by default): maximum primary reserve that can be provided by a unit. Can be a deterministic timeserie.
- **MaxSecondaryPower** (optional, 0 by default): maximum secondary reserve that can be provided by a unit (optional, 0 by default). Can be a deterministic timeserie.
- **DeltaRampDown** (optional, MaxPower by default): maximum gradient when the power is decreased from one time step to the other, MW per hour. Can be a deterministic timeseries.

---

[7] 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.
[8] Consistency between MinPower and MaxPower is ensured, MinPower can never be greater than MaxPower.
[9] Consistency between MinVolume and MaxVolume is ensured. MinVolume can never be greater than MaxVolume

This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)

- **DeltaRampUp** (optional, Maxpower by default): maximum gradient when the power is increased from one time step to the other, MW per hour. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **Cost** (optional, 0 by default): proportional cost (€/MWh). Can be a deterministic timeseries.
- **Inflows**: (optional, 0 by default). Inflows to the upstream reservoir (MWh). If negative it is considered to be a consumption linked to the unit (eg consumption of EV). Can be a deterministic timeseries.
- **VolumeLevelTarget**[10]: (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.

3.4.1 File RES_RenewableUnits.csv

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

Table 11: RES_RenewableUnits.csv

Name	Zone	NumberUnits	MaxPower	MinPower	MaxPowerProfile
Wind|Onshore	DolAmroth	1	9923.33	0	ONSHORE_OPT_DolAmroth.csv
Wind|Offshore	RoGonDor	1	18113.33	0	Offshore-Deep_RoGonDor.csv

It contains the following data:

- **Name**: name of the technology, or name of the plant. The list of names must be present in the settingsCreateInputPlan4res_*xxx*.yml configuration file (technos section, thermal), as this list is used in particular for post-treatments of results. Whenever the user wishes to add a new technology or plant, it must be added to the list in the settingsCreateInputPlan4res_*xxx*.yml configuration file.
- **Zone**: the values in this column must be included in the list of **nodes**, which is given in the first column of ZP_ZonePartition.csv, and corresponds to the lowest granularity of regions.
- **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.)
- maximum power of a single unit in MW. When NumberUnits=1, MaxPower is equal to Capacity.
- **MaxPowerProfile** (optional): deterministic or stochastic timeseries which will be applied to MaxPower. This allows accounting for in particular the variability of the potential power, given its correlation with climate variables (sun, wind…).

[10] If an InitialStorage is not provided, VolumeLevelTarget will be used as InitialStorage.

- **MinPower** (optional, 0 by default)[11] : minimum power of a single unit in MW. Can be a deterministic timeseries. *This constraint is relaxed in SSV and CEM, as well as in SIM when the user requires to use continuous relaxation (see parameter in SMS++ configuration files)*
- **Inertia** (optional, 0 by default): maximum inertia that can be provided by a unit in MWs/MWA. Can be a deterministic timeserie.
- **Gamma** (optional, 1 by default): this parameter is used by the model to determine the maximum available primary and secondary reserve. In practice, the model accounts for the following constraint: at each timestep, the sum of the primary and secondary reserves is lower than the maximum power minus the generated power.

3.5 Additional columns for dealing with Investments (CEM)

To run the Capacity Expansion Model (CEM), 4 columns need to be added to the files corresponding to assets in which one wants to invest (see below), ie to the files describing the generation units (apart from Seasonal Storages) as well as to the file describing the interconnection (see):

Table 12: additional columns for capacity expansion

Zone	Name	....	InvestmentCost	DecommissionCost	MaxAddedCapacity	MaxRetCapacity
DolAmroth	Biomass|w/o CCS		1980000	100	5000	0
RoGonDor	Biomass|w/o CCS		1980000	100	4000	0
Harad	Biomass|w/o CCS		1980000	100	500	0
DolAmroth	Coal|Hard coal|w/o CCS		1082250	100	0	2000

- **MaxAddedCapacity**: this is the maximum capacity that may be added, in MW.
- **MaxRetCapacity**: this is the maximum capacity that may be decommissioned, in MW.
- **InvestmentCost**: this is the cost for investing into the given capacity in the given zone, in €/MW. Note that these are yearly costs, computed as Capital Cost / LifeTime (in years) + Fixed Cost.
- **DecommissionCost**: this is the cost for decommissioning the given capacity in the given zone, in €/MW.

Costs are not discounted.

These columns may then 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.

[11] Note that data consistency is ensured: if 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 started.

4 Plan4res outputs

The results of plan4res are detailed in the following subsections

4.1 Results of SSV

The SSV runs 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[12]: *
  - BellmanValuesOUT.csv: redundant cuts have been pruned out.
  - BellmanValuesAllOUT.csv: contains all the cuts found by the algorithm.
- 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.

