SILAMv5 User Manual

Thu 06 Jul, 2023

Contents

1 Substances and their transformation in SILAM

SILAM is capable of computation of dispersion of up to 496 different nuclides, together with their radioactive decays and transformations; inert and chemical y active size-specific aerosol; biological material (pollen grains), chemically active gases. And this is so called the *forward* mode.

The model also computes probabilities (*backward* mode) where the source represents the measurements of a measurement site and the result is the probability of a certain grid cell to be contributing to that measurement.

Silam has the following structure:

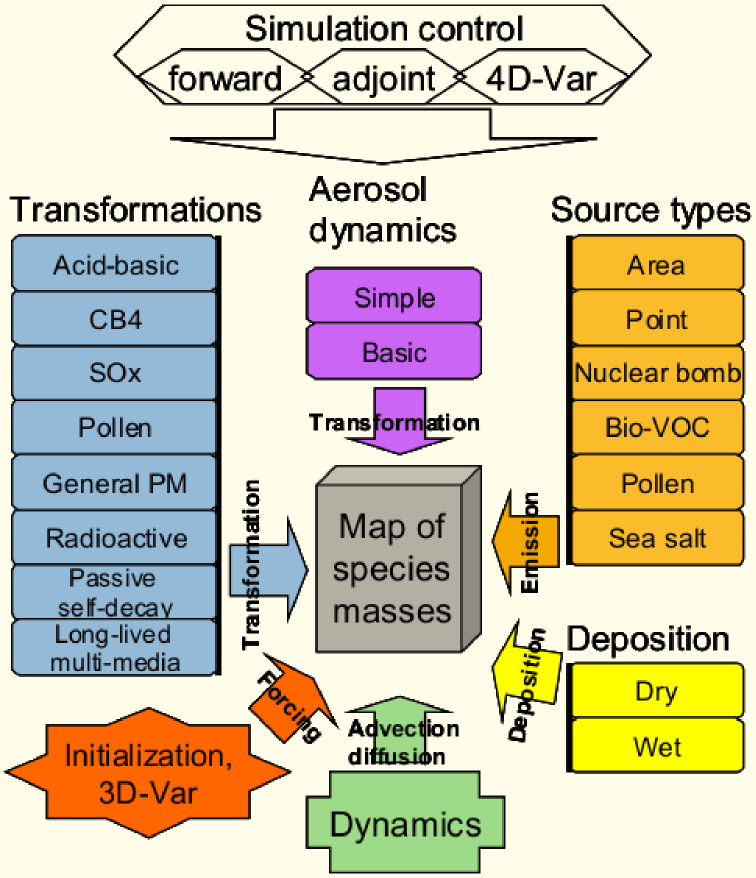

Fig. 1. Structure of SILAM.

The principles implemented in the model enable handling of virtually any species with any types of interactions between them. A single specie or a mixture of species transported in air is called *cocktail*. Each cocktail has specific species and characteristics regarding its composition. The chemical and physical transformations that a cocktail can endure are:

Transformation	Description	Emissions requested
PASSIVE	used for probability computations (*backwards* mode)	
PM_GENERAL	transport and deposition (no chemistry involved)	PM
DMAT_SULPHUR	linear chemistry for SO2 and SO4, transport and deposition.	SOx
ACID_BASIC	inorganic chemistry, transport and deposition.	CO, NOx, SOx & NH3
CB4	inorganic and organic chemistry, transport and deposition.	CO, NMVOC & NOx
POP_GENERAL		
RADIOACTIVE	radioactive transport and deposition features.	

2 Outline of the initialization/configuration files

SILAM may have up to ten input files depending on the complexity of the setup.

```
control file
├── general_parameters
│   └──internal model setup
│         ├── nuclide_database_fnm
│         ├── chemical_database_fnm
│         ├── standard_cocktail_fnm
│         ├── grib_code_table_fnm
│         ├── netcdf_name_table_fnm
│         ├── optical_properties_meta_data_file
│         ├── photolysis_data_file
│         └── timezone_list_fnm
│
├── emission_parameters
│   └── source term file/s
│
├── dispersion_parameters
│
├── meteo_parameters
│
├── transformation_parameters
│
├── initial_and_boundary_conditions
│   └── boundary_header_filename
│
├── optical_density_parameters
│
└── output_parameters
    └── output configuration file
```

Fig. 2. Structure of SILAM configuration files.

The mandatory files, for any run configuration, are:

- **control file:** general user-defined parameters of the run.
- **internal model setup:** sets the internal model features, usually read-only or fully invisible for users.
  - **standard cocktails file:** defines the standard cocktails that can be used in the source description[1].
  - **GRIB or NetCDF code table**: definitions for handle files, invisible for users.
- **source term file:** describes the emission sources.
- **output configuration file:** describes the output setup.

Depending of the configuration of the run, there are different files that should be included in the setup configuration:

- **nuclide data file:** for radioactive simulations, invisible for users, referred from the internal setup file.
- **nuclide decay data file:** for radioactive simulations, invisible for users, referred from the internal setup file.
- **land-use data:** for chemical simulations of biogenic emissions, invisible for users, referred from the internal setup file.

[1] Users are free to create their own cocktails, adding to the existent file.

- **optical properties:** for chemical and aerosol simulation, describes the optical properties of substances, invisible for users, referred from the internal setup file;
- **chemical properties:** describes the chemical properties of the species available in SILAM, invisible for users, referred from the internal setup file;

  Do NOT alter internal model files referred from the internal setup file: their modification may lead the model to malfunction. Only the source terms, control file, output configuration file, boundary file and the standard cocktails file should be modified in a standard run.

