ARIA Technologies
8-10 rue de la Ferme
92100 Boulogne-Billancourt

ARIANET
via Gilino 9, 20128 Milano

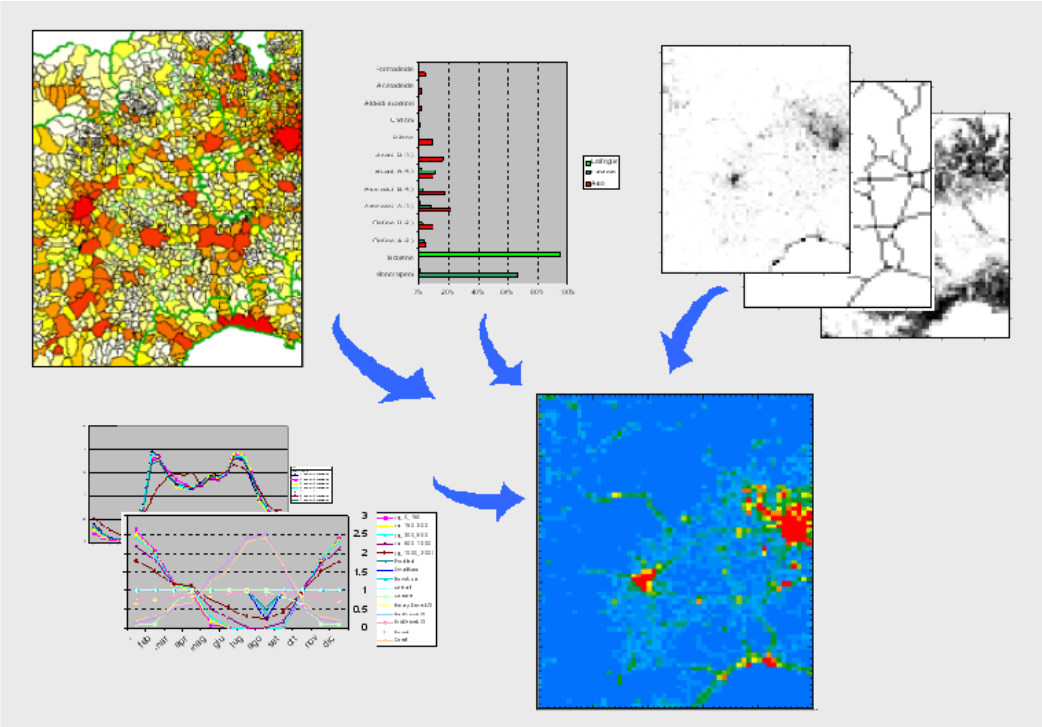
"Un seul métier, L'environnement Atmosphérique"

NOVEMBER 2013

Emission Manager

Processing system for model-ready emission input

User's guide



References: ARIA/ARIANET R2013.19

Document: Documentation

CONTENTS

MODULES GLOSSARY

1 INTRODUCTION

Emission Manager (EMGR) is a modular pre-processing system allowing to prepare model-ready emission inputs starting from one or multiple emission data bases. It can manage sources with different types of geometry: point (stacks, …), area (residential heating, …) and line (traffic, …).

Emission Manager relies on a set of input files describing the different emission sources. Since the format of the input files depends on the source type, the point (LPS), area (SRF) and line (LIN) sources are separated into individual files. Based on these input files, Emission Manager generates emissions in the form of Substance/Source file "couples". The file containing information about the substances included in the inventory is the so-called "pemspe" file and the second file including the source information and the actual emission data has the name "pemtim". These files can be further used by ARIA dispersion models. Moreover, a certain number of operations can be applied to the inventories (vector to raster data gridding, time modulation, chemical speciation, Savi3D export…), depending on the needs of the target air quality model (e.g. SPRAY, CHIMERE, FARM, IMPACT).

This document describes the emission input files and the set of modules allowing tobuild an emission inventory with Emission Manager (the name and format of I/O files, execution arguments).

1.1 EXAMPLES

The following figures give some examples of the chain of modules to be applied in order to obtain a model-ready emission input.

Figure 1 illustrates the general procedure employed to transform "client" files for the different source types into emission files in "pemtim" format, as well as how to modulate the emissions in time for a specific time period.

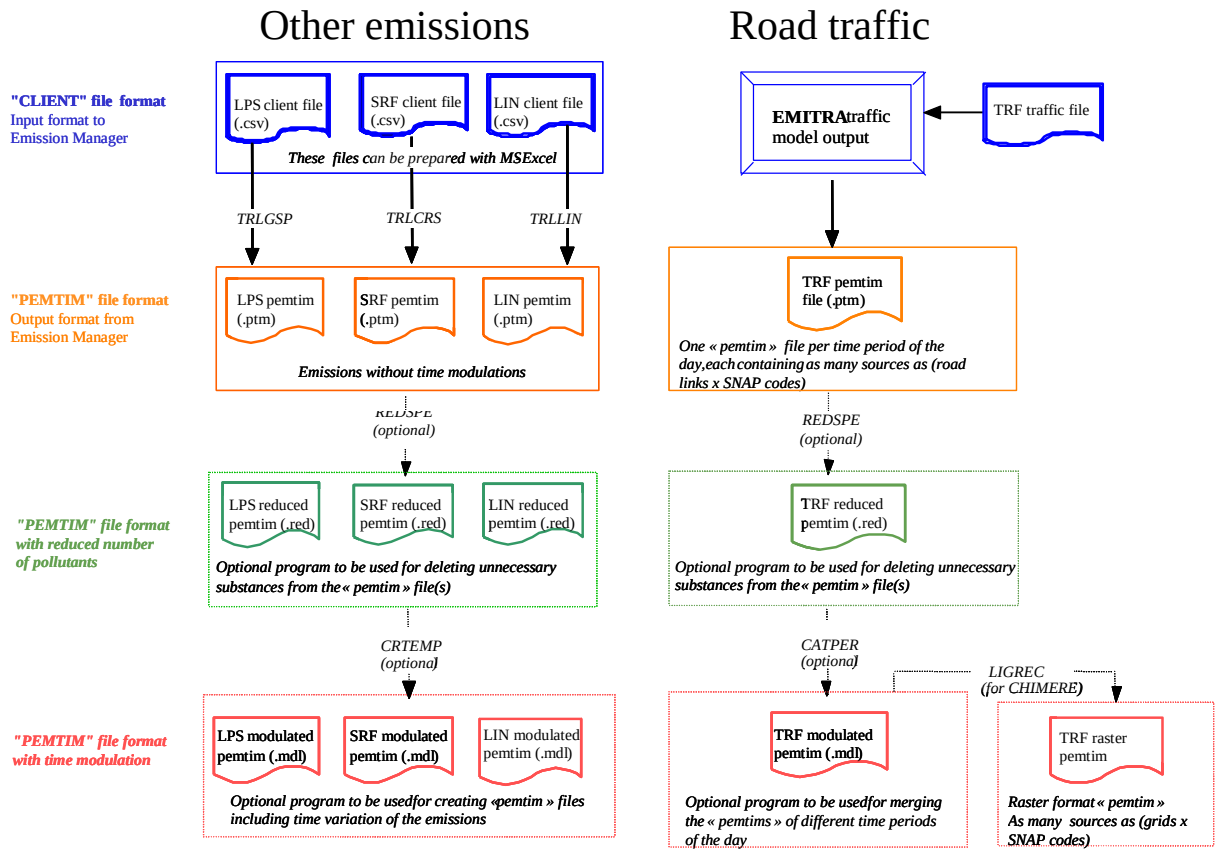


Figure 1: Example procedure to prepare an emission input for a specific time period.

Figure 2 shows the methodology for generating input files for CHIMERE based on the modulated emission files.

Throughout the chains, two kinds of files are used to store the emission data:

- "client files": mainly used as initial input files for the Emission Manager modules; these files have to be directly filled in by the user
- "pemtim"/ "pemspe" files: used either as model input files or to store intermediate results during the processing chain; these files are generated by EMGR modules, described in Chapter 3. The description of "pemtim"and "pemspe" files is given in A.1

These two kinds of files are described in this document.

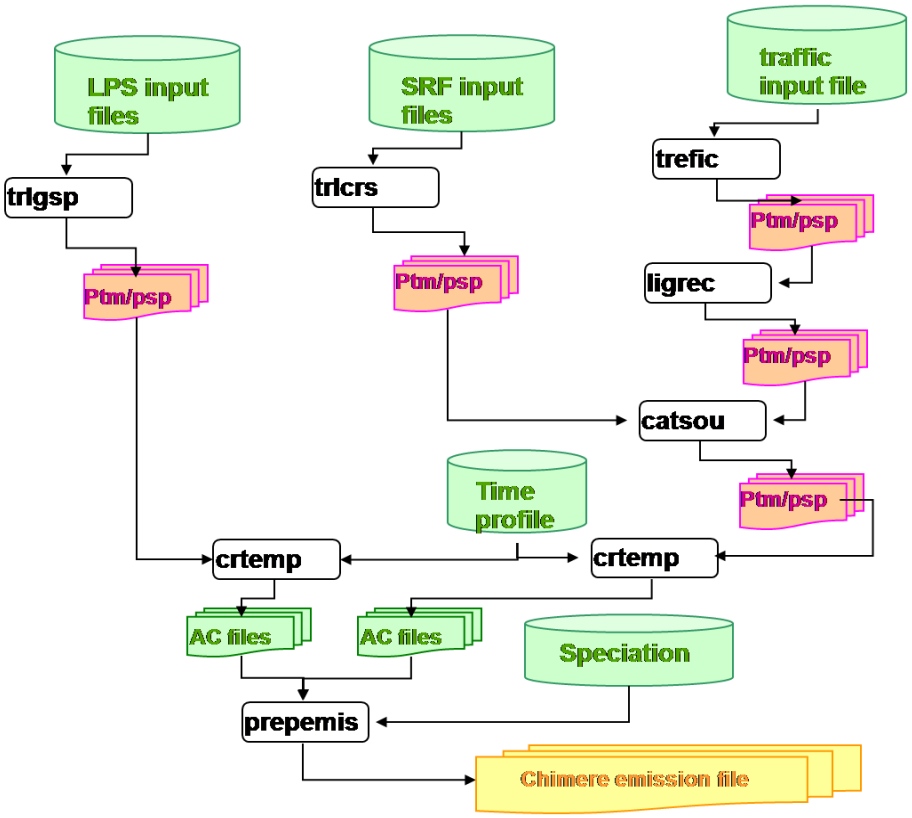


Figure 2: Methodology for generation of input files for CHIMERE.

Emission Manager modules can be anyway used in many combinations, depending on data availability (e.g. point/line/area sources, single vs. multiple inventories, etc.), target model and purpose. The user is free to organize the data processing flow according to its needs, invoking the sequence of modules that realize the required operations. Such calling sequence can then be put in a **script**, written under the preferred command shell, allowing to repeat the processing steps under different conditions (e.g. different time interval, different input data, etc.). This way possibly gives the highest degree of freedom and possibility of customization, at the price of writing and managing directly the calls of the individual modules. The reference **calling syntaxes** are given in section 3 of this manual.

On the other side, two other possibilities exist, both implementing the calling sequences that are most typically used for a specific target air quality model:

- the **Emission Manager GUI**: through the interface, the user can interactively pick the input files and execute the processing steps, either individually or as a whole.
- **EMMA**: a series of macro commands implementing typical calling sequences, to be embedded in simulation system performing repetitive operations involving the same datasets (e.g. forecast systems or long-term runs). EMMA commands are illustrated in Chapter 4 of this manual.

1.2 CHANGES FROM PREVIOUS VERSIONS

The features progressively implemented are summarized here, as differences of each version with respect to the previous one.

This version with respect to (2008):

- optimized generation of time-dependent binary input for FARM: combined modulation and conversion to FARM input binary format of multiple pemtims, bypassing .mdl pemtims;
- possibility of generating also 3D gridded input for FARM, with grid allocation of area/line sources and optionally point sources with plume rise;
- INEMOS: possibility of using also 'new gspdef' file; multiple formats for target dates;
- SPICE: possibility to process also (time modulated) pemtims; longer species names;
- CRTEMP: correct end-of-the-year crossing;
- PEM2GRID: possibility of processing also non-modulated pemtims; usage of geographic (lat/lon) coordinates;
- MODGRID: combined time modulation and FARM binary input (2D/3D gridded or FARM "pointemi"); generation of 3D gridded emissions including point sources, with plume rise based on hourly meteorology; category-based vertical allocation of diffuse sources on the 3D grid based either on simple layers or full arbitrary profiles.

This version with respect to (August 2013):

- integration of EMS emissions in SPRAY input.

2 INPUT DATABASE

2.1 INTRODUCTION

The input database of Emission Manager is made by several components, stored in files with specific formats, prepared by the user either manually or through other processors and systems.

The main components are:

- a reference classification scheme adopted for all the emitting activities of interest for a given application (e.g. the SNAP nomenclature by the EEA, or any other set of categories);
- emission data associated to point, line and area sources; such emission data can come from an inventory (e.g. on a yearly basis), from automatic monitoring system, or from other modelling systems (e.g. a traffic emissions module);
- geographic information in vector and raster format, describing the geometry of complex sources (area sources polygons and networks of line sources) or to be used when disaggregating in space the emission data for area sources;
- time modulation data, to describe the typical behaviour of the emitting sources when the emissions are coming from an inventory on e.g. yearly basis;
- species split / speciation data, to break the emissions of the species coming from the inventory (e.g. "NOx", "total NMVOC") into the individual species needed by the target model (e.g. "NO and NO_2", "lumped hydrocarbons").

Input emissions can be assigned in different ways:

- using the so-called **"time-independent client files"**, containing emissions in terms of average or total amounts released over a reference time period (e.g. one year, one hour);
- using the so-called **"time-dependent client files"**, containing time series of the parameters varying in time (mainly flow rates);
- or also through the files coming from **"emission monitoring systems"** (EMS), giving directly emission rates from stacks on a e.g. hourly basis.

In the first case, the emissions reference values are complemented by the so-called **"modulation files"**, describing the evolution of emission along the months of the year, the days of the week and the hours of the day.

These components can be articulated in many ways, according to the problem specifications and the available information. The formats used in the files of the input database are explained in the following sections.

2.2 TIME-INDEPENDENT "CLIENT FILES"

Here we describe the emission input files containing time-independent values of the emissions. As the source parameters are not the same for all source types, we distinguish three types of "client files":

- point sources "client files" (GSP)
- line sources "client files" (LIN)
- area sources "client files" (SRF)

2.2.1 *COMMON FEATURES AND CONVENTIONS*

"Client files" are ASCII files of .csv type (compatible e.g. with Excel), following the rules:

- the **character separating each field** must be a semicolon (";")
- a **header** containing the fields' names; fields names can be enclosed by inverted commas (e.g. "IDCLA1")
- fields' column **positions** are **free**
- fields can be **mandatory** or **optional**, as described below
- in text fields, the **characters** allowed are lower case letters from "a" to "z", upper case letters from "A" to "Z", numbers 0 to 9, the character "_", and space " "; inverted commas (") cannot be used.
- in numeric fields, the **decimal separator** must be a point ".".
- when giving the pollutant names, only characters from "a" to "z" or "A" to "Z" will be accepted; the whole processing is **case sensitive**.

In the tables describing each file, the following **conventions** have been used:

- Background colour of field name:

- red: key field of the source
- cyan: classification and naming fields
- light blue: SNAP category fields
- green: geographic fields
- yellow: source physical parameters
- light pink: species emission rates

- Text style of field name:

- fields written Bold fields are **mandatory** ones
- fields written in Italic are *optional* ones; such fields maybe present or not; if present, they must be filled with valid values

- "Default value" column:

  - a default value is specified, if meaningful or appropriate
  - if no default value is applicable, a "–" is shown

- "Type" column:

  - "I": indicates integer value
  - "R": indicates real value
  - "Cxx": text value, made of up to xx characters

- "Time dependency" column:

  - "C" denotes a constant field
  - "V" denotes a field that can be time dependent; such fields can appear in time-dependent emission files

Common features

- If ASCII "pemtims" are to be generated, the source identifier (SRCEID) can contains up to 8 digits.

- The codes indicating the administrative units (IDADM1 and IDADM2) can indicate whatever is meaningful for the territorial units of interest: (e.g., country, state, region, county, province)

- A few text fields (NAMADM, NAMGP1, NAMGP2, NAMGP3) can be used to document the data records; typically choices can be the name of the territorial unit (NAMADM) and three levels of level description of the source (e.g. "Plant1", "Group2", "EDF")

- The **classification scheme** of emission-related categories (activities) is the same for all sources (different classifications cannot be combined); examples classification schemes are EEA SNAP94, US EPA SCC, but user-defined classification schemes can also be used.

- The **projection type** (IDPROJ) must be the same for all sources. The following options are currently available:

  0 = Lambert Conformal Conic
  1 to 60 = Universal Transverse Mercator (Gauss-Krüger)
  99 = geographic (longitude/latitude)

- **Coordinates** units are expressed in decimal degrees if geographic (longitude/latitude) projection is used; in metres in all other cases.

- An **anchor point** (origin of the reference system) can be specified through ORIGX and ORIGY fields (it must be the same for all sources); in such case, all sources coordinates

are given respect to this point; when ORIGX = 0 and ORIGY = 0, absolute coordinates are used.

- **Emission rates** for an arbitrary set of species are assigned through fields named "Q_x", where "x" is the name of the species.

- Emissions rates can be assigned according to different **units**; the adopted unit can be specified either through:
  - IDUNIT field, that is related to all species in client file
  - a set of IDUNIT_x fields, one each for each species

IDUNIT	Definition	Type	Conversion Factor
1	ton/year	mass	3.169 10^{12} µg/s
2	ton/day	mass	1.157 10^{12} µg/s
3	Kg/h	mass	2.778 10^{5} µg/s
4	µg/s	mass	1 µg/s
5	ng/s	mass	1. 10^{-3} µg/s
6	uo/h	odour	-
7	Bq/s	activity	-
8	Ci/s	activity	3.7 10^{10} Bq/s

2.2.2 POINT SOURCES (GSP)

Field	Description	Type	Default value	Time dep.	Example
SRCEID	Unique identifier of the point source	I	-	C,V	100
IDADM1	Fist level administrative code of the territorial unit where the point source is located	I	-999	C	7
IDADM2	Second level administrative code of the territorial unit where the point source is located	I	-999	C	135
NAMADM	Name of the territorial unit where the source is located.	C20	NULL	C	Somewhere
NAMGP1	1st level description of the source	C8	NULL	C	Plant1
NAMGP2	2nd level description of the source	C8	NULL	C	Group2
NAMGP3	3rd level description of the source	C16	-	C	EDF
IDCLA1	1st level code of the emitting category in the adopted classification scheme	I	-999	C	01
IDCLA2	2nd level code of the emitting activity in the adopted classification scheme	I	-999	C	01
IDCLA3	3rd level code of the emitting category in the adopted classification scheme	I	-999	C	01
IDCLA4	4th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDCLA5	5th level code of the emitting category in the	I	-999	C	-999

	adopted classification scheme				
IDPROJ	Code of projection used for coordinates (see above for codes)	I	-	C	0
SHIFTX	coordinates shift for X axis (only with UTM)	R	0.	C	0
ORIGX	X coordinate of anchor point	R	0.	C	0
ORIGY	Y coordinate of anchor point	R	0.	C	0
POSX	X coordinate of point source	R	-	V	645000
POSY	Y coordinate of point source	R	-	V	2399000
DX	Size of the source along x (meters)	R	-	V	5
DY	Size of the source along y (meters)	R	-	V	5
DZ	Size of the source along z (meters)	R	-	V	5
HEIGHT	Above-ground height (metres) at which emission occurs. In the case of a stack, HEIGHT is equal to stack height. The value is either positive or 0.	R	-	V	150
DIAMET	Diameter (metres) of point source. In the case of a stack, DIAMET is equal to stack diameter. The value is either positive or 0.	R	-	V	4
TEMPER	Emission temperature (Kelvin degrees). In the case of a stack, TEMPER is equal to the exit gas temperature. If the entered value is –999, the computation will consider ambient temperature. Otherwise, the value must be positive.	R	-	V	300
SPEED	Emission speed (in metres per second). In the case of a stack, SPEED is equal to the stack exit velocity. The value is either positive.	R	-	V	50
IDUNIT(_x)	Unit in which emission rates reported in field Q_x, or (in case of IDUNIT) in which emission rates of all species are expressed (see table above for units codes).	I	-	C	1
Q_x	Emission rate of species x, expressed according to the unit specified in IDUNIT(_x). The value is either positive or 0.	R	-	V	200.3

2.2.3 *LINE SOURCES (LIN)*

Two types of files can be provided by the user, according to the complexity of the geometry of the line sources:

- **simplified** line sources file, where sources are described as 2-points segments
- **complex** approach with two files: a CSV file describing the general information of the sources, including the emission rates, and a geographic file, describing the geometry of the polylines to which the emissions are referred

Simplified line sources files

Field	Description	Type	Default	Time	Example

			value	dep.	
SRCEID	Unique identifier of the line source	I	-	C	100
IDADM1	Fist level administrative code of the territorial unit where the line source is located	I	-999	C	7
IDADM2	Second level administrative code of the territorial unit where the line source is located	I	-999	C	135
NAMADM	Name of the territorial unit where the source is located.	C20	NULL	C	Somewhere
NAMGP1	1st level description of the source	C8	NULL	C	Traffic
NAMGP2	2nd level description of the source	C8	NULL	C	Cars
NAMGP3	3rd level description of the source	C16	-	C	Highway
IDCLA1	1st level code of the emitting category in the adopted classification scheme	I	-999	C	7
IDCLA2	2nd level code of the emitting activity in the adopted classification scheme	I	-999	C	1
IDCLA3	3rd level code of the emitting category in the adopted classification scheme	I	-999	C	1
IDCLA4	4th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDCLA5	5th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDPROJ	Code of projection used for coordinates (see above for codes)	I	-	C	0
SHIFTX	coordinates shift for X axis (only with UTM)	R	0.	C	0
ORIGX	X coordinate of anchor point	R	0.	C	0
ORIGY	Y coordinate of anchor point	R	0.	C	0
IDZH	Type of points height ZHi in input : 0 : altitude 1 : height above ground	R	-	C	0
X1	X coordinate of the 1st point of the segment	R	-	V	645000
Y1	Y coordinate of the 1st point of the segment	R	-	V	2399000
ZH1	Height (in metres) at which emission occurs. ZH1 represents height above sea level or above ground according to IDZH value. The value is either positive or 0.	R	-	V	0
X2	Same as X1 for the 2nd point of the segment	R	-	V	646000
Y2	Same as Y1 for the 2nd point of the segment	R	-	V	2399050
ZH2	Same as ZH1 for the 2nd point of the segment	R	-	V	0
SLOPE	Slope (%) of the line source in forward direction (X1, Y1 → X2, Y2) (e.g. direction in which vehicles move on roadway). The sign of the slope indicates if it is upward (positive) or downward (negative).	R	-999	V	4
WIDTH	Width (in metres) of line source	R	-999	V	10
DZ	Thickness of the source (meters)	R	-	V	5
IDUNIT(_x)	Unit in which emission rates reported in	I	-	C	1

	field Q_x, or (in case of IDUNIT) in which emission rates of all species are expressed (see table above for units codes).				
Q_x	Emission rate of species x, expressed according to the unit specified in IDUNIT(_x). The value is either positive or 0.	R	-	V	2589.2

Complex line sources files

Two files must be supplied for complex line sources:

- general information file: it is a client file of LIN type, containing all information about the sources, except their geometric layout
- cartographic file: a DONCAR format file, describing the geometry of the polylines to which the emissions specified in the general information file are referred

The elements of the two files are linked through the IDGEOM field of each polyline.

❖ **General information file**

Field	Description	Type	Default value	Time dep.	Example
SRCEID	Unique identifier of the line source.	I	-	C	100
IDADM1	Fist level administrative code of the territorial unit where the point source is located	I	-999	C	7
IDADM2	Second level administrative code of the territorial unit where the point source is located	I	-999	C	135
NAMADM	Name of the territorial unit where the source is located.	C20	NULL	C	Somewhere
NAMGP1	1st level description of the source	C8	NULL	C	Traffic
NAMGP2	2nd level description of the source	C8	NULL	C	Cars
NAMGP3	3rd level description of the source	C16	-	C	Highway
IDCLA1	1st level code of the emitting category in the adopted classification scheme	I	-999	C	7
IDCLA2	2nd level code of the emitting activity in the adopted classification scheme	I	-999	C	1
IDCLA3	3rd level code of the emitting category in the adopted classification scheme	I	-999	C	1
IDCLA4	4th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDCLA5	5th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDPROJ	Code of projection used for coordinates (see above for codes)	I	-	C	0
SHIFTX	Coordinates shift for X axis (only with UTM)	R	0.	C	0
IDGEOM	Unique identifier of the polyline. Each value must correspond to a single	I	-	C	200

	polyline in the geographic file.				
SLOPE	Slope (%) of the line source in forward direction (X1, Y1 → X2, Y2) (direction in which vehicles move on roadway). The sign of the slope indicates if it is upward (positive) or downward (negative).	R	-999	V	4
WIDTH	Width (in metres) of line source	R	-999	V	10
DZ	Thickness of the source (meters)	R	-	V	5
IDUNIT(_x)	Unit in which emission rates reported in field Q_x, or (in case of IDUNIT) in which emission rates of all species are expressed (see table above for units codes).	I	-	C	1
Q_x	Emission rate of species x, expressed according to the unit specified in IDUNIT(_x). The value is either positive or 0.	R	-	V	2589.2

❖ **Cartographic file**

It is a DONCAR file (see *§ 2.5.1* for a description of file format) describing all the polylines that correspond to the line sources. The coordinates must be specified according to the projection system declared in IDPROJ field of the general information file.
Each polyline must have a unique value for the associated IDGEOM field. All records in the general information files must have a corresponding polyline object, to be linked through IDGEOM; multiple records in a general information file can be associated to the same polyline.