The format of this file is, as shown in Table 13, aiming to represent a polyhedral function at each SSV timestep. The first column (Timestep) gives the index of the SSV Timestep. There are a number of rows for each timestep, representing the different 'cuts' of the function at this timestep. The function at timestep $i is: $\max_{cuts}\left(\sum_{j=0}^{R} a\_j + b\right)$, where a_j and b are the values in the corresponding columns, and R is the number of reservoirs (the number of columns a_j).

Table 13: BellmanValuesOUT.csv

Timestep	a_0	a_1	a_2	b
0	0	0	0	3.9825E+10
0	-113.40944	-10000	-10000	7.4137E+10
0	-113.40944	-10000	-81.517916	1.2033E+11
0	-81.786066	-3	-150.94929	5.6955E+10
0	-0.036	-3.816	-8	5.3972E+10
0	-8	-3	-113.20533	5.5282E+10
0	-0.03249	-0.03249	-8	5.4175E+10
0	-0.036	-10000	-81.517916	1.1965E+11
0	-0.036	-8	-81.517916	5.5241E+10
1	0	0	0	3.9637E+10
1	-113.40944	-10000	-10000	1.3613E+11

4.2 Results of SIM

The simulation produces 2 sets of results:

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

- Raw results of the simulation. They correspond to the outputs of the model, without any post-treatment (the post-treatment is performed when running POSTTREAT).
- Post-treated results. These are results computed by the POSTTREAT function of plan4res.

4.2.1 Raw results of SIM

All row results follow the same format: the index column gives the indexes of the simulation timesteps, while the other columns give the values corresponding to the name in the header (which can be a region, a technology….)

For each scenario $i, the following files are produced.

- Demand_Scen$i_OUT.csv: 1 column per node with the power demand in the node.
- ActivePower_Scen$i_OUT.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). Note that all other units with storages comprise a unique column with both positive and negative (pumping) values.
- Primary_Scen$i_OUT.csv: primary reserve schedules. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part).
- Secondary_Scen$i_OUT.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)
- Inertia_Scen$i_OUT.csv : inertia provided by each unit. 1 column per unit (apart from the Seasonal Storage units which are duplicated in 2 columns: one for the generation part, and the other one for the pumping part).
- Volume_Scen$i_OUT.csv: volumes of each storage (seasonal storages and short-term storages).
- Flows_Scen$i_OUT.csv: flows between zones. 1 column per transmission line.
- MarginalCostActivePowerDemand_Scen$i_OUT.csv: marginal costs of the demand constraint. 1 column per node (ch the *ActivePowerDemand* constraint is always applied to Nodes).
- MarginalCostPrimary_Scen$i_OUT.csv: marginal costs of the primary reserve constraint. 1 column per region (at the partition level of the *PrimaryDemand* constraints, note that these constraints may apply to different regional partitions than the Demand. See section 5.1).
- MarginalCostSecondary_Scen$i_OUT.csv: marginal costs of the secondary reserve constraint. 1 column per zone (at the partition level of the *SecondaryDemand* constraints).
- MarginalCostInertia_Scen$i_OUT.csv: marginal costs of the Inertia constraint (at the partition level of the *InertiaDemand* constraints). 1 column per zone.
- MarginalCostFlows_Scen$i_OUT.csv: marginal costs of the lines. 1 column per line.
- MaxPower_Scen$i_OUT.csv: maximum available power. 1 column per technology.

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

4.2.2 Post-Treated resuls of SIM

4.2.2.1 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 node. (the generation units are attached to nodes)

Table 14: InstalledCapacity.csv

	Hydro|Reservoir	**Biomass|w/o CCS**	**Coal|Hard coal|w/o CCS**	**Coal|Lignite|w/o CCS**	**Gas|CCGT|w/o CCS**
RoGonDor	25814.05	19573.5	16445.95488	5339.784093	83694.0023
DolAmroth	5524.239444	367.12	11.923392	0	2000
Harad	6000	469.1416144	1783.371951	0	15000

4.2.2.2 Generation

Generation.csv and AggrGeneration.csv: total average (on scenarios) generation on the whole simulation time horizon (aggregated generation, with aggregations such as defined in `settingsPostTreatPlan4res_XXX.py`) from the different available technologies per node. 1 column per technology, 1 row per node. Note that the Non Served column corresponds to the non-served electricity.