The structure of the mandatory files will be described in this document. Nuclide data file, nuclide decay file, chemical and optical properties files, land use data and GRIB/NetCDF code table files must **NOT** be altered by the user in any circumstances, and therefore are not included in the document.

3 Configuration files

3.1 General rules for the configuration files:

Configuration files are ASCII files, lines are *case-sensitive* and trailing blanks are ignored. Empty lines and commented lines are ignored. All characters after signs # or `!` are considered as comments[2].

When paths to files are requested, templates for dates and time are supported. > Example: `/path/to/data/my` `where`/path/to/data/`is thepathto the files and`%ay4%am2%ad2%ah2%f2‘ `is file name` itself. The name depends on the analysis time and forecast length of the fields stored in it.

Templates pointing to the analysis time (not allowed for the output files):

format	description	example
`%ay4`	4-digit year of the analysis time	*2002*
`%am2`	2-digit month of the analysis time	*01*
`%ad2%`	2-digit day of the analysis time	*05*
`%ah2`	2-digit hour of the analysis time	*07*
`%f2`	2-digit number of hours of the forecast	*015*

More about templates:

Analysis time, Forecast base time or first guess verification time (all usually at synoptic hours: 00, 06, 12 and 18). Templates pointing to the analysis time:

format	description	example
`%ay2`, `%ay4`	firmly 2- and 4-digit year of the analysis time	*02* or *2002*
`%am1`, `%am2`, `%amc`	1 or 2-digit; firmly 2-digit; 3-character month of the analysis time	*1*; *01* or *JAN*
`%ad1`, `%ad2%`	1 or 2-digit; firmly2-digit day of the analysis time	*5*; *05*
`%ah1`, `%ah2`, `%ah3`	1,2 or 3; 2 or 3; firmly 3-digit hour of the analysis time	*7*; *07*; *007*
`%an2`,	2-digit minutes of the analysis time	*15*
`%f2`, `%f3`	2- and 3-digit number of hours of the forecast length	*15*; *015*

Observation time (any combination in hours and minutes is valid, subject to data availability in the archive). Templates pointing to the valid time of the fields are constructed in the same way but without the *a*, e.g. %y2; %y4 – firmly 2- and 4-digit year of the analysis time (e.g. *02* or *2002*).

[2] Note: sign # always starts comments, while sign `!` starts comments ONLY if it is placed at the beginning of line or preceded by the empty space.

3.2 Rules for the namelist-type format:

A single file can include a group of namelists, placed one-by-one in arbitrary order. Each namelist starts from the line `LIST = <namelist_name>` and ends with the line `END_LIST = <namelist_name>`[3]. The `namelist_name` must be understood by the model.

The namelist content is placed between the `LIST` and `END_LIST` lines with the following format:`<item_name> = <item_value>` [4]. The `item_name` must be understood by the model and the `item_value` format and meaning fully depends on the `item_name` itself. The `item_value` may vary from a single number to a complicated line with several space-separated fields. The order of the namelist lines is arbitrary and unnecessary lines or lines with unknown `item_name` will be skipped by the model.

3.3 Control file

The control file is the main configuration file, where the model set-up is described. This file will also provide the link between the model and other necessary input files. A control file is always starting and ending with `CONTROL_V5` and `END_CONTROL_V5`. A control file is a namelist group that contains eight namelists:

- `general_parameters`
- `emission_parameters`
- `dispersion_parameters`
- `meteo_parameters`
- `transformation_parameters`
- `initial_and_boundary_conditions`
- `optical_density_parameters`
- `output_parameters`

each section starts and ends respectively by `LIST = <namelist>` and `END_LIST = <namelist>`. The model will only read what is stated between these two command lines.

Below sections describe the `item_names` for each namelist.

3.3.1 GENERAL_PARAMETERS

Here we set the name of the run, dates, time step, and type of run:

```
LIST = general_parameters
   case_name    = prueba
   system_setup = /home/rama/SILAM/ini/standard_eulerian.setup
   direction_in_time = FORWARD           !FORWARD/INVERSE
   start_time   = 2009 03 15 19 00 00
   end_time     = 2009 03 16 18 00 00
   time_step    = 60                     ![minutos]
   !computed_period = 23 hour
   !nbr_of_out_trajectories = 0          !solo si es lagrangiano.
   !progress_file_name =                 !archivo para debugging
   computation_accuracy = 5
END_LIST = general_parameters
```

[3] the blank spaces around the = character are mandatory

[4] the blank spaces around the = character are mandatory

variable	description	format / value
`case_name`	name of the run	%s
`system_setup`	path to standard setup file	%s
`direction_in_time`	direction in time of the run.	`FORWARD/INVERSE`
`start_time`	-	`%Y %m %d %H %M %s`
`end_time`	-	`%Y %m %d %H %M %s`
`computed_period`	-	`%d (hr/day/mon/yr)`
`time_step`	number of minutes (min)	`%d min`
`computation_accuracy`	[0..10]	%d

3.3.2 EMISSION_PARAMETERS

In this section define sources to be considered by the model.

```
LIST = emission_parameters

  emission_source = EULERIAN  emis/dust-simple/src_simple_dust.ini
  cut_area_source_if_outside_meteo_grid = YES

END_LIST = emission_parameters
```

variable	description	format / value
`emission_source`	type of source and path	
`cut_area_source_if_outside_meteo_grid`		YES/NO

The type of source and file depends if the emissions are computed by SILAM or not. SILAM's state-of-the-art is that natural PM emissions, such as sea salt (`SEA_SALT`), pollen (`POLLEN`), biogenic volatile organic compounds - VOC (`BIOGENIC_VOC`) and dust (`DESERT_DUST`) are computed by the model. When these types of sources are stated, the model request specific initialization files found in the `silam_v5_0/ini directory` (see section 3.6). Wild land fires source are currently obtained by the Fire Assimilation System at the Finnish Meteorological Institute (FMI) and has specific physical and chemical information for this type of emissions. Therefore, if using FMI wild-land fire emissions, `WILD_LAND_FIRE` should be the type to be stated.