2.2.4 AREA SOURCES (SRF)

Two types of files can be provided by the user, according to the complexity of the geometry of the area sources:

- **simplified** area sources files: a single file where sources are described as polygons with four points (quadrangles)
- **complex** approach with two files: described as 2-points segments
- **complex** approach with two files: a CSV file describing the general information of the sources, including the emission rates, and a geographic file, describing the geometry of the polygons to which the emissions are referred

Simplified area sources files

Field	Description	Type	Default value	Time dep.	Example
SRCEID	Unique identifier of the area source	I	-	C	100
IDADM1	Fist level administrative code of the territorial unit where the point source is located	I	-999	C	7
IDADM2	Second level administrative code of the territorial unit where the point source is located	I	-999	C	135
NAMADM	Name of the territorial unit where the source	C20	NULL	C	Somewhere

	is located.				
NAMGP1	1st level description of the source	C8	NULL	C	Combust
NAMGP2	2nd level description of the source	C8	NULL	C	Resident
NAMGP3	3rd level description of the source	C16	-	C	Gas
IDCLA1	1st level code of the emitting category in the adopted classification scheme	I	-999	C	2
IDCLA2	2nd level code of the emitting activity in the adopted classification scheme	I	-999	C	2
IDCLA3	3rd level code of the emitting category in the adopted classification scheme	I	-999	C	2
IDCLA4	4th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDCLA5	5th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDPROJ	Code of projection used for coordinates (see above for codes)	I	-	C	0
SHIFTX	coordinates shift for X axis (only with UTM)	R	0.	C	0
ORIGX	X coordinate of anchor point	R	0.	C	0
ORIGY	Y coordinate of anchor point	R	0.	C	0
IDZH	Type of points height ZHi in input : 0 : altitude 1 : height above ground	R	-	C	1
X1	X coordinate of first point of the quadrangle	R	-	V	125.4
Y1	Y coordinate of first point of the quadrangle	R	-	V	43.2
ZH1	Height (in metres) at which emission occurs. ZH1 represents height above sea level or above ground according to IDZH value. The value is either positive or 0.	R	-	V	0
X2	Same as X1 for the 2nd point	R	-	V	125.4
Y2	Same as Y1 for the 2nd point	R	-	V	44.3
ZH2	Same as ZH1 for the 2nd point	R	-	V	0
X3	Same as X1 for the 3rd point	R	-	V	127.2
Y3	Same as Y1 for the 3rd point	R	-	V	44.3
ZH3	Same as ZH1 for the 3rd point	R	-	V	0
X4	Same as X1 for the 4th point	R	-	V	127.2
Y4	Same as Y1 for the 4th point	R	-	V	43.2
ZH4	Same as ZH1 for the 4th point	R	-	V	0
DZ	Thickness of the source (in meter)	R	-	V	5
IDUNIT(_x)	Unit in which emission rates reported in field Q_x, or (in case of IDUNIT) in which emission rates of all species are expressed (see table above for units codes).	I	-	C	1
Q_x	Emission rate of species x, expressed according to the unit specified in IDUNIT(_x). The value is either positive or 0.	R	-	V	200.3

Complex area sources files

Two files must be supplied for complex area sources:

- general information file: it is a client file of SRF type, containing all information about the sources, except their geometric layout
- cartographic file: a DONCAR format file, describing the geometry of the polygons to which the emissions specified in the general information file are referred

The elements of the two files are linked through the IDGEOM field of each polygon.

❖ **General information file**

Field	Description	Type	Default value	Time dep.	Example
SRCEID	Unique identifier of the area source	I	-	C,V	100
IDGEOM / IDSURF	Unique identifier of the polygon.	I	-	C	100
IDADM1	Fist level administrative code of the territorial unit where the point source is located	I	-999	C	7
IDADM2	Second level administrative code of the territorial unit where the point source is located	I	-999	C	135
NAMADM	Name of the territorial unit where the source is located.	C20	NULL	C	Somewhere
NAMGP1	1st level description of the source	C8	NULL	C	Combust
NAMGP2	2nd level description of the source	C8	NULL	C	Resident
NAMGP3	3rd level description of the source	C16	-	C	Gas
IDCLA1	1st level code of the emitting category in the adopted classification scheme	I	-999	C	2
IDCLA2	2nd level code of the emitting activity in the adopted classification scheme	I	-999	C	2
IDCLA3	3rd level code of the emitting category in the adopted classification scheme	I	-999	C	2
IDCLA4	4th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDCLA5	5th level code of the emitting category in the adopted classification scheme	I	-999	C	-999
IDPROJ	Code of projection used for coordinates (see above for codes)	I	-	C	0
SHIFTX	coordinates shift for X axis (only with UTM)	R	0.	C	0
DZ	Thickness of the source (meters)	R	-	V	5
IDUNIT(_x)	Unit in which emission rates reported in field Q_x, or (in case of IDUNIT) in which emission rates of all species are expressed (see table above for units codes).	I	-	C	1
Q_x	Emission rate of species x, expressed according to the unit specified in IDUNIT(_x). The value is either positive or	R	-	V	200.3

	0.				

❖ **Cartographic file**

It is a DONCAR file (see *§ 2.5.1* for a description of file format) describing all the polygons that correspond to the area sources. The coordinates must be specified according to the projection system declared in IDPROJ field of the general information file.
Each polygon must have a unique value for the associated IDGEOM field. All records in the general information files must have a corresponding polygon object, to be linked through IDGEOM; multiple records in a general information file can be associated to the same polygon.

2.3 TIME-DEPENDENT FILES

The time variation of emissions can be represented in two ways:

- by a set of **modulation** files, describing the way the emission rate varies according to the hour in the day, the day of the week and the month of the year
- through time-**dependent client files,** directly specifying time series of the time-varying parameters for each source

and also

- through a set of files from **emission monitoring system (EMS),** also directly specifying time series of the emission flow rates.

2.3.1 MODULATION FILES

Following this approach, the emitted quantities are provided in the "client files" for a reference time period (e.g. on annual basis), and then modulated in time on an hourly basis according to a series of profiles.
The modulation criteria are assigned through 5 files (all of them are mandatory):

- three files containing a database of time profiles:
  - yearly profiles, on a monthly basis
  - weekly profiles, on a daily basis
  - daily profiles, on a hourly basis
- one file giving the correspondence between specific sources and the triplet of (yearly, weekly, daily) profiles to be used; through this file is thus possible to apply specific temporal modulations to individual sources
- one file giving the correspondence between the emission categories (e.g. SNAP) and the triplet of (yearly, weekly, daily) profiles to be used; the time modulations are then applied to all the sources belonging to a given categories and for which a specific modulation has not been defined.

Yearly modulation profiles file

Yearly modulation profiles are stored in a CSV file delimited by semicolon (";"). The file has a header, containing the fields names, followed by data records, each one corresponding to a yearly profile. The file must not contain more than blank lines at the end.
An example of such file is given below.

ID_PRF	NOM_PROFIL	PRF_1	PRF_2	PRF_3	PRF_4	PRF_5	PRF_6	PRF_7	PRF_8	PRF_9	PRF_10	PRF_11	PRF_12	TOTAL
1	AIRPARIF01	1.10	1.10	1.10	1.00	0.90	0.90	0.90	0.90	0.90	1.00	1.10	1.10	12.00
2	F(1/T)	2.49	1.14	1.32	0.77	0.56	0.47	0.36	0.40	0.57	0.54	1.22	2.16	12.00
3	365J/A	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	12.00
4	AIRPARIF02	1.55	1.55	1.55	1.00	0.45	0.45	0.45	0.45	0.45	1.00	1.55	1.55	12.00
5	AIRPARIF03	1.10	1.12	1.12	1.12	1.00	1.00	0.91	0.70	0.98	1.00	1.00	0.95	12.00
6	AOUT=	1.09	1.09	1.09	1.09	1.09	1.09	1.09	0.00	1.09	1.09	1.09	1.09	12.00
7	50%_AOUT	1.04	1.04	1.04	1.04	1.04	1.04	1.04	0.52	1.04	1.04	1.04	1.04	12.00
8	INDUS_PAPIER	1.35	1.41	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	1.01	1.28	12.00
9	F(T)	0.28	0.60	0.52	0.89	1.23	1.45	1.93	1.73	1.20	1.27	0.57	0.32	12.00
10	AIRPARIF06	0.60	0.70	0.80	1.00	1.30	1.40	1.40	1.20	1.30	1.00	0.70	0.60	12.00
11	AEROPORT	0.97	0.88	0.98	1.01	1.07	1.04	1.07	1.04	1.04	1.03	0.94	0.92	12.00
12	6MOIS/12	0.50	0.50	0.50	1.50	1.50	1.50	1.50	1.50	1.50	0.50	0.50	0.50	12.00

Fields content is as follow:

ID_PRF	numerical code of the profile
NOM_PROFIL	name given to profile (selected by user, limited to16 characters)
PRF_1	multiplier applied to average rate for month of January
PRF_2	multiplier applied to average rate for month of February
PRF_3	multiplier applied to average rate for month of March
PRF_4	multiplier applied to average rate for month of April
…	
PRF_11	multiplier applied to average rate for month of November
PRF_12	multiplier applied to average rate for month of December
TOTAL	sum of factors (must be equal to the sum of the 12 monthly values)

When, for a given profile, the sum of factors is equal to 12, the total emitted mass is conserved during modulation; otherwise, the mass is altered for a factor corresponding to TOTAL/12 ratio.

Weekly modulation profiles file

Weekly modulation profiles are stored in a CSV file delimited by semicolon (";"). The file has a header, containing the fields names, followed by data records, each one corresponding to a weekly profile. The file must not contain more than one blank line at the end.
An example of such file is given below.

ID_PRF	NOM_PROFIL	PRF_1	PRF_2	PRF_3	PRF_4	PRF_5	PRF_6	PRF_7	TOTAL
1	AIRPARIF01	1.06	1.06	1.06	1.06	1.06	0.85	0.85	7.00
2	7J_7	1.00	1.00	1.00	1.00	1.00	1.00	1.00	7.00
3	AIRPARIF03	1.08	1.08	1.08	1.08	1.08	0.80	0.80	7.00
4	LU_SA	1.17	1.17	1.17	1.17	1.17	1.17	0.00	7.00
5	TRAFIC	1.09	1.09	1.09	1.09	1.09	0.82	0.71	7.00
6	LU_VE	1.40	1.40	1.40	1.40	1.40	0.00	0.00	7.00
7	WE_S	0.63	0.63	0.63	0.63	0.63	1.92	1.93	7.00
8	GRIF	1.12	1.40	1.40	1.40	1.40	0.28	0.00	7.00

Fields content is as follow:

ID_PRF	numerical code of the profile
NOM_PROFIL	name given to profile (selected by user, limited to16 characters)
PRF_1	multiplier applied to average rate for day 1 (can be any day)
PRF_2	multiplier applied to average rate for day 2
PRF_3	multiplier applied to average rate for day 3
PRF_4	multiplier applied to average rate for day 4
PRF_5	multiplier applied to average rate for day 5
PRF_6	multiplier applied to average rate for day 6
PRF_7	multiplier applied to average rate for day 7
TOTAL	sum of factors (must be equal to the sum of the 7 daily values)

When, for a given profile, the sum of factors is equal to 7, the total emitted mass is conserved during modulation; otherwise, the mass is altered for a factor corresponding to TOTAL/7 ratio.

Daily modulation profiles file

Daily modulation profiles are stored in a CSV file delimited by semicolon (";"). The file has a header, containing the fields names, followed by data records, each one corresponding to a daily profile. The file must not contain more than one blank line at the end.
An example of such file is given below.

ID_PRF	NOM_PROFIL	PRF_23	PRF_24	PRF_1	PRF_2	PRF_3	PRF_4	PRF_5	PRF_6	PRF_7	PRF_8	PRF_9	PRF_10	PRF_11	PRF_12	PRF_13	PRF_14	PRF_15	PRF_16	PRF_17
1	AIRPARIF01	0.90	0.90	0.90	0.90	0.90	0.90	1.00	1.00	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.00
2	24H/ 24	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
3	AIRP02_1H	0.50	0.50	0.50	0.50	0.50	1.00	1.00	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.50	1.00	1.00
4	6H-22H_MAX	0.27	0.27	0.27	0.27	0.27	0.27	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37	1.37
5	8H-20H	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
6	TRAFIC	0.11	0.06	0.05	0.09	0.22	0.59	1.53	1.74	1.32	1.22	1.31	1.48	1.44	1.49	1.45	1.74	1.98	1.88	1.62
7	AIRPARIF06	0.80	0.80	0.80	0.80	0.80	0.80	1.00	1.00	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.20	1.00
8	F(T)	0.75	0.71	0.69	0.66	0.64	0.68	0.80	0.91	1.01	1.10	1.17	1.22	1.25	1.28	1.29	1.29	1.27	1.23	1.17

Fields content is as follow:

ID_PRF	numerical code of the profile
NOM_PROFIL	name given to profile (selected by user, limited to16 characters)
PRF_1	multiplier applied to average rate for the 1st hour (from 00 to 01)
PRF_2	multiplier applied to average rate for the 2nd hour (from 01 to 02)
PRF_3	multiplier applied to average rate for the 3rd hour (from 00 to 01)
PRF_4	multiplier applied to average rate for the 4th hour (from 00 to 01)
…	
PRF_22	multiplier applied to average rate for 22nd hour (from 21 to 22)
PRF_23	multiplier applied to average rate for 23rd hour (from 22 to 23)
PRF_24	multiplier applied to average rate for 24th hour (from 23 to 24)
TOTAL	sum of factors (must be equal to the sum of the 24 hourly values)

When, for a given profile, the sum of factors is equal to 24, the total emitted mass is conserved during modulation; otherwise, the mass is altered for a factor corresponding to TOTAL/24 ratio.

Souces-specific modulation profiles mapping file

This CSV file delimited by semicolon (";") is used to assign the correspondence between specific sources and the triplet of (yearly, weekly, daily) profiles to be used for time modulation. The file has a header, containing the fields names, followed by data records, each one corresponding to a specific source. The file must not contain more than one blank line at the end.
An example of such file is given below.

```
IDSRCE; NOM_PROFIL; TYP_OBJET; ID_PRF_M; ID_PRF_J; ID_PRF_H
113; EDF_TR1_2; 0; 2; 3; 14
227; Refinery1; 0; 1; 1; 26
3502; highway32; 2; 7; 4; 40
5521; highway33; 2; 7; 4; 41
5502; highway12; 2; 7; 4; 42
```

Fields content is as follow:

IDSRCE	numerical code corresponding to a specific source
NOM_PROFIL	name given to the combination (selected by user, limited to 16 characters)
TYP_OBJET	source geometry code (0 = point; 2 = line; 3 = area)
ID_PRF_M	yearly profile code
ID_PRF_J	weekly profile code
ID_PRF_H	daily profile code

The fields in this file are separated by a semicolon ";". The file must not contain more than one blank line at the end.

Categories-based modulation profiles mapping file

This CSV file delimited by semicolon (";") is used to assign the correspondence between emission categories (e.g. SNAP) and the triplet of (yearly, weekly, daily) profiles to be used for time modulation of emissions. The file has a header, containing the fields names, followed by data records, each one corresponding to a specific activity. The file must not contain more than one blank line at the end.
An example of such file is given below.

ID_CLA1	ID_CLA2	ID_CLA3	ID_CLA4	ID_CLA5	NOM_PROFIL	ID_PRF_M	ID_PRF_J	ID_PRF_H
1	1	1	0	0	SNAP_111	1	1	1
1	1	2	0	0	SNAP_112	1	1	1
1	1	3	0	0	SNAP_113	1	1	1
1	1	4	0	0	SNAP_114	1	1	1
1	1	5	0	0	SNAP_115	1	1	1
1	2	1	0	0	SNAP_121	2	1	1
1	2	2	0	0	SNAP_122	2	1	1
1	2	3	0	0	SNAP_123	2	1	1
1	2	4	0	0	SNAP_124	2	1	1
1	2	5	0	0	SNAP_125	2	1	1
1	3	1	0	0	SNAP_131	3	2	2
1	3	2	0	0	SNAP_132	3	2	2
1	3	3	0	0	SNAP_133	3	2	2

Fields content is as follow:

ID_CLA1	1st level code of the emission category
ID_CLA2	2nd level code of the emission category
ID_CLA3	3rd level code of the emission category
ID_CLA4	4th level code of the emission category
ID_CLA5	5th level code of the emission category
NOM_PROFIL	mnemonic name given to the combination
ID_PRF_M	yearly profile code
ID_PRF_J	weekly profile code
ID_PRF_H	daily profile code.

2.3.2 TIME-DEPENDENT CLIENT FILES

Time-dependent client files contain the sources parameters varying in time. Emission rates and positions are represented by a constant value between two dates: DATEDEB, the beginning of the time interval, DATEFIN, the end of the time interval.

The rules described in §1.2 are available for these files. The following added rules must be respected:

- Dates are given through the format : DD-MM-YYYY HH:MN:SS where DD is the day, MM the month, YYYY the year, HH the hour, MN the minutes, SS the second. For example, the 2004,1st April at 12 should be written as : 01-04-2004 12:00:00.
- The time series are ordered by time
- If a source is not present at a given time, the default behaviour of the application will be to calculate the corresponding modulated rates if modulation files are given, take into account the nominal rate given by the constant emission file if not. The position will be given by the constant emission file.
- All the fields for a given source and a given time must be filled with a valid value.
- Only fields with a "V" value in the "time dependency" column can be present.
- Emissions rates for a given specie are given using the corresponding IDUNIT of the constant emission file.

Example of a time series file for a point source (GSP) :

SRCEID	DATEDEB	DATEFIN	Q_NOx
1	01-04-2004 12:00:00:00	01-04-2004 15:00:00	1000.
2	01-04-2004 12:00:00:00	01-04-2004 14:00:00	2000.
2	01-04-2004 14:00:00:00	01-04-2004 16:00:00	1500.
1	01-04-2004 15:00:00:00	01-04-2004 16:00:00	1500.

In that case, if SO_2 emissions have been defined in the time independent file, they will be taken into account from the modulation file if present or with the nominal rate if not.

As for time independent files, you must have one file for each type of source (GSP, LIN or SRF).

2.3.3 EMISSION MONITORING SYSTEMS (EMS) FILES

EMS files (ingested by INEMOS) are ASCII files delimited by semicolon (CSV) containing time-varying (e.g. hourly) data of a set of monitored species and stack exit parameters.
An example of such files is given below.

```
IDSRCE;Date/hour;CO instrumental;NOx instrumental (as NO2);Fuel flow;Dry smoke
flow;Smoke temperature;Plant status
2;08/04/2005 0.00;1.8;24;42929;1274441;95;
2;08/04/2005 1.00;1.8;24;42983;1274911;95;
2;08/04/2005 2.00;1.8;24;42963;1274657;95;
2;08/04/2005 3.00;1.8;24;42914;1273398;95;
2;08/04/2005 4.00;1.9;24;42633;1265427;95;
2;08/04/2005 5.00;N.A.;45;30283;1279991;96;
2;08/04/2005 6.00;N.A.;60;27961;1343244;108;
2;08/04/2005 7.00;N.A.;60;28115;1342794;109;
2;08/04/2005 8.00;N.A.;N.A.;31219;1273641;108;MN
2;08/04/2005 9.00;12.8;28;47202;1392022;101;
2;08/04/2005 10.00;1.1;35;62153;1712014;104;
2;08/04/2005 11.00;0.5;18;31144;855520;52;
2;08/04/2005 12.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 13.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 14.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 15.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 16.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 17.00;N.A.;N.A.;N.A.;N.A.;N.A.;AC
2;08/04/2005 18.00;1;38;62176;1719536;103;
2;08/04/2005 19.00;0.9;38;62094;1702999;104;
2;08/04/2005 20.00;0.9;37;62261;1710650;104;
2;08/04/2005 21.00;0.9;37;62317;1715176;104;
2;08/04/2005 22.00;0.9;37;62332;1717601;104;
2;08/04/2005 23.00;0.9;37;62392;1719649;104;
2;09/04/2005 0.00;0.9;37;62457;1721963;104;
2;09/04/2005 1.00;0.9;37;62450;1721856;104;
2;09/04/2005 2.00;0.8;36;62531;1724390;104;
2;09/04/2005 3.00;0.8;36;62611;1726059;104;
2;09/04/2005 4.00;0.8;36;62650;1726156;104;
2;09/04/2005 5.00;0.8;36;62648;1725982;104;
2;09/04/2005 6.00;0.8;36;62671;1727371;104;
2;09/04/2005 7.00;0.8;36;62816;1730859;104;
2;09/04/2005 8.00;0.8;37;62782;1728718;104;
2;09/04/2005 9.00;0.8;37;62705;1727157;104;
2;09/04/2005 10.00;0.7;37;62669;1727923;108;
2;09/04/2005 11.00;0.7;37;62611;1728595;108;
2;09/04/2005 12.00;0.7;38;62464;1764841;108;
2;09/04/2005 13.00;0.8;37;62701;1744307;54;
2;09/04/2005 14.00;0.8;38;62724;1730901;105;
2;09/04/2005 15.00;0.8;38;62736;1733137;104;
2;09/04/2005 16.00;0.8;38;62718;1732126;105;
2;09/04/2005 17.00;0.8;37;63162;1741328;104;
2;09/04/2005 18.00;0.9;37;63234;1748298;104;
2;09/04/2005 19.00;1.3;36;49536;1834468;101;
2;09/04/2005 20.00;5.4;18;10922;1251449;119;
2;09/04/2005 21.00;3;0;0;0;84;MN
2;09/04/2005 22.00;1.6;0;0;0;88;MN
2;09/04/2005 23.00;1.6;1;1112;75013;81
```

Fields content is as follow:

- The first column must contain the stacks code (matching the ones used in the corresponding point sources “client file” - GSP file).

- The second column must contain date and time, separated by a blank; date can be specified either as DD/MM/YYYY or as DD.MM.YYYY; time cen be specified either as HH.MM.SS or as HH:MM:SS (minutes and seconds can be omitted).
- Other columns can be freely used; their interpretation is specified in INEMOS initialization file (see Annex 0).

2.4 SPECIES SPLIT/SPECIATION

2.4.1 SPLITTING PROFILES DATABASE

The database of split/speciation profiles used for a given chemical species is stored in an ASCII file delimited by semicolon (CSV). The file has a header, containing the fields names, followed by data records.
An example of such file is given below.

```
PROFILE_ID;HCHO;CCHO;RCHO;…;ISOP;TRP1;XC;COMMENTS
0;1.680;0.907;2.150;…;0.432;0.680;32.601;Over All Average
1;47.200;0.000;0.000;…0.000;0.000;15.000;Ext. Combustion Boiler - Residual Oil
2;46.886;0.000;0.000;…;0.000;0.000;0.000;Ext. Combustion Boiler – Distill. Oil
3;18.202;0.000;0.000;…;0.000;0.000;3.440;Ext. Combustion Boiler - Natural Gas
4;8.236;0.000;0.000;…;0.000;0.000;21.915;Ext. Combustion Boiler - Refinery Gas
5;0.000;0.000;0.000;…;0.000;0.000;18.802;Ext.Combustion Boiler - Coke Oven Gas
```

Each record corresponds to a profile:

- the first column (PROFILE_ID) contains the numerical code identifying the profile
- a series of columns (one for each target species) contain the weights of each species, as percentage of the species respect to total mass; the names of the target species correspond to the names of those columns
- the last column (COMMENTS) contains free comments.

2.4.2 CATEGORIES-BASED SPLIT PROFILES MAPPING FILE

This CSV file delimited by semicolon (";") is used to specify the correspondence between emission categories (e.g. SNAP) and the profiles to be used to split/speciate the emissions of a given input species.
The file has a header, containing the fields names, followed by data records, each one corresponding to a specific category.
An example of such file is given below.

```
IDCLA1;IDCLA2;IDCLA3;IDCLA4;IDCLA5;PROFILE_ID;SNAP_Desc;Profile_Desc
1;0;0;0;0;2;Combustion: energy & transform.;Ext. Combustion Boiler - Coal
1;1;0;0;0;2;Production of electricity;Ext. Combustion Boiler - Coal
1;1;1;0;0;2;Boilers >= 300 MW;Ext. Combustion Boiler - Natural Coal
1;2;0;0;0;3;District heating;Ext. Combustion Boiler - Natural Gas
1;2;1;0;0;3;Boilers >= 300 MW;Ext. Combustion Boiler - Natural Gas
```

Fields content is as follow:

- IDCLA1: 1st level code of the emission category
- IDCLA2: 2nd level code of the emission category
- IDCLA3: 3rd level code of the emission category
- IDCLA4: 4th level code of the emission category
- IDCLA5: 5th level code of the emission category
- PROFILE: numerical code of the split/speciation profile
- two optional fields at the end containing comments

2.4.3 SOURCES-SPECIFIC SPLIT PROFILES MAPPING FILE

This CSV file delimited by semicolon (";") is used to assign the correspondence between individual sources and the profiles to be used to split/speciate the emissions of a given input species.
The file has a header, containing the fields names, followed by data records, each one corresponding to a specific category.
An example of such file is given below.

```
ID_SRCE; TYP_OBJET; PROFILE_ID
113; EDF_TR1_2; 21
227; Refinery1; 11
3502; highway32; 32
5521; highway33; 32
5502; highway12; 32
```

Fields content is as follow:

- ID_SRCE: numerical code corresponding to a specific source
- TYP_OBJET: source geometry code (0 = point; 2 = line; 3 = area)
- PROFILE_ID: numerical code of the split/speciation profile

2.5 GEOGRAPHIC REFERENCES

2.5.1 VECTOR CARTOGRAPHY FILE (DONCAR)

This file describes the geometrical contours of area source emissions and can be generated from a DXF file with the help of “Site Manager" (Data\Map\Import and Export). The objects in the "doncar" file are defined as polygons. The objects are defined by the coordinates Xi, Yi for each point.

The file can also contain point and line (vectors) objects but they will only be regarded as cartographical objects and not as point or line emission sources.

The cartography file is an ASCII file with the following structure:

1. Version number: IVENUM 201
2. Number of points
3. Number of lines and maximum number of points on the line
4. Number of polygons and maximum number of points per polygon
5. Site name
6. Type of projection
   0 = Lambert II
   NN = UTM zone
   99 = Latitude–Longitude
   For UTM coordinates, the zone is indicated by XDECAL.
7. Site name
8. Selection of unit
   0 = kilometre
   1 = metre
   99 = degrees
9. Specification of numbers
   2 = 2 digits considered
   4 = 4 digits considered
10. Selection of coordinates
    0 = absolute
    1 = relative
11. Selection of height
    1 = above-ground height
    0 = altitude
    –999 = none

12. Southwest point coordinates

13. Domain size

14. Point specifications

Example of point specification (PTS):

```
ELT. TYPE  =  PTS.
ELT.  NUM. / OBJ. :    10    10
POSIT   POINT
501.0000  2485.0000
```

A number is indicated for identification of each point. When the number is repeated twice on the same line, it cannot be used again for another point. The coordinates defining the position of the point is indicated by using the projection and unit previously defined.

15. Line specifications

Example of line specification (FIL)

```
ELT. TYPE  =  FIL.
ELT.  NUM. / OBJ. :    4    4
ROUTENAT LINE
12(1)  4(2)
502.5411  2497.6917    501.7886  2496.6692    503.2123  2496.1868    504.6909  2495.5132
505.5246  2493.2295    506.5469  2492.2886    508.2405  2491.5623    507.9992  2489.2500
507.6481  2487.3677    506.6281  2485.8364    505.7133  2484.4392    504.8271  2483.0681
```

A number is indicated for identification of each line. When the number is repeated twice on the same line, it cannot be used again for another line. Next, the user adds a comment on the line name representing:

- the number of points on the line
- the number of coordinate couples on each line

The coordinate couples of the different points setting up the line are indicated at the end.

16. Polygon specifications

Example of polygon specification (POL)

```
ELT. TYPE  =  POL.
ELT.  NUM. / OBJ. :    6    6
CPN    ZONE IND
7(3)  4(4)
506.0907  2494.2495    507.0581  2493.4971    507.9706  2494.3044    507.7843  2494.8420
507.1919  2494.8420    506.7356  2495.0305    506.1695  2494.3569
```

A number is indicated for identification of each polygon. When the number is repeated twice on the same line, it cannot be used again for another polygon. Next, the user adds a comment on the polygon name representing:

- the number of points on the polygon
- the number of coordinate couples of each polygon

The coordinate couples corresponding to the different points setting up the polygon are indicated at the end.

Example

```
* IVENUM
         201
 * NB. PTS
           5
 * NB. FIL - NB. MAX PTS/FIL.
           3           13
 * NB. POL - NB. MAX PTS/POL.
           3           46
 * ------------------------------------------------------------------------
 *  CARTOG . SIT.
 * ------------------------------------------------------------------------
 ************************************************************************
 *  SIT. NOM.
           NOWHERE
 * ELT. PROJ =>ICOGEO=0:LII =NN:UTM ZONE =99:LATLONG
    ICOGEO   XDECAL: X SHIFT
           0      0.0000
 * ELT. UNIT   =  0:KM  1:M  99:DEG
           0
 * ELT. PRECISION  2=F11.2  4=F11.4
           4
 * ELT. RELAT. FOR LINES AND POLYGONS =
 *      0=ABSOLUTE  1:RELATIVE TO SW PT
           0
 * ELT. HEIGHT TYPE =1:HEIGHT/GROUND, =0:ALTITUDE, =-999: NO HEIGHT
    -999
    COORD. X - Y S-W KM
           0.0000
           0.0000
    DIM.   X - Y     KM
           0.0000          0.0000
 * ELT. TYPE   =
    PTS.
    ELT.    NUM. / OBJ.  :     10     10
    POSIT    POINT
           501.0000   2485.0000
    ELT.    NUM. / OBJ.  :      1      1
    POSIT    RECEPT2
           509.0407   2493.8481
    ELT.    NUM. / OBJ.  :      2      2
    POSIT    POINT
           504.6192   2492.1785
    ELT.    NUM. / OBJ.  :      3      3
    POSIT    POINT
           510.2924   2491.5168
    ELT.    NUM. / OBJ.  :     11     11
    POSIT    POINT
           514.2266   2490.3345
 * ELT. TYPE   =
    FIL.
    ELT.    NUM. / OBJ.  :      4      4
    ROUTENAT LINE
           12    4
502.5411 2497.6917 501.7886 2496.6692 503.2123 2496.1868 504.6909 2495.5132
```

```
505.5246 2493.2295 506.5469 2492.2886 508.2405 2491.5623 507.9992 2489.2500
507.6481 2487.3677 506.6281 2485.8364 505.7133 2484.4392 504.8271 2483.0681
    ELT.    NUM. / OBJ.  :      5       5
    ROUTEDEP LINE
           9    4
514.6924 2493.3894 513.9399 2493.7119 512.7838 2494.0083 511.3052 2494.2783
510.3115 2493.7671 509.3704 2492.6636 508.9117 2491.6697 508.3743 2491.6147
508.2405 2491.5359
    ELT.    NUM. / OBJ.  :      9       9
    ROUTEDEP RD134
           13    4
500.3912 2493.6475 500.9860 2492.2168 502.2759 2490.7524 503.6709 2490.5779
505.4505 2489.8447 506.2173 2488.5190 507.9276 2488.3447 509.2175 2488.6599
510.7176 2488.9036 511.9048 2487.8572 512.5665 2486.6367 513.9280 2486.4265
514.3460 2486.8110
 * ELT. TYPE   =
    POL.
    ELT.    NUM. / OBJ.  :      6       6
    CPN      ZONE IND
           7    4
506.0907 2494.2495 507.0581 2493.4971 507.9706 2494.3044 507.7843 2494.8420
507.1919 2494.8420 506.7356 2495.0305 506.1695 2494.3569
    ELT.    NUM. / OBJ.  :      7       7
    VILLE    POLYG
           6    4
512.0577 2492.5845 512.6214 2490.9985 514.1549 2491.2922 513.8325 2492.4485
512.5688 2492.8784 511.8976 2493.0408
    ELT.    NUM. / OBJ.  :      8       8
    VILLE    POLYG
           46    4
506.4251 2491.6626 506.1456 2491.3472 505.7610 2490.7524 505.5150 2489.6704
505.5150 2489.1807 505.4816 2488.6553 505.4816 2488.1296 505.6559 2487.8167
505.7610 2487.5708 506.0764 2487.3606 506.2841 2487.1167 506.5995 2486.8374
506.9506 2486.6272 507.1584 2486.5220 507.8942 2486.3118 508.0685 2486.3118
508.3146 2486.3118 508.6275 2486.3813 509.1530 2486.8015 509.3274 2487.0476
509.3967 2487.5015 509.5017 2488.5142 509.5734 2488.6909 509.6427 2489.0398
509.7478 2489.4243 509.8171 2489.7729 509.8171 2489.9832 509.6761 2490.3679
509.2581 2490.6140 509.0837 2490.7524 508.8735 2491.0320 508.8735 2491.2087
508.9428 2491.5933 508.9786 2491.7317 508.7684 2492.1165 508.4173 2492.3625
508.0327 2492.3625 507.6839 2492.5728 507.3686 2492.7112 506.8455 2492.8162
506.5995 2492.7805 506.3200 2492.5010 506.2507 2492.2573 506.3893 2492.0112
506.4609 2491.9063 506.4251 2491.7317
```

2.5.2 GRIDDED THEMATIC DATA (LDU)

Used by TRLCRS. LDU are ASCII files storing gridded thematic data. An LDU file is organized in sections, each one containing a matrix of gridded data. Each section starts with a header, followed by data records. An example of a file section is given below.

```
#- NUMVER ----------------------------------------------------------------
 300
#- NOMLDU ----------------------------------------------------------------
D:\PROJECTS\PIEMONTE\LANDUSE\PM_layers.cos
#- NOMSIT ----------------------------------------------------------------
ALSAC
#- DDCSLU ----------------------------------------------------------------
 29 11 99
#- DDVAL1 ----------------------------------------------------------------
 1 1 70 0 0 0
#- DDVAL2 ----------------------------------------------------------------
```

```
 31 12 10 24 0 0
#- NOMRLU ----------------------------------------------------------------
AF
#- INFUSC ----------------------------------------------------------------
CORINE LAND COVER
#- INFUPR ----------------------------------------------------------------
#- VSONAM ----------------------------------------------------------------
BATI
#- VSOUNI ----------------------------------------------------------------
PERCENT
#- IVSOCO ----------------------------------------------------------------
 910
#- IUSOCO ----------------------------------------------------------------
 1
#- VSOMIN ----------------------------------------------------------------
 0.0000000000E+00
#- VSOMAX ----------------------------------------------------------------
 90.00000000
#- IVARIN ----------------------------------------------------------------
 0
#- ICOGEO ----------------------------------------------------------------
 32
#- XDECAL ----------------------------------------------------------------
    0.0000
#- VAXNAM ----------------------------------------------------------------
OX
#- VAXUNI ----------------------------------------------------------------
KM
#- IVAXSO ----------------------------------------------------------------
 102
#- IUNXSO ----------------------------------------------------------------
 201
#- VAYNAM ----------------------------------------------------------------
OY
#- VAYUNI ----------------------------------------------------------------
KM
#- IVAYSO ----------------------------------------------------------------
 103
#- IUNYSO ----------------------------------------------------------------
 201
#- XSEIGN ----------------------------------------------------------------
  319.0000
#- YSEIGN ----------------------------------------------------------------
 5244.0005
#- RELMDA ----------------------------------------------------------------
    6.6048
#- REPHI  ----------------------------------------------------------------
   47.3235
#- IREIGN ----------------------------------------------------------------
 1
#- JREIGN ----------------------------------------------------------------
 1
#- XSOIGN ----------------------------------------------------------------
  459.0000
#- YSOIGN ----------------------------------------------------------------
 5456.0005
#- TOLMDA ----------------------------------------------------------------
    8.4366
#- TOPHI  ----------------------------------------------------------------
   49.2544
#- ITOIGN ----------------------------------------------------------------
 720
#- JTOIGN ----------------------------------------------------------------
```

```
 1080
#- XSXIGN ----------------------------------------------------------------
  140.0000
#- YSYIGN ----------------------------------------------------------------
  212.0000
#- I1     ----------------------------------------------------------------
 36
#- J1     ----------------------------------------------------------------
 54
#- XRES2D ----------------------------------------------------------------
0.0000E+00 4.000 8.000 12.00 16.00 20.00 24.00 28.00 32.00 36.00 40.00 44.00
48.00 52.00
56.00 60.00 64.00 68.00 72.00 76.00 80.00 84.00 88.00 92.00 96.00 100.0 104.0
108.0 112.0
116.0 120.0 124.0 128.0 132.0 136.0 140.0
#- YRES2D ----------------------------------------------------------------
 0.0000E+00 4.000 8.000 12.00 16.00 20.00 24.00 28.00 32.00 36.00 40.00 44.00
48.00 52.00
56.00 60.00 64.00 68.00 72.00 76.00 80.00 84.00 88.00 92.00 96.00 100.0 104.0
108.0 112.0
116.0 120.0 124.0 128.0 132.0 136.0 140.0 144.0 148.0 152.0 156.0 160.0 164.0
168.0 172.0
176.0 180.0 184.0 188.0 192.0 196.0 200.0 204.0 208.0 212.0
#- VSOL2D ----------------------------------------------------------------
       0.0000     0.0000      0.0000      0.2300      0.0000      0.1100
       0.6700     0.0000
       0.0000     0.0000      1.0000      0.4300      0.0000      0.0000
       0.1070     0.0000
       0.6300     0.1200      0.0000      0.0000      0.3200      0.0000
       0.0000     0.0000
       0.0000     0.5400      0.0000      0.8500      1.0000      0.9800
       0.0000     0.0000
       0.1230     0.0000      0.0000      0.0000
       0.0000     0.0000      0.0000      0.4300      0.0000      0.2200
       0.0300     0.0000
       0.4600     0.0000      0.0000      0.0000      0.0000      0.0000
       0.0000     0.0000
       ...
       ...
       0.8700     0.5600      0.0000      0.6700      0.0000      0.0000
       0.6000     0.0000
       0.0000     0.1200      0.4300      0.0000      0.9800      0.0000
       0.3400     0.0000
```

Comment lines begins with a "#". Variables have the following meaning:

- NUMVER: version code
- NOMLDU: name of the originating file
- NOMSIT: site name
- DDCSLU: creation date of the file
- DDVAL1: starting date and hour of the data validity period
- DDVAL2: ending starting date and hour of the data validity period
- NOMRLU: name of file creator
- INFUSC: type of source database
- INFUPR: type of processing
- VSONAM: variable name

- VSOUNI: variable unit
- IVSOCO: variable code
- IUSOCO: variable unit code
- VSOMIN: minimum value of the data matrix
- VSOMAX: maximum value of the data matrix
- IVARIN: flag indicating to the presence of the variance matrix
- ICOGEO: coordinate system code
- XDECAL: false East
- VAXNAM: x-coordinate variable name
- VAXUNI: name of the x-coordinate unit
- IVAXSO: code of the x-coordinate variable
- IUNXSO: code of the x-coordinate unit
- VAYNAM: y-coordinate variable name
- VAYUNI: name of the y-coordinate unit
- IVAYSO: code of the y-coordinate variable
- IUNYSO: code of the y-coordinate unit
- XSEIGN: x-coordinate of SW cell centerpoint (km)
- YSEIGN: y-coordinate of SW cell centerpoint (km)
- RELMDA: longitude of SW cell centerpoint (deg, from GM)
- REPHI: latitude of SW cell centerpoint (deg, from GM)
- IREIGN: index I of SW corner in source file
- JREIGN: index J of SW corner in source file
- XSOIGN: x-coordinate of NE cell centerpoint (km)
- YSOIGN: y-coordinate of NE cell centerpoint (km)
- TOLMDA: longitude of NE cell centerpoint (°, from GM)
- TOPHI: latitude of NE cell centerpoint (°, da GM)
- ITOIGN: index I of NE corner in source file
- JTOIGN: index J of NE corner in source file
- XSXIGN: West-East span of the domain (unit: VAXUNI)
- YSYIGN: South-North span of the domain (unit: VAYUNI)
- I1: number of points of the grid along x
- J1: number of points of the grid along y
- XRES2D: vector of x grid points coordinates, starting from XSEIGN
- YRES2D : vector of y grid points coordinates, starting from YSEIGN
- VSOL2D: matrix of values (I1 x J1 numbers, from North to South and then from West to East).

2.5.3 CATEGORIES - THEMATIC LAYERS MAPPING FILE (SNAP-CRS)

This fixed format ASCII file is used by TRLCRS to assign the correspondence between emission categories and gridded thematic layers to be used for spatial disaggregation of area sources.
Each record corresponds to a emission category (e.g. SNAP). The first 10 columns contain the numerical code of the category (two characters for each level of classification), followed by a blank and by a 4-character label identifying the thematic layer to be used (such label must match the VSONAM label of a layer in the employed LDU file). Data records must be sorted accorded to categories numerical codes.
An example of such file is given below.

```
 1 5 4 0 0 INDUS
 1 5 5 0 0 INDUS
 1 5 6 0 0 INDUS
 2 0 0 0 0 BATI
 2 2 2 0 0 BATI
 2 2 3 0 0 BATI
 2 2 4 0 0 BATI
 2 2 5 0 0 BATI
 2 3 0 0 0 CULT
 2 3 1 0 0 CULT
 8 4 3 0 0 EAU
 8 4 4 0 0 EAU
 8 5 0 0 0 BATI
 8 6 0 0 0 CULT
 8 7 0 0 0 CONI
 8 8 0 0 0 INDUS
 8 9 0 0 0 BATI
 810 0 0 0 PRAI
1011 7 0 0 CONIF
```

3 MODULES REFENCE

In this section each Emission Manager module is synthetically described, with the list of input and output files, its calling syntax from the command line and the details on the individual arguments.

The following **conventions** are adopted in command line arguments tables:

- Type: indicates whether the argument refers to:
  - a program input ("I");
  - output ("O");
  - or a calculation flag ("FLAG");
- Format: it can be:
  - "integer";
  - "real" (with or without exponent);
  - "char(x)" (a string of maximum "x" characters);
  - a special formatted value (e.g. a date given as "YY/MM/DD");
  - or a value chosen from a fixed list (e.g. "0 / 1").

Examples of running scripts are given in Annex B.

3.1 GENERAL MODULES

3.1.1 GENERATION OF "PEMTIM" FILES FROM "CLIENT FILES"

TRLGSP, **TRLLIN** and **TRLCRS** modules allow to generate "pemtim" files starting from "client files".

In case of line sources, emissions related to sources polylines are also allocated to the individual segments which make each polyline, according to their lengths.

In case of area sources, emissions related to sources polygons are also distributed to the cells of a user-defined target grid, optionally taking into account gridded proxies.

As for time aspects, the output "pemtim" will contain the share of the input emitted mass corresponding to the specified target time interval (e.g. if input emissions are in mass/year and the target time period is three days, the output "pemtim" will contain emissions corresponding to 3/365 of the input quantities). In most cases a proper time modulation is performed in a further processing step: in such cases, when generating the initial "pemtim" a conventional target time interval of 24 hours is usually employed.

TRLGSP *v.2.0*

TRLGSP performs the ingestion of inventory data for point sources, assigned as a "point sources client file" (a CSV file, see § 2.2.2), producing in output a "pemtim" file in ASCII or binary format.
Data filling of stacks parameters that are specified as "missed" in input data file is performed by TRLGSP on the basis of a category-based file (see Annex A.4), specifying for each category typical stack height, diameter, exit speed and temperature. This allows to maintain the missing data in the original database while keeping trace of the assumptions made to fill them.

Input/output files

INPUT	• A "point sources client file" • A file defining the categories nomenclature (e.g. SNAP) • *A file defining the default values of the stack parameters, by category (optional)* • *A DONCAR file defining the sources coordinates (optional)*
OUTPUT	• A "pemtim" emission file, with a number of sources equal to the number of records in the "client file" that are located inside the target domain defined by the user • A "pemspe" file containing all the substances listed in the input "client file" • A mass balance file reporting the total emission of each substance, by category

Calling syntax

```
trlgsp      <gsp_file> <ref_class> <pemtim_out> <pemspe_out> <ascii_bin>
            <xorig> <yorig> <nx > <ny> <dx> <dy>
            <starting_date> <starting_hour> <ending_date> <ending_hour>
            <time_res> [optional_arguments]
```

Command-line arguments

Argument	Type	Description	Format	Unit
gsp_file	I	name of the "client file" containing point sources emissions	Char(256)	-
ref_class	I	file containing the reference nomenclature (classification of categories)	Char(256)	-
pemtim_out	O	name of the output "pemtim"	Char(256)	-
pemspe_out	O	name of the output pemspe	Char(256)	-
ascii_bin	FLAG	type of the output "pemtim" (0 = ASCII; 1 = binary)	0/1	-
xorig	I	X coordinate of the SW point of the target grid	Real	Km
yorig	I	Y coordinate of the SW point of the target	Real	Km

Argument	Type	Description	Format	Unit
		grid		
nx	I	number of point in X of the target grid	Integer	-
ny	I	number of point in X of the target grid	Integer	-
dx	I	grid cell size along X	Integer	Km
dy	I	grid cell size along Y	Integer	Km
starting_date	I	starting day of the target simulation period	YYYY/MM/DD	-
starting_hour	I	starting hour of the target simulation period	HH/MM/SS	-
ending_date	I	ending day of the target simulation period	YYYY/MM/DD	-
ending_hour	I	ending hour of the target simulation period	HH/MM/SS	-
time_res	I	time resolution (max = 999/59/59)	HH/MM/SS	-

Optional arguments

-wascii <ascii_output_file>

ascii_output_file	I	Contains the summary of total emissions	Char(256)	-

Allows to obtain an output an ASCII file reporting the total emission of each substance, by category

-nosnap

Allows to obtain a "pemtim" output file without a categories (e.g. SNAP) detail: contributions of all categories insisting on a given source are summed up

-carto <doncar_file>

doncar_file	I	geographical file containing the position of each emission source	Char(256)	-

Allows referring to an external DONCAR file for sources coordinates
See § 2.5.1 for a description of the DONCAR file format.

-projection <icocib> <xdecib>

icocib	I	code of the projection system of the target mesh	Integer	-
xdecib	I	shift along x	Real	km

Allows to change the projection of the target mesh.
Possible codes of the projection system are:

0:	French Lambert 2
NN > 0:	UTM Zone NN in northern hemisphere
NN < 0:	UTM Zone NN in southern hemisphere
99:	Longitude - Latitude in decimal degrees

-csvspefile

Cause the creation of an output species file in CSV format.

-check <ficdef>

ficdef	I	file containing the permitted range and default stack parameters by category ("gspdef.txt")	Char(256)	-

Check the values in the client files against the permitted range by category.
See A.3 for "gspdef.txt" file format.

–default <ficdef>

ficdef	I	file containing the permitted range and default stack parameters by category ("gspdef.txt")	Char(256)	-

Force the adoption of the default stack parameters by category for the sources in the client files having those parameters outside the permitted range.
See A.3 for "gspdef.txt" file format.

TRLCRS *v.2.0*

TRLCRS performs the ingestion of inventory data for area sources, with spatial disaggregation on the target grid with optional use of thematic layers. Input emissions data are in the form of "area sources client file" (a CSV file, see 2.2.4), while disaggregated data in output are stored in a "pemtim" file in ASCII or binary format.
Spatial disaggregation is performed intersecting the emission polygons (assigned through an additional DONCAR file) with the target simulation grid: for each polygon, emissions are partitioned according to the share of the polygon-grid cell intersection area respect to the area of the polygon. Such partition can be optionally weighted by the information contained in a gridded thematic layer (e.g. obtained from land-use or other type of cartographic data), specifying for each cell the percentage of the area occupied by a selected feature (e.g. built-up areas, vegetation, etc.). A different thematic layer can be employed for each emission category.

Input/output files

INPUT	• An "area sources client file" • A file defining the categories nomenclature (e.g. SNAP) • *A file defining the boundaries of the polygons associated to the area sources (DONCAR format; see § 2.5.1 for format description)* • *A file defining the correspondence between emission categories and thematic layers (see § 2.5.3)* • *A file the file containing the gridded thematic layers (LDU format; see § 2.5.2 for format description)*
OUTPUT	• A "pemtim" emission file, with emissions disaggregated on the cells of the target grid defined by the user • A "pemspe" file containing all the substances listed in the input "client file

Calling syntax

```
trlcrs      <srf_file> <ref_class> <pemtim_out> <pemspe_out> <ascii_bin>
            <xorig> <yorig> <nx > <ny> <dx> <dy>
            <starting_date> <starting_hour> <ending_date> <ending_hour>
            <time_res> [optional_arguments]
```

Command-line arguments

Argument	Type	Description	Format	Unit
srf_file	I	name of the "client file" containing area sources emissions	Char(256)	-
ref_class	I	file containing the reference nomenclature (classification of categories)	Char(256)	-
pemtim_out	O	name of the output "pemtim"	Char(256)	-
pemspe_out	O	name of the output pemspe	Char(256)	-
ascii_bin	FLAG	type of the output "pemtim" (0 = ASCII; 1 = binary)	0/1	-
xorig	I	X coordinate of the SW point of the target grid	Real	Km
yorig	I	Y coordinate of the SW point of the target grid	Real	Km
nx	I	number of point in X of the target grid	Integer	-
ny	I	number of point in X of the target grid	Integer	-
dx	I	grid cell size along X	Integer	Km
dy	I	grid cell size along Y	Integer	Km
starting_date	I	starting day of the target simulation period	YYYY/MM/DD	-
starting_hour	I	starting hour of the target simulation period	HH/MM/SS	-
ending_date	I	ending day of the target simulation period	YYYY/MM/DD	-
ending_hour	I	ending hour of the target simulation period	HH/MM/SS	-
time_res	I	time resolution (max = 999/59/59)	HH/MM/SS	-

Optional arguments

-nosnap

Allows to obtain a "pemtim" output file without a categories (e.g. SNAP) detail: contributions of all categories insisting on a given source are summed up

-carto <doncar_file>

doncar_file	I	Geographical file containing the boundaries of all the emitting polygons	Char(256)	-

Allows referring to an external DONCAR file for polygons geometry
See § 2.5.1 for a description of the DONCAR file format.

-projection <icocib> <xdecib>

icocib	I	code of the projection system of the target mesh	Integer	-

xdecib	I	shift along x	Real	km

Allows to change the projection of the target mesh.
Possible codes of the projection system are:

0:	French Lambert 2
NN > 0:	UTM Zone NN in northern hemisphere
NN < 0:	UTM Zone NN in southern hemisphere
99:	Longitude - Latitude in decimal degrees

-csvspefile

Cause the creation of an output species file in CSV format.

-keepinfo

Keeps the code of the originating polygon (field IDSURF/IDGEOM) in each source of the output "pemtim"

-check <ficdef>

ficdef	I	file containing the permitted range and default stack parameters by category ("gspdef.txt")	Char(256)	-

Check the values in the client files against the permitted range by category.
See A.3 for "gspdef.txt" file format.

–default <ficdef>

ficdef	I	file containing the permitted range and default stack parameters by category ("gspdef.txt")	Char(256)	-

Force the adoption of the default stack parameters by category for the sources in the client files having those parameters outside the permitted range.
See A.3 and A.4 for "gspdef.txt" file format.

–crs <snap_crs> <ldu_file>

snap_crs	I	file containing the corrispondence table between emission categories and thematic layers	Char(256)	-
ldu_file	I	file containing the gridded thematic layers (LDU format; see § 2.5.2)	Char(256)	-

Activate the spatial disaggregation using also gridded thematic layers.

-filetime <time_dependent_file_name>

Name of the optional time-dependent client file in input, containing the sources parameters varying in time.

TRLLIN *v.2.0*

TRLLIN performs the ingestion of inventory data for line sources. Input emissions data are in the form of a "line sources client file" (a CSV file, see § 2.2.3), related either to individual segments or to complex polylines (assigned through an additional DONCAR file). Output data are stored in a "pemtim" file in ASCII or binary format

Input/output files

INPUT	• A "line sources client file" • A file defining the categories nomenclature (e.g. SNAP)
OUTPUT	• A "pemtim" emission file, with emissions referred on individual segments • A "pemspe" file containing all the substances listed in the input "client file"

Calling syntax

```
trllin      <lin_file> <pemtim_out> <pemspe_out>
            <starting_date> <starting_hour> <ending_date> <ending_hour>
            <time_res> <ascii_bin> [optional_arguments]
```

Command-line arguments

Argument	Type	Description	Format	Unit
lin_file	I	name of the "client file" containing area sources emissions	Char(256)	-
pemtim_out	O	name of the output "pemtim"	Char(256)	-
pemspe_out	O	name of the output pemspe	Char(256)	-
starting_date	I	starting day of the target simulation period	YYYY/MM/DD	-
starting_hour	I	starting hour of the target simulation period	HH/MM/SS	-
ending_date	I	ending day of the target simulation period	YYYY/MM/DD	-
ending_hour	I	ending hour of the target simulation period	HH/MM/SS	-
time_res	I	time resolution (max = 999/59/59)	HH/MM/SS	-
ascii_bin	FLAG	type of the output "pemtim" (0 = ASCII; 1 = binary)	0/1	-

Optional arguments

-nosnap

Allows to obtain a "pemtim" output file without a categories (e.g. SNAP) detail: contributions of all categories insisting on a given source are summed up

-carto <doncar_file>

doncar_file	I	geographical file containing the boundaries of all the emitting polygons	Char(256)	-

Allows referring to an external DONCAR file for polygons geometry
See § 2.5.1 for a description of the DONCAR file format.

-projection <icocib> <xdecib>

icocib	I	code of the projection system of the target mesh	Integer	-
xdecib	I	shift along x	Real	km

Allows to change the projection of the target mesh.
Possible codes of the projection system are:

0: French Lambert 2
NN > 0: UTM Zone NN in northern hemisphere
NN < 0: UTM Zone NN in southern hemisphere
99: Longitude - Latitude in decimal degrees

-csvspefile

Cause the creation of an output species file in CSV format.

-limit <xsorig> <ysorig> <nx> <ny> <dx> <dy>

xsorig	I	X coordinate of the SW point of the target grid	Real	-
ysorig	I	Y coordinate of the SW point of the target grid	Real	-
nx	I	number of point in X of the target grid	Integer	-
ny	I	number of point in X of the target grid	Integer	-
dx	I	grid cell size along X	Real	-
dy	I	grid cell size along Y	Real	-

Removes the segments which are not included in the target grid

3.1.2 GENERATION OF "PEMTIM" FILES FROM TIME SERIES OF DATA

INEMOS ***v.1.3.1***

INEMOS (INgestion of Emissions MOnitoring System data) produces a modulated "pemtim" (and the associated "pemspe") from one or multiple files containing time-varying (e.g. hourly) data coming from e.g. stacks emissions monitoring systems (EMS), on a time frame specified by the user.
Such data are emissions rates of a given set of species, optionally associated to monitored stack exit parameters: exit temperature and speed, volumetric flow (normalized or not, dry or wet), O_2, and H_2O (humidity) contents. Stacks fixed parameters (coordinates, stack height and diameter, as well as temperature and speed, if not given by the EMS) are otherwise taken from a standard "client file" for point sources, also given in input.

Input EMS files are in csv format. Columns containing monitored species and stack exit parameters can be freely assigned (either on the basis of column number or name).

A number of columns containing flags specifying missing/invalid data can be also specified (each one either on the basis of column number or name); such columns can mark:

A. overall plant status;
B. validity of individual time-varying data (emissions rates and stack exit parameters).

In all cases, list of flags to be used are user-defined.
In case A., user-defined flags can mark data records corresponding to 'non-operational' plant status (e.g. plant stopped or out of order); such records will produce zero emissions in the output "pemtim".
In case B. otherwise, the user-defined flags can mark individual invalid/missing data, which can be subsequently filled according to different methods (see below). Such data validity flags can be contained in specific columns or in the data columns themselves (e.g. when the adopted flags are not numeric and/or can be confused with valid data).

Source data can be assigned in different modes:

1. EMS species emissions rates in [kg h^{-1}], with exit speed [m s^{-1}];
2. EMS species emissions rates in [mg Nm^{-3}], with dry volumetric flow [Nm^3 h^{-1}];
3. as in 2., but with non-normalized volumetric flow [m^3 h^{-1}];
4. as in 2., but with correction of humid volumetric flow with O_2 [%] and H_2O [%];
5. as in 4., but with non-normalized volumetric flow [m^3 h^{-1}].

The produced "pemtim" contains all sources falling inside the target calculation domain, specified on command line. Species of output "pemtim" and pemspe correspond to all species specified in the input "client file": for the continuously monitored species, time-varying emissions rates are taken from EMS file(s); for all other species, time-varying emissions rates are taken from the "client file", using the time modulation pattern of a "pivot species" in the EMS file(s).

Missing data in EMS file(s) (emissions and time-varying stack parameters) can be filled according to different methods:

- substitution with zero;
- persistence (up to a user-defined time interval);
- interpolation (up to a user-defined time interval);
- use of default value in "client file".

Missing data in "client file" are filled according to category-based (e.g. SNAP) values given in an additional file ("stacks default parameters file"; see also module TRLGSP).

An initialization file (see Annex 0) contains all program options not specified in the command line: columns and flags used to control of missing/invalid data, time-varying parameters to be processed, source data assignment mode, data conversion factors, treatment mode for missing/invalid data, pivot species.

Input/output files

INPUT	• One or more files containing time-varying data from an emissions monitoring system (EMS) • A "point sources client file" containing sources constant parameters • A file defining the default values of the stack parameters, by category • A configuration file, specifying EMS files interpretation and data treatment options

OUTPUT	• A “pemtim” containing time-varying emissions for all input sources

Calling syntax

```
inemos      <ems_file_spec> <gsp_file> <gsp_def_file> <ini_file>
            <x0> <y0> <dx> <dy> <nx> <ny>
            <from_time to_time> <dt>
            <out_pemtim> <out_pemspe> <pemtim_format>
```

Command-line arguments

Argument	Type	Description	Format
ems_file_spec	I	EMS input file(s) specification; can be: -f <ems_file> -l <ems_files_list>	
ems_file	I	name of file containing EMS data	Char(256)
ems_files_list	I	name of a text file containing the list of EMS data files (one file name for each row)	Char(256)
gsp_file	I	name of the “client file” containing stacks data and other default emissions data	Char(256)
gsp_def_file	I	name of the file with category-based default stacks parameter	Char(256)
ini_file	I	name of initialization file	Char(256)
x0	I	X coordinate of the SW point of the target grid (m)	Real
y0	I	Y coordinate of the SW point of the target grid (m)	Real
dx	I	grid cell size along X of the target grid (m)	Real
dy	I	grid cell size along X of the target grid (m)	Real
nx	I	number of point in X of the target grid	Integer
ny	I	number of point in Y of the target grid	Integer
from_time	I	beginning of output time frame	DD/MM/YYYY HH:MM:SS
to_time	I	end of output time frame	DD/MM/YYYY HH:MM:SS
dt	I	output time resolution	Real
out_pemtim	O	name of output “pemtim”	Char(256)
out_pemspe	O	name of output pemspe	Char(256)
pemtim_format	FLAG	type of the output “pemtim” (0 = ASCII; 1 = binary)	0 / 1

3.1.3 DEALING WITH SPECIES

SPICE v.1.3.3

SPICE splits the emissions of species of an input "client file" (or a time-modulated "pemtim") according to an arbitrary number of "splitting schemes", writing the results in a second "client file" (or time-modulated "pemtim"). Examples of "splitting schemes" are the ones pertaining to:

- NMVOC speciation
- PM speciation
- PM subdivision in dimensional classes
- NO_x split into NO and NO_2
- SO_x split into SO_2 and sulphate

A "splitting scheme" is defined through:

- the species to be splitted
- a collection of profiles, splitting that species into a set of arbitrary target species
- a table mapping splitting profiles to individual sources of interest ("sources-based mapping table")
- a table defining the default mapping of profiles to emission categories (e.g. SNAP; "categories-based mapping table")

Splitting schemes are hierarchically applied according to a user-defined sequence (e.g. PM firstly split according to size classes, then to chemical components, ...).

Input/output files

INPUT	• A "client file" (point, line, or area sources) or a time-modulated "pemtim" and the related "pemspe" And, for each "splitting scheme": • A collection of splitting profiles • A file assigning splitting profiles to individual sources ("sources-based mapping table") • A file defining default splitting profiles for each emission category ("categories-based mapping table")
OUTPUT	• A "client file" (or time-modulated "pemtim" and related "pemspe", if the input emissions file is a "pemtim") with splitted species

Calling syntax

```
spice       <input_output_emi_specif>
            -s <splitting_scheme(1)>
            -s <splitting_scheme(2)>
            ...
            -s <splitting_scheme(n)>
```

where `<input_output_emi_specif>` is the specification of input/output emissions files:

in case of "client files":

```
<input_client_file> <output_client_file>
```

in case of "pemtims":

```
-pemtim <input_pemtim> <input_pemspe> <output_pemtim> <output_pemspe>
```

and where each `splitting_scheme(i)` is made by the following four elements:

```
<species(i)> <prof_db(i)> <src_mapping_table(i)> <categ_mapping_table(i)>
```

Command-line arguments

Argument	Type	Description	Format
input_client_file	I	name of the "client file" containing point, line or area sources emissions to be splitted	Char(256)
output_client_file	O	name of the output "client file" containing splitted emissions	Char(256)
input_pemtim	I	name of the input "pemtim" containing emissions to be splitted	Char(256)
input_pemspe	I	name of the "pemspe" associated to input "pemtim"	Char(256)
output_pemtim	O	name of the output "pemtim" containing splitted emissions	Char(256)
output_pemspe	O	name of the "pemspe" containing the list of species of output "pemtim"	Char(256)
species(i)	I	name of the input species to be splitted; in case of the first scheme, the species must be present in the input "client file"/"pemtim"; in case of subsequent schemes, the species can also be the result of the application of one of the preceding splitting schemes	Char(16)
prof_db(i)	I	name of the file containing the collection of splitting profiles ("profiles database") for splitting scheme i; the names of the output species of the splitting scheme are given by the names of the columns from 2 t n-1 in the "profiles database" file	Char(256)
src_mapping_table(i)	I	name of the file assigning splitting profiles to individual sources ("sources-based mapping table"), for splitting scheme I	Char(256)
categ_mapping_table(i)	I	name of the file defining default splitting profiles for each emission category ("categories-based mapping table"), for splitting scheme i	Char(256)

Remarks

Inside input "client files", species fields can appear in any position, marked by 'Q_' prefix. Splitting schemes are sequentially applied, according to the order given by their appearance in the command line.

An output species of a given splitting scheme can be already present in the input "client file"/"pemtim" or as the result of the application of one of the preceding splitting schemes; in such cases, all the emissions associated to the species in question are summed up.
For each record of a "splitting profiles database", the sum of all weights should ideally give 100. Anyhow, the sum can also be different if e.g. adjustments of molecular weights between input and output species are needed (this depends on the nature of the species and the related chemical mechanism). In such circumstances, when processing "profiles databases" SPICE issues a warning message on standard output.

REDSPE *v.2.0*

REDSPE allows to eliminate unneeded species in a given "pemtim", keeping only the emissions of the species desired in a specific context.
The set of the desired target species is specified through a "pemspe" file. The index of each substance can be changed in the output "pemspe" file, while the names of the substances cannot be altered. If a given target species is not present in the input "pemtim", in the output "pemtim" is added with all emissions equal to zero.
Using REDEPE is possible to homogenize the species lists present in different "pemtim" files, allowing e.g. subsequent merge.

Input/output files

INPUT	• A "pemspe" file corresponding to the input "pemtim" file • A "pemtim" file • A "pemspe" file defining the target species of interest
OUTPUT	• A "pemtim" file containing only the emissions of the target species

Calling syntax

```
redspe      <pemspe_target> <pemtim_out> <pemspe_in> <pemtim_in>
```

Command-line arguments

Argument	Type	Description	Format
pemspe_target	O	name of "pemspe" file defining the target species	Char(256)
pemtim_out	O	name of output "pemtim" file	Char(256)
pemspe_in	I	name of input "pemspe" file	Char(256)
pemtim_in	I	name of input "pemtim" file	Char(256)

3.1.4 DEALING WITH TIME

CRTEMP *v.2.0.2*

CRTEMP modulates emissions in time using yearly, weekly and daily profiles.

The database of modulation profiles is contained in a set of csv files, freely defined by the user (see *§ 2.3.1*). A daily profile can be used for all days of the week, but also a set of up to 7 daily profiles, can be associated to the days of the week in different ways (e.g. a different profile for each day of the week, or profile for Saturdays, one for Sundays and a third for all working days, end so on).
Modulation profiles can be assigned either to individual sources (identified through IDSRCE identifying code) or by emission category (e.g. SNAP). Source-specific assignments have the priority over categories-based ones (i.e. for a given source, the set of profiles of the corresponding category is used only if a specific modulation has not been defined).
The occurrence of daylight saving time can be taken into account, by shifting in time the daily profiles that have been specified to be subject to that regime.
Input emissions are contained in a time-independent "pemtim" file, while output emissions are written either in a time-dependent "pemtim" or in a "time dependent client file" (see *§ 2.3.2*).
The smallest resolution of output file is 1 hour.

Input/output files

INPUT	• A "pemtim" file with non-modulated emissions • A "pemspe" file corresponding to the "pemtim" file • A file defining the reference nomenclature of categories (e.g. SNAP) • A file assigning time profiles to individual sources • A file assigning time profiles to categories • A file defining the yearly modulation profiles • A file defining the weekly modulation profiles • A file defining the daily time modulation profiles
OUTPUT	• A "pemtim" file with modulated emissions

Calling syntax

```
crtemp      <pemtim_in> <pemspe_in> <file_type> <out_file>
            [optional_arguments]
```

Command-line arguments

Argument	Type	Description	Format
pemtim_in	I	name of non-modulated "pemtim" file	Char(256)
pemspe_in	I	name of "pemspe" emission file	Char(256)
file_type	FLAG	option for creation of output pemtim (1 = creation of modulated pemtim; 2 = ARIA Impact files)	Integer
out_file	O	name of file containing modulated emissions	Char(256)

Optional arguments

-projection <icocib> <xdecib>

icocib	I	code of the projection system of the target mesh	Integer	-

xdecib	I	shift along x	Real	km

Allows changing the projection of the target mesh

-modulation <mod_src> <mod_cat> <yearly_prof> <weekly_prof> <hourly_prof> <classif> <reference_day>

mod_src	I	file containing the association between individual sources and (yearly, weekly, daily) modulation profiles	Char(256)	-
mod_cat	I	file containing the association between categories and (yearly, weekly, daily) modulation profiles	Char(256)	-
yearly_prof	I	file of the yearly profiles	Char(256)	-
weekly_prof	I	file of the weekly profiles	Char(256)	-
hourly_prof	I	file of the daily profiles	Char(256)	-
classif	I	file containing the reference nomenclature (classification of categories)	Char(256)	-
reference_day	I	reference day for the daily modulations (first day of the week corresponding to the day 1 in the profiles)	Char(256)	monday / tuesday / … / sunday

-dates <from_time> <to_time>

from_time	I	beginning of modulation time interval	YYYY/MM/DD HH	-
to_time	I	end of modulation time interval	YYYY/MM/DD HH	-

Time modulation is applied between these two time stamps.

-dst <sum_ref_day> <nb_sum> <summer_month> <win_ref_day> <nb_win> <winter_month> <decdst>

sum_ref_day	I	day of the week when DST takes place (winter-to-summer transition)	Char(256)	monday / tuesday / … / sunday
nb_sum	I	number of the week in the month	Integer	1…5
summer_month	I	number of the month	Integer	1…12
win_ref_day	I	day of the week when DST end (summer-to-winter transition)	Char(256)	monday / tuesday / … / sunday
nb_win	I	number of the week in the month	Integer	1…5
winter_month	I	number of the month	Integer	1…12
decdst	I	time shift when DST takes place		+/-HH:MM

Allows taking into account the summer/winter change of hour.

Remark
UNCLASSE: a source is considered non-classified if all SNAP levels are equal to -999 or 0.

CATPER

CATPER allow to merging "pemtim" files containing emissions from different time periods (e.g. hours) into a single, modulated "pemtim". All input "pemtims" must be referred to the same set of sources.

Input/output files

INPUT	• The "pemtim" files to be merged, corresponding to different time periods of the day • A "pemspe" file corresponding to "pemtim" files
OUTPUT	• A time modulated "pemtim" file

Calling syntax

```
catper      <pemspe_in> <pemtim_out> <iverif> <pemtims_in(1…n)>
```

Command-line arguments

Argument	Type	Description	Format
pemspe_in	I	name of "pemspe" file	Char(256)
pemtim_out	O	name of time modulated "pemtim" emission file	Char(256)
Iverif	FLAG	control flag *(NO = 0 ; YES = 1)*	0/1
pemtims _in (1…n)	I	names of "pemtim" files to be merged, in chronological order	Char(256)

Remark
To save time, it's recommended to set the control FLAG to 0.

MODGRID *v.2.2*

MODGRID combines time modulation of emissions (as done in CRTEMP) with generation of binary input files for FARM (as done in PEM2GRID): from one or more "pemtim" files it creates either a point sources file (FARM binary format, a.k.a. "pointemi") or a diffuse sources file (a.k.a. "diffemi", either in ADSO/bin or netCDF/COARDS format).
Module details are given in section 3.1.5.

3.1.5 GRID ALLOCATION

LIGREC v.2.0

LIGREC projects line sources on a target regular grid. The emissions associated to input segments are allocated on the corresponding intersecting grid cells according to the length of the intersection. The output file contains emissions referred to rectangular cells. Input and output emissions files are in "pemtim" format.

Input/output files

INPUT	• A "pemtim" file with emissions in linear (vector) format • A "pemspe" file corresponding to the "pemtim"file • A file defining the SNAP nomenclature
OUTPUT	• A "pemtim" file with emissions in raster format

Calling syntax

```
ligrec      <pemspe_in> <pemtim_lin> <ref_snap> <xorig> <yorig> <nx> <ny>
            <dx> <dy> <pemtim_rec> <snapmode>
```

Command-line arguments

Argument	Type	Description	Format
pemspe_in	I	name of "pemspe" file corresponding to the "pemtim"	Char(256)
pemtim_lin	I	name of the input "pemtim" containing line sources	Char(256)
ref_snap	I	file containing the reference nomenclature (classification of categories)	Char(256)
xsorig	I	X coordinate of the SW point of the target grid (km)	Real
ysorig	I	X coordinate of the SW point of the target grid (km)	Real
nx	I	number of points along X of the target grid	Integer
ny	I	number of points along Y of the target grid	Integer
dx	I	target grid resolution along X (m)	Real
dy	I	target grid resolution along Y (m)	Real
pemtim_rec	O	name of output raster "pemtim" file	Char(256)

Optional arguments

-projection <icogeo> <xdecal> <icocib> <xdecib>

icogeo	I	system of the initial projection	Integer	-
xdecal	I	shift in km in the initial system	Real	-
icocib	I	system of the projection for the target mesh	Integer	-
xdecib	I	shift in km for the target mesh	Real	-

Allows changing the projection of the target mesh.

-nosnap

Allows obtaining a "pemtim" output file without categories detail (on each grid cell, contributions of all categories are summed up).

-keepinfo

Keeps the information about the source polygon.

PEM2GRID *v.1.4*

PEM2GRID put the emissions from rectangular sources contained in one or more "pemtim" files (either modulated in time or not) into an archive of gridded fields (ADSO/bin or netCDF format). By default, output fields are 2D (ie. without differentiating the sources according to their height); optionally, sources can be allocated to 3D gridded fields, using either source-specific heights or default heights by category specified in an additional file.

Input/output files

INPUT	• One or more "pemtim" files containing (time-varying) emissions • The related "pemspe" file • A file containing the orography of the domain (REL format)
OUTPUT	• An ADSO/bin or netCDF archive of gridded emissions

Calling syntax

```
pem2grid    <input_emi> <pemspe> <orog_file> <idproj>
            <output_arch> <output_format>
            [-3d <src_param_file> <nlevs_user> <z1 ... zn>]
```

where `input_emi` can be either:

```
-f <filename>
```

or:

```
-l <listfile>
```

Command-line arguments

Argument	Type	Description	Format
filename	I	name of "pemtim" file containing emissions from all sources of interest	Char(256)
listfile	I	name of the file defining the list of "pemtims" containing the emissions from all sources to be considered; the list file is an ASCII file with one file name (including full path) per row	Char(256)
orog_file	I	name of the file containing the orography of the domain (REL format)	Char(256)

idproj	I	projection code: 1 to 60 = UTM; 99 = geographic (lat/lon)	Integer
output_arch	I	name of output archive	Char(256)
output_format	I	output archive format: 0 = ADSO/bin, 1 = netCDF (COARDS)	Integer

Optional arguments

-3d <src_param_file> <nlevs_user> <z1 z2 ... zn>

src_param_file	I	name of file containing vertical parameters by category (see Annex A.4 for format)	Char(256)	-
nlevs_user	I	number of output vertical levels	Integer	-
z1 z2 … zn	I	blank-delimited list of levels heights (grid cells midpoints)	Real	m above orog.

Allows to allocate emissions on a 3D output mesh (if not specified, emissions are by default allocated as 2D matrices, i.e. at "surface").
The mass emitted from a given sources is split among the vertical levels of the output grid according to the intersections of the emission layer with the grid cells. The emission layer extension is taken directly from the "pemtim" when present, otherwise the categories-based thicknesses are used, as specified by the user in the "stacks default parameters file" in terms of a min and max emission heights (see A.3 for file format).
In the assignment by category, a "generalization criterium" is applied: when a given emission in the input "pemtim" refers to a category that is not present in the file containing vertical parameters by category, then the first category available at higher levels of the classification tree is used (e.g. if an emission in the input "pemtim" is associated to category 3.4.2, and such category is not present in the vertical parameters file, then the assignment made for category 3.4 is used; if also category 3.4 is not present in the vertical parameters file, profiles associated to category 3 are then used). When no match can be established also at higher levels of the classification tree, then the overall default (specified for the conventional category "0;0;0;0;0") is used.

Remarks
The following assumptions are made:

- the parameters of the horizontal output grid are taken form orography file
- if the output archive already exists, its grid is checked against the one of the orography given in input; if the match fails, the output archive is overwritten
- when the output archive already exists:
  - ADSO/bin: it is overwritten with data records from input "pemtims"
  - netCDF: it is updated with data records from input "pemtims" (only if grids are matching, otherwise output archive is overwritten)
- all input "pemtims" must be referred to the same set of species
- all input "pemtims" are supposed to be time modulated, with a hourly interval, and must start at the same time; the exception are "pemtims" defined over a single time interval (either 1 hour, or longer)

MODGRID *v.2.2*

MODGRID combines time modulation of emissions (as done in CRTEMP) with generation of binary input files for FARM (as done in PEM2GRID), creating either a point sources file or a gridded diffuse sources file from one or more input "pemtims".
Output point sources file are in FARM binary format (a.k.a. "pointemi"), while output gridded diffuse sources file (a.k.a. "diffemi") are either in ADSO/bin or netCDF/COARDS format.
By default, gridded output files are 2D, ie. without differentiating the sources according to their height. Alternatively, sources can be allocated to 3D gridded fields, using either source-specific heights or default heights assigned by category. 3D gridded fields can also include contributions from point sources, with allocation to grid cells taking into account plume rise; in such case, hourly meteorological fields for the period of interest must be also given in input.
For a given MODGRID run, if either FARM "pointemi" or 2D gridded output is selected, all input files must contain sources of the same geometry (point or rectangular, respectively). Otherwise, sources of different geometry (point or rectangular) can be processed in the same MODGRID run when a 3D gridded output is selected.

Input/output files

INPUT	• One or more "pemtim" files containing non-modulated emissions • The related "pemspe" files • A file defining the reference nomenclature of categories (e.g. SNAP) • A file assigning time profiles to individual sources • A file assigning time profiles to categories • A file defining the yearly modulation profiles • A file defining the weekly modulation profiles • A file defining the daily time modulation profiles • (only in case of gridded output) a file containing the orography of the domain (REL format) • (only in case of gridded 3D output containing also point sources) one or multiple files containing fields of meteorological variables
OUTPUT	• An ADSO/bin or netCDF/COARDS archive of 2D/3D gridded emissions or • A point sources file in FARM binary format

Calling syntax

```
modgrid  <input_emi> <pemspe>
         -modulation <ref_snap> <src_prof> <snap_prof>
         <yearly_prof> <weekly_prof> <daily_prof> <ref_dow>
         -dates <from_time> <to_time>
         -dst <sum_ref_day> <nb_sum> <summer_month>
              <win_ref_day> <nb_win> <winter_month> <dst_shift>
         <out_specs>
```

where:

```
<input_emi>  input pemtim file(s) specification; it can be:
             -f <filename>   to process a single file
             -l <listfile>   to process a set of files, where
                             'listfile' contains the files list
<pemspe>          input pemspe file

<ref_snap>         reference SNAP classification
<src_prof>         sources-modulation profiles correspondence table
<snap_prof>        SNAP-modulation profiles correspondence table
<yearly_prof>      yearly modulation profiles
<weekly_prof>      weekly modulation profiles
<daily_prof>       day-by-day modulation profiles
<ref_dow>          reference day of week for daily modulations
                   (first day of the week corresponding to the day 1
                   in the profiles): "monday"/"tuesday"/ ... /"sunday"

<from_time>        beginning of target time interval (YYYY/MM/DD HH)
<to_time>          end of target time interval (YYYY/MM/DD HH)

<sum_ref_day>      day of the week when DST takes place (winter-to-summer
                   transition): "monday"/"tuesday"/ ... /"sunday"
<nb_sum>           number of the week in the month
<summer_month>     number of the month
<win_ref_day>      day of the week when DST ends (summer-to-winter
                   transition: "monday"/"tuesday"/ ... /"sunday"
<nb_win>           number of the week in the month
<winter_month>     number of the month
<dst_shift>        time shift when DST takes place (+/-HH:MM)
```

and where where <out_specs> can be either (in case of diffuse sources):

```
-gridding <orog_file> <idproj>
<output_arch> <output_format>
[-3d <src_param_file> <nlevs_user> <z1 ... zn>]
[-minlevs]
[-meteo <wind_file> <temp_file> <tcc_file>]
```

or (in case of point sources):

```
-pointemi <idproj> <output_arch>
```

with:

```
<orog_file>        input orography file
<idproj>           projection ID (UTM zone, in case of UTM;
                   99 if lat/lon)
<output_arch>      name of output archive
<output_format>    output archive format: 0 = ADSO/bin, 1 = netCDF

[-3d ..... ]       (optional) 3D output; default is 2D
<src_param_file>   name of file containing vertical ranges by category
<nlevs_user>       number of output vertical levels
<z1 ... zn>        blank-delimited list of levels heights [m above orog.]
[-minlevs]         (optional) automatically adopt the min # of levels
                   allowing to represent all sources heights

[-3d ..... ]       (optional) 3D output; default is 2D
<src_param_file>   name of file containing vertical ranges by category
```

Command-line arguments

Argument	Type	Description	Format
filename	I	name of "pemtim" file containing emissions from all sources of interest	Char(256)
listfile	I	name of the file defining the list of "pemtims" containing the emissions from all sources to be considered; the list file is an ASCII file with one file name (including full path) per row	Char(256)
Time modulation			
ref_snap	I	file containing the reference nomenclature (classification of categories)	Char(256)
src_prof	I	file containing the association between individual sources and (yearly, weekly, daily) modulation profiles	Char(256)
snap_prof	I	file containing the association between categories and (yearly, weekly, daily) modulation profiles	Char(256)
yearly_prof	I	file of the yearly profiles	Char(256)
weekly_prof	I	file of the weekly profiles	Char(256)
dayly_prof	I	file of the daily profiles	Char(256)
ref_dow	I	reference day for the daily modulations (first day of the week corresponding to the day 1 in the profiles) : "monday"/"tuesday"/ ... /"sunday"	Char(256)
from_time	I	beginning of modulation time interval	YYYY/MM/DD HH
to_time	I	end of modulation time interval	YYYY/MM/DD HH
sum_ref_day	I	day of the week when DST takes place (winter-to-summer transition): "monday"/"tuesday"/ ... /"sunday"	Char(256)
nb_sum	I	number of the week in the month: 1…5	Integer
summer_month	I	number of the month: 1…12	Integer
win_ref_day	I	day of the week when DST end (summer-to-winter transition): "monday"/"tuesday"/ ... /"sunday"	Char(256)
nb_win	I	number of the week in the month: 1…5	Integer
winter_month	I	number of the month: 1…12	Integer
dst_shift	I	time shift when DST takes place	+/-HH:MM
Diffemi/pointemi output			
orog_file	I	name of the file containing the orography of the domain (REL format)	Char(256)
idproj	I	projection code: 1 to 60 = UTM; 99 = geographic (lat/lon)	Integer
output_arch	I	name of output archive (gridded or FARM binary "pointemi")	Char(256)

output_format	I	output gridded archive format: 0 = ADSO/bin, 1 = netCDF/COARDS	Integer

Optional arguments

-3d <src_param_file> <nlevs_user> <z1 z2 ... zn> [-minlevs]

src_param_file	I	name of file containing vertical parameters by category (see Annex A.4 for format)	Char(256)	-
nlevs_user	I	number of output vertical levels	Integer	-
z1 z2 … zn	I	blank-delimited list of levels heights (grid cells midpoints)	Real	m above orog.
-minlevs	I	(optional flag) automatically adopt the min number of vertical levels (a subset of the `z1, z2 … zn` specified) allowing to include all emissions sources heights listed in `src_param_files`	Char	-

The "-3d" clause allows to allocate emissions on a 3D output mesh (if not specified, emissions are by default allocated as 2D matrices, i.e. at "surface"). The emissions vertical distributions are categories-based, and are specified by the user in the "stacks default parameters file" (see A.4 for file format); for each category, they can be defined either as a single layer (through a min and max height), or more freely through a full arbitrary vertical profile. In both cases, the mass emitted from a given source is split among the vertical levels of the target grid according to the intersections of the each specified emission layer with the target grid cells.
In the vertical assignment of emissions by category, a "generalization criterium" is applied, as explained with PEM2GRID.

-meteo <wind_file> <temp_file> <tcc_file>

wind_file	I	name of file containing 3D fields of horizontal components of the wind; the variables must be named "U" and "V" and its values given in [m s^{-1}]	Char(256)	-
temp_file	I	name of file containing 3D fields of air temperature; the variable must be named "T" and its values given in [K]	Char(256)	-
tcc_file	I	name of file containing 2D fields of total cloud cover; the variable must be named "TCC" and its values given either in tenths (recognised as [0..10] or [tenths]) or octals (recognised as [0..8] or [octals])	Char(256)	-

The "-meteo" clause allows to allocate emissions from point sources on the 3D output mesh taking into account hour-by-hour plume rise. The following assumptions are made:

- input varables can be freely arranged in one or multiple files (e.g. all variables in a unique file, or each variable in a different file, etc.)
- input meteo file(s) can be either in ADSO/bin or netCDF(COARDS/CF) format;

- input meteo file(s) must be defined on the grid chosen for output gridded emissions, ie. on the same horizontal mesh implied by the input `<orog_file>` and the same vertical levels assigned through the "-3d" clause;
- for a given MODGRID run, all input meteorological variables must be available over the whole time interval between `<from_time>` and `<to_time>`.

At each time step, the effective equilibrium height of each plume (h_e) due to thermal and inertial rise, is computed according to (Briggs, 1975), taking also into account partial penetration into an elevated inversion and stack-tip downwash correction, starting from the actual values of the meteorological parameters (wind, temperature and stability) at the grid points of the affected column:

$$h_e = h_s + \Delta H$$

where h_s is the stack height; heights of plume bottom (h_{pb}) and top (h_{pt}) are then computed as:

$$h_{pb} = h_e - 0.6 \cdot \Delta H$$
$$h_{pt} = h_e + 0.6 \cdot \Delta H$$

and the emitted mass is injected into the vertical array of the affected grid cells, partitioned according to the intersections of the plume layer with cells.

Remarks

The following assumptions are made:

- in case of gridded output, the parameters of the hrizontal output grid are taken form the orography file
- if the output gridded archive already exists, its grid is checked against the one of the orography given in input; if the match fails, the output archive is overwritten
- when the gridded output archive already exists:
  - ADSO/bin: it is overwritten with data records from input "pemtims"
  - netCDF: it is updated with data records from input "pemtims" (only if grids are matching, otherwise output archive is overwritten)
- all input "pemtims" must be referred to the same set of species
- all input "pemtims" are supposed to be time modulated, with a hourly interval, and must start at the same time; the exception are "pemtims" defined over a single time interval (either 1 hour, or longer)
- all sources in input "pemtims" with a geometry not compatible with the chosen type of output are discarded (ie. only point sources are kept if the output is a FARM binary "pointemi", while only rectangualr sources are kept when the output is a gridded archive)

BINRAS

BINRAS allows to translate a "pemtim" into an ADSO/bin grid-based file, which can further be used for graphical display with SAVI3D by using BINMERAF.

Input/output files

INPUT	• A "pemtim" file

	• A “pemspe” file corresponding to the “pemtim”file • An orography file
OUTPUT	• An ADSO/bin gridded file

Calling syntax

```
binras      <fic”pemtim”> <ficpemspe> <ficmail> <idconcat> <kb> <plafoh>
            <igrids> <zsurf> <ficecrbin> <typ_index> <fclist> <check_input>
            <code_order> <fic_code> <hour_min_sec>
```

Command-line arguments

Argument	Type	Description	Format
fic”pemtim”	I	name of “pemtim” file to project	Char(256)
ficpemspe	I	name of “pemspe” file	Char(256)
ficmail	I	name of topographical file	Char(256)
idconcat	FLAG	generation of one file by SNAP code or one global file *(one by SNAP code = 0 ; one global = 1)*	0/1
kb	I	number of vertical layers	Integer
plafoh	I	top (height) of domain (m)	Real
igrids	I	type of vertical levels (from 1 to 7 depending on formulation)	Integer
zbsurf	I	height of surface layer (from 0 to 40m)	Real
ficecrbin	O	root name of output files (without extension)	Char(256)
typ_index	I	type of index for vertical layers *(MINERVE = 1; HERMES = 2)*	Integer
ficlist	O	name of file in the output file list	Char(256)
check_input	FLAG	control flag *no = 0 ; yes = 1)*	0/1
code_order	FLAG	generation of a file listing the SNAP codes *(no = 0 ; yes = 1)*	0/1
fic_code	O	name of file containing SNAP classification	Char(256)
hour_min_sec	I	time resolution in the binary file	HH/MM/SS

Remarks

The different formulations possible for the generation of vertical layers are as follows (IGRIDS flag):

1. Nickerson
2. Quadratic
3. Sinus
4. Logarithmic
5. Double Sinus
6. Identity – constant levels
7. Geometric

The top (height) of the domain (PLAFOH) should be at least 4 times the height of the max altitude in the topography file.

FIC_CODE should only be documented when CODE_ORDER = 1.

The time resolution (RESOLUTION) is an optional argument. If there is no value specified, the time resolution of the output file will be the same as in the "pemtim" file.

3.1.6 OTHER MODULES

CATSOU v.2.0

CATSOU merge all the sources contained in a set of input "pemtim" files into a single output "pemtim". The order of sources in the output file corresponds to the order of the files in the input list.

Input/output files

INPUT	• The "pemtim" files related to different source types, that should be merged A "pemspe" file corresponding to the "pemtim"file
OUTPUT	• A "pemtim" traffic file

Calling syntax

```
catsou      <pemspe_out> <pemtim_out> <iverif> <pemspe_in(1)> <pemtim_in(1)>
            <pemspe_in(2)> <pemtim_in(2)> … <pemspe_in(n)> <pemtim_in(n)>
```

Command-line arguments

Argument	Type	Description	Format
pemspe_out	O	name of output "pemspe" file	Char(50)
pemtim_out	O	name of output "pemtim" file	Char(50)
iverif	FLAG	control flag *(NO = 0 ; YES= 1)*	0/1
pemspe_in(1)	I	name of input "pemspe" file #1	Char(50)
pemtim_in(1)	I	name of input "pemtim" file #1	Char(50)
pemspe_in(2)	I	name of input "pemspe" file #2	Char(50)
pemtim_in(2)	I	name of input "pemtim" file #2	Char(50)
pemspe_in(n)	I	name of input "pemspe" file #n	Char(50)
pemtim_in(n)	I	name of input "pemtim" file #n	Char(50)

Remarks

It is recommended to set the control FLAG to 0 in order to save time.

The number of input "pemtim" files that can be merged is unlimited.

Optional arguments

-update

Source IDs will be updated according to file number.

PEMSOM *v.2.0*

PEMSOM allows to remove the detail of individual activities in a given "pemtim". The result is another "pemtim" containing for each source (i.e. each geometric entity) the total emissions coming from all the activities insisting on it.

Input/output files

INPUT	• A "pemtim" file with emissions detailed by category • A "pemspe" file corresponding to the "pemtim" file
OUTPUT	• A "pemtim" file containing total emissions for each source

Calling syntax

```
pemsom      <pemtim_in> <pemspe_in> <pemtim_out>
```

Command-line arguments

Argument	Type	Description	Format
pemtim_in	I	name of "pemtim" file containing emissions detailed by category	Char(256)
pemspe_in	I	name of "pemspe" file	Char(256)
pemtim_out	O	name of "pemtim" file containing total emissions for each source	Char(256)

BILSOU

BILSOU creates a text file containing the assessment of the total emissions by substance and divided by SNAP classification.

Input/output files

INPUT	• A "pemtim" file • A "pemspe" file corresponding to the "pemtim" file • A file defining the SNAP nomenclature
OUTPUT	• A file containing a summary of the emissions in the "pemtim" file

Calling syntax

Command-line arguments

Argument	Type	Description	Format
ficspe	I	Name of "pemspe" file	Char(256)
fictim	I	Name of "pemtim" file	Char(256)
ficncl	I	Name of file with SNAP profiles	Char(256)
ficbil	O	Name of file output text file	Char(256)

PEMMOD

PEMMOD allows to convert a "pemtim" file from ASCII to binary format and vice-versa (binary "pemtims" are recommended when handling large files, since they are processed by EMGR modules much faster than their ASCII equivalents).

Input/output files

INPUT	• A "pemtim" file • A "pemspe" file corresponding to the "pemtim" file
OUTPUT	• The converted "pemtim" file

Calling syntax

```
pemmod      <pemtim_in> <pemspe_in> <bintim_out> <pemtim_out> <version>
            <pemspe_new> <binspe_out> <pemspe_out>
```

Command-line arguments

Argument	Type	Description	Format
pemtim_in	I	name of "pemtim" file to convert	Char(256)
pemspe_in	I	name of "pemspe" file	Char(256)
bintim_out	FLAG	Format of output file *(ASCII = 0 ; BIN = 1)*	0/1
pemtim_out	O	name of output "pemtim" file	Char(256)
version	FLAG	version number of output "pemtim" *(No modification = 0 ; Other ≥ 800)*	Real
pemspe_new	FLAG	generation of a new "pemspe" file *(NO = 0 ; YES = 1)*	0/1
binspe_out	FLAG	format of output "pemspe" file *(ASCII = 0 ; BIN = 1)*	0/1
pemspe_out	O	name of output "pemspe" file	Char(256)

POLMOD

POLMOD allows to alter emissions in a given "pemtim" by species and category. For each species-category combination, emissions can be modified using a linear transformation, of type: "output = A * input + B". Modifications can be applied either to masses/time interval (e.g. using only B with emissions in μg m^{-3}) or to masses (e.g. using only A with emissions in μg).

Input/output files

INPUT	• A "pemtim" file • A "pemspe" file corresponding to the "pemtim" file • A file containing the transformation parameters by category and pollutant
OUTPUT	• A "pemtim" file with modified emissions • A file containing a summary of the modifications that have made based on categories (e.g. SNAP)

Calling syntax

```
polmod      <pemtim_in> <pemspe_in> <snap_mod_file> <cat_mod> <typ_modul>
            <pemtim_out> <mod_log_file>
```

Command-line arguments

Argument	Type	Description	Format
pemtim_in	I	name of "pemtim" to modify	Char(256)
pemspe_in	I	name of "pemspe" file	Char(256)
snap_mod_file	I	name of file containing the list of A and B factors to be applied to each species-category combination (format as described in Annex *A.6*)	Char(256)
cat_mod	FLAG	flag for handling classification levels sub-categories treatment mode (0 = no action; 1 = treat sub-levels -999 as 0)	0/1
typ_modul	FLAG	type of modulation *(modulation applies to emission rate = 1* *modulation applies to mass emitted = 2)*	1/2
pemtim_out	O	name of modulated "pemtim"	Char(256)
mod_log_file	O	name of output file containing the balance of emissions modifications	Char(256)

Remarks

Each SNAP code (containing 5 classification levels "X X X X X") can only be associated once to a particular pollutant.

The universal SNAP code "0 0 0 0 0" allows to modulate a pollutant whatever the real SNAP classification might be. This code must however appear in the last position of the file FICSNAPMOD and the associated pollutants shouldn't be associated to a real SNAP classification.

The emissions in the "pemtim" file are supplied in the form of mass (e.g. µg) for a certain time period (e.g.1 or 24 hours) from which it's possible to calculate the emission rate (e.g. µg/hour…). TYP_MODUL controls the type of modulation that should be used, either by applying the factors A and B on the emission rate (a constant emission can for example be added) or on the mass (e.g. division of total emissions by factor A for a particular SNAP category).

PEM2MIF *v.1.5.2*

PEM2MIF converts a "pemtim" into MIF+MID (MapInfo export format). A separate layer is generated for each type of source geometry (point, line and area), for easier GIS visualization and processing of sources emission data.

Input/output files

INPUT	• A "pemtim" file • The corresponding "pemspe" file
OUTPUT	• A MIF+MID for point sources (if such sources exist in input "pemtim") • A MIF+MID for line sources (same as above) • A MIF+MID for area sources (same as above)

Calling syntax

```
pem2mif     <pemtim_file> <pemspe_file> <base_mif_name> <ex_date> <ex_time>
```

Command-line arguments

Argument	Type	Description	Format
pemtim_file	I	name of input "pemtim" file	char(128)
pemspe_file	I	name of input pemspe file	char(128)
base_mif_name	I	base name of output mif+mid files	char(128)
ex_date	I	extraction date	DD/MM/YYYY
ex_time	I	extraction time	HH:MM:SS

Remarks

If ex_date = "00/00/0000" and ex_time = "00:00:00", total emissions over the whole period are extracted for each source; otherwise, only emissions for the specified date and time are extracted.

In all cases output units correspond to units of the input "pemtim" (e.g. mass in [µg]) over the extraction time period.

PEMSTAT *v.1.4*

PEMSTAT returns some statistics about the sources contained in a given pemtim:

- sources number and geometry, with an indication of their order inside the file;

- time references;
- stacks parameters and total emissions by species, in case of point sources;
- hourly time series of total emitted mass by species;
- a summary of total emitted mass by species and category (e.g. SNAP).

The hourly emitted mass data can be either the totals for all the sources contained in the input pemtim, or can be referred to a single source of interest.

Input/output files

INPUT	• A "pemtim" file • The corresponding "pemspe" file
OUTPUT	• An ASCII file containing pemtim statistics

Calling syntax

```
pemstat     <pemtim_file> <pemspe_file> <stat_file_out> [option]
```

Command-line arguments

Argument	Type	Description	Format
pemtim_file	I	name of input "pemtim" file	char(128)
pemspe_file	I	name of input pemspe file	char(128)
stat_file_out	O	name of the output statistics file (an ASCII text file)	char(128)
option	I	it can be: '-t': to be used if sources in input pemtim are referred to different time intervals '-s': output of stack parameters and total emissions for each point source '-i' <num>: retrieve only data related to source number 'num' '-n' <name>: retrieve only data related to source named 'name'	

3.1.7 PRE-PROCESSING MODULES

FCSPLIT ***v.1.3.3***

FCSPLIT allows to split point sources and area sources "client files" according to emission category or administrative unit.
Split by category can be performed on the basis of a given classification level: a different client file will be produced for each combination of codes currently present in the input file at the selected classification level.

Split by administrative unit (available only for area files) can be performed in different ways: on the basis of IDADM1 or IDSURF (a different client file will be produced for each IDADM1 / IDSURF currently present in the input file), or user-defined 'administrative sets', specified in an additional file (see Appendix; in such case, a different client file will be produced for each user-defined administrative set).
Splitting client files allows easier manipulation of emissions data in subsequent pemtims.

Input/output files

INPUT	• A "client file" to be splitted
OUTPUT	• A series of "client files" containing emissions splitted

Calling syntax

```
fcsplit     <input file> <output_path>
            [nclasslev_match | 'idadm1' | 'idsurf'] [adm_sets_file]
```

Command-line arguments

Argument	Type	Description	Format
input file	I	name of the input client file	Char(128)
output_path	O	root name for output client files	Char(128)
nclasslev_match	I	(optional) number of classification levels to be used to split in multiple client files	Integer
'idadm1'	I	(optional, available only for area sources files) when indicated, client file is splitted on the basis of idadm1	(-)
'idsurf'	I	(optional, available only for area sources files) when indicated client file is splitted on the basis of idsurf, using the administrative sets specified in 'adm_sets_file'	(-)
adm_sets_file	I	(optional; must be specified only if 'idsurf' has been selected) administrative sets to be used to split input area sources file	Char(50)

Administrative sets file

It is an ASCII file providing the specification of the administrative sets according to which the emissions records inside an area sources "client file" have to be subdivided, and the subset of emission categories to which this operation has to be performed.
An example of such file is given below, with in-line comments. Records are free-format. Comment lines begin with an exclamation mark ("!") in the first column.

```
!=========================================================================
! Administrative sets split file
!=========================================================================!
!=========================================================================
! List of SNAP categories to which the split
! into administrative sets has to be applied
!=========================================================================
  12   ! # of SNAP categories
```

```
!
! List of SNAP categories codes
! Each record: codes of levels 1 to 5
!-------------------------------------------------------------------------
  7    1    3    0    0
  7    1    4    0    0
  7    1    5    0    0
  7    1    6    0    0
  7    2    3    0    0
  7    2    4    0    0
  7    2    5    0    0
  7    2    6    0    0
  7    3    3    0    0
  7    3    4    0    0
  7    3    5    0    0
  7    3    6    0    0
!
!=========================================================================
! Administrative sets definition
!=========================================================================
  2    ! # of administrative sets
!
!-------------------------------------------------------------------------
! Sets codes (to be used as categories codes in the output client files)
! and labels (to be used to build names of output client files)
!
! Each record: set #, set code, set label
!-------------------------------------------------------------------------
  1    50   'set1'
  2    51   'set2'
!
!-------------------------------------------------------------------------
! Label to be used in the name of the file
! containing the unsplitted records
!-------------------------------------------------------------------------
'others'
!
!-------------------------------------------------------------------------
! Administrative sets composition
!
! For each record: IDSURF, set code
!-------------------------------------------------------------------------
   12009      50
   12019      50
   12020      50
   12021      50
   12099      50
   98050      50
   98051      50
   98059      50
   98060      50
   12075      51
   12109      51
   15011      51
   15075      51
   15154      51
   15234      51
   15239      51
   15241      51
   15242      51
   97042      51
   98031      51
```

MIF2LIN *v.1.1.1*

MIF2LIN allows to split into individual segments the line emissions associated to polylines, producing a segment-based "client file" for line sources.
Input data are stored in a MIF+MID files couple (MapInfo export format): the MIF file describes the polylines ("multiple polylines" are also possible), while the corresponding MID is a "line sources client file" (with the only exception of being comma delimited) containing associated emissions. MIF2LIN breaks all polylines in elementary arcs (segments), splitting emissions according to arcs length.
In output client files, segments have source id codes (IDSRCE) obtained by shifting the IDSRCE of the originating polyline by a user-defined number of digits and adding a progressive number along the polyline.

Input/output files

INPUT	• A "client file" to be splitted
OUTPUT	• A series of "client files" containing emissions splitted

Calling syntax

```
mif2lin     <base_mif_name> <lin_file> [id_factor]
```

Command-line arguments

Argument	Type	Description	Format
base_mif_name	I	base name of intput MIF/MID files	Char(128)
lin_file	O	name of output line sources client file	Char(128)
id_factor	I	(optional) code factor to be applied to source id (default = 1000)	Integer

3.2 MODEL-SPECIFIC MODULES

ARQRAS/ARQRASLITE

ARQRAS/ARQRASLITE are emission pre-processors for CHIMERE model.

Based on a "pemtim" file containing all type of sources, ARQRAS generates one output file per SNAP code. These files serve as input for the emission pre-processor of the CHIMERE model.

Arqras-lite version (only to be used for area and traffic "pemtim" files)
ARQRAS also exist in a LITE version. This LITE version is used for speeding up the treatment of very large traffic and/or area source "pemtim" files (>100 Mb). It can also be used for the same reason when the number of grid points of the CHIMERE domain is very large. However, it should be observed that the LITE version doesn't account for point sources why two separate "pemtims" distinguishing the point sources from the rest of the sources first needs to be prepared.

In this case ARQRAS is executed for the “pemtim” containing the point sources while ARQRAS-LITE is executed for the “pemtim” containing the other sources (area sources, traffic…).

Nevertheless, if the traffic and/or area source “pemtim” file is not very large or the study domain is rather small, the easiest is to only use the ARQRAS routine with a “pemtim” file containing all the sources.

Input/output files

INPUT	• A “pemtim” file • A “pemspe” file corresponding to the “pemtim”file • A topographical file corresponding to the CHIMERE domain
OUTPUT	• CHIMERE input file

Calling syntax

An example running script is presented in Appendix B.

Command-line arguments

Argument	Type	Description	Format
ficpemtim	I	Name of “pemtim” file	Char(256)
ficpemspe	I	Name of “pemspe” file	Char(256)
ficmail	I	Name of topographical file	Char(256)
dirarq	O	Directory where the output files will be stored	Char(256)
connectiv	FLAG	Creation of a connection table for use of none regular grid (NO = 0; YES = 1)	0
ficlist	O	Name of file listing the output files	Char(256)
check_input	FLAG	Control Flag *(NO = 0 ; YES = 1)*	0/1
code_order	FLAG	Generation of a file with listing the SNAP classifications *(NO = 0 ; YES = 1)*	0/1
fic_code	O	Name of file listing the SNAP classifications	Char(256)

Remark

The ”CONNECTIV” flag is an old option associated to the generation of emission inventories over non-regular grids. This option has been replaced by nesting in the CHIMERE model. CONNECTIV should therefore always be set to “0”.

CALEMIS

CALEMIS is an emission pre-processor for CHIMERE model, generating of emission input files for the MODEL model based on ARQRAS data.

Remark: the substance nomenclature used here is prepared by the IER (Institut für Energiewirtschaft und Rationelle Energieanwendung - Universität Stuttgart) and IS NOT public. It can only be used for CHIMERE applications according to the ADEME-IER convention. Please contact ADEME (Agence française de l'Environnement et de la Maitrise de l'Energie) for further details and requests.

Input/output files

INPUT	• IER-CATEGS: the SNAP categories of the IER speciation • SPECIATION: the IER speciation • AGGREGATION: the lumping matrix for CHIMERE • WEEKEND: weekend modulations for traffic • VOC-TABLE-IER: a table of mass and reactivity for VOC
OUTPUT	• EMISSIONS-<domain>.<month>

Calling syntax

```
calemis <number of month>
```

Name and link of final emission files

CHIMERE uses ASCII files named "EMISSIONS-<domain>.<number of month>". The output file of CALEMIS has the name "EMISSIONS" which should be renamed to CHIMERE standard. If the output file from CALEMIS is not renamed or copied before running the program one more time, the previous "EMISSIONS" file will be deleted. One must pay attention to the 1h-shift due to summertime, do not use April data for March emissions e.g.

First stage: Create a correspondance file ("CORRESP")

1. Edit the make_corresp shell and name the directory where the ARQRAS files (srf & spt files) are located. Do not forget to first remove all biogenic emission files, since they are calculated by CHIMERE directly. The biogenic emissions correspond to SNAP code 11 and certain SNAP code 10 (managed forestry for instance). Normally it's sufficient to do this for only one month, since the files for the other months have the same structure.
2. Run the make_corresp shell. It will build a "CORRESP" file which must be corrected manually. The "CORRESP" file should contain:
    - 1st column: the codes of all the SNAP code emission encountered in the data base
    - 2nd column: the corresponding IER speciation profile. If no obvious corresponding profile has been found, the text string "-----" is written. The user should replace this text string with a profile number of her/his choice.
    - 3rd column: A flag for weekend modulation of traffic. For SNAP codes corresponding to traffic and for which WE modulation is available, the flag should be set to 2. Otherwise the flag should be set to 1.
    - 4th column: A flag for methane. Should be set to 1 if methane is accounted for in the VOC profiles, otherwise 0.

- 5th column: A flag describing the order of pollutants in the ARQRAS file. This flag is used when reading data with the program "calemis.f". The flag should by default be set to 1.

The "CORRESP" file must be checked manually and carefully before launching the script "calemis".

Second stage: Edit and modify calemis and calemis.f

1. Edit calemis.f, and change NXX, NYY, the dimensions of the original grid. If a subgrid is to be extracted as output, then change NXSTART, NXEND, NYSTART, NYEND.
2. CHANGE THE UNIT FACTOR, so that the output is in molec/cm2/s. This factor may depend on the grid size and the original emission units. 36.e6 is the factor when the original emissions (of ARQRAS) are given in Ton/h/km2.
3. The lines below the "Reading data" comment should be verified, in particular the flag "iorde" (5th column), defining in which order the pollutants of the ARQRAS file should be read. The flag ITYPE tells whether the source is a point or surface source.

Remark: *In this version, methane emissions are set to 0*

BOTTARGA *v.4.1*

BOTTARGA is an emission pre-processor for FARM model, performing the lumping of NMVOC speciation profiles for SAPRC mechanisms. Lumped profiles are calculated from a database of profiles of elementary hydrocarbons, the specification of the aggregation of elementary hydrocarbons in lumped species, and representative case-specific emissions (e.g. domain total emissions).

Input/output files

INPUT	• Emissions from all sources to be considered, either in the form of a couple of "pemtim" + "pemspe" files, or in a set of "client files" (point, line, area sources) • A file containing the database of VOC speciation profiles (SAROAD classification) • A file assigning speciation profiles to emissions categories (e.g. SNAP) • A lumping control file (.LPC) • A file containing detailed model species list (.DMS) • A file containing the assignment of DMS to SAROAD classes (.PRM)
OUTPUT	• A file containing the VOC speciation profiles for the selected target chemical mechanism + auxiliary files

Calling syntax

```
bottarga    <input_emi> <proffile> <snap2prof>
            <lpcfile> <dmsfile> <dms2sarfile>
            <VOC_name> <NROG_name> <CH4_flag> <out_pref>
```

where `input_emi` can be either:

```
<pemtim_in> <pemspe_in>
```

or:

```
-cf <cf_list>
```

Command-line arguments

Argument	Type	Description	Format
pemtim_in	I	name of "pemtim" file containing emissions from all sources	Char(200)
pemspe_in	I	name of "pemspe" file corresponding to input "pemtim"	Char(200)
cf_list	I	name of the file defining the list of "client files" (point, line, area sources) containing the emissions from all sources to be considered; the list file is an ASCII file with one file name (including full path) per row	Char(200)
proffile	I	name of data file containing profiles weights (SAROAD classification)	Char(200)
snap2prof	I	name of the file assigning speciation profiles to categories (e.g. SNAP)	Char(200)
lpcfile	I	name of the lumping control file (.LPC)	Char(200)
dmsfile	I	name of the file containing detailed model species list (.DMS)	Char(200)
dms2sarfile	I	name of the file containing the assignment of DMS to SAROAD classes (.PRM)	Char(200)
VOC_name	I	name of the species in input file(s) corresponding to VOC	Char(8)
NROG_name	I	name of the species in input file(s) corresponding to NROG	Char(4)
CH4_flag	FLAG	treatment of methane: .false. = normalize speciation profiles excluding methane (ROG = NMVOC) .true. = VOCs include methane (ROG = VOC)	.true. / .false.
out_pref	O	prefix for files created in output	Char(200)

MPEM2FARM *v.3.3*

MPEM2FARM is an emission pre-processor for FARM model, converting one or more "pemtims" into FARM input files: a point sources file (FARM binary format) and a diffuse sources file (ADSO/bin or netCDF (COARDS) format). For a given run, all input files must contain sources of the same geometry (either point or rectangular).

Input/output files

INPUT	• One or more "pemtim" files containing (time-varying) emissions • The related "pemspe" file • A file containing the orography of the domain (REL format)
OUTPUT	• An input binary file for FARM, containing either point or gridded emissions

Calling syntax

```
mpem2farm   <source_type> <utmzone> <list_file> <pemspe> <relief>
            <farm_file> <out_arch_type>
```

Command-line arguments

Argument	Type	Description	Format
source_type	I	type of sources to be processed (1 = point; 2 = diffuse)	1 / 2
utmzone	I	UTM zone	Integer
list_file	I	name of file containing the list of "pemtim" files to be converted	Char(256)
pemspe	I	name of "pemspe" file	Char(256)
relief	I	name of orography file (REL format)	Char(256)
farm_file	O	name of output emission file for FARM	Char(256)
out_arch_type	O	format of FARM diffuse emission file (0 = ADSObin; 1 = NetCDF (COARDS))	0 / 1

4 EMMA SHELL COMMANDS

EMMA (EMission MAker), is a set of shell commands and procedures helping the automated creation of the emission input for air quality models using Emission Manager (EMGR) modules.

The **version** described here is **6.0**.

4.1 SYSTEM DESCRIPTION AND ARCHITECTURE

4.1.1 INTRODUCTION

As stated before, Emission Manager is a modular system, where the computational modules described in section 3 are the building blocks. The output of each module can be used as input of another module, allowing to set up different calculation chains performing a broad spectrum of emissions manipulation operations. In complex cases otherwise, setting up the system for a given area can be a time consuming operation, as it is also keeping track of all changes made while tuning or adjusting input data during setup. EMMA addresses these needs, allowing a coordinated use of the **Emission Manager modules**, according to a set of **pre-arranged sequences** that meet the most typical needs of a set of models. The processing sequences also optimize the computational effort, avoiding the repetition of unneeded calculation steps when only part of input data are changed.

Target air quality models currently considered are SPRAY Lagrangian particle model, and FARM Eulerian chemical-trasport model. The two codes have their specific requirements about input emissions:

- SPRAY needs hourly emissions for an arbitrary number of (non-reactive) species; emissions can be related to point, line and area (rectangular) sources;
- FARM needs hourly emissions for the set of species considered by the adopted chemical mechanism, typically with speciated hydrocarbons (VOC) and PM; emissions can be related to point and diffuse (gridded) sources.

Starting **input data** are the emissions either from inventories (e.g. on yearly basis) or from time-varying monitoring/modelling systems (as stacks emissions monitoring systems or hourly traffic emissions systems) and the related "ancillary" data used for disaggregation operations (as cartography, spatial proxies on the target grid, time and speciation profiles, etc.).

Within EMMA, data files are organized according to "cases". A "**case**" corresponds to a tree of directories containing the set of input and output files related to a given geographic domain and scenario. For a given *case*, the typical procedure to prepare a model-ready emission input can be outlined as follows:

- the *case* directories tree is firstly built according to EMMA conventions;
- users' emission data related to different geographic entities are organized in sets of "client files", and placed in the corresponding directories of the *case*;

- data for space, time and species disaggregation are also organized and stored in the designated directories;
- a configuration file is prepared, listing input files and calculation options chosen for the case;
- once the input dataset of the *case* is complete, model-ready emission input can be obtained, issuing EMMA commands that drive coordinated sequences of calls to EMGR modules.

EMMA commands are a combination of shell scripts and makefiles, so a "**make**" utility and "**bash shell**" must be available on the machine in use (both are usually available under all Unix systems, and *cygwin* can be used under MS Windows). Following "make" paradigm, EMMA final output (input emission file(s) for the chosen model) and all the intermediate files are considered as **targets**. Each target corresponds to a logical stage in the calculations flow and can be obtained by applying a given EMGR module (with related options and parameters) to a set *pre-conditions*. Pre-conditions are either input data files prepared by the user, or the intermediate targets produced by previous processing steps. The capability of "make" utility to keep track of the **files timestamps** is also exploited in the calculations workflow: when an input data file is changed by the user, all the intermediate or final targets potentially affected by that change are re-generated, while the other ones remain unchanged. In this way, the processing sequence is minimized, and individual modules are called only where needed (i.e. only where the user has changed some of the input data affecting the targets).

EMMA shell commands are organized to perform either:

- **step-by-step calculations**, when the input data are progressively added and the intermediate or final results are checked until a satisfactory configuration is reached, or
- **comprehensive model input preparation**, when input dataset is consistent and has reached the desired configuration and the system is used in "production mode".

Organization of input data in a *case* is described in section 4.1.2, while the syntax of individual shell commands is explained in the reference section 4.2.

Changes from previous versions

Version 5.0 (2008) with respect to Version 4.0 (2006) implies a major revision:

- use of new/updated **EMGR modules** (reduction of the number of computational modules and of model-specific modules)
- simplified and **unified commands**: at top level, one for each target model (the correspondences between old and new commands are mapped in Figure **Errore. L'origine riferimento non è stata trovata.**)
- re-organized **operations order**
- enhanced **error tracking**
- simplified **time management**
- added **(multiple) networks** of line sources
- no more use of TMPDIR and EXEDIR**: multiple instances** of EMGR can be run at the same time
- better "system" vs. "case" data separation

- optional generation of FARM input file for diffuse sources by assembling a set of "typical days of the week" (for each month), specified according to a user-specified scheme
- compatibility with unix-like pathnames used by later versions of "make" utility under cygwin

Version 6.0 (August 2013) with respect to Version 5.0 (2008):

- optimized generation of time-dependent binary input for FARM: combined modulation and conversion in FARM input binary format of multiple pemtims, bypassing .mdl pemtims;
- possibility of generating also 3D gridded input for FARM, with grid allocation of area/line sources and optionally point sources with plume rise;
- possible use of 'dummy' doncar in subcases;
- use of new/updated EMGR modules.

Version 6.1 (this manual) with respect to Version 6.0 (August 2013):

- integration of EMS emissions in SPRAY input.

Installation

The system components reside in EMMA subdirectory of the "`ADSO`" models tree (Figure 1):

- EMMA system commands are placed in "EMMA" main directory and in its "mak", "msg" and "utils" subdirectories; normally, they are not modified by the user;
- EMGR computational modules are typically located in "`bin`" subdirectory (if needed, the default location can be changed modifying the EMMA configuration file `emgr_install.mak`, explained later);
- the files containing the database of VOC speciation profiles and the descriptions of lumping schemes according to SAPRC mechanisms, are placed in subdirectory "`SAPRC`" (needed only when the target model is FARM with a SAPRC mechanism).

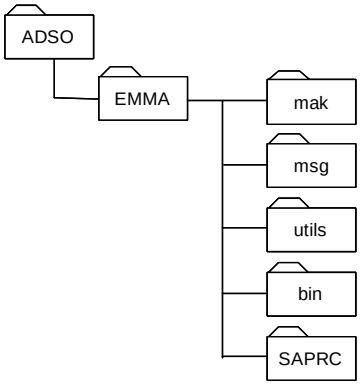


Figure 1. EMMA / Emission Manager system directories tree.

System configuration: emma_config.mak

This file, placed in "EMMA/mak" system directory, contains the specifications related to the installation of the modules of the Emission Manager and EMMA itself on the computer. As a rule, it is part of the default installation and **should not be changed** by the user. Anyway, a commented example of this file is given below for reference; comment lines begin with "#".

```
# EMMA/EMGR installation settings
# Arianet 2001-13

# --- EMGR modules

SPICE = spice.exe
TRLGSP = trlgsp.exe
TRLLIN = trllin.exe
TRLCRS = trlcrs.exe
INEMOS = inemos.exe
CRTEMP = crtemp.exe
REDSPE = redspe.exe
CATSOU = catsou.exe
PEMSOM = pemsom.exe
LIGREC = ligrec.exe
SELREC = selrec.exe
BOTTARGA = bottarga.exe
MPEM2FARM = mpem2farm.exe
PEM2GRID = pem2grid.exe
MODGRID = modgrid.exe
ARSETTIME = arsettime.exe
ARCONV = arconv.exe

# -- case subdirs names

POINTSUBDIR = point
LINESUBDIR = line
AREASUBDIR = area
EMSSUBDIR = ems
HTRAFSUBDIR = htraffic
NOMSUBDIR = tables
CARSUBDIR = cartog
LDUSUBDIR = cartog
SPECSUBDIR = spec

SPRAYSUBDIR = SPRAY
FARMSUBDIR = FARM
IMPACTSUBDIR  = IMPACT
CHIMERESUBDIR = CHIMERE

# --- BOTTARGA NMVOC files

NMVOC_PROFILES_DB = PROFWTwithnames.csv
SAROAD_SPEC_ASSIGNMENT = SAROAD_NEW.PRM

# --- other fixed-names system files

CURTIME = curtime.dat

PHASECOMFILE = current_command.tmp
PHASELOGFILE = current_phase.tmp
ERRSTEPS = error_steps.log
GLOBALERRLOG = error.log

# --- output files
```

```
# merged pemtims prefix

MERGED = all

# output files for SPRAY

SPRAYPTM_NAME = all.ptm
SPRAYPSP_NAME = all.psp

# output files for FARM

CF_FILELIST = list.txt

POINTEMI_FILELIST = filelist.pointemi.txt
DIFFEMI_FILELIST = filelist.diffemi.txt

FARMPOINTEMI_NAME = pointemi.dat
FARMDIFFEMI_NAME = diffemi.bin

# output files for IMPACT

IMPACTPOINT_NAME = imptpoint.gsp
IMPACTLINE_NAME = imptline.lin
IMPACTAREA_NAME = imptarea.srf

# output files for CHIMERE

CHIMEREPSP_NAME = pointchimere.psp
CHIMEREPOINT_NAME = pointchimere.ptm
CHIMEREAREA_NAME = area.ptm
CHIMERELINE_NAME = line.ptm

# --- reference parameters

# time interval for client files data ingestion

DSTART_INGEST = 99/01/25
HSTART_INGEST = 00/00/00
DSTOP_INGEST = 99/01/25
HSTOP_INGEST = 24/00/00
DELTAT_INGEST = 24/00/00

# reference day of the week in weekly modulation profiles
REF_DAY_OF_WEEK = monday

# date of a reference day (e.g. a monday, according of what chosen above)
REF_MONDAY = 99/01/25

# DST parameters:

# (DST-1) day-of-the-week for winter -> summer transition ("monday" | ... |
"sunday")
SUM_REF_DAY = sunday

# (DST-2) number, inside the month, of SUM_REF_DAY (1..5; where 5 is the last
SUM_REF_DAY)
NBSUM = 5

# (DST-3) month of winter -> summer transition (1 = January .. 12 = December)
SUMMER_MONTH = 3

# (DST-4) day-of-the-week for summer -> winter transition ("monday" | ... |
"sunday")
```

```
WIN_REF_DAY = sunday

# (DST-5) number, inside the month, of WIN_REF_DAY (1..5; where 5 is the last
WIN_REF_DAY)
NBWIN = 5

# (DST-6) month of summer -> winter transition (1 = January .. 12 = December)
WINTER_MONTH = 10

# (DST-7) DST time shift, in winter time ("+/-HH:MM")
DECDST = +01:00

# output time resolution for EMS data
DTSEC = 3600
```

4.1.2 "CASE" STRUCTURE

As mentioned before, within EMMA data files are organized according to "cases". A *case* corresponds to a tree of directories containing the set of input and output files related to a given geographic domain and scenario, and completed with ancillary information about the target grid, time behavior, species splitting and target model.

Input files and directories structure

User-prepared **input data files** are (Figures 2 and 3; refer to section 1.2 and Annex A for files formats description):

A. a categories file containing the reference classification scheme adopted for emitting activities (e.g. the SNAP nomenclature by the EEA, or any other one set of categories);
B. a set of "client files", containing inventory (e.g. yearly) emission data associated to point, line and area sources;
C. cartographic files in vector format ("DONCAR"), describing the geometry of sources, when is complex (area sources polygons and networks of line sources);
D. a file specifying default parameters for point sources, by category;
E. a file ("PEMSPE") specifying the subset of species used by the target model;
F. time modulation data:
    1. a set of files containing hourly, daily and monthly modulation factors;
    2. a file specifying the modulation factors to be applied to each category;
    3. a file (optionally) specifying the modulation factors to be applied to selected individual sources;

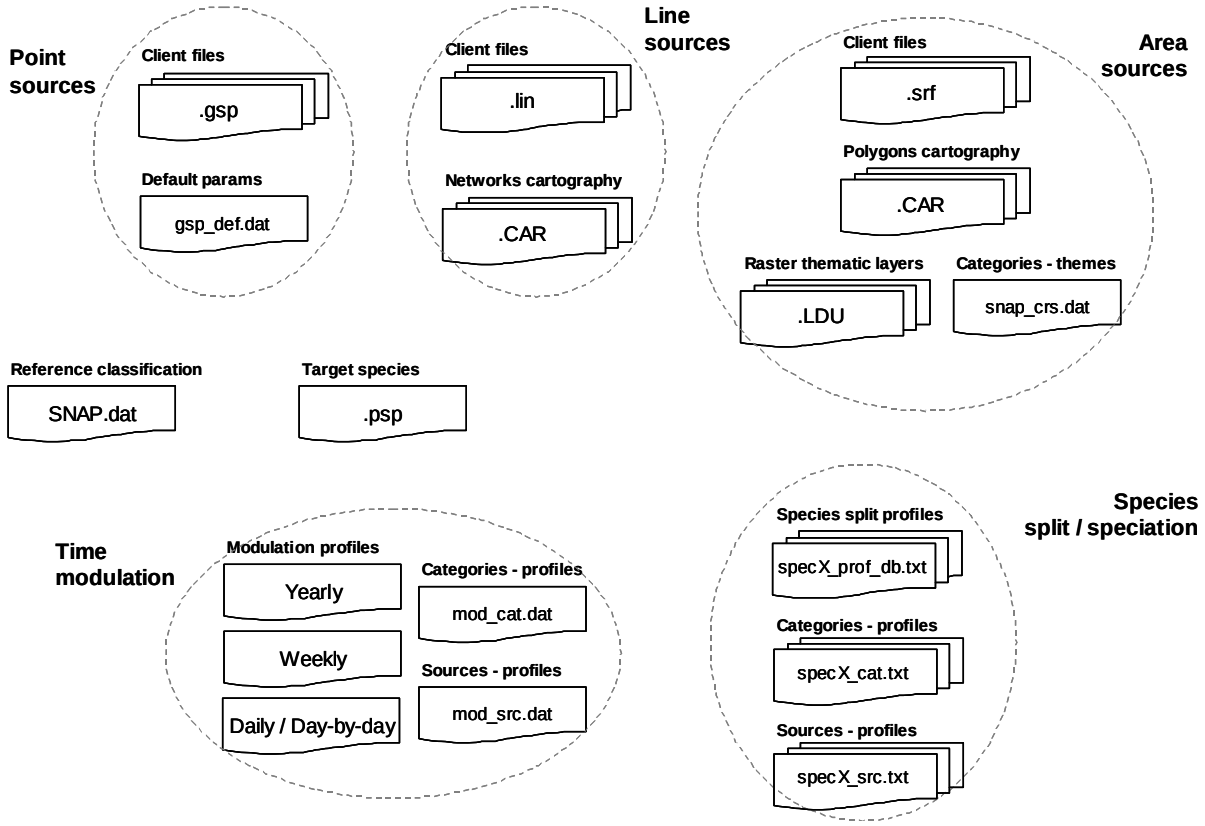


Figure 2. Emission Manager database managed by EMMA.

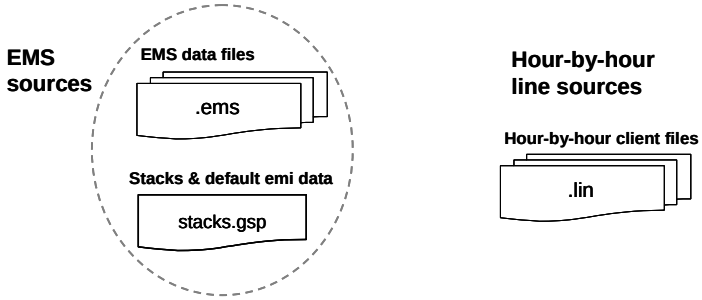


Figure 3. Emission Manager database managed by EMMA – Time-varying sources.

G. space disaggregation data for area sources:
   1. a file containing thematic layers (e.g. land-use) in raster format ("LDU");
   2. a file specifying the space disaggregation criterium to be applied to each category;

H. species split / speciation data, according to an arbitrary number of "schemes" (the rules to be applied to split a given aggregated chemical species in more detailed ones); for each scheme:
   1. a database of splitting/speciation profiles;
   2. a file specifying the profile to be applied to each category;
   3. an optional file specifying the profile to be applied to selected individual sources;

I. (only when FARM is the target model) an orography file in "REL" format.

J. when data from emissions monitoring systems are used, also:

1. EMS files containing time-varying emissions and stacks data;
2. a point sources "client file" containing stacks data and other default emissions data.

K. when hourly line sources are used, also hour-by-hour "client files" containing emissions data.

Data files for a given *case* are organized according to the **directory tree** of Figure 4.

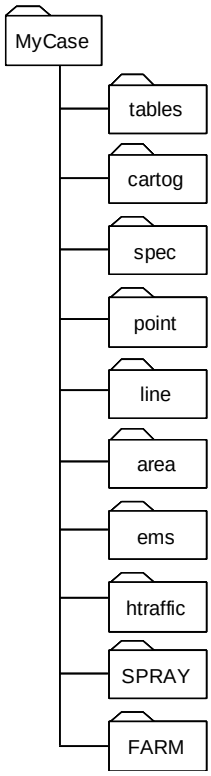


Figure 4. Standard directory tree for an EMMA "case".

The *case* root directory (indicated in figure with "MyCase") can be placed anywhere on the available disks / file systems, according to users' needs; subdirectories of the *case* main directory, will contain:

- "`tables`": the files of sets A., D., E., F., G.2 and H;
- "`point`", "`line`" and "`area`": "client files", according to the corresponding source geometry;
- "`ems`": files related to emissions monitoring systems (set J);
- "`htraffic`": hour-by-hour line sources "client files" (set K);
- "`cartog`": files of sets C., G.1 and I;
- "`spec`": target files created during the lumping of NMVOC profiles for FARM;
- "`SPRAY`" and "`FARM`": model-specific output files generated during all the processing phases.

For a given source geometry, emission data can be freely organized in a single "client file" (all area emissions from all categories) or in **multiple "client files"**, related to *subsets of categories* (e.g. one file containing all emissions related to urban heating, another file containing all emissions related to solvent use, etc.), according to the user specific needs.

Client files can have arbitrary names but must have standard extensions: "`.gsp`", "`.lin`" or "`.srf`" for point, line or area sources, respectively. The usage of **standard files extensions** is

also adopted throughout all the rest of the processing chain, to identify emissions data files referred to different calculation stages and keep track of the correct dependencies among them. So each type of target has its own extension, according to the following table.

Standard file extensions used in EMMA.

gsp	input client file for point sources
lin	input client file for line sources
srf	input client file for area sources
spc	intermediate csv file (client file format) containing speciated emissions
ptm0	intermediate file (pemtim format) containing ingested client file data, with spatial disaggregation of input geometry (polylines broken into segments and polygons projected on grid cells)
rec	intermediate file (pemtim format) containing line sources projected on grid cells
psp	intermediate file (pemspe format) containing the list of species of ptm0 file with the same name
red	intermediate file (pemtim format) containing only emission data for the target species of interest
mdl	intermediate file (pemtim format) containing emission data modulated at hourly level
nosnap	intermediate file (pemtim format) where the detail of emitting categories has been suppressed
merged	intermediate file (pemtim format) containing emission data coming from multiple input pemtims

Subcases and networks

In case of complex datasets, client files containing emissions data for area and line sources can be partitioned in logical subsets.

Emissions from area sources can be organized according to "**subcases**", each one corresponding to different administrative units of intermediate level (e.g. regions, provinces, districts, etc.), different inventories or to subsets of categories. In such circumstances, the single subdirectory "`area`" , containing the "client files" of the regular case, is substituted by the directory tree of Figure 5: each *subcase* corresponds to a different subdirectory (in Figure 5 indicated as "region1", "region2", etc., but names are free), containing in turn a subdirectory named "`area`"; the "client files" of a given region are then stored in the corresponding "`area`" subdirectory. All client files of a given *subcase* can then have associated their own set of ancillary input data, e.g. a specific cartographic file, describing the geometry of the related administrative unit, but also other specific information that may differ from the other *subcases*. These possibilities are discussed in the paragraph explaining *subcases* configuration.

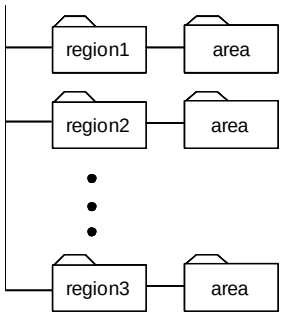


Figure 5. Directory tree for area sources in case of multiple "subcases".

Similarly, emissions from line sources can be organized according to "**networks**", corresponding to e.g. different transportation modes, different road types, geographic area. In such circumstances, the directory tree showed in Figure 6 substitutes the subdirectory "`line`" of the regular case without multiple networks (Figure 4): as for *subcases*, also each network corresponds to a different subdirectory (in Figure 6 indicated as "network1", "network2", etc., but names are free), containing in turn a subdirectory named "`line`"; the "client files" of a given network are then stored in the corresponding "`line`" subdirectory. As in the cases of area sources subsets (*subcases*), then also client files of a given network can have associated their own set of ancillary input data, e.g. a specific cartographic file, describing the geometry of the related network of polylines. The specification of networks is also explained in detail in the following.

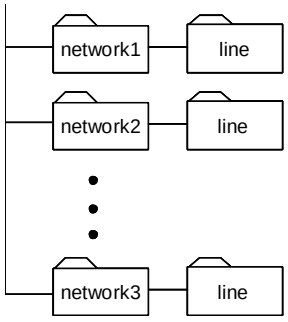


Figure 6. Directory tree for line sources in case of multiple "networks".

4.1.3 CONFIGURATION FILES

Configuration files allow to describe to EMMA the structure of a given *case*, including input files and main calculation options. Their syntax is explained in the following paragraphs.

"Case" configuration: case.mak

This file, placed in the case root directory, contains the specifications related to the current *case*: sources types, input emissions files, their (optional) partition in subsets, and all other input files needed to perform species, space and time disaggregations.
Specifications are given through statements in the form "keyword = value(s)", arranged in lines placed in free order. Keywords names and predefined values are case-sensitive, while filenames

are free (except for the extensions of input client files, that must follow the "`.gsp`", "`.lin`" or "`.srf`" convention). Comment lines begin with "`#`", and can be placed anywhere in the file.

An example of this file is given below, while keywords are explained in the tables immediately following.

```
# EMMA case settings - Ver. 6.1

# name

CASENAME=Test

# sources types

SOURCES_TYPES = point ems line area

# subcases dir names (if present)

SUBCASES = region1 region2
NETWORKS = main_roads secondary_roads

# --- input files

# (optional) user-specified subsets of client files to be processed

#SELECTED_GSP_NAMES = one.gsp two.gsp
#SELECTED_EMS_NAMES = stack1.ems stack2.ems
#SELECTED_LIN_NAMES = one.lin
#SELECTED_SRF_NAMES = one.srf two.srf

# reference nomenclature

NOMFILE = snap.txt

# species split / speciation

N_SPEC_SPLIT = 3
SPEC_SPLIT_SCHEME_1 = NOX nox_db.txt nox_src.txt nox_cat.txt
SPEC_SPLIT_SCHEME_2 = SO2 SOX_split_prof.txt dummy overall_split_cat.txt
SPEC_SPLIT_SCHEME_3 = NMVOC NMVOC_saprc90_prof.txt dummy NMVOC_spec_cat.txt

# target species (after split & speciation)

TARGET_SPEC = test.psp

# target grid

X0 = 360
Y0 = 4954.
NX = 81
NY = 81
DXY = 1

# point & area sources default params file,
# and action for point sources (check / default)

GSPDEF = gspdef.txt
SRC_PARAM = default

# EMS data

EMS_INI=inemos.mycase.ini
EMS_STACKS=stacks.gsp
```

```
# cartography & land-use

CARFILE = city.car
LDUFILE = landuse.ldu
SNAP_CRS_FILE = snap_crs.txt
#NETFILE = roadnet.car

# time modulation

MOD_SRC = mod_src.txt
MOD_CAT = mod_cat.txt
MOD_YEAR = yearly_mod.txt
MOD_DAY = weekly_mod.txt
MOD_HOUR = daily_mod.txt

# pemtims file format: 0 = ASCII; 1 = binary

FILE_FORMAT = 0

# logfiles saving option: 0 = overwrite; 1 = keep individual logfiles

KEEPLOGS = 1

# --- FARM options and files ---

# projection code/UTM zone

UTMZONE = 32

# orography

RELFILE = test.rel

# 3D gridded output

OUT3D = 1
ZPARAMS = gspdef.txt
NZ = 5
ZLEVS = 10 25 50 100 200
MINLEVS = 1

# weekly modulation scheme

REPEATING_WEEKS = 1
WEEKLY_SCHEME = 1 1 1 1 1 6 7

# combined modulation and generation of FARM binary input files

FASTMOD = 1

# meteo data for point sources plume rise

GSP_PLUME_RISE = 1
WIND_FILE = /mnt/data/meteo/meteo.nc
TEMPER_FILE = /mnt/data/meteo/meteo.nc
TCC_FILE = /mnt/data/meteo/meteo.nc

# diffuse emissions file format: 0 = ADSO/bin; 1 = netCDF

DIFFEMI_FORMAT = 1

# NMVOC profiles lumping
```

```
GASMECH = saprc99
SPEC_CAT = NMVOC_spec_cat.txt

# --- SPRAY options ---

# optional sources treatment

#LINE_TREATMENT = gridded
KEEP_SINGLE_ACTIV = true
KEEP_MERGED_ACTIV = true

# vertical params for line and area sources

#TRLLIN_DAT = trllin.test.dat
#TRLCRS_DAT = trlcrs.test.dat
#LIGREC_DAT = ligrec.test.dat
```

General keywords

Variable	Meaning
CASENAME	label identifying the case
SOURCES_TYPES	choice of sources types to be processed: can be any combination of "point", "ems", "line" and "area" keywords, as a blank-delimited list (e.g. "point area", "line area", or, in the most complete case: "point ems line area")
SUBCASES	(only if area sources data are subdivided in subsets) blank-delimited list of *subcases*; each element of the list must correspond to the name of a subdirectory of the case root directory: that subdirectory will hold the corresponding subcase.mak configuration file and an area subdirectory where the related "client files" will be placed; if area sources client files are not partitioned in subsets, the keyword must be absent (in such case, client files will then be placed in area subdirectory of the case root directory)
NETWORKS	(only if line sources data are subdivided in subsets) blank-delimited list of networks; each element of the list must correspond to the name of a subdirectory of the case root directory: that subdirectory will hold the corresponding network.mak configuration file and a line subdirectory where the related "client files" will be placed; if area sources client files are not partitioned in subsets, the keyword must be absent (in such case, client files will then be placed in line subdirectory of the case root directory)
SELECTED_GSP_NAMES, SELECTED_EMS_NAMES, SELECTED_LIN_NAMES, SELECTED_SRF_NAMES	(optional) blank-delimited lists of "client files" (point, line, area) or EMS input files to be processed; if left blank or absent, all "client files" placed in the corresponding directories matching the standard extensions (".gsp", ".lin", ".srf", ".ems") will be processed
NOMFILE	file containing the reference nomenclature of categories (e.g. SNAP or other)
N_SPEC_SPLIT	number of species splitting/speciation "schemes"; can be 0 if species in

	input client files are to be kept as they are
SPEC_SPLIT_SCHEME_x	specification of species splitting/speciation "schemes", where x goes from 1 to N_SPEC_SPLIT; each splitting scheme is assigned through the following four elements: (case-sensitive) name of the input species to be splitted (e.g. "NOX") name of the file containing the database of splitting/speciation profiles, expressed in terms of target species name of the file optionally mapping splitting/speciation profiles to individual sources of interest; if only categories-based mapping is used, it can be set to "dummy" name of the file mapping splitting/speciation profiles to categories splitting/speciation "schemes" will be applied following the order they have been declared, with x going progressively from 1 to N_SPEC_SPLIT
TARGET_SPEC	name of "PEMSPE" file specifying the subset of species used by the target model (after applying all species splitting/speciation schemes)
X0, Y0	coordinates of the SW corner of the target grid (km)
NX, NY	# of point of the target grid in each directions
DXY	cells size of the target grid (km)
GSPDEF	file containing default parameters for point sources, by category
SRC_PARAM	"check", to perform only a check of stacks parameters against default parameters specified in GSPDEF file "default", to use stacks default parameters specified in GSPDEF file to actually fill missed data or adjust the values outside the valid range
EMS_INI	initialization file for INEMOS module
EMS_STACKS	file containing EMS stacks data and other default emissions data (used by for INEMOS module)
CARFILE	cartographic file ("DONCAR" format) containing all polygons to which area sources are referred; when set to "dummy", area sources are associated to rectangles, whose coordinates are directly specified inside the records of the input "client file" when *subcases* are used, each of them may use a different cartographic file, specified through a CARFILE statement in the corresponding subcase.mak file; also CARFILE in subcase.mak can be set to "dummy"
LDUFILE	file containing gridded cartographic layers ("LDU" format) to be used for spatial disaggregation of area sources
SNAP_CRS_FILE	file specifying the space disaggregation criterium (ie. the gridded cartographic layer) to be applied to each individual category
NETFILE	cartographic file ("doncar" format) containing all polylines to which line sources are referred;

	when multiple networks are used, each of them may use a different cartographic file, specified through a NETFILE statement in the corresponding network.mak file
MOD_SRC	name of file mapping optionally modulation profiles to applied to individual sources of interest
MOD_CAT	name of file specifying the modulation factors to be applied to each individual category
YEARLY_PROF, WEEKLY_PROF, DAILY_PROF	(former variables names: MOD_YEAR, MOD_DAY, MOD_HOUR – now deprecated) names of files containing databases of yearly, weekly and daily modulation profiles
FILE_FORMAT	"pemtims" format flag: 0 = all "pemtim" files will be in ASCII format 1 = all "pemtim" files will be in binary format
KEEPLOGS	flag controlling storage of logfiles generated by multiple executions of a given EMGR module over multiple files: 0 = logfiles will be overwritten, and named r_xxxxxx.log, where xxxxxx is the name of the EMGR module 1 = all individual logfiles will be kept, and named yyyyyy.zzz.log, where yyyyyy.zzz is the name of the corresponding generated output file

FARM-specific keywords: emissions input generation

UTMZONE	1 to 60 = UTM zone (e.g. "32"); 99 = geographic coordinates (lat/lon); NOTE: coordinates in all input files (ie. client files, cartographic files containing polygons or gridded layers, orography) must be in the same projection.
RELFILE	name of the orography file ("REL" format), to be put in output diffuse emissions file
OUT3D	flag controlling the number of spatial dimensions of the output diffuse emissions file: 0 = 2D file (ZPARAMS, NZ end ZLEVS are not needed) 1 = 3D file (ZPARAMS, NZ end ZLEVS need to be specified)
ZPARAMS	name of file containing sources heights parameters, by category
NZ	number of vertical levels of the output diffuse emissions file
ZLEVS	blank-delimited list of the vertical level heights
MINLEVS	flag controlling the minimization of the number of output vertical levels: 1 = automatically adopts the min number of vertical levels (a subset of the specified ZLEVS) that allows to include all emissions sources having the heights listed in the ZPARAMS file 0 = uses the NZ vertical levels specified through ZLEVS
REPEATING_WEEKS	flag controlling whether in building the output diffuse emissions file each day has to be generated independently or each week is assembled using a

	set of "typical days of the week" (for each month) according to a specified scheme: 0 = each day of the diffuse emissions file has to be generated independently 1 = diffuse emissions file is assembled from a set of "typical days" (for each month) according to a specified scheme; this option speeds-up the calculations, but has to be selected considering the modulation profiles actually employed and the result to be obtained
WEEKLY_SCHEME	scheme of the days of the week to be adopted in the generation of output diffuse emissions file: a series of 7 numbers, each indicating the "typical day" to be used (1 = Monday, 2 = Tuesday, … 7 = Sunday) examples: "1 1 1 1 1 6 7" means that within each week of for a given month there will be 3 types of days: all days from Monday to Friday (treated as Monday), Saturdays and Sundays "1 2 3 4 5 6 7" means that within each week of for a given month there each day will be different from the others in all cases, all weeks within a month will repeat according to the specified scheme
GSP_PLUME_RISE	flag controlling the inclusion of point sources emissions in output 3D gridded file, with plume rise calculation: 1 = enable point source inclusion 0 = disable
WIND_FILE	name of input archive (ADSO/bin or netCDF (COARDS/CF) format) containing 3D fields of horizontal components of the wind over the whole time interval of interest; the variables must be named "U" and "V" and its values given in [m s-1]
TEMPER_FILE	name of input archive (ADSO/bin or netCDF (COARDS/CF) format) containing 3D fields of air temperature over the whole time interval of interest; the variable must be named "T" and its values given in [K]
TCC_FILE	name of input archive (ADSO/bin or netCDF (COARDS/CF) format) containing 2D fields of total cloud cover over the whole time interval of interest; the variable must be named "TCC" and its values given either in tenths (recognised as [0..10] or [tenths]) or octals (recognised as [0..8] or [octals])
DIFFEMI_FORMAT	flag controlling the format of the output diffuse emissions file: 0 = ADSO/bin 1 = netCDF (COARDS)

FARM-specific keywords: generation of lumped NMVOC profiles

GASMECH	label of the selected SAPRC chemical mechanism: currently "`SAPRC90p`" or "`SAPRC99`"
SPEC_CAT	name of the file mapping NMVOC speciation profiles to categories

SPRAY-specific keywords

LINE_TREATMENT	if set to "gridded", forces the gridding of all line sources on the specified target grid; in any other case (i.e. if not specified or set to any other value), line sources are kept as individual segments
AREA_GRIDDING	if set to "geometric", forces the gridding of all area sources on the specified target grid, using a purely geometric method (ie. emissions from a given polygon are splitted among the intersecting grid cells basing only on the area of the polygon-cells intersections); in any other case (i.e. if not specified or set to any other value), area sources are gridded using a set of user-specified gridded proxies
KEEP_SINGLE_ACTIV	if set to "true", skip the suppression of individual activities in "pemtims" produced after the modulation phase; in any other case (i.e. if not specified or set to any other value), activities info are suppressed
KEEP_MERGED_ACTIV	if set to "true", skip the suppression of activities in merged "pemtims" for line and area sources; in any other case (i.e. if not specified or set to any other value), activities info are suppressed

Obsolete keywords (used in previous versions, not needed anymore)

MOD_YEAR, MOD_DAY, MOD_HOUR	names of files containing databases of yearly, weekly and daily modulation profiles, now substituted by YEARLY_PROF, WEEKLY_PROF, DAILY_PROF (warning: backward compatibility is checked only on assignments made on case configuration file case.mak, but not on subcases configuration files subcase.mak)

"Subcase" configuration: subcase.mak

Files of this type contain the specifications related to a given *subcase*, a subset of area sources client files containing emissions from a specific geographic region, inventory or group of activities. Each *subcase* directory must contain its own subcase.mak file, that can be used to assign input files that differ from the general ones of case.mak file, and that need to be applied only to the client files of the *subcase*. As a general rule, any specification given in a subcase.mak file supersedes what has been specified in case.mak file for the whole *case*.

An example of this file type is given below (format conventions are the same of case.mak file). In the example, a specific cartographic file ("DONCAR" format) containing all polygons to which area sources of the *subcase* are referred is specified through the CARFILE keyword. That file will be used when processing all the client files of the *subcase*, regardless of what specified through the CARFILE keyword in case.mak file (in fact, when all *subcases* have their own specific CARFILE, the CARFILE keyword in case.mak can be omitted).

Similarly, other specific input files involved in species, space and time disaggregation can be also assigned for each *subcase*, by assigning in subcase.mak the corresponding keyword (see the list explained for case.mak file), and providing that the associations among the involved data (assigned either to the *subcase* or to the whole *case*) are consistent.

As for client files involved in calculations, the same convention of the main *case* holds: by default, all client files placed in "area" subdir of the *subcase* are processed, unless specific client files are specified (as in the example below, through SELECTED_SRF_NAMES).

```
# Region1 subcase settings

# cartography

CARFILE = region1_polygons.car

# (optional) user-specified subset of client files to be processed

SELECTED_SRF_NAMES = region1_activity1.srf region1_activity2.srf
```

"Network" configuration: network.mak

In full analogy with subcase.mak files, network.mak files contain the specifications related to a given *network*, a subset of line sources client files containing emissions from a specific geographic network (roads, railway lines, shipping lanes, etc.), inventory or group of activities. Each *network* directory must contain its own network.mak file, that can be used to assign input files that differ from the general ones of case.mak file, and that need to be applied only to the client files of the *network*. As with *subcases*, any specification given in a network.mak file supersedes what has been specified in case.mak file for the whole *case*.

An example of this file type is given below (format conventions are the same of case.mak file). In the example, a specific cartographic file ("DONCAR" format) containing all polylines to which line sources of the *network* are referred is specified through the NETFILE keyword. That file will be used when processing all the client files of the *network*, regardless of what specified through the NETILE keyword in case.mak file (in fact, when all *networks* have their own specific NETFILE, the NETFILE keyword in case.mak can be omitted).

Similarly, other specific input files involved in species and time disaggregation can be also assigned for each *network*, by assigning in network.mak the corresponding keyword (see the list explained for case.mak file), and providing that the associations among the involved data (assigned either to the *network* or to the whole *case*) are consistent.

As for client files involved in calculations, the same convention of the main case holds: by default, all client files placed in "line" subdir of the *network* are processed, unless specific client files are specified (as in the example below, through SELECTED_LIN_NAMES).

```
# Primary roads network

# cartograpy

NETFILE = roadnet.car

# (optional) user-specified subset of client files

SELECTED_LIN_NAMES = cars.lin trucks.lin
```

4.1.4 CALCULATIONS WORKFLOWS

The schemes in the following sections detail the EMMA calculation workflow for each target model, with the emission data files and EMGR computational module involved in each phase ("ancillary" input files are not reported, for simplicity).

Each scheme corresponds to one or more EMMA commands, also cited (their syntax is documented in section 4.2).

In the schemes, the following conventions are used:

- a grey box denotes a file type (i.e., an intermediate or final target);
- a green box denotes a processing phase;
- a white box denotes an Emission Manager module;
- the icon denotes an emission file (an intermediate or final target, either in "client file" or "pemtim" format);
- the icon denotes a set of emission files (an intermediate or final target), each one corresponding to a subset of categories or sources;
- the thin arrow → denotes a flow concerning a single file;
- the thick arrow ➡ denotes a flow concerning a set of files.

SPRAY

The generation of an emissions input for SPRAY conceptually involves two phases (illustrated in Figures 7-8):

- **sources ingestion and disaggregation** (spatialization and time modulation);
- **data merge**.

Both include processing flows for point, EMS, line and area sources, with the possibility of organizing line and area sources in subsets (*networks* and *subcases*, respectively), if needed.

The workflow is organized to avoid repeating the space disaggregation when preparing the emissions for different time periods. The following options are also possible:

- line sources treatment: line sources can be either gridded on the specified target grid or kept as they are (individual segments);
- area sources gridding: can be either performed on a purely "geometric" basis (ie. emissions from a given polygon are splitted among the intersecting grid cells basing only on the area of the polygon-cells intersections) or using a set of user-specified gridded proxies;
- categories detail: in output "pemtims", the detail of individual activities can be either suppressed or maintained (either inside individual "pemtims" representing different sources sets or when "pemtims" are merged).

According to SPRAY needs, generated "pemtims" are always in ASCII format.

Sources disaggregation

For each type of sources (point, line and area), this phase involves (Figure 7):

- **spatialization & "pemtim" creation**: the *base "pemtims"* (extension `.ptm0`) ad the related *pemspes* (extension `.psp`) are generated from the *client files* (extension `.gsp`, `.lin`, `.srf`, according to source geometry), using TRLGSP, TRLLIN and TRLCRS modules; for line sources, polylines are broken in individual segments, and associated emissions distributed according to lengths; for area sources, space disaggregation on the target grid is performed, distributing the emissions associated to polygons on the intersecting grid cells, either with the help of gridded proxy layers specified in input or through simpler polygons-grid cells geometric intersection; for each source, the mass stored in output "pemtims" corresponds to the one emitted in a conventional period of 1 day (ie. considering constant emission rates);
- **species selection**: using REDSPE module, redundant species that may be present in emission files are eliminated; the *reduced "pemtims"* in output (extension `.red`) contain only the "target species" that are meaningful for the current simulation;
- optional **line sources grid projection**: using LIGREC module, segments contained in *redcuced "pemtims"* (extension `.red`) are intersected with the target simulation grid, and written as "square sources" in the output *projected "pemtims"* (extension `.rec`); if line sources are needed to maintained as segments, this operation is skipped and `.red` file is simply copied into `.rec` file;
- **time modulation**: using CRTEMP module, the mass emitted by each source is disaggregated into a series hourly emissions values covering all the current simulation time period, using modulation profiles assigned in input; the *modulated "pemtims"* (extension `.mdl`) are created from the *reduced "pemtims"* (extension `.red`), except for line sources, where the *projected "pemtims"* are used (extension `.rec`).

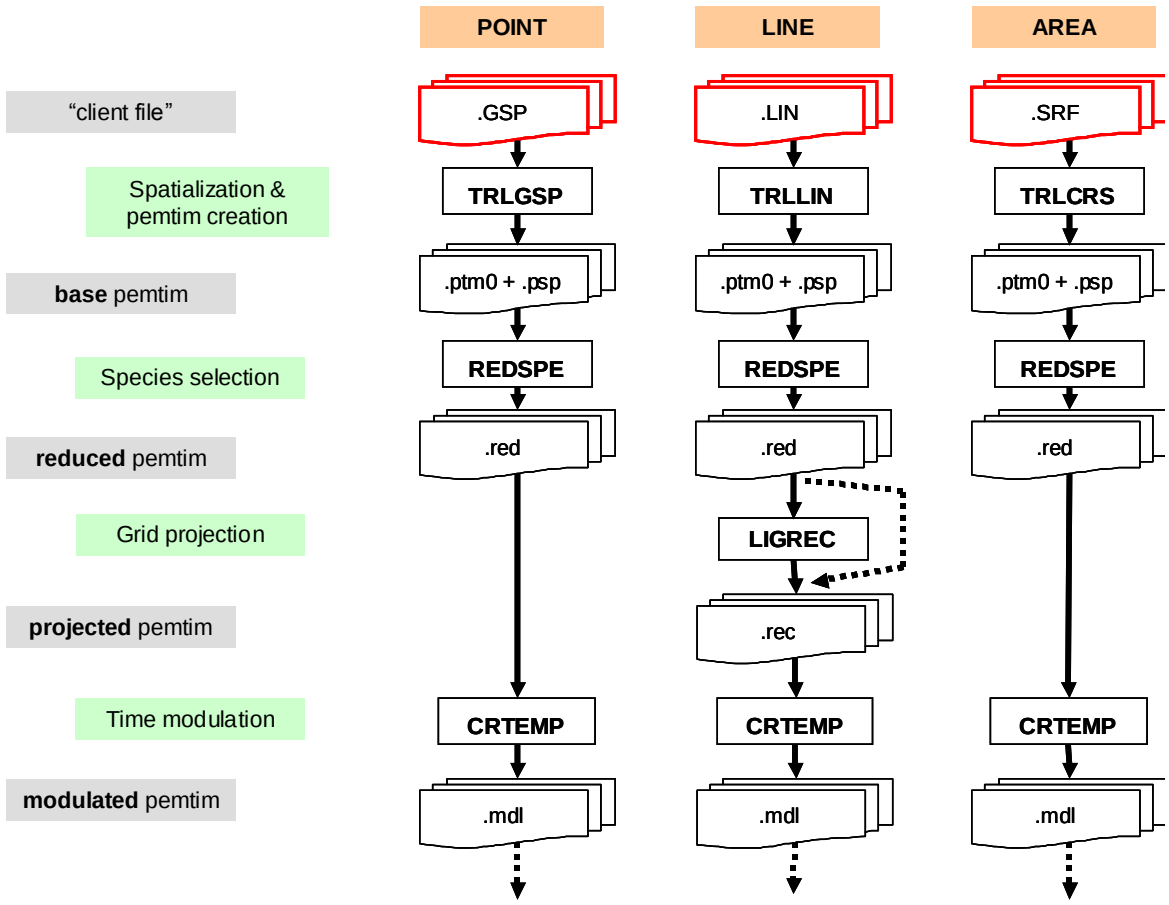


Figure 7. Calculation workflow for SPRAY: spatialization and modulation.

Emission data coming from continuous emissions monitoring systems (EMS) deserve a different treatment; for each input file related to EMS sources, the processing steps are as follows (Figure 7a):

- the first stage of the **EMS data ingestion**: the generation of the "*modulated "pemtims"* " (extension `.mdl`) and the related "*pemspes*" (extension `.psp`) from the "EMS hourly data files" (extension `.ems`); the Emission Manager module INEMOS is invoked for each "EMS hourly data file"; for each individual source, the output file contains hourly emission rates for the current time period, extracted from the corresponding input file and filled with stack data and (optional) default emissions from missed species or data; such modulated pemtims can also include time-varying stacks exit temperature and speed, if such information is present in the corresponding input `.ems` files;
- the **selection** of the **target species** for the specific model: the "*reduced "pemtims"* " (extension `.red`) are created from the "modulated "pemtims" ", using REDSPE module.

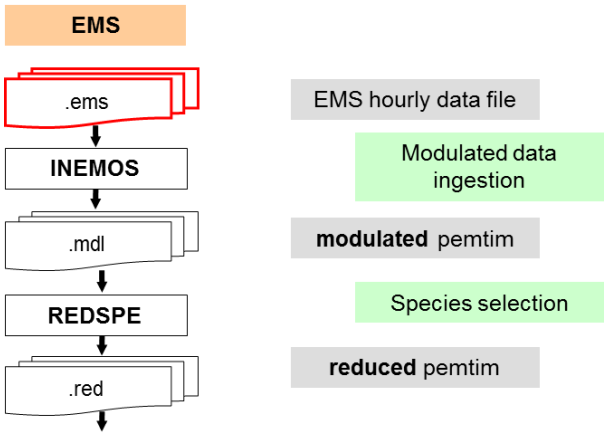


Figure 7a. Calculation workflow for SPRAY: ingestion of EMS data.

Data merge

Once emission data have been disaggregated in space and time, this phase leads to the creation of the input "pemtim" for SPRAY, through (Figure 8):

- **suppression of the individual categories** inside each "pemtim": using PEMSOM module, the *no SNAP "pemtims"* (extension `.nosnap`) are created from the *modulated "pemtims"* ; for each source (i.e. each geometric entity) the emissions from all the categories insisting on it are summed up; when needed, this operation can be skipped, keeping the information of individual categories related to each source;
- **concatenation** of all the emission files related to different **subsets**: using CATSOU module, a single *merged "pemtim"* is created for each source type/geometry (files `all.gsp.spray.merged`, `all.ems.spray.merged`, `all.lin.spray.merged` and `all.srf.spray.merged`, for point, line and area sources, respectively), merging all the corresponding *no SNAP "pemtims"* ;
- further **suppression of the individual categories** in the *merged "pemtims"* : using PEMSOM module, the *merged, no SNAP "pemtims"* (files `all.gsp.spray.nosnap`, `all.ems.spray.nosnap`, `all.lin.spray.nosnap` and `all.spray.srf.nosnap`) are created from `all.gsp.spray.merged`, `all.ems.spray.merged`, `all.lin.spray.merged` and `all.srf.spray.merged` files; when needed, this operation can be skipped, keeping the information of categories coming from the different input files;
- **final concatenation** of all the emission files related to different source geometries: using CATSOU module, the *merged, no SNAP "pemtims"* are combined into a single file, the *emissions input file for SPRAY* (`all.ptm`).

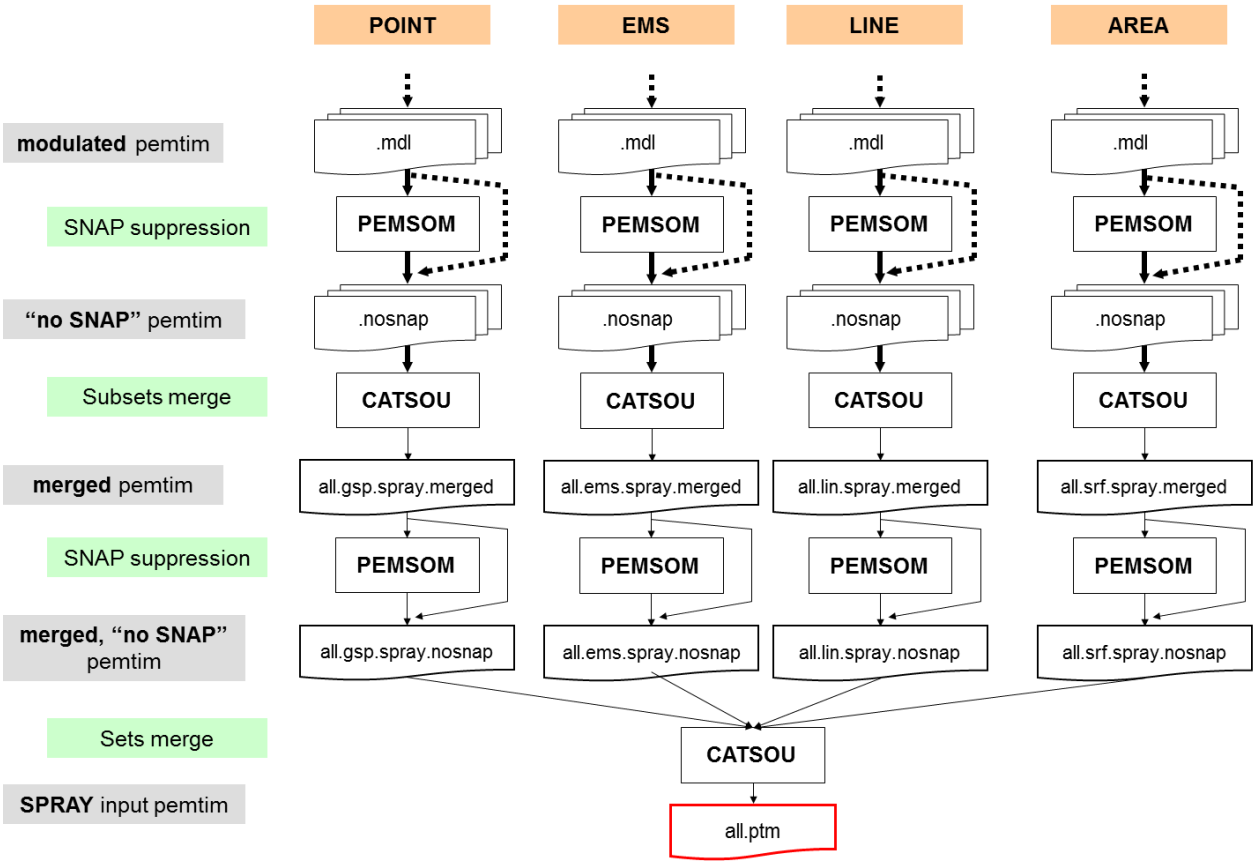


Figure 8. Calculation workflow for SPRAY: data merge.

FARM

The generation of an emissions input for FARM is organized in **two phases** (illustrated in Figures 9-11):

- **pre-processing** (time-independent);
- **time-dependent part**.

Both parts involve specific processing flows for point, line and area sources, with the possibility of organizing line and area sources in subsets (*networks* and *subcases*, respectively), if needed.

The workflow is optimized to minimize calculations when preparing the emissions for long-term runs or for forecast systems performing e.g. day-by-day operations. The pre-processing part manages all the calculations that are independent from the current time period (species split/speciation and space disaggregation); for a given inventory, target computational grid, and a given set of species and space disaggregation criteria this part is intended to be run only once. The second part instead is run at each time period, managing essentially time modulation operations and the final generation of input files for FARM model.

Generated "pemtims" files can be either in ASCII or binary format; typically, the latter is used in production mode, while ASCII format can be useful during a *case* setup or debug.

A further calculation chain allows to generate *case*- or domain-specific lumped NMVOC speciation profiles for SAPRC chemical mechanisms, starting from a database of speciation profiles expressed in terms of elementary hydrocarbons, a description of the selected chemical mechanism and the related lumping rules, and the mix of emissions resulting from all sources of the *case* of interest.

Pre-processing part

This part manages all the processing steps that are independent from the current time period (Figure 9). For each emissions file related to point, line and area inventory-based sources, it involves:

- **species split and NMVOC & PM speciation**: through SPICE module, emissions of aggregated species in *input client files* (extension .gsp, .lin, .srf, according to source type) are split in more detailed target chemical species, according to the needs of the chemical mechanism and PM module employed in FARM; the *speciated client files* (extension .spc) are then created in output; an arbitrary number of splitting schemes can be specified, including e.g. PM subdivision in dimensional fractions, NMVOC and PM speciation, NO_x split into NO and NO_2, and so on;
- **spatialization & "pemtim" creation**: the *base "pemtims"* (extension .ptm0) and the related *"pemspes"* (extension .psp) are generated from the *speciated client files* (extension .spc), using TRLGSP, TRLLIN and TRLCRS modules according to source geometry; for line sources, polylines are broken in individual segments, and associated emissions distributed according to lengths; for area sources, space disaggregation on the target grid is performed, distributing the emissions associated to polygons on the intersecting grid cells, with the help of gridded proxy layers specified in input; for each source, the mass stored in output "pemtims" corresponds to the one emitted in a conventional period of 1 day (i.e. considering constant emission rates);
- **line sources grid projection**: using LIGREC module, segments contained in *base "pemtims"* (extension .ptm0) are intersected with the target simulation grid, and written as "square sources" in the output *projected "pemtims"* (extension .rec);
- **species selection**: using REDSPE module, redundant species that may be present in emission files are eliminated; the *reduced "pemtims"* in output (extension .red) contain only the "target species" that are meaningful for the current simulation.

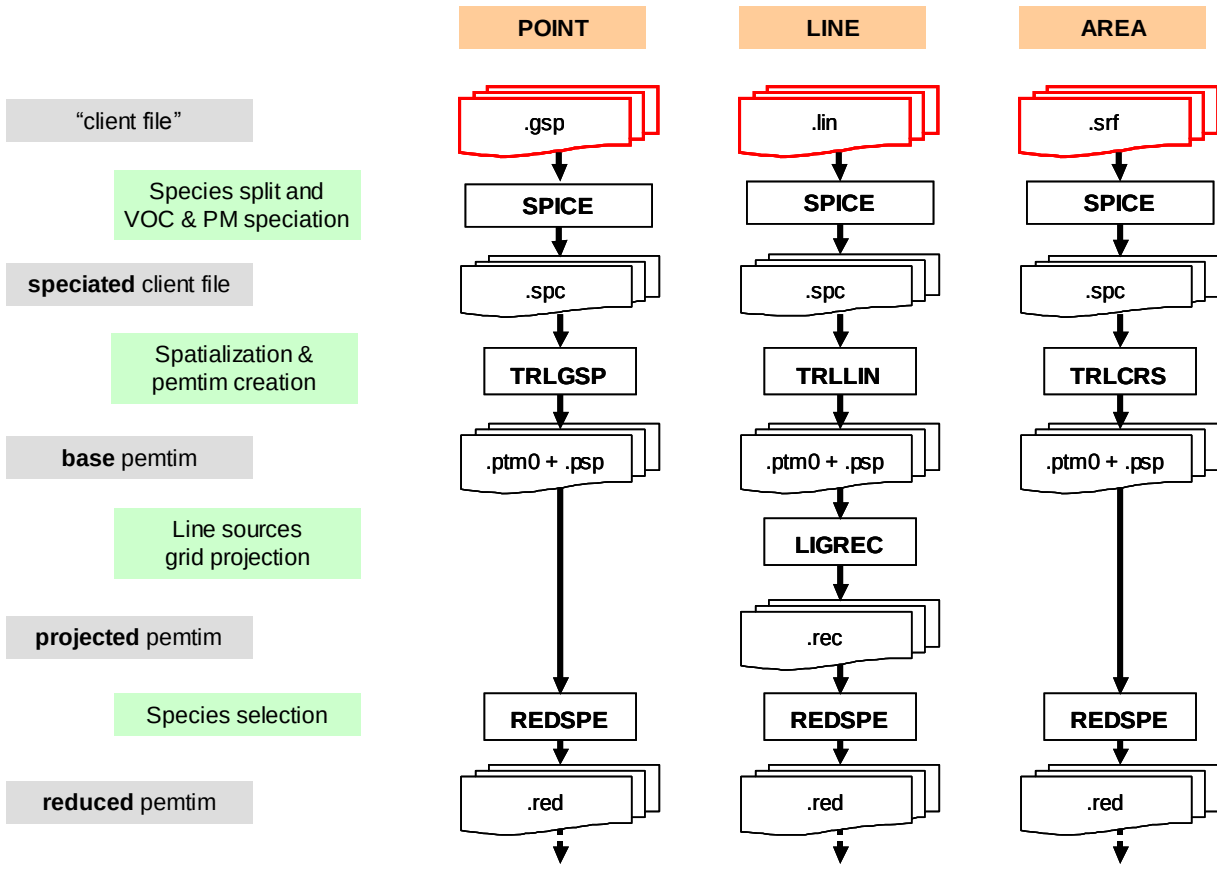


Figure 9. Calculation workflow for FARM: time-independent part.

This part is managed by the follwing commands:

- `FARM_emi`: invoked with `preproc` argument, executes all calculations involved in Figure 9 scheme;
- `FARM_point_sources`, `FARM_line_sources`, `FARM_area_sources`: execute the calculations showed in Figure 9 involving only point, line and area sources, respectively; the specification of `speciated / base / reduced / preproc` on command line allows to perform only the calculations until the desired stage.

Time-dependent part

This part manages all the processing steps depending on the current time period (Figures 10 and 11). The treatment is different if the emissions come from an inventory referred to a standard time period (e.g. year, day, hour) and need to be disaggregated into hourly values, or come from automatic monitoring systems and are already expressed as hourly values. Moreover, the sequence of processing steps changes when FASTMOD option is activated (combined time modulation and conversion into FARM input binary format).

In the traditional sequence (FASTMOD option not specified in case.mak, or set to 0), for each emissions file related to point, line and area inventory-based sources, the time-dependent phase involves (Figures 10a and 11a):

- **time modulation**: using CRTEMP module, the mass emitted by each source is disaggregated into a series of hourly emissions values covering all the current simulation

time period, using modulation profiles assigned in input; the *modulated "pemtims"* (extension `.mdl`) are created from the *reduced "pemtims"* (extension `.red`).

For each file related to EMS sources, the processing steps are as follows (Figure 10a):

- the first stage of the **EMS data ingestion**: the generation of the "*modulated "pemtims"* "(extension `.mdl`) and the related "*pemspes*" (extension `.psp`) from the "EMS hourly data files" (extension `.ems`); the Emission Manager module INEMOS is invoked for each "EMS hourly data file"; for each individual source, the output file contains hourly emission rates for the current time period, extracted from the corresponding input file and filled with stack data and (optional) default emissions from missed species or data;
- the **selection** of the **target species** for the specific model: the "*reduced "pemtims"* " (extension `.red`) are created from the "modulated "pemtims" ", using REDSPE module;
- the **NMVOC and PM speciation** according to the target chemical mechanism and PM module: the "*speciated "pemtims"* " (extension `.spc`) are created from the "reduced "pemtims" ", using BOTTARGA module.

After these steps, all "pemtims" contain speciated, gridded (in case of line and area sources) and modulated emissions, ready to be transformed in FARM input files, the last processing step:

- **"pemtims" merge & translation to gridded format**: using MPEM2FARM module, the `pointemi.dat` file is created, containing hourly emissions for the selected time period from all stacks; using PEM2GRID module, the `diffemi.bin` file is created, containing hourly emissions for the selected time period from all line and area sources.

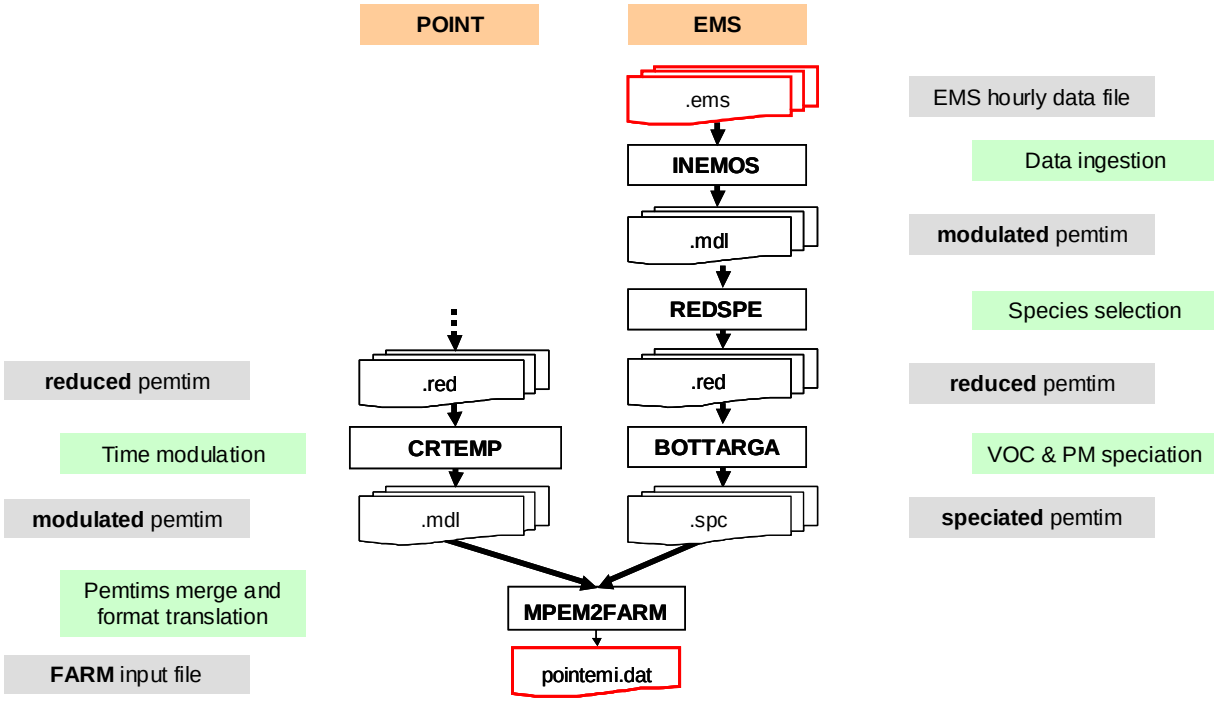


Figure 10a. Calculation workflow for FARM: time-dependent part (point sources), traditional way (FASTMOD=0, or not specified).

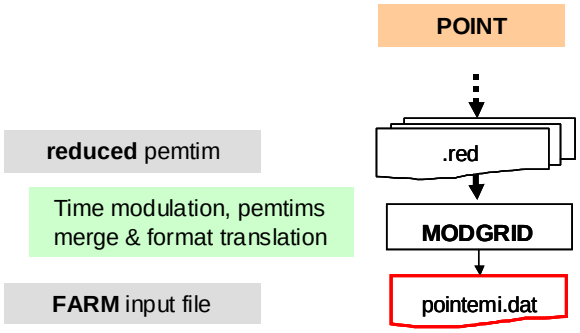


Figure 10b. Calculation workflow for FARM: time-dependent part (point sources), with FASTMOD=1.

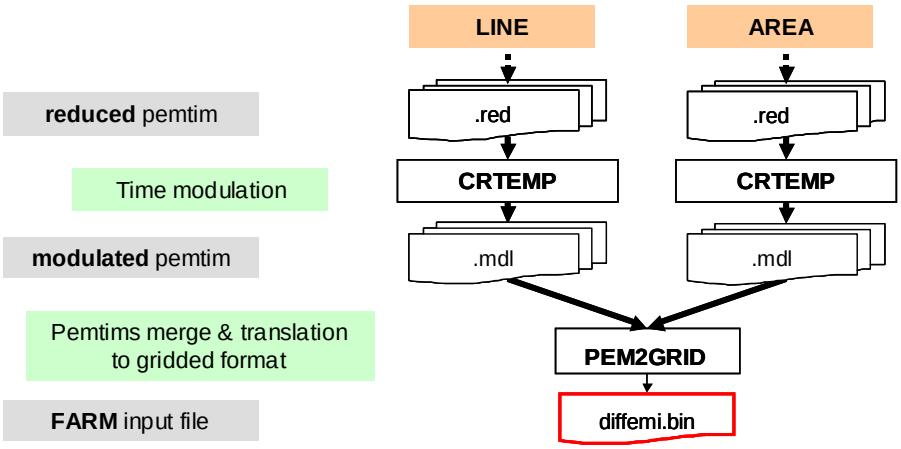


Figure 11a. Calculation workflow for FARM: time-dependent part (diffuse sources), traditional way (FASTMOD=0, or not specified).

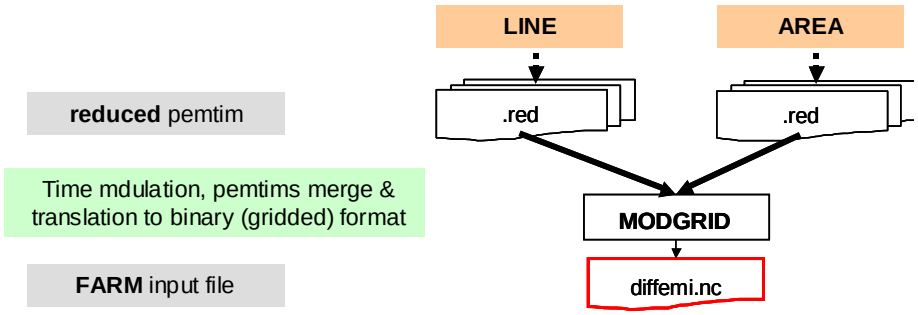


Figure 11b. Calculation workflow for FARM: time-dependent part (diffuse sources), with FASTMOD=1.

Otherwise, when `FASTMOD` option is set to 1 in case.mak, the time-dependent phase involves only (Figures 10b and 11b):

- **"pemtims" modulation, merge & translation to binary (gridded) format**: using MODGRID module, emissions in all *reduced "pemtims"* (extension `.red`) are firstly disaggregated into hourly time series and then written into a FARM binary input file (either a `pointemi.dat`, in case of point sources, or a gridded archive `in case of` line and area sources).

Using this option, the creation of intermediate *modulated "pemtims"* (extension `.mdl`) is avoided, allowing a susbtantial reduction of processing time and disk working space. Emission data from EMS are inherently modulated, so when EMS sources are present, the `FASTMOD` option is automatically disabled by the system during the generation of the `pointemi.dat` (ie. the workflow of Figure 10a is followed, instead of the one of Figure 10b).

When "fast modulation" is used, 3D gridded emissions are selected as output and emissions from point sources are injected in the 3D emission matices (by setting to 1 `FASTMOD`, `OUT3D` and `GSP_PLUME_RISE` options case.mak), the gridded emissions archive for FARM is created from all *reduced "pemtims"* (extension `.red`) through a single call of MODGRID module (Figure 12).

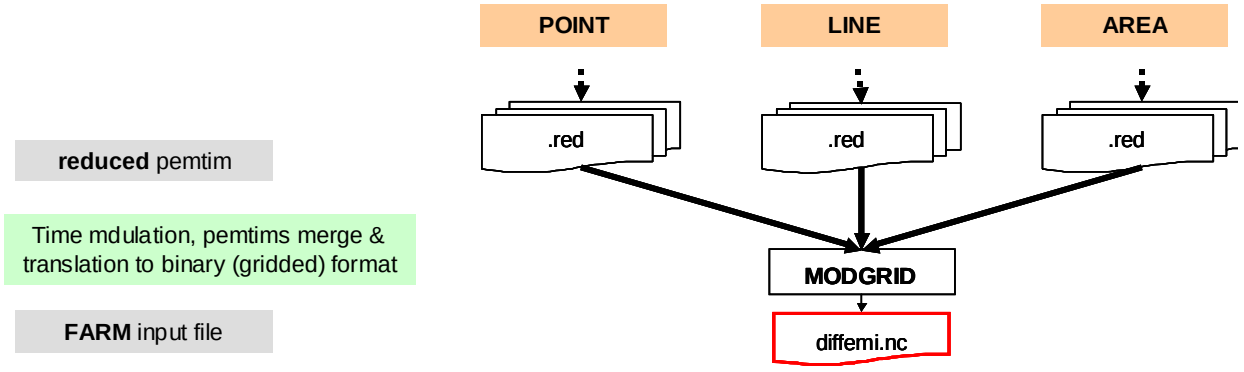


Figure 12. Calculation workflow for FARM: time-dependent part (diffuse & point sources), with FASTMOD=1, OUT3D=1 and GSP_PLUME_RISE=1.

The time-dependent part is managed by the following commands:

- `FARM_emi`: invoked with the desired time interval in the command line, executes all calculations involved in Figures 10 and 11 schemes (and the ones of Figure 9 scheme, if not already performed);
- `FARM_point_sources`, `FARM_line_sources`, `FARM_area_sources`: modulate only point, line and area sources, respectively, on the target time interval specified in the command line (and also perform the steps of Figure 9 scheme, if not already completed).

The generation of FARM input file for **diffuse sources** can be optionally speed-up by assembling a set of **"typical days of the week"** (for each month), specified according to a user-specified scheme. This option is activated through the `REPEATING_WEEKS` flag in case.mak, while the scheme is specified through `WEEKLY_SCHEME` keyword. The scheme tells which days of the week are different from each other and which ones share the same time modulation factors (e.g. within a week, all days from Monday to Friday can be considered to have the same emissions, while Saturday and Sunday can have different values; this would be specified as "1 1 1 1 1 6 7"). The scheme of the days of the week has to be selected considering the modulation profiles actually employed and the result to be obtained (no cross-check is performed against the set of modulation profiles actually employed). When this option is activated, FARM input files for diffuse sources for each independent day and month are generated only once and only if implied by the current time interval. These files are stored in the FARM subdirectory of the current *case*, and are named:

```
mXX.dY.diffemi.nc
```

where XX is the month (01, 02, … 12) and Y the index of the day of the week (1…7). Once generated the first time that the combination (XX&Y) is needed, each of these files is then possibly reused when generating FARM input files for other target time intervals involving the same combination days of the week and months. This feature can save computational time in e.g. long-term runs (monthly, yearly) or in day-by-day operations of a forecast system.

Generation of lumped NMVOC profiles

The `FARM_lump_profiles` allows to generate *case*- or domain-specific lumped NMVOC speciation profiles for a given SAPRC chemical mechanism (Figure 13): through BOTTARGA module, the speciation profiles of an input database expressed in terms of elementary hydrocarbons are converted in profiles expressed in terms of the lumped species of the selected SAPRC chemical mechanism; the hydrocarbons mixture used for the operation is obtained using all the emissions contained in all *input client files* (extension `.gsp`, `.lin`, `.srf`) of the *case* of interest; the procedure also uses a description of the selected chemical mechanism and the related lumping rules, specified in files that are usually part of the installation kit (stored in subdirectory "`SAPRC`" of the "`ADSO`" models tree).

The output database of speciation profiles is the `xxxxxxx.split` file (where xxxxxxx is the name of the selected SAPRC chemical mechanism), stored in the "`spec`" subdirectory of the *case*. The choice of using this database for the actual speciation of NMVOC emissions of a given *case* (either the one used to generate the lumped profiles or another one) is left to the user. This can be done by copying the resulting `xxxxxxx.split` file in the "`tables`" subdirectory of the desired *case* and citing that file in the splitting scheme employed for NMVOC (second argument of the involved "`SPEC_SPLIT_SCHEME_y = …`" statement in `case.mak`).

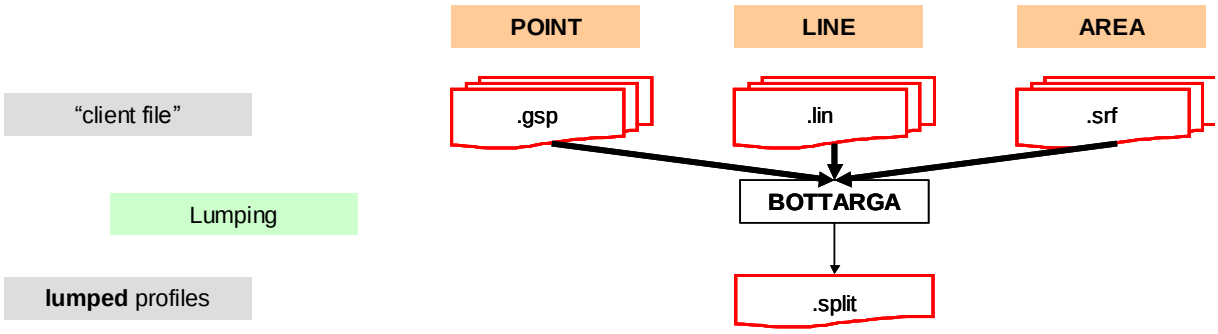


Figure 13. Calculation workflow for FARM: generation of lumped profiles for NMVOC speciation.

Logfiles and error tracking

Each time an EMMA command is executed, a **command logfile** (named <command>.log) is written in *case* root directory. That logfile contains the sequence of module calls.
The standard output of each module run is then also redirected to a **module logfile** (named r_<module>.log; e.g. "r_trlgsp.log"), written in the directory of the target file being created. By default, a given module logfile is overwritten when the corresponding module is run on multiple files.

Optionally, **individual logfiles** corresponding to each individual module run can be also saved (option `KEEPLOGS = 1` in `case.mak`). Such logfiles are named <target_file_name>.log, and are saved in the same directory of the target file just created.
All logfiles are standard ASCII files under the hosting operating system (DOS/Windows or Unix/Linux).

Possible errors appearing during the execution of EMMA commands and the implied computational modules are tracked through a series of mechanisms:

- the phase in which the error occurred is reported in the **command logfile**
- a **global error file** (error.log) appears in *case* root directory, containing the list of failed modules, the target files that have not been created and the names of the individual log files of failed modules
- logfiles related to individual failed modules are saved as **modules error files**, named <name_of_target_file>.err
- if an **incomplete target file** exists, this is saved as <name_of_target_file>.wrong

4.2 SHELL COMMANDS REFERENCE

Once a unix/cygwin shell is opened, all EMMA commands are invoked from EMMA system directory (subdirectory "`EMMA`" of "`ADSO`"), usually following the syntax (please note the leading dot and blank):

```
. < EMMA_command > < casedir > < command_option(s) >
```

where:

- **`casedir`** is the name of the main directory of the current *case*;
- **`command_option`** specifies the operation to be performed (i.e. the processing phase to be achieved, usually corresponding to a set of target emission files to be created); command-specific options are listed explained in detail in the following paragraphs (please also refer to the descriptions and schemes of calculation workflows in the previous section of the manual).

EMMA shell commands can be grouped in three sets:

- **top-level**: comprehensive commands, managing all the operations needed to create input emissions files for a given model;
- **low-level**: these commands are invoked by the first set, but that can be also used separately for specific purposes (e.g. to do step-by-step processing when setting up or debugging a *case*);
- **utilities**: commands performing case-maintenance tasks.

4.2.1 TOP-LEVEL COMMANDS

SPRAY_emi

The command drives all the processing steps needed to generate an emission file for SPRAY model.

Syntax

```
. SPRAY_emi < casedir >
            < base / reduced / modulated D1 D2 /
              nosnap D1 D2 / merged D1 D2 / merged_nosnap D1 D2 /
              SPRAY D1 D2 / purge / clean >
```

where:

base	generates base “pemtims” from “client files”
reduced	selects species from base “pemtims”
modulated D1 D2	modulates “pemtims” for time interval between dates D1 & D2 (specified as "DD MM YYYY")
nosnap D1 D2	generates “pemtims” (optionally) without SNAP codes
merged D1 D2	generates merged “pemtims”
merged_nosnap D1 D2	generates merged “pemtims” (optionally) without SNAP codes
SPRAY D1 D2	generates input “pemtim” for SPRAY
purge	deletes intermediate files of current time period
clean	deletes all target files

FARM_emi

The command drives all the processing steps needed to generate the emission files needed by a run of FARM model for a given time period.

Syntax

```
. FARM_emi  < casedir >
            < preproc / < D1 D2 > /
              purge / clean / cleanall >
```

where:

preproc	performs all the pre-processing steps needed to create the time-independent part of FARM input file(s)
D1 D2	generates all FARM input files for time interval between dates D1 & D2 (specified as "DD MM YYYY")
purge	deletes intermediate files of current time period
clean	deletes target files of current time period
cleanall	deletes all target files (including pre-processing phase and reference diffuse emission files by days of the week & months)

FARM_lump_profiles

The command drives all the processing steps needed to generate a database of case-specific lumped profiles for NMVOC speciation according to SAPRC mechanisms.

Syntax

```
. FARM_lump_profiles  < casedir > < all | purge | clean >
```

where:

all	generate a set of (case-specific) lumped NMVOC profiles, using all client files of current case
purge	delete lists of client files used for lumping
clean	delete also output lumped NMVOC profiles

4.2.2 LOW-LEVEL COMMANDS

SPRAY_point_sources

The command drives all the processing steps needed to generate the emission files for point sources related to SPRAY.

Syntax

```
. SPRAY_point_sources  < casedir >
                       < base / reduced / modulated D1 D2 /
                         nosnap D1 D2 / merged D1 D2 /
                         merged_nosnap D1 D2 / SPRAY D1 D2 /
                         purge / clean >
```

where:

base	generates base “pemtims” from “client files”
reduced	selects species from base “pemtims”
modulated D1 D2	modulates “pemtims” for time interval between dates D1 & D2 (specified as "DD MM YYYY")
nosnap D1 D2	generates “pemtims” (optionally) without SNAP codes
merged D1 D2	generates merged “pemtims”
merged_nosnap D1 D2	generates merged “pemtims” (optionally) without SNAP codes
SPRAY D1 D2	generates input “pemtim” for SPRAY
purge	deletes intermediate files of current time period
clean	deletes all target files

SPRAY_ems_sources

The command drives all the processing steps needed to generate the emission files for EMS sources related to SPRAY.

Syntax

```
. SPRAY_ems_sources  < casedir >
                       < modulated D1 D2 / reduced D1 D2 /
                         nosnap D1 D2 / merged D1 D2 /
                         merged_nosnap D1 D2 / SPRAY D1 D2 /
                         purge / clean >
```

where:

modulated D1 D2	generates modulated "pemtims" from EMS files, for time interval between dates D1 & D2 (specified as "DD MM YYYY")
reduced D1 D2	selects species from modulated "pemtims"
nosnap D1 D2	generates "pemtims" (optionally) without SNAP codes
merged D1 D2	generates merged "pemtims"
merged_nosnap D1 D2	generates merged "pemtims" (optionally) without SNAP codes
SPRAY D1 D2	generates input "pemtim" for SPRAY
purge	deletes intermediate files of current time period
clean	deletes all target files

SPRAY_line_sources

The command drives all the processing steps needed to generate the emission files for line sources related to SPRAY.

Syntax

```
. SPRAY_point_sources  < casedir >
                       < base / reduced / modulated D1 D2 /
                         nosnap D1 D2 / merged D1 D2 /
                         merged_nosnap D1 D2 / SPRAY D1 D2 /
                         purge / clean >
```

where:

base	generates base "pemtims" from "client files"
reduced	selects species from base "pemtims"
modulated D1 D2	modulates "pemtims" for time interval between dates D1 & D2 (specified as "DD MM YYYY")
nosnap D1 D2	generates "pemtims" (optionally) without SNAP codes
merged D1 D2	generates merged "pemtims"
merged_nosnap D1 D2	generates merged "pemtims" (optionally) without SNAP codes
SPRAY D1 D2	generates input "pemtim" for SPRAY
purge	deletes intermediate files of current time period
clean	deletes all target files

SPRAY_area_sources

The command drives all the processing steps needed to generate the emission files for area sources related to SPRAY.

Syntax

```
. SPRAY_point_sources  < casedir >
                         < base / reduced / modulated D1 D2 /
                           nosnap D1 D2 / merged D1 D2 /
                           merged_nosnap D1 D2 / SPRAY D1 D2 /
                           purge / clean >
```

where:

Base	generates base “pemtims” from “client files”
Reduced	selects species from base “pemtims”
modulated D1 D2	modulates “pemtims” for time interval between dates D1 & D2 (specified as "DD MM YYYY")
nosnap D1 D2	generates “pemtims” (optionally) without SNAP codes
merged D1 D2	generates merged “pemtims”
merged_nosnap D1 D2	generates merged “pemtims” (optionally) without SNAP codes
SPRAY D1 D2	generates input “pemtim” for SPRAY
purge	deletes intermediate files of current time period
clean	deletes all target files

FARM_point_sources

The command drives all the processing steps needed to generate the emission files for point sources related to FARM.

Syntax

```
. FARM_point_sources   < casedir >
                       < speciated / base / reduced / preproc |
                         modulated D1 D2 / FARM D1 D2 |
                         purge | clean >
```

where:

speciated	speciates the aggregated species in “client files”
base	generates base “pemtims” from speciated “client files”
reduced	selects species from base “pemtims”
preproc	does all the steps listed above
modulated D1 D2	modulates “pemtims” for time period between D1 & D2 (and ending date of the target time period, specified as "DD MM YYYY")
FARM D1 D2	generates all files needed by FARM, for time interval between D1 & D2 (same as 'modulated')
purge	deletes intermediate files of last time period that has been run
clean	deletes all target files

FARM_ems_sources

The command drives all the processing steps needed to manage the EMS emission files needed to build an emission input for FARM.

Syntax

```
. FARM_ems_sources   < casedir >
                     < modulated | reduced | speciated | mdl_list |
                       FARM | purge | clean >
```

where:

modulated	generates modulated “pemtims” from EMS files
reduced	selects species from modulated “pemtims”
speciated	speciates VