Using the plan4res models may require the following steps (depending on user’s needs):

* **Create a dataset with the plan4res data format:** This dataset, composed of excel and csv files and easily readable, can be created manually or by using the openEntrance linkage scripts which are available in <https://github.com/openENTRANCE/linkages>
* **Run the formatting tool** : the formatting tool is part of the plan4res container. It is used to convert the plan4res (excel+csv) dataset into SMS++ (NetCdF) datasets.
* **Run the optimisation (SSV)** : the SSV tool is part of the plan4res container.
* **Run the investment (CEM)** : the CEM tool is part of the plan4res container.
* **Run the simulation (EUC)** : the simulation tool is part of the plan4res container.
* **Run the UnitCommitment** : the UC tool is part of the plan4res container.
* **Posttreat results** : the outputs of the EUC can be analysed using the python scripts available in <https://github.com/openENTRANCE/linkages>

## Running the formatting tool

The formatting tool reads the excel+csv plan4res input files, and creates a serie of NetCdF files (SMS++ format) :

* SDDPBlock.nc4 : this file, used for optimisation, investment and simulation, describes the full problem, apart from investment  ;
* InvestmentBlock.nc4 : this file, used for investment only, describes the investment problem.
* N files Block\_i.nc4 : each one describes the assets for the SSV timestep i (usually week) ;

The formatting tool can be run with the following command :

Formatting.exe file="excelfile.xlsx" indir="myInputDir" outdir="myOutputDir" vu="0" mode="1" csv="csvfile.csv"

With :

* excelfile.xlsx: path and name of the excel file
* myInputDir: timeseries directory
* myOutputDir: netcdf files pathway
* csvfile.csv : path and name of the csv file
* Vu : 0
* Mode :
  + 0 : generates 1 UCBlock for solving the unit commitment on the first SSV timestep
  + 1 : generates a serie of UCBlocks for each period (each SSVTimestep) of the dataset, WITHOUT scenarised data
  + 2 : generates a serie of UCBlocks for each period (each SSVTimestep) of the dataset, WITH scenarised data
  + 3 : generates a SDDPBlock for the whole period, without UCBlocks
  + 4 : generates a SDDPBlock for the whole period, and all UCBlocks, WITH scenarised data
  + 5 : generates a SDDPBlock for the whole period, and all UCBlocks, WITHOUT scenarised data
  + 6 : generates a InvestmentBlock and a SDDPBlock for the whole period, without UCBlocks
  + 7 : generates a InvestmentBlock and a SDDPBlock for the whole period, and all UCBlocks, WITH scenarised data
  + 8 : generates a InvestmentBlock and a SDDPBlock for the whole period, and all UCBlocks, WITHOUT scenarised data
  + 9 : generates a InvestmentBlock only

A full description of the SMS++ NetCdF input files is available in the SMS++ GitLab: <https://gitlab.com/smspp/smspp-project/-/blob/develop/doc/SMS++%20File%20Format%20Manual/ffm.pdf>

## Running the Unit Commitment

The unit commitment can be run with the following command :

ucblock\_solver [options] Block.nc4

with the following set of options :

-B <file>, --blockcfg <file> Block configuration.

-S <file>, --solvercfg <file> Solver configuration.

-n <file>, --nc4problem <file> Write nc4 problem on file.

-v, --verbose Make the solver verbose.

-h, --help Print this help.

Block.nc4 is the netcdf input file ; it can be generated by the formatting tool, mode 0.

Detailed documentation as well as examples of configuration files can be found on the SMS++ repository : [SMS++ / Tools · GitLab](https://gitlab.com/smspp/tools/-/tree/master/)

## Running the SSV or the EUC

sddp\_solver [options] SDDPBlock.nc4

with the following set of options :

-B, --blockcfg <file> Block configuration.

-c, --configdir <path> The prefix for all config filenames.

-e, --eliminate-redundant-cuts Eliminate given redundant cuts.

-h, --help Print this help.

-i, --scenario <index> The index of the scenario.

-l, --load-cuts <file> Load cuts from a file.

-m, --num-simulations <number> Number of simulations to be performed.

-n, --num-blocks <number> Number of sub-Blocks per stage.

-p, --prefix <path> The prefix for all Block filenames.

-r, --relax Relax integer variables.

-s, --simulation Simulation mode.

-S, --solvercfg <file> Solver configuration.

-t, --stage <stage> Stage from which initial state is taken.

The option -s corresponds to the EUC, it requires to choose a scenario with option -i ; without this option the SSV will be run. -t is not needed. -l is necessary for running the EUC as well as for hot-starting the SSV ; -r is used for a run with continuous relaxation  (without this option, SMS++ runs with integer constraints)

SDDPBlock.nc4 is the netcdf input file ; it can be generated by the formatting tool, mode 5.

Detailed documentation as well as examples of configuration files can be found on the SMS++ repository : [SMS++ / Tools · GitLab](https://gitlab.com/smspp/tools/-/tree/master/)

## Running the CEM

investment\_solver [options] InvestmentBlock.nc4

with the following set of options :

-B, --blockcfg <file> Block configuration.

-c, --configdir <path> The prefix for all config filenames.

-e, --eliminate-redundant-cuts Eliminate given redundant cuts.

-h, --help Print this help.

-l, --load-cuts <file> Load cuts from a file.

-n, --num-blocks <number> Number of sub-Blocks per stage.

-o, --output-solution Output the solutions.

-p, --prefix <path> The prefix for all Block filenames.

-r, --relax Relax integer variables.

-S, --solvercfg <file> Solver configuration.

-s, --simulate Simulate the given investment.

-x, --initial-investment <file> Initial investment.

Detailed documentation as well as examples of configuration files can be found on the SMS++ repository : [SMS++ / Tools · GitLab](https://gitlab.com/smspp/tools/-/tree/master/)