3.3.3 DISPERSION_PARAMETERS

Mainly grid definitions and configuration.

```
LIST = dispersion_parameters
  grid_method = OUTPUT_GRID
  vertical_method = OUTPUT_LEVELS

END_LIST = dispersion_parameters
```

All geographical values are in degrees and decimal parts of a degree, NO MINUTES/SECONDS.

- `grid_type` = `lon_lat` Geographical coordinates grid is so far the only available
- `grid_title`. A name for the grid.
- `lon_start` and `lat_start` Area source's longitude and latitude of the first grid cell -ksec2(5), ksec2(4).

- dx and dy x- and y-direction increment (lon and lat) - ksec2(9), ksec2(10).10
- nx and ny Number of cells along the parallel ksec2(3), ksec2(2)
- lon_end andlat_end Area source's longitude and latitude of the last grid cell (ksec2(8), ksec2(7)). Not needed if nx and ny are defined. and meridian (varying lon and lat) - dx, lon_start are defined
- lat_s_pole Latitude of the south pole of rotation (-90. for geo) - ksec2(13)
- lon_s_pole Longitude of the south pole of rotation (0. for geo) - ksec2(14)
- lat_pole_stretch Latitude of pole of stretching (0 so far) - ksec2(15)
- lon_pole_stretch Longitude of pole of stretching (0 so far) - ksec2(16)
- resol_flag Resolution flag. DEFAULT: 128 = regular grid - ksec2(6),
- ifReduced Regular/reduced grid flag. DEFAULT: 0=regular - ksec2(17),
- earth_flag Earth-flag, 0=sphere, 64=oblate spheroid. DEFAULT: 0 - ksec2(18),
- wind_component Wind flag, 0=u,v relate to east/north, 8=u,v relate to x/y growing - ksec2(19),
- reduced_nbr_str Number of elements along the reduced direction, in one line - ksec2(23+)
- vertical_method OUPUT_LEVELS/METEO_LEVELS/CUSTOM_LEVELS.
  - If OUTPUT_LEVELS it assumes the same vertical levels defined for the output.
  - If METEO_LEVELS it assumes the same vertical level as the meteorological files
  - If CUSTOM_LEVELS the user has to set the levels by defining the following namelists:
    * level_type = HEIGHT_FROM_SURFACE / ALTITUDE_FROM_SEA / PRESSURE / HYBRID. There are 3 types of the output vertical allowed: z-, p- and hybrid systems, with corresponding units as: metres, hectoPascals or hybrid relative numbers. If the hybrid layers are selected, they MUST exist in the meteodata. The difference between the levels and layers is that levels are defined at one altitude, while layers cover the whole range between two levels. Dispersion output must be made into layers, while meteorology makes sense at levels too. Rules: z-, p- systems accept both THICKNESS of the layers and their CENTRAL POINTS; hybrid system accepts the NUMBER of the meteo hybrid and model will get the central point.
    * layer_thickness = Thickness of the output levels in [m]/[pa]/[hybrid_nbr] depending on the level type.

3.3.4 METEO_PARAMETERS

Mainly paths to meteorological data.

```
LIST = meteo_parameters
  dynamic_meteo_file = GRIB meteo/F4D%am2%ad2%ah200%m2%d2%h2001
  static_meteo_file = GRIB  meteo/ecglob100_VEG_%ay4%am2%ad2%ah2+00.sfc
  static_meteo_file = GRIB  meteo/era5_glob_physiography.sfc
  static_meteo_file = GRADS emis/sslt/salinity_map_global_1deg.fld_water_salinity.
  static_meteo_file = NETCDF:TZ_index meteo/tz_index_02deg.nc4
  meteo_time_step = 3 hr
  if_wait_for_data = NO
  abl_parameterization_method = FULL_PARAM
  number_of_precipitation_fields = 2
  max_hole_in_meteo_data = 0 hr
  use_lai = STATIC2
END_LIST = meteo_parameters
```

- dynamic_meteo_file = <file type> <file name>. where file type could take the values: GRIB / ASCII / NETCDF and is time dependent.
- static_meteo_file = <file type> <file name>. If static_meteo_file = -, the dynamic file is used. These files are not varying in time.
- meteo_time_step. Weather data time interval: number and unit, integer > 0.
- if_wait_for_data = YES/NO, if yes, model will waits for the missing meteorological files.

- abl_parameterization_method = DRY_ABL/FULL_PARAM. Sets the methodology for the boundary layer height computation. The methods available for the computation are DRY_ABL and FULL_PARAM.
  - DRY_ABL parameterization is computing atmospheric boundary layer without humidity correction
  - FULL_PARAM includes humidity correction. DRY_ABL is the common used method.
- number_of_precipitation_fields = 1/2. If only large-scale rain is required and available the user should use 1; if both convective and large-scale rain required and available the user should use 2. Typically both fields are required.

3.3.5 TRANSFORMATION_PARAMETERS

Sets the chemical and physical processes undergoing during the computation, depending on the emissions available [5].

```
LIST = transformation_parameters
  transformation = CB5_SOA EULERIAN
  aerosol_dynamics = SIMPLE  EULERIAN
  dry_deposition_scheme = KS2011_TF
  surface_resistance_method = WES2013
  wet_deposition_scheme = 2018_SCAVENGING
  max_scav_rate_depends_on = CAPE
  use_dynamic_albedo = YES
  if_actual_humidity_for_particle_size = YES
  default_relative_humidity = 0.8
  passive_subst_ref_lifetime = 1000000 day
  passive_subst_ref_tempr = 288
  passive_subst_dLifeTime_dT = 0 min/K
  passive_ones_tracer = NO
  mass_low_threshold = STANDARD_ACCURACY
  oh_param_method = FROM_MASSMAP
  biogenic_SOA_aging_rate = 1.2E-11
  anthropogenic_SOA_aging_rate = 4.0E-11
  intermediate_volatility_OC_aging_rate = 4.0E-11
  if_monoterpene_products = 1.0
  if_full_acid_chemistry = YES
  make_coarse_no3 = sslt   0.03
  methylchloroform_OH_rate_factor = 1.0
  photolysis_affected_by_o3col = YES
  photolysis_affected_by_aod = YES
  photolysis_AOD_wavelength = 550 nm
  cloud_model_for_photolysis = SIMPLE_CLOUD
  cbm_tolerance = FAST
END_LIST = transformation_parameters
```

- transformation = transformations schemes already discused.
  - PASSIVE
  - PM_GENERAL
  - DMAT_SULPHUR
  - CB4
  - POP_GENERAL
  - ACID BASIC

[5] Notice that several can be co-existing except the chemical transformations.

- `aerosol_dynamics` = `SIMPLE`, sets the methodology for including aerosol dynamics processes.
- `dry_depostion_scheme` Method for the dry deposition: settling and viscous sub-layer resistance.
  - `SIMPLE_DIFFUSION_ONLY` is only considering viscous sub-layer resistance.
  - `FULL_DIFFUSION_ONLY`
  - `GRAVITATIONAL_ONLY` gravitational settling.
  - `GRAVITATIONAL_AND_SIMPLE_DIFFUSION`
  - `GRAVITATIONAL SETTLING_AND_FULL_DIFFUSION`
  - `GRAVITATIONAL_AND_FULL_DIFFUSION` is typically used.
- `wet_depostion_scheme` = `STANDARD_3D_SCAVENGING`. The only wet deposition method available.
- `if_actual_humidity_for_particle_size` = `YES/NO`. Sets if humidity is time resolving or not.
- `default_relative_humidity`. default value for relative humidity (typically number is 0.8).
- `compute_thermodiffusion` = `YES/NO`. Sets if the model computes thermodiffusion or not[6].
- `mass_low_threshold = CRUDE_ACCURACY / STANDARD_ACCURACY / HIGH_ACCURACY`. Sets the accuracy for the computation of the low-mass threshold for the Eulerian setup [7].
- `if_full_acid_chemistry` = YES/NO. Sets if nitrogen chemistry is computed or not; method to compute biogenic VOC emissions (only for transformations `ACID_BASIC` and `CB4`)[8].

If `PASSIVE` transformation is set some parameters such as lifetime, temperature and degradation with temperature should be specified: - `passive_subst_ref_lifetime` - `passive_subst_ref_tempr` - `passive_subst_dLifeTime_dT`

If aerosol dynamics is taken into account, some items should be specified: - `ADB_if_compute_nucleation` - `ADb_nucleation_scheme`, - `ADB_if_compute_coagulation` - `ADB_if_compute_condensation` - `ADB_if_compute_cloud_activation` - `ADB_if_compute_recalcu_wet_d`

3.3.6 INITIAL_AND_BOUNDARY_CONDITIONS

```
LIST = initial_and_boundary_conditions
  initialize_quantity = concentration
  initialize_quantity = advection_moment_x
  initialize_quantity = advection_moment_y
  initialize_quantity = advection_moment_z
  initialization_file = GRADS ${OUTPUT_DIR}/%ay4%am2%ad2/%y4%m2%d2%h2_%y4_%m2_%d2_

  boundary_type = DIRICHLET
  if_lateral_boundary = YES
  if_top_boundary = YES
  if_bottom_boundary = NO
  boundary_time_step = 3 hr
  boundary_header_filename = boundary_CB5_globalCB4.ini

END LIST = initial_and_boundary_conditions
```

For initializing a run the user can set:

- `initialize_quantity`. Describes which quantity is being initialized. The typical case is concentration.
- `initialization_file = <file type> <file name>`. can be `GRIB`, `GRADS` and `POINT_DATA`. If GRADS type the file to be used is a super ctl file. This file is the standard output of any SILAM run.

[6] Normally set to NO.

[7] Normally set to `HIGH_ACCURACY`.

[8] Normally set as YES.

For setting boundary conditions:

- boundary_type = ZERO/DIRICHLET. Boundaries can be static (ZERO) or timeresolving (DIRICHLET)
- if_lateral_boundary = YES/NO. If lateral boundary is or not set to the values prescribed in the boundaries file.
- if_top_boundary = YES/NO. If top boundary is or not set to the values prescribed in the boundaries file
- if_bottom_boundary = YES/NO. If bottom boundary is or not set to the values prescribed in the boundaries file
- boundary_time_step =
- boundary_header_filename. Filename of the file describing the concentrations at the boundaries.

The boundary file itself maps input data concentration for boundaries and transport species.

3.3.7 OPTICAL_DENSITY_PARAMETERS

This namelist describes the parameters needed for the optical density calculation:

```
LIST = optical_density_parameters
  optical_coefficients_depend_on_relative_humidity = YES
  optical_coefficients_depend_on_temperature = YES
  if_split_aerosol_modes = YES
  if_narrow_wave_bands = YES
END_LIST = optical_density_parameters
```

- optical_coefficients_depend_on_relative_humidity = YES/NO dependency the optical properties on relative humidity.
- optical_coefficients_depend_on_temperature = YES/NO dependency the optical properties on temperature.
- if_split_aerosol_modes not working yet.
- if_narrow_wave_bands not working yet.

3.3.8 OUTPUT_PARAMETERS

```
LIST = output_parameters
   source_id = NO_SOURCE_SPLIT  # SOURCE_NAME  SOURCE_SECTOR  SOURCE_NAME_AND_SECT

   output_time_step = 1 hr
   output_times = REGULAR
   output_format = NETCDF3
   time_split = ALL_IN_ONE

   template =  output/%case  !(if time spliting should give names with template fo
   variable_list = output_config.ini

   !custon grid
   grid_method = CUSTOM_GRID
   grid_type = lon_lat
   grid_title = GEMS output grid
   resol_flag = 128
   ifReduced = 0
   earth_flag = 0
   wind_component = 0
   reduced_nbr_str = 0
   nx = 50
   ny = 63
   lon_start = -72.0
   lat_start =-54.0
   dx = 0.4
   dy = 0.4

   lat_s_pole = -90.
   lon_s_pole = 0.
   lat_pole_stretch = 0.
   lon_pole_stretch = 0.

   !vertical layers:
   vertical_method = CUSTOM_LAYERS
   level_type = HEIGHT_FROM_SURFACE
   reference_4_low_mass_threshold = CONST  !EMISSION or DEFAULT
   layer_thickness = 25. 50. 100. 200. 400. 750. 1200. 2000. 2000   # output level

END_LIST = output_parameters
```

- `source_id` = `NO_SOURCE_SPLIT / SOURCE_NAME / SOURCE_SECTOR / SOURCE_NAME_AND_SECT` Controls mixing or splitting of the plumes from individual sources in the output files. In case of MIX_SOURCES, the plumes are mixed, so that all the sources create a single output field or trajectory set. If sources are split – each plume from the corresponding source is put into its own file, thus creating a surrogate for the source-receptor matrix computations. The source may have name and sector – and they both can be used for the creation of the source ID (NO_SOURCE_SPLIT) or according to source name and/or sector.

- `vertical_method` = `OUPUT_LEVELS/METEO_LEVELS/CUSTOM_LEVELS`

- `output_time_step`. Output timestep and unit

- `output_times` = REGULAR (standard)

- `file_type` = `GRIB_YES/NO TRAJECTORY_YES/NO GRADS_YES/NO ENSEMBLE_YES/NO NETCDF_YES/NO`. This namelist defines the type of output file 16required, by setting the type of output to YES or NO. The type of output can be GRIB, GRADS, NETCDF and ensemble for Eulerian setup and trajectories for Lagragian setup.

- `time_split` = `ALL_IN_ONE / HOURLY_NEW_FILE / DAILY_NEW_FILE / MONTHLY_NEW_FILE/ YEARLY_NEW_FILE`, depending of how the user wants these files to be stored, bearing in mind that this is just to store since the ouput averaging is set by `output_time_step`.

- `template`. %case%case_%y4%m2%d2%h2 time template depends on the `time_split` chosen

- `variable_list`. Path for output_config file.

- `grid_method` = `EMIS_GRID / METEO_GRID / AREA_BASED / CUSTOM_GRID`. Grid definition for the output files. The same definition as emission or meteorological files (EMIS or METEO_GRID) or according to specific needs. If AREA_BASED, the output area and required resolution have to be defined:

- `area_borders` = ; North positive, east positive; all real.

- `area_title`. A name for the area defined

- `resolution`. Horizontal grid size of output grid, [km]/[m]/[deg], real

If `grid_method = CUSTOM_GRID` is set a full definition of the grid has to be described. If `vertical_method = CUSTOM_LEVELS` is set, then `level_type` and `layer_thickness` should be specified.