Table 15: AggrGeneration.csv

	Hydro	**WindPower**	**PV**	**Biomass**	**Nuclear**
RoGonDor	126850389	578839017	185813311	0	249852103
DolAmroth	29932305.8	26161752.2	11697614.5	2944912.11	29463408.7
Harad	37280301.6	10847753.5	117436876	4098421.14	346617336

Generation-NODE.csv and ***AggrGeneration-NODE.csv***: average generation (aggregated generation) from the different available technologies per node. 1 file per node, 1 column per technology, 1 row per simulation time step.

Table 16: Generation-Harad.csv

	Hydro|Reservoir	Biomass|w/o CCS	Coal|Hard coal|w/o CCS
02/07/2030	886.9210309	11259.39875	42800.92681
03/07/2030	0	11259.39875	42800.92681
04/07/2030	0	11259.39875	42800.92681
05/07/2030	0	11259.39875	42800.92681
06/07/2030	0	11259.39875	42800.92681
07/07/2030	0	11259.39875	42800.92681
08/07/2030	0	11259.39875	42800.92681
09/07/2030	19775.91482	11259.39875	42800.92681

Generation-NODE-$i.csv: generation from the different available technologies in the node 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 node. 1 file per node. 1 column per scenario, 1 row per time step.

Table 17: Slack-Harad.csv

	Harad-0	Harad-1	Harad-2	Harad-3
02/07/2030	0	0	0	0
03/07/2030	0	0	0	0
04/07/2030	0	0	0	0
05/07/2030	0	0	0	0
06/07/2030	0	0	0	0
07/07/2030	0	0	0	0

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

4.2.2.3 Costs

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

Table 18: MeanVariableCost.csv

	RoGonDor	DolAmroth	Harad
Hydro|Reservoir	0	0	0
Biomass|w/o CCS	0	11237784.6	15639575.1
Coal|Hard coal|w/o CCS	330248836	305191.142	46738612.1
Coal|Lignite|w/o CCS	107785456	0	0

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

Table 19: VariableCostPerScenario.csv

	RoGonDor-0	RoGonDor-1	RoGonDor-2	RoGonDor-3	DolAmroth-0
Hydro|Reservoir	0	0	0	0	0
Biomass|w/o CCS	0	0	0	0	11073780.1

4.2.3 Marginal Costs

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

Table 20: meanScenCmar.csv

	RoGonDor	DolAmroth	Harad
02/07/2030	0.036	8	81.517916
03/07/2030	0.036	7.415	81.517916
04/07/2030	0.036	8	81.517916
05/07/2030	0.036	5.999648	81.517916
06/07/2030	0.036	5.999648	81.517916
07/07/2030	0.036	5.999648	77.5439176
08/07/2030	0.036	5.999648	81.517916
09/07/2030	0.036	8	81.517916
10/07/2030	0.036	8	81.517916
11/07/2030	0.036	5.999648	81.517916

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

Table 21: meanTimeCmar.csv

	RoGonDor	DolAmroth	Harad
Base	256.72732	3439.75537	1521.54793
PVminus10	257.017259	3583.34174	1793.57264
Demandplus10	442.220462	4878.66198	3086.87072
PVminus10AndDemandplus10	451.32022	4977.57685	3259.88472

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

Table 22: SortedCmar.csv

	RoGonDor	DolAmroth	Harad
0	10000	10000	10000
1	10000	10000	10000
2	10000	10000	10000
3	10000	10000	10000
4	9512.5	10000	10000
5	9154.88089	10000	10000
6	9154.88089	10000	10000
7	8749.78	10000	10000
8	8506.03	10000	10000

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

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

4.2.3.1 Flows

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

Table 23: meanImportExport-NODE.csv

	Export	Import
02/07/2030	0	22488
03/07/2030	0	22488
04/07/2030	0	22488
05/07/2030	0	22488
06/07/2030	0	22488
07/07/2030	0	22488

ImportExport-NODE-$i.csv: import and exports to/from the node 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.

Table 24: MeanImportExport.csv

	DolAmroth>Harad	RoGonDor>DolAmroth	RoGonDor>Harad
02/07/2030	1704	52800	20784
03/07/2030	1704	52800	20784
04/07/2030	1704	52800	20784
05/07/2030	1704	52800	20784
06/07/2030	1704	52800	20784
07/07/2030	1704	52800	20784

5 Appendix

5.1 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. A partition defines a set of regions which all together cover the whole area in the dataset.

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), where the regions are **also called nodes**. The partition 'Level2' is a higher level partition, meaning that the regions in 'Level2' are larger than the nodes in 'Level1', and each region in 'Level2' is composed of a list of nodes in 'Level1'. 'Level3' regions again are bigger than in 'Level2'. At each level we may have different partitions obtained by regrouping regions differently, as exemplified in Figure 2 below.

