User-guide for SILAM chemical transport model

Contents

1 Substances and their transformation in SILAM v5

SILAM v5 is capable of computation of dispersion of up to 496 different nuclides, together with their radioactive decays and transformations; inert and chemically active size-specific aerosol; biological material (pollen grains); chemically active gases. And this is so called the “forward” mode. The model also computes probabilities, 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.

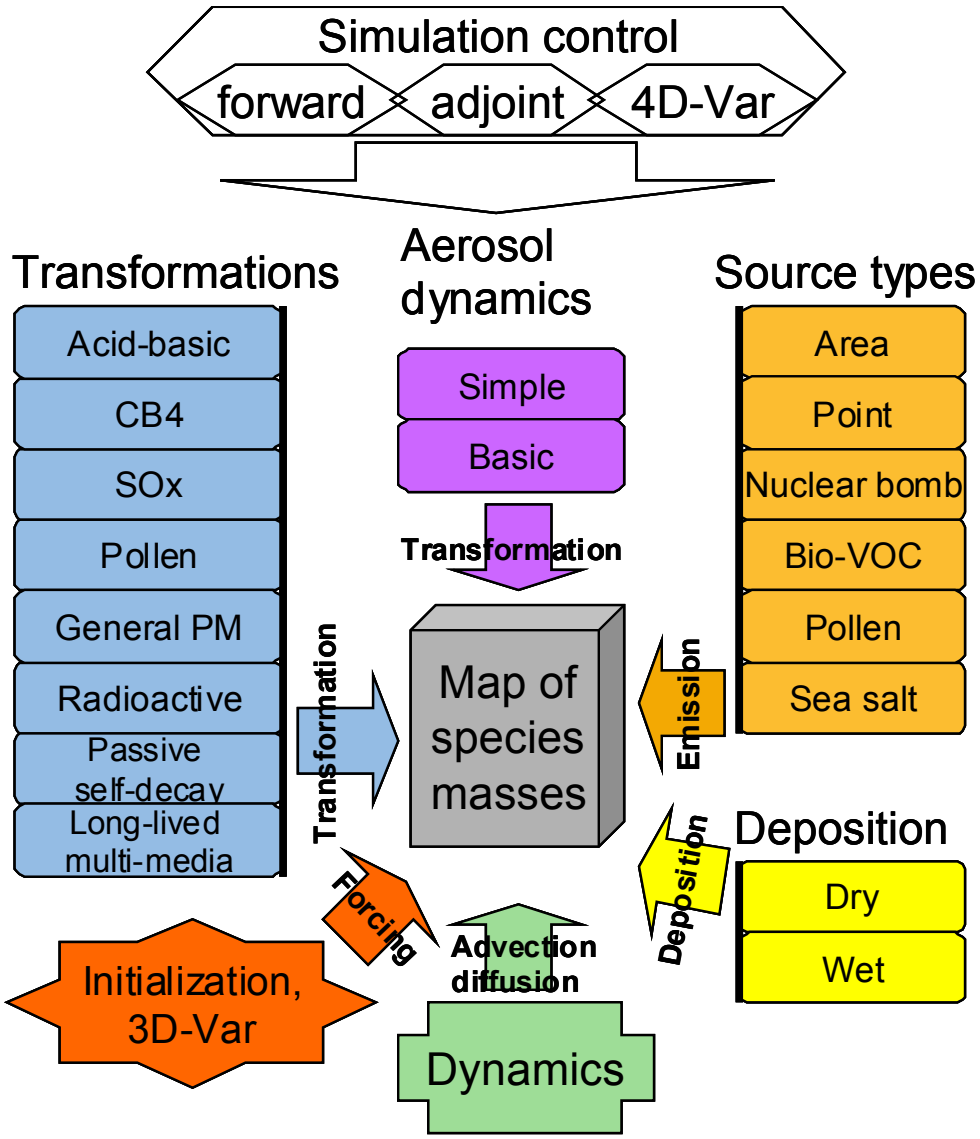

Figure 1. Structure of SILAM v5.

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 (see section 3.4.1). The chemical and physical transformations that a cocktail can endure are:

- PASSIVE – used for probability computations. In the case of probabilities, the model is then set in the “backwards” mode, where the source is representing the measurements of a measurement site and the result is the probability of probability of a certain grid cell to be contributing to that measurement.

- PM_GENERAL - no chemical reactions involved; only transport and deposition. Emissions requested: PM
- DMAT_SULPHUR – linear chemistry for SO2 and SO4, transport and deposition. Emission requested: SOx
- ACID_BASIC – inorganic chemistry, transport and deposition. Emissions requested: CO, NOx, SOx and NH3
- CB4 – inorganic and organic chemistry, transport and deposition. Emissions requested: CO, NMVOC and NOx
- POP_GENERAL
- RADIOACTIVE - cocktail has inherent radioactive transport and deposition features

2 Outline of the initialization/configuration files

SILAM v4.5.4 may have up to ten input files depending on the complexity of the setup (see Figure 2).

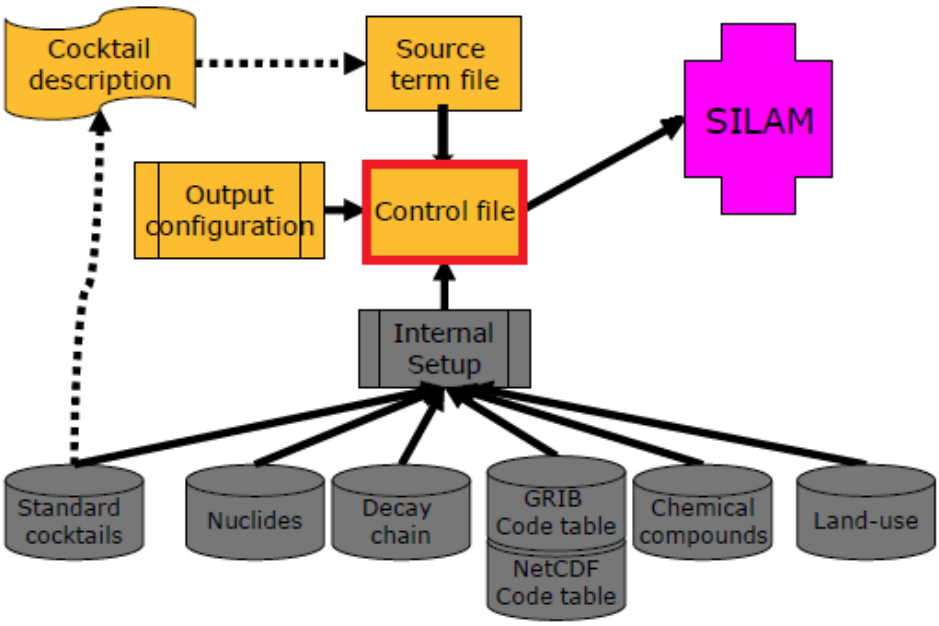

Figure 2. A structure of SILAM v5 configuration files.

The mandatory files, for any run configuration, are:

- **control file**: sets the user-defined parameters of the run;
- **source term file**: describes the emission sources: referred from the control file;
- **output configuration file**: description of the output setup, referred from the control file;
- **internal model setup**: referred from the control file, 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; referred from the internal setup file. Users are free to create their own cocktails, adding to the existent file;
- **GRIB or NetCDF code table** (depending of the type of files): mandatory, invisible for users, referred from the internal setup file.

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;
- **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, the control file, the output configuration file, boundary file and the standard cocktails file are open to everyone**. 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:

- All configuration files are in standard text (ASCII) files following one of the two standard formats: fixed-structure file or namelist-type file. Output configuration and internal files follow the fixed-structure format, and control and source files are following the namelist-type format.

- Each file consists of a set of lines, with leading and trailing blanks ignored.

- Lines are case-sensitive.

- Empty lines and commented lines are ignored. All characters after signs "#" or " !" are considered as comments. 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.

- Path and names of files are written in a usual format <full_path><file_name>, where both <path> and <name> can vary depending on their content. In case of including templates (commonly recognized by having "%" character):

  Example: /data/hirfields/fc%ay4%am2%ad2%ah2%f2

  → /data/hirfields/ is the path to the files

  → %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 (<u>not allowed for the output files</u>):

  → %ay4 – 4-digit year of the analysis time (e.g. "2002")

  → %am2– 2-digit month of the analysis time (e.g. "01" or "JAN")

  → %ad2% - 2-digit day of the analysis time (e.g. "05")

  → %ah2 - 2-digit hour of the analysis time (e.g. "07")

  → %f2 – 2-digit number of hours of the forecast length (e.g. "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:

  → %ay2; %ay4 – firmly 2- and 4-digit year of the analysis time (e.g. “02” or “2002”)

  → %am1; %am2; %amc – 1 or 2-digit; firmly 2-digit; 3-character month of the analysis time (e.g. “1”; “01” or “JAN”)

  → %ad1; %ad2% - 1or 2-digit; firmly2-digit day of the analysis time (e.g. “5”; “05”)

  → %ah1; %ah2; %ah3 – 1 or 2 or 3-digit; 2 or 3-digit; firmly 3-digit hour of the analysis time (e.g.“7”; “07”; “007”)

  → %an2 – 2-digit minutes of the analysis time (e.g. “15”)

  → %f2; %f3 – 2- and 3-digit number of hours of the forecast length (e.g. “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”)

3.2 Rules for the namelist-type format:

- A single file includes a group of namelists, placed one-by-one in arbitrary order.

- A single namelist starts from the line “LIST = <namelist_name>” and ends with the line “END_LIST = <namelist_name>”, the blank spaces around the “=” character are mandatory. The namelist_name must be understood by the model.

- The namelist content is placed between the LIST – END_LIST lines with the following format: <item_name> = <item_value>, the blank spaces around the “=” character are mandatory. The item_name must be understood by the model and the item_value format and meaning fully depends on the item_name. 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_V4 and END_CONTROL_V4. These lines mandatory!! The model will only read what is stated between these two command lines. Grid and vertical definitions follow the standards of the GRID format.

A control file is a **namelist** group that contains eight namelists:

- general_parameters
- emission_parameters
- dispersion_parameters
- meteo_parameters
- transphormation_parameters
- initial_and_boundary_conditions
- optical_density_parameters
- output_parameters

that start and end 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 Namelist general_parameters

```
CONTROL_v4_7

LIST = general_parameters
   case_name = <case_name>
   system_setup =  d:\model\silam_v5_0\ini\standard_eulerian.setup
   direction_in_time = FORWARD
   start_time = 2007 7 25 0 0 0.
   end_time = 2009 7 1 0 0 0.
#   computed_period = 65 day
   time_step = 15 min
   nbr_of_particles = 100
   nbr_of_out_trajectories = 0
#   progress_file_name = /scratch/tmp/silam2/out_air_quality/GEMS_cb4/progress
 END_LIST = general_parameters
```

Figure 3. SILAM's control file: namelist general parameters

- **case_name**: name of the run

- **system_setup**: full path and name of the lagragian or eulerian standard setup file in the silam_v5_0/ini directory, depending of running an eulerian or lagragian type of dispersion.
- **direction_in_time** = FORWARD / INVERSE. The direction in time of the run.
- **start_time** = <year> <month> <day> <hour> <minute> <sec>.
- **end_time** = <year> <month> <day> <hour> <minute> <sec>.
- **computed_period** = number of hours (hr), days (day), months (mon) or years (yr), if end_time was not stated. User can choose one of the item_names.
- **time_step** = number of minutes (min), time-step of the model
- **nbr_of_particles** and **nbr_of_trajectories**: if the user intends to use the lagragian dispersion, the number of particles (nbr_of_particles) has to be stated and and number of trajectories (nbr_of_trajectories) has to be different than zero.
- **progress_file_name** = path and name of the file for debugging purposes, common user should comment it out
- **computation_accuracy** = [0..10], the user should state if the results should be computed with low (0) or high accuracy (10). 5 is commonly used since the higher the accuracy the longer the computation time is

Additional notes:

1) **If direction_in_time = FORWARD**

- If start and duration of the simulations are defined: the model will follow them.
- If source release is longer than the simulations: the source will be cut out.
- If source starts later than the simulations: the cloud will be empty until the release start time.
- Source start term “NOW” means that the release will start from the closest 6-hour meteotime – 00, 06, 12, 18 UTC of the current day.

2) **If direction_in_time = INVERSE**

- The “Source” actually becomes a receptor receiving the pollution cloud: inverse (adjoint) task.

- The main time direction in the model is backward.

- If the release time defines the period when the receptor receives the particles (“receptor is active”). It is from start till start+duration.

- If at least part of the receptor active time must be covered by the simulation time. So, the simulation start must be AFTER the source (receptor) starts. Then the model will go backward in time till the source(receptor) start.

- If the source start term “NOW” means that the receptor activity starts from the closest 6-hour meteotime – 00, 06, 12, 18 UTC of the current day. Note that the simulation must start AFTER this moment. This is dictated by the inverse advection time in the model.

- If the simulation start time is undefined, it is set automatically to <source(receptor) start> + <release duration>. So, the model will go from the end of the active receptor time backward. Covered period is determined by the duration of the simulations.

3.3.2 Namelist emission_parameters

```
 LIST = emission_parameters
   emission_source = INVENTORY <list_source_files_name>
#   emission_source = INVENTORY <area/point/bomb_source_file_name>
#   emission_source = VOID_SOURCE
#   emission_source = SEA_SALT  d:\model\silam_v5_0\ini\src_sea_salt.ini
#   emission_source = BIOGENIC_VOC
#   emission_source = WILD_LAND_FIRES  <fire_file_name>
#   emission_source = DESERT_DUST
#   emission_source = POLLEN  d:\model\silam_v5_0\ini\src_pollen_birch.ini
#   emission_source = POLLEN  d:\model\silam_v5_0\ini\src_pollen_olive.ini

   cut_area_source_if_outside_meteo_grid = YES

#   if_technical_source_dump = ORIGINAL_GRID   # NONE / ORIGINAL_GRID / DISPERSION_GRID
 END_LIST = emission_parameters
```

Figure 4. SILAM’s control file: namelist emission parameters

- emission_source = <type of source> <path and file name>

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.

INVENTORY type is normally used for emissions stated in emission inventories. These files can be area, point or bomb sources (see section 3.4). These files can be stated individually or by a list of emission files (see section 3.4).

VOID_SOURCE is

- **cut_area_source_if_outside_meteo_grid** = YES/NO, depending if the user wants the model to reduce the source area to the meteo grid, in case of the source are being bigger than the meteo grid. Typically is YES.

- **if_technical_source_dump**, for debugging purposes, common user should comment it out (# if_tecnical….)

3.3.3 Namelist dispersion_parameters

```
LIST = dispersion_parameters
  grid_method = OUTPUT_GRID  #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 = 150
  ny = 100
  lon_start = -10.
  lat_start = 35.
  dx = 0.4
  dy = 0.3
  lat_s_pole = -90.
  lon_s_pole = 0.
  lat_pole_stretch = 0.
  lon_pole_stretch = 0.
  vertical_method = OUTPUT_LEVELS    # METEO_LEVELS / OUTPUT_LEVELS / CUSTOM_LEVELS
  !
  ! If CUSTOM_LEVELS - fill-in the below two lines
  !
  level_type = HEIGHT_FROM_SURFACE    # PRESSURE, HEIGHT_FROM_SURFACE, ALTITUDE_FROM_SEA, HYBRID
  layer_thickness = 100. 500. 1400. 4000.   # output levels [m]/[pa]/[hybrid_nbr], reals
END_LIST = dispersion_parameters
```

Figure 5. SILAM's control file: namelist 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).

- **nx** and **ny**. Number of cells along the parallel and meridian (varying lon and lat) - ksec2(3), ksec2(2)
- **lon_end** and **lat_end**. Area source's longitude and latitude of the last grid cell (ksec2(8), ksec2(7)). Not needed if nx and ny are defined.

  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 Namelist meteo_parameters

```
 LIST = meteo_parameters
!   dynamic_meteo_file = NETCDF d:\!project\ncdf\nc_stuff\WRF_OUT\wrfout_d01
!   dynamic_meteo_file = GRIB d:\data\meteo\ec_oper\%ay4\ec%ay4%am2%ad2+%ah2+%f2.ml
!   dynamic_meteo_file = GRIB d:\data\meteo\ec_oper\%ay4\ec%ay4%am2%ad2+%ah2+%f2.sfc
!   static_meteo_file = GRIB d:\data\meteo\hirlam\2006\fc20060419_18+006
!   static_meteo_file = ASCII_V1 e:\data\emission\4SILAM\salinity_map.fld
!   static_meteo_file = -

   meteo_time_step = 3 hr

   if_wait_for_data = NO

   abl_parameterization_method = DRY_ABL    # DRY_ABL, FULL_PARAM
   number_of_precipitation_fields = 2
 END_LIST = meteo_parameters
```

Figure 6. SILAM's control file: namelist meteorological parameters

- **dynamic_meteo_file** = <file type> <file name>. File type: GRIB / ASCII / NETCDF and is time dependent. The filename format is described in section 3.1.
- **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 and 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 Namelist transformation_parameters

```
 LIST = transformation_parameters
!   transformation = PASSIVE
!   transformation = PM_GENERAL
!   transformation = DMAT_SULPHUR
!   transformation = CB4
!   transformation = POP_GENERAL
   transformation = ACID_BASIC
!   transformation = RADIOACTIVE
   aerosol_dynamics = SIMPLE
   dry_deposition_scheme = GRAVITATIONAL_AND_FULL_DIFFUSION
   wet_deposition_scheme = STANDARD_3D_SCAVENGING
   if_actual_humidity_for_particle_size = NO
   default_relative_humidity = 0.8
   compute_thermodiffusion = NO
   mass_low_threshold = HIGH_ACCURACY  # CRUDE_ACCURACY, STANDARD_ACCURACY, HIGH_ACCURACY
   if_full_acid_chemistry = YES
   passive_subst_ref_lifetime = 1 day
   passive_subst_ref_tempr = 288
   passive_subst_dLifeTime_dT = -1 min/K
   ADB_if_compute_nucleation = YES
   ADB_nucleation_scheme = KINETIC    !BINARY, TERNARY, KINETIC, ACTIVATION
   ADB_if_compute_coagulation = YES
   ADB_if_compute_condensation = YES
   ADB_if_compute_cloud_activation = NO
   ADB_if_recalc_wet_d = YES
 END_LIST = transformation_parameters
```

Figure 7. SILAM's control file: namelist transformation parameters

- **transformation** = PASSIVE / PM_GENERAL / DMAT_SULPHUR / CB4 / POP_GENERAL / ACID BASIC, sets the chemical and physical processes undergoing during the computation, depending on the emissions available (see section 1). 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** = GRAVITATIONAL_AND_FULL_DIFFUSION/ GRAVITATIONAL_ONLY/SIMPLE_DIFFUSION_ONLY/FULL_DIFFUSION_ONLY/GRAVITATIONAL_AND_SIMPLE_DIFFUSION/. Sets the method for the dry deposition and includes gravitational settling or/and diffusion. SIMPLE_DIFFUSION_ONLY is only considering viscous sub-layer resistance. 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**. Sets the default value for relative humidity, with typically number of 0.8.
- **compute_thermodiffusion** = YES/NO. Sets if the model computes thermodiffusion or not. Normally set to NO.

- **mass_low_threshold** = CRUDE_ACCURACY / STANDARD_ACCURACY / HIGH_ACCURACY. Sets the accuracy for the computation of the low-mass threshold for the Eulerian setup. Normally set to HIGH_ACCURACY.
- **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). Normally set as YES.
- **passive_subst_ref_lifetime,passive_subst_ref_tempr,passive_subst_dLifeTime_dT** are setting different parameters for the run with a passive tracer: lifetime, temperature and degradation with temperature. If PASSIVE transformation is not set, then these values are irrelevant. Typical values are shown in Figure 7
- **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** are set if aerosol dynamics is taken into account. Figure 7 is showing the standard setting if aerosol dynamics is requested.

3.3.6 Namelist initial_and_boundary_conditions

Both initial and boundary conditions can be void. If existent the user must include a list of initialised/constrained quantities and the list of input files.

```
LIST = initial_and_boundary_conditions
# initialize_quantity = temperature_2m_acc          ! if no such line, initial conditions are void
# initialize_quantity = daily_mean_temperature_2m   ! if no such line, initial conditions are void
# initialize_quantity = degree_day   ! if no such line, initial conditions are void
 #initialize_quantity = concentration              ! if no such line, initial conditions are void
 #initialization_file = GRADS f:\!model\2011\silam_v4_9\output\tst_acid_adv_v2\tst_acid_adv_v2_ALL_SRCS_20070829.grads.super_ctl

 boundary_type =  ZERO            ! ZERO / DIRICHLET
 if_lateral_boundary = YES        ! YES/NO
 if_top_boundary =  NO            ! YES/NO
 if_bottom_boundary =  NO         ! YES/NO
 boundary_time_step =  1 hr       ! timestep unit
 boundary_header_filename = d:\!model\2009\silam_v4_5_4\ini\boundary.ini
 END_LIST = initial_and_boundary_conditions
```

Figure 8. SILAM's control file: namelist 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>. The file type 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 v5 run.

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** = <timestep> <unit>
- **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. See section 3.6 for the description of boundary header file.

3.3.7 Namelist optical_density_parameters

```
LIST = optical_density_parameters ##############################################################################
  optical_coefficients_depend_on_relative_humidity = YES
  optical_coefficients_depend_on_temperature = YES
  if_split_aerosol_modes = YES            ! doesn't work yet
  if_narrow_wave_bands = YES              ! doesn't work yet
END_LIST = optical_density_parameters
```

Figure 9. SILAM's control file: namelist optical density parameters

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

- **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 namelist.

This namelist sets the output configuration: variables and temporal- and spatial-resolution.

```
 LIST = output_parameters
   source_id = NO_SOURCE_SPLIT  # SOURCE_NAME  SOURCE_SECTOR  SOURCE_NAME_AND_SECTOR
   vertical_method = CUSTOM_LAYERS
   level_type = HEIGHT_FROM_SURFACE
   layer_thickness = 100. 400. 1000. 2000. 3000.   # output levels [m]/[pa]/[hybrid_nbr], reals
   output_time_step = 1 hr
   output_times = REGULAR
   file_types = GRIB_NO TRAJECTORY_NO GRADS_YES ENSEMBLE_NO NETCDF_NO
   time_split = DAILY_NEW_FILE
   template =  f:\!model\2011\silam_v4_9\output\%case\%case_%source_%y4%m2%d2
   variable_list = d:\!model\2011\silam_v4_9\ini\output_config.ini

   grid_method = CUSTOM_GRID

   # If AREA_BASED grid then:
   area_borders =  54. 57. 33.5 41.5 # south, north, west, east; North positive, east positive
   area_title = out_area
   resolution = 5. km    # horizontal gridsize of output grid, [km]/[m]/[deg], real

   # This is for 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 = 265
   ny = 195
   lon_start = -17.
   lat_start = 33.
   dx = 0.2
   dy = 0.2
   lat_s_pole = -90.
   lon_s_pole = 0.
   lat_pole_stretch = 0.
   lon_pole_stretch = 0.
 END_LIST = output_parameters
END_CONTROL_v4_7
```

Figure 10. SILAM's control file: namelist output parameters

- **source_id** = NO_SOURCE_SPLIT / SOURCE_NAME / SOURCE_SECTOR/ SOURCE_NAME_AND_SECTOR. 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**, **level_type** and **layer_thickness** are explained in section 3.3.3
- **output_time_step**. Output timestep and unit
- **output_times** = REGULAR (standard)
- **file_types** = GRIB_YES/NO TRAJECTORY_YES/NO GRADS_YES/NO ENSEMBLE_YES/NO NETCDF_YES/NO. This namelist defines the type of output file

required, 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**. <Path for output dumping>\%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** = <south> <north> <west> <east>; 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 **CUSTOM_GRID**, a full definition of the grid has to be described, see section 3.3.3.

3.4 Source term files

The source file for SILAM v5 consists of a list of individual sources, following one-by-one. Each source is treated totally independently from the others. The source is always started from the **Header line** and ends by **End line**. There are three types of sources supported: bomb source, point source and area source. They all can appear in the same emission file. In SILAM v5 there is no limitation on the type of emitted species, except if the species are chemically active, where there can be only one type of chemistry involved: sulphate chemistry (DMAT_SULPHUR), inorganic chemistry (ACID_BASIC) or inorganic and organic chemistry (CB4).

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 # First point-source starts

source_name = ETEX
source_sector_name =         # source sector name, e.g. SNAP_10. May be empty

#
# ETEX-1 source coordinates:
#
source_longitude = -2.008         # start geograph. lat., degrees and decimals, N positive
source_latitude = 48.058          # start geograph. lon., degrees and decimals, E positive

# Plume characteristics: horizontal size, gas vertical velocity and temperature
# Used only if plume rise routine is activated and ignored otherwise.
# Above source height boundaries are also involved in the plume rise computation

plume_rise = OFF         # ON/OFF of the buoyant pume rise routine
release_rate_unit = g/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]

# Time-strength-composition data.
# Arbitrary number of lines. The last line determines the end of the release
# If there is only one line, it may be just:
# NOW  <duration [min]> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>
# Otherwise each line contains:
#  y4 m2 d2 h2 m2 sec(real)  <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name1><rate1>.

par_str_point = 2007 6 25 0 0 0.    1.    800. 1010.   0.   273.  PASSIVE_COCKTAIL 2.
par_str_point = 2007 6 25 1 0 0.    1.    800. 1010.   0.   273.  PASSIVE_COCKTAIL 3.

#
# Extra rate adjustment is possible: hour_in_day, day_in_week, month_in_year
#
# 24 hours in day: duirnal relative intensity
# 7 days in week: week-day relative intensity
# 12 months in year: monthly relative intensity
#
hour_in_day_index = 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
day_in_week_index = 1. 1. 1. 1. 1. 1. 1.
month_in_year_index = 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.

END_POINT_SOURCE_5   # MANDATORY
```

Figure 10. SILAM's point source file

- **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** = <mass>/<time> (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]> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>
- **par_str** = [NOW]/[LAST_METEO_TIME] <duration [min]> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>

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** = <year> <month> <day> <hour> <minute> <sec> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>
- **par_str** = <year> <month> <day> <hour> <minute> <sec> <rate> <xy_size> <bottom> <top> <z-velocity> <tempr> <cocktail_name>

The release rate (<rate>) is the value of the release in the units defined by release_rate_unit (above). The horizontal size (<xy_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.

3.4.2 Area source v.3

This form represents a SILAM source term type: a spatially distributed emission source. Following the general standards, it is defined in some 3-dimensional grid, while the time dimension is represented in a very similar way as par_str in the above point sources. Grid and vertical definitions follow the standards of the GRID format. The source file consists of five main parts: general parameters, grid definition, vertical definition, time definitions and grid cell values. A template of the file is below and the namelists are described. 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: AREA_SOURCE_3 and END_AREA_SOURCE_3

```
AREA_SOURCE_2
source_name = AM
source_sector_name = S1p
release_rate_unit = kmole/yr
vertical_unit = m
nx = 800
ny = 1000
dx = 0.1
dy = 0.05
grid_type = lon_lat
lon_s_pole = 0
lat_s_pole = -90
lon_pole_stretch = 0.
lat_pole_stretch = 0.
lon_start = -10
lat_start = 30
resol_flag = 128
ifReduced = 0
earth_flag = 0
wind_component = 0
reduced_nbr_str = 0
cocktail_composition = SPECIFIC
size_class_split = SPECIFIC
emitted_substance = NOX
emitted_size_mode_nbr = 2
hour_in_day_index = 0.79 0.72 0.72 0.71 0.74 0.8 0.92 1.08 1.19 1.22 1.21 1.21 1.17 1.15 1.14 1.13 1.1 1.07 1.04 1.02 1.02 1.01 0.96 0.88
day_in_week_index = 1.0385222  1.0280915  1.0375451  1.0411919  1.0900258  0.8872839  0.8773400
month_in_year_index = 1.307 1.216 1.087 1.050 0.899 0.788 0.752 0.740 0.823 0.996 1.124 1.232
vert_level = HEIGHT_FROM_SURF 180.  320.   0.08
vert_level = HEIGHT_FROM_SURF 320.  520.   0.46
vert_level = HEIGHT_FROM_SURF 520.  780.   0.29
vert_level = HEIGHT_FROM_SURF 780. 1100.   0.17
vertical_distribution = MULTI_LEVEL_FIXED    ! SINGLE_LEVEL_DYNAMIC or MULTI_LEVEL_FIXED
par_str = 1900 12 25 0 0 0.   1.   1. 10.  ACID_BASIC_COCKTAIL
par_str = 2100 1 1 0 0 0.   1.   1. 10.  ACID_BASIC_COCKTAIL
coordinate_of_values = GEOGRAPHICAL  # GEOGRAPHICAL or GRID
val = 44.5  40.15  1241.51
val = 44.5  40.2  766.813
val = 44.5  40.75  101.976
val = 44.5  40.8  248.567
val = 44.6  40.15  11173.6
val = 44.6  40.2  6901.32
val = 44.6  40.75  917.785
val = 44.6  40.8  2237.1
val = 44.7  40.15  1241.51
val = 44.7  40.2  766.813
val = 44.7  40.75  101.976
val = 44.7  40.8  248.567
val = 44.8  40.45  11791
val = 44.8  40.5  14149.2
val = 44.9  40.45  8474.78
val = 44.9  40.5  10169.7
END_AREA_SOURCE_2
```

Figure 11. SILAM's area source description

- **source_name**. Source name. The source name has to be different.
  - **source_sector_name**. Normally according to EMEP's sector denomination. May be empty.

- definition of the area source grid: location, resolution and rotation, explained in section 3.3.3.
- **cocktail_composition** = COMMON/SPECIFIC. The cocktail composition (cocktail_composition), describes the fractionation of the emission between the species can be defined in each time slot for all the cells (COMMON) or be specific for each cell (SPECIFIC). If cocktail_composition is SPECIFIC, then the number of values is n_substances*(n_aerosol_modes+n_gaseous_modes
- **emitted_substance** = WHOLE_COCKTAIL/"substance". This defines if the definition of the source is for the whole cocktail or for just one substance
- **emitted_size_mode_nbr**. Number of sizes describing the aerosol.
- **vert_level** = <type> <layer_bottom> <layer_top> <fraction>. The type of vertical level (type) can be defined as: HEIGHT_FROM_SURFACE (height measured from the surface), ALTITUDE_FROM_SEA (height measured from the sea level) and PRESSURE (pressure levels). The bottom (layer_bottom) and top (layer_top) of the layer sets how the model defines the position of the layer.
- **vertical_unit**. [hpa] or [m] according to the type chosen
- **Time-strength-composition data** and **time variation coefficients,** see Point Source definition, Section 3.4.1
- **coordinate_of_values** = COORDINATES
- **val** = <lon> <lat> <value>. All coordinates and emission value are REAL

or

- **coordinate_of_values** = GRID_INDICES
- **val** = <hor index> <vert index> <value>. All grid indexes are INTEGER and emission value are REAL

There are a few critical differences between the above area source definition and the point source files. They all originate from one more dimension of parameter variations – spatial – that has to be taken into account. In the point source definition, there is only one vertical layer where the emission goes to. All sophisticated considerations are supposed to be solved via a plume rise routine. Such approach does not work with the area sources. Therefore, there are two ways allowed for the description of the vertical distribution: time-varying single layer defined in par_str for corresponding times (resembling the approach of point sources), and multi-layer distribution that is fixed in time but allows split of emission between the layers (see vertical_distribution and vertical_layer in the above example).

Another ambiguity is connected with the composition of the release. Species mass fractions in cocktail may vary between the grid cells. To take this into account, another two-option

selection is introduced (switcher is the cocktail_composition line). The first option is the same as in point source: the cocktail name is taken from par_str, its composition is taken from the cocktail description file (section 3.8) and assumed the same for all grid cells.

Time variation of the composition is then reproduced via cocktail definition – as is done in the point source. The second option is to use fixed-in-time but varying-in-space cocktail composition. In this case, the cocktail name in the par_str lines defines only lists of species and aerosol size classes, while the mass fractions are written in the val lines – specifically for each grid cell. In the latter case, there must be an agreement between the number of mass fractions in the val lines and the number of species in the cocktail descriptors references in the par_str lines.

It is also possible to create sources with dynamical emission rates computed with regard to meteorological parameters, which is mandatory for biogenic emission. This is the case of, e.g., sea salt, as explained in the following section.

3.4.3 Sea salt initialisation file

The emission map of sea salt is computed internally by the SILAM model. This type of source is so called a map source, where the emission map is created by utilising GIS data and source functions. In the case of the sea salt the GIS data is a, a map for the salinity distribution and the source function is dependent of sea surface temperature and salinity. The source contain the aerosol distribution of the substance emitted as well as other parameters describing the sea salt aerosol. The example shown below is a standard file, where the user just needs to change the path of the **source_mask_area**. This file is provided within SILAM v5 package.

```
SEA_SALT_SOURCE_V5

source_name = sea_salt_standard
source_sector_name = natiral_emission    ! free sector name

source_area_mask = ASCII_V1 e:\data\emission\4SILAM\v5\sea_salt_source_europe_10km.sa2

sea_salt_emission_method = HYBRID_WIND_10M    ! Own development, the only one available so far
water_temperature_input_type = FIXED_VALUE   # FIXED_VALUE / FIXED_MAP / MONTHLY_CLIMATOLOGY / DYNAMIC
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

sea_salt_substance_name = sslt              ! must be in chemical database
aerosol_mode = 1  0.01 0.1  0.03 mkm  1000 kg/m3    ! mode_number Dmin, Dmax, Daver D_unit, density, density_unit
aerosol_mode = 2  0.1  1.   0.3  mkm  1000 kg/m3
aerosol_mode = 3  1.   2.5  1.5  mkm  1000 kg/m3
aerosol_mode = 4  2.5  10.  6.   mkm  1000 kg/m3
aerosol_mode = 5  10.  30.  20.  mkm  1000 kg/m3
aerosol_distribution_shape = FIXED_DIAMETER   ! later also: GAMMA_FUNCTION
aerosol_default_density = 1000 kg/m3          ! if mode-specific is not given

END_SEA_SALT_SOURCE_V5
```

Figure 12. SILAM's sea salt initialisation file

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

```
OUTPUT_CONFIG_3_7

LIST = OUTPUT_CONFIG_3_7

#-----------------------------------------------------------------
#
# General characteristics of the output variables
#

aerosol_size_modes = SEPARATE     # SUM or SEPARATE - a way to report aerosol size modes in the output


#-----------------------------------------------------------------
#
#  SILAM dispersion model variables
#  Specific names of substanses see AFTER the meteorological variables
#  there are too many of them to put them here
#
#  Emission fields
out_var =   0  emission_flux     TOTAL_WHOLE_PERIOD   #TOTAL_WHOLE_PERIOD AS_IS

#  Permanent fields (physiography)
out_var =   0   physiography_field_set  AS_IS

#  Particle counter and vertically integrated particle counter
out_var =  0   particle_counter  INSTANT
out_var =  0   areas_of_risk     AS_IS

#  Nuclides existing in the source inventory - concentr. and deposition
out_var =  0   concentration  [SOURCE_INVENTORY] AVERAGE
out_var =  0   drydep        [SOURCE_INVENTORY] AVERAGE
out_var =  0   wetdep        [SOURCE_INVENTORY] AVERAGE

#  All species from source inventory AND from transformation chain - conc and dep.
out_var =   2   concentration  [FULL_INVENTORY]   AVERAGE
out_var =   2   drydep        [FULL_INVENTORY]   AVERAGE
out_var =   2   wetdep        [FULL_INVENTORY]   AVERAGE

# Diagnostic optical depth
out_var = 0 optical_density      [FULL_INVENTORY]  AVERAGE  %WAVE_LENGTH nm 300. 320. 500.
out_var = 0 optical_column_depth [FULL_INVENTORY]  AVERAGE  %WAVE_LENGTH nm 550.
#out_var =  2 optical_column_depth [NO2]  AVERAGE  %WAVE_LENGTH nm 550.
#out_var =  2 optical_column_depth [SO4]  AVERAGE  %WAVE_LENGTH nm 550.
#out_var =  2 optical_column_depth [NH4NO3]  AVERAGE  %WAVE_LENGTH nm 550.
#out_var =  2 optical_column_depth [NH4_S]  AVERAGE  %WAVE_LENGTH nm 550.
#out_var =  2 optical_column_depth [NO3rad]  AVERAGE  %WAVE_LENGTH nm 550.

#-----------------------------------------------------------------
#  SILAM meteorological variables
#
out_var =  0  temperature               AVERAGE
out_var =  0  temperature_2m             AVERAGE
#-----------------------------------------------------------------
#  Full list of SILAM nuclides - you are free to choose each of them
#
out_var =  0  silam_cocktail      [FM_256]    AVERAGE
out_var =  0  silam_cocktail_dep  [FM_256]    AVERAGE

END_LIST = OUTPUT_CONFIG_3_7

END_OUTPUT_CONFIG_3_7
```

Figure 13. SILAM's area output configuration file

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 (see Figure 13), 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 index** to 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 = d:\data\bc\2003_monmean_raqbc.nc
file_format = NETCDF          ! GRIB/ASCII/GRADS/NETCDF
boundary_names = NSEW         ! NSEWTB
ifClimatology = YES           ! YES/NO
climatologyTimestep = MONTHLY  ! MONTHLY/STATIC;  only used if ifClimatology = YES, otherwise dynamic and timestamp has to be correct
nBoundSpecies = 7        ! Number of species to be read from the files
! Each par_strline should have the following fields:
! par_str = <boundary_species_subst_name> <boundary_species_mode> <transport_species_subst_name> <transport_species_mode> <factor>
par_str = O3 O3 0. 0. 1.
par_str = CO CO 0. 0. 1.
par_str = HCHO HCHO  0. 0. 1.
par_str = NO NO 0. 0. 1.
par_str = NO2 NO2 0. 0. 1.
par_str = PAN PAN 0. 0. 1.
par_str = C5H8  ISOP 0. 0. 1.
```

Figure 14. SILAM's boundary conditions header file

- **boundary_file** = <file path and file name>
- **file_format** = GRIB/ASCII/GRADS/NETCDF. Format of the input files
- **boundary_names** = NSEWTB. Description of which boundaries of the domain are emitting: N = north, S = south, E = east, W = west, T = top and B = bottom. the example here is for the case that all the boundaries are emitting.
- **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** = <nro of species>, number of species to be read from the boundary files.
- **par_str** = <boundary_substance_name> <model_substance_name> <boundary_substance_mode> <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 v5 package and are essential for the model to run.

```
BEGIN_STANDARD_SETUP

advection_method = EULERIAN_3D_V2    # dimension of advection, [LAGRANGIAN_2D, LAGRANGIAN_3D]

random_walk_method = FULLY_MIXED    # Random-walk method [IDIOT_SIMPLE, FULLY_MIXED, BULK_GAUSSIAN]

abl_height_method = COMBINATION     # [HIRLAM_ABL, CONSTANT, COMBINATION, RICHARDSON, PARCEL, TURBULENT_ENERGY]

horizontal_interpolation = LINEAR   # [NEAREST_POINT, SECOND_ORDER, CUBIC, LOG_LINEAR]
vertical_interpolation = LINEAR     # [NEAREST_POINT, SECOND_ORDER, CUBIC, LOG_LINEAR]
time_interpolation = LINEAR         # [NEAREST_POINT, SECOND_ORDER, CUBIC, LOG_LINEAR]

nuclide_database_fnm = /lustre/apps/silam/silam_v5_0/ini/silam_nuclides.dat

chemical_database_fnm = /lustre/apps/silam/silam_v5_0/ini/silam_chemicals.dat

standard_cocktail_fnm = /lustre/apps/silam/silam_v5_0/ini/standard.cocktails

grib_code_table_fnm = /lustre/apps/silam/silam_v5_0/ini/grib_code_table_v5.silam

netcdf_name_table_fnm = /lustre/apps/silam/silam_v5_0/ini/netcdf_name_table.silam

land_use_data_meta_file = /lustre/apps/silam/silam_v5_0/ini/land_use_features_USGS_Eurasia.dat

optical_properties_meta_data_file = /lustre/apps/silam/silam_v5_0/ini/optical_properties.dat

allow_zero_forecast_length = NO      # Whether zero-long forecasts may be used in the simulations
                                     # Beware of zero-valued accumulated fields

precipitation_low_limit = 0.1 mm/hr  # Cut-off limit for precipitation rate (separate for large-sc and conv)

print_debug_info = DEBUG_INFO_YES    # DEBUG_INFO_YES, DEBUG_INFO_NO

cloud_report_interval = 1

disregard_meteo_data_sources = YES

END_STANDARD_SETUP
```

Figure 15. SILAM's standard Eulerian setup file

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_DESCRIPTION_V3_2

cocktail_name = FIRE_FULL_COCKTAIL

mass_unit = kg
gas_phase = YES

aerosol_mode = 1  0.01   2.5  1.5  mkm   1100  kg/m3
aerosol_mode = 2  2.5   10.   6.   mkm   1500  kg/m3
aerosol_mode = 3  10.   30.   18.  mkm   2000  kg/m3

aerosol_distribution_shape = FIXED_DIAMETER ! or GAMMA_FUNCTION - later

component_fraction = CO 0. 0. 0. 0.834
component_fraction = HCHO 0. 0. 0. 0.012
component_fraction = NO2 0. 0. 0. 0.026
component_fraction = NH3 0. 0. 0. 0.012
component_fraction = SO2 0. 0. 0. 0.004
component_fraction = SO4 0. 0. 0. 0.
component_fraction = SOX 0. 0. 0. 0.
component_fraction = C5H8 0. 0. 0. 0.
component_fraction = C5H8_2 0. 0. 0. 0.

component_fraction = OC 0.066 0.019 0.009 0.
component_fraction = EC 0.006 0.002 0.001 0.
component_fraction = SIA 0.006 0.002 0.001 0.

END_COCKTAIL_DESCRIPTION
```

Figure 16. SILAM's standard cocktail description

- **cocktail_name** = random name
- **mass_unit** = Bq/number/mass
- **gas_phase** = YES/NO
- **aerosol_mode** = <min> <max> <average diameter> <diameter_unit> <density> <density_unit>
- **aerosol_distribution_shape** = FIXED_DIAMETER # so far the only available
- **component_fraction** = <Component name> <mass fraction in the mixture>, 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** = <Component name> number_of modes*<mass fraction in the aerosol mixture> <mass fraction in the gas mixture>

If **gas_phase** = NO,

- **component_fraction** = <Component name> number_of modes*<mass fraction in the aerosol mixture>

If **gas_phase** = YES and no aerosol phase,

- **component_fraction** = <Component name> <mass fraction in the gas mixture>

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 control path and or file name:

1) > <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>

2)> <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.

3)> <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.