3.4 Source term files

3.4.1 Point source v.5

This source term is compatible for forward and backward runs. The source file may contain several sources of this type, as well other types, as long as each source is defined by starting and ending with: PONIT_SOURCE_5 and END_POINT_SOURCE_5, these lines are mandatory!!.

```
POINT_SOURCE_5
source_name = TOYPOINT
source_sector_name =        # source sector name, e.g. SNAP_10. May be empty

source_longitude = -70      # start geograph. lat., degrees and decimals, N positi
source_latitude = -50       # start geograph. lon., degrees and decimals, E positi

plume_rise = NO
release_rate_unit = kg/sec  # Unit of the release rate: <mass>/<time>
                            # [kg][g][t][bq][mole] - mass(radioactivity);
                            # [yr][mon][day][hr][min][sec] - time units
vertical_unit = m  #hpa     # unit of the vertical release boundaries [hpa] or [m]
vertical_distribution =  SINGLE_LEVEL_DYNAMIC # SINGLE_LEVEL_DYNAMIC, MULTI_LEVEL_
stack_height = 10 m

par_str_point = 2009 03 15 19 00 0.0    1.    500. 1000.   5.0   450.  PASSIVE_COC
par_str_point = 2009 03 15 20 00 0.0    1.    500. 1000.   5.0   450.  PASSIVE_COC

# Time variation indices - separate set for every cocktail
hour_in_day_index = PASSIVE_COCKTAIL 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
day_in_week_index = PASSIVE_COCKTAIL 1. 1. 1. 1. 1. 1. 1.
month_in_year_index = PASSIVE_COCKTAIL 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.

END_POINT_SOURCE_5
```

- source_name. Source name. The source name has to be different if there are other sources.
- source_sector_name. Normally according to EMEP's sector denomination. May be empty.
- source_longitude. Source's geographical longitude, degrees and decimals, N positive, E positive.
- source_latitude. Source geographical latitude, degrees and decimals, N positive, E positive.
- plume_rise = PLUME_RISE_YES / PLUME_RISE_NO. Activates the buoyant plume rise routine
- release_rate_unit = / (no spaces!!): mass: kg][g][t][Bq][mole][number] time: [yr][mon][day][hr][min][sec]
- vertical_unit. Unit of the vertical release boundaries [hpa] or [m]
- par_str is the time definition of the source if time of release is fixed-in-time source, fixed-in-time release is defined via two lines with identical parameters and with start and end time of the release. The source is activated at current moment ("NOW") or at last-most meteorological time ("LAST_METEO_TIME") and will continue constant-in-time release during the given duration.
- par_str = [NOW]/[LAST_METEO_TIME] <duration [min]>
- par_str = [NOW]/[LAST_METEO_TIME] <duration [min]>

if time of release is varying source, the first line determines the start of the release and last line determines the end of the release. There are an arbitrary number of lines and if two sequential lines have different release parameters, every parameter will be linearly interpolated between these times. A varying source is defined by a 4-digit year and a 2-digit month, day, hour and minute, seconds is a real value with mandatory decimal dot. - par_str = - par_str =

The release rate () is the value of the release in the units defined by release_rate_unit (above). The horizontal size () is the diameter of the source since sources are assumed to be circles. Bottom and top are the vertical boundaries of the emitted cloud (unit: meters or hPa). If the plume-rise routine is activated, the boundaries must be the same and correspond to physical height of the source. The vertical velocity (z-velocity) is the velocity of the plume at the top of stack (unit: meters per second). Temperature at the top of the stack of outgoing gases is defined by (tempr). The release composition (cocktail_name) points to one of the standard cocktails. - hour_in_day_index. Diurnal relative intensity considering 24 hours in day. - day_in_week_index. Week-day relative intensity considering 7 days in a week. - month_in_year_index. Monthly relative intensity considering 12 months in a year.

3.4.X Wind-blown dust sources

```
WIND_BLOWN_DUST_SOURCE_V1
    source_name = wb_dust
    source_sector_name = natural

    wind_blown_dust_emission_method = SIMPLE_DUST    # GILLETTE_DMAT or SANDBLASTI
    wind_blown_dust_spectrum = LOGNORMAL_FOUR_MODES  # internal, to be projected t

    wind_blown_dust_substance_name = dust
    aerosol_mode = 1  0.01 1.   0.3  mkm
    aerosol_mode = 2  1.   2.5  1.5  mkm
    aerosol_mode = 3  2.5  10.  6.   mkm
    aerosol_mode = 4  10.  30.  20.  mkm
    mode_distribution_type = FIXED_DIAMETER   ! later also: GAMMA_FUNCTION

    ## For some reason no grads hats supported here...
    supplementary_file = NETCDF:dust_emis_0  ^dust_emis_0_v3.nc4

END_WIND_BLOWN_DUST_SOURCE_V1
```

3.4.X Biogenic VOC sources

```
BIOGENIC_VOC_SOURCE_V1

source_name = bio_voc_standard           ! free source name
source_sector_name = natural_emission    ! free sector name

source_mask_file = GRIB e:\data\meteo\EC_OPER\ec_land_use_global.sfc  ! essentiall

bvoc_emission_method = GUENTHER_MODIFIED_V1    ! the only one available so far
land_use_meta_data_file = d:\!model\silam_v5_7\ini\land_use_features_USGS_Eurasia.

if_emit_isoprene = YES
if_emit_monoterpene = NO

END_BIOGENIC_VOC_SOURCE_V1
```

3.4.X Fires sources

```
FIRE_SOURCE_V1
```

```
  source_name = fire20100501
  source_sector_name = fire
  number_of_fires = 5  ! different fires
  max_number_of_same_fire_observations = 2
  fire_metadata_file = d:\data\emission\fires\fire_metadata_ecodata_Acagi_PM_v5_5.
  mode_distribution_type = FIXED_DIAMETER   ! later also: GAMMA_FUNCTION

  aerosol_mode = 1  0.01  0.1  0.05 mkm  ! mode_number Dmin, Dmax, Daver D_unit
  aerosol_mode = 2  0.1   1.5  0.5  mkm
  aerosol_mode = 3  1.5   6.   3    mkm
  aerosol_mode = 4  6.   15.   9.   mkm
  aerosol_mode = 5  15.  30.  20.   mkm

  frp_dataset = d:\public\2018\ozone_depletion_Finland\Fire_MOD_MYD_coll_6__FRP_fa

END_FIRE_SOURCE_V1
```

3.4.X Sea salt sources

```
SEA_SALT_SOURCE_V5

source_name = sea_salt_standard
source_sector_name = natural_emission    ! free sector name
source_area_mask = GRADS ^eco_collection_water_bodies.ctl.super_ctl

#sea_salt_emission_method = HYBRID_WIND_10M
sea_salt_emission_method = HYBRID_AND_SPUME_WIND_10M
water_temperature_input_type = DYNAMIC  ! FIXED_VALUE / FIXED_MAP / MONTHLY_CLIMAT
sea_salt_emis_depend_on_water_salinity = YES   ! YES / NO
sea_salt_emis_depend_on_ice_fraction = NO      ! YES / NO
default_water_salinity = 0.033                 ! as a fraction
default_water_temperature = 288                ! K
min_open_water_area_fraction = 0.0             ! fraction
wind_selection = WIND_LEVEL_1

sea_salt_substance_name = sslt                 ! must be in chemical database

# PM2.5 = mode 1 + mode 2
# PM10  = PM2.5  + mode 3
aerosol_mode = 1  0.01  0.1  0.05 mkm          ! mode_number Dmin, Dmax, Daver D_u
aerosol_mode = 2  0.1   1.5  0.5  mkm
aerosol_mode = 3  1.5    6.  3    mkm
aerosol_mode = 4  6.   15.   9.   mkm
aerosol_mode = 5  15.  30.   20.  mkm

mode_distribution_type = FIXED_DIAMETER        ! later also: GAMMA_FUNCTION

END_SEA_SALT_SOURCE_V5
```

3.4.X Dimethylsulfide (DMS) sources

```
DMS_SOURCE_V5
```

```
# Name and sector, optional
# source_name = dms
# source_sector_name = natural_emission

source_area_mask = NETCDF:dms_clim sslt_emission_global_50km.fld.nc

# DMS climatology: needs be in mol/dm3 (=M), in a specific format
dms_map_filename = NETCDF:dms_clim dms_lana_2011.nc

# Emitted substance: any gas, DMS by default
emitted_substance = DMS

# Yield: by default 1.0 for DMS, otherwise need to give explicit.
yield = 1.0

END_DMS_SOURCE_V5
```

3.5 Output configuration file

The output post-processor allows the user to select flexible averaging for each dispersion variable and to include any SILAM internal meteorological variable to the output. The output variable categories are:

- general characteristics of the output variables
- dispersion
- meteorological
- nuclides

The output configuration file should be starting and ending with OUT_CONFIG_3_7 and END_OUT_CONFIG_3_7. These lines mandatory!! This file has a single namelist that should be started and ended by LIST = OUT_CONFIG_3_7 and END_LIST = OUT_CONFIG_3_7. The content between the namelist defines the output available.

The general characteristics of the output variables category basically describes how to report the aerosol sizes: as one size (SUM) or different sizes, as described in the cocktail description (SEPARATE), see section 3.8. - aerosol_size_mode = SEPARATE/SUM

The remaining categories have arbitrary number of lines containing three or four or five fields, depending of the output variable category requested. The general format goes:

- out_var = <necessity_index> <variable_name> <substance_name/lists> <averaging> with optical properties:
- out_var = <necessity_index> <variable_name> <substance_name/lists> <averaging> <wave_lenght> with meteorological variables:
- out_var = <necessity_index> <variable_name> <averaging>

To request or not a variable, there is a necessity index that is placed after the out_var item list:

- 0 : quantity is not needed
- 1 : quantity is desirable, but if is not available the model run will not be discontinued
- 2 : mandatory variable for the output, if the variable is not available, the model run will be interrupted.

The variable name is fixed by the model, and the user just has to use the necessity indexto switch on or off that variable output request.
The substance name/lists is set according to the availability of substances and the user necessity. If the run is not for an individual substance, there can be requested:

- SOURCE_INVENTORY just the substances emitted.
- FULL_INVENTORY when requested all the substances present in the dispersion cloud. The averaging type for the particular variable is set by the user according to the user's needs. The available types of averaging are:
- AS_IS – the field comes to the output exactly as it was stored in SILAM internal buffers at the moment of output collection
- INSTANT – cumulative field is converted to their mean rates between the last two model time steps, while the instant variables go as they are
- CUMULATIVE – the variable is accumulated since the beginning of the simulations
- AVERAGE – the variable is averaged from the previous to the current output time
- MEAN_LAST_**_HR – the field is averaged over the given period preceding the current output. The period must not be longer than the interval between the outputs.

The wavelength (units: nm) is set by the user. The optical properties of the substance name/list are set for this specific wavelength.

3.6 Boundary header file

The boundary header file describes the information about the boundary fields to be used by the model; the user should edit this file accordingly. The figure below shows an example of a boundary header file. *This file does not need a beginning and end namelist.*

```
boundary_file = bnd/out/bcon.nc
file_format = NETCDF      ! GRIB/ASCII/GRADS/NETCDF
boundary_names = NSEWT    ! NSEWTB
ifDynamic = YES           ! YES/NO
ifClimatology = NO        ! YES/NO

par_str = dust dust 0.3e-6 0.3e-6 1.0
par_str = dust dust 1.5e-6 1.5e-6 1.0
par_str = dust dust 6e-6 6e-6 1.0
par_str = dust dust 20e-6 20e-6 1.0
```

- boundary_file =
- file_format = GRIB/ASCII/GRADS/NETCDF. Format of the input files
- boundary_names = NSEWTB. Description of which boundaries of the domain that are emitting: N = north, S = south, E = east, W = west, T = top and B = bottom.
- ifDynamic = YES/NO
- ifClimatology = YES/NO, is the time resolution of the boundaries is climatological or not.
- climatologyTimestep = MONTHLY/STATIC this item will only be used ifClimatology = YES, and varies if the files are time dependent (MONTHLY) or not (STATIC).
- nBoundSpecies = , number of species to be read from the boundary files.
- par_str = <boundary_substance_name> <model_substance_name> <boundary_substance_m <model_substance_mode> <conversion_factor>

The same substance might have different name in the boundary fields and in the model, therefore it is necessary to define the name of the substances required, as well as their mode. In case of gases the mode is zero. the conversion factor might be necessary if the user finds it more suitable to convert the emissions to a, e.g. SI unit.

3.7 Internal model setup

The internal setup file is the file that provides other configuration files that are needed for running SILAM model. This file is only open for user to write the correct path for the files mentioned in this file, see Figure

15. These files are included in SILAM package and are essential for the model to run.

```
 LIST = STANDARD_SETUP
  advection_method_eulerian = EULERIAN_V5           !EULERIAN_V4/EULERIAN_3D_BULK/
  advection_method_lagrangian = LAGRANGIAN_WIND_ENDPOINT_3D
  advection_method_default = EULERIAN
  continuity_equation = anelastic_v2     !incompressible|incompressible_v2|anelast
  abl_height_method = COMBINATION    !richardson_method/parcel_method/combination
  kz_profile_method = SILAM_ABL_EC_FT_KZ !SILAM_RESISTANCE/SILAM_ABL_EC_KZ
  random_walk_method = FULLY_MIXED   !NONE/FIXED/FULLY_MIXED/BULK_GAUSSIAN
  mass_distributor = TRIANGLE_SLAB   !TRIANGLE_SLAB/RECTANGLE_SLAB/STEP_SLAB
  diffuse_vert_cm = YES
  reference_4_low_mass_threshold = CONST

  horizontal_interpolation = LINEAR
  vertical_interpolation = LINEAR
  time_interpolation = LINEAR

  standard_setup_directory = ini
  nuclide_database_fnm = ^silam_nuclides.dat
  chemical_database_fnm = ^silam_chemicals_95_OC.dat
  standard_cocktail_fnm = ^standard_aerosols_cocktails.ini
  standard_cocktail_fnm = ^standard_auxillary_cocktails.ini
  grib_code_table_fnm = ^grib_code_table_v5.silam
  netcdf_name_table_fnm = ^netcdf_name_table.silam
  optical_properties_meta_data_file = ^optical_properties.dat
  photolysis_data_file = ^photolysis_finrose.dat
  timezone_list_fnm = ^tzindex.dat
  allow_zero_forecast_length = NO
  print_debug_info = DEBUG_INFO_YES
  cloud_report_interval = 1
  max_hole_in_meteo_data = 6 hr
  disregard_meteo_data_sources = YES
 END_LIST = STANDARD_SETUP
```

3.8 Standard cocktails

Cocktail description files contain lists of cocktails. Cocktail description consists of the cocktail name, type, unit of fractions and then a list of species with their fractions (in corresponding unit) in the cocktail. The description starts from header and ends with end line: COCKTAIL_DESCRIPTION_V3_2 and END_COCKTAIL_DESCRIPTION_V3_2. The cocktail may contain the gas and/or aerosol description. Standard cocktails can be used by their names in the source term files. An example of cocktail description is given in Figure 16. Depending on whether the aerosol size classes are defined, the fractions have somewhat different meaning. A total mass fraction of each substance in the mixture comes as a sum of fractions of the substance in the aerosol classes and/or gas phase.

- cocktail_name= random name
- mass_unit = Bq/number/mass
- gas_phase = YES/NO
- aerosol_mode =
- aerosol_distribution_shape = FIXED_DIAMETER (so far the only available).
- component_fraction = , there should be as many component_fraction lines as the number of substances that the user is trying to simulate. Only substances available in silam_chemicals.ini file should be added to the cocktail. If gas_phase = YES and aerosol modes coexist,

- `component_fraction` = `number_of modes`* If gas_phase = NO,
- `component_fraction` = number_of modes* If `gas_phase` = YES and no aerosol phase,
- `component_fraction` =

4 Running the model

There is only one argument to be given to run the model, the control file name. This can be done via one of the following command line constructions in a command prompt window.

Notations below are:

- `<program>` is the path and or name of the SILAM executable.
- `<control_file>` is the path or name of the control file.

```
> <program>
```

No arguments. The program will open the file `silam.ini` in the working directory and read the name of the control file from the namelist: `control_file = <control_file>`

```
> <program> <ini_file_name>.
```

One argument, which is treated as a main ini-filename instead of `silam.ini`. This file must contain the namelist as described above.

```
> <program> <control_file>
```

The file is given explicitly as an argument.

The user can simply click on the model executable if the silam.ini file is available, but it is recommend using command prompt for a better reporting of possible errors.

In case of Linux-based users, a run with SILAM can be set with several threads since the model is by default compiled with OpenMP based parallelization enabled. By default, the code will then use the default number of threads, which is usually the number of physical or logical cores.