- Level1 = identifier of the first level zone

- Level2 = identifier of the second level partition
- Level3Part1 = identifier of a first partition at third level
- Level3Part2 = identifier of a second partition at the third level
- Etc.

A row with values L1, L2, L31, L32 … means that the region L1 (in level 1) belongs to the region L2 (in level 2), as well as to regions L31 and L32 (both in level 3), …;

The Figure 2 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 nodes in level1, aggregated in 4 regions at level 2: BE, FR1, FR2 and ES. Each level1 node belongs to a unique level2 region. At level3 there are 2 different groupings: Level3-grouping1 has 2 regions: North and South while Lenel3-grouping 3 has 3 regions: BE, FR, ES.

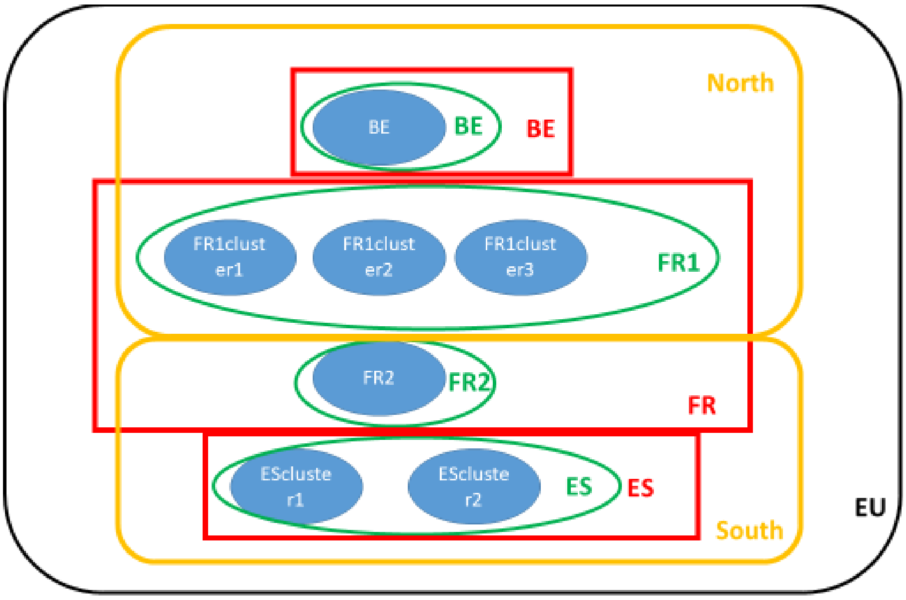

Figure 2: partitions

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 25: example of ZP_ZonePartition file

5.2 Coupling Constraints

The constraints are described in the plan4res settings file: settingsCreateInputPlan4res.yml[13]. An example is available in the toyDataset available here https://github.com/plan4res/toyDataset .

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 settingsCreateInputPlan4res.yml[14] 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 3: 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 = Total Demand in MWh * TimeSeries . The TimeSeries is a profile 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 = Demand for heating in TWh * ElecHeating Timeseries
    - AirCondition= Demand for cooling in TWh * AirCondition Timeseries
    - ElecVehicle= Demand for transport * ElecVehicle Timeseries
    - OtherExclHeatTransp= Other demands in TWh * OtherExclHeatTransp Timeseries

- PrimaryDemand: to represent the primary reserve requirement in each region. Note that the regions to which this constraint apply may be different than the nodes (see the definition of the partitions). This demand can be a timeserie.

- SecondaryDemand: to represent the primary reserve requirement in each region. Note that the regions to which this constraint apply may be different than the nodes (see the definition of the partitions). This demand can be a timeserie.

[13] Two different settings file may exist: 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 information for the capacity expansion

[14] Two different settings file may exist: 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 information for the capacity expansion.

- <u>InertiaDemand</u>: to represent the inertia requirement in each zone. Note that the regions to which this constraint apply may be different than the nodes (see the definition of the partitions). This demand can be a timeserie.

  <u>PollutantBudget</u>: to represent the max quantity of polluting emissions of a specific pollutant in each region. Note that the regions to which this constraint apply may be different than the nodes (see the definition of the partitions). This demand can be a timeserie. The user can define a list of pollutants and constraints for each pollutant. There must then exist a column for each pollutant in the TU_ThermalUnits.csv file, with the emission rate of this 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 *3.1*), 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).

5.3 Units

The units in the plan4res input files are MW, MWh, €, E/MW, €/MWh. When creating these datafiles from e.g. Scenario data, conversions must be explicitly defined in the settings files (in `CreateInputPlan4res_XXX.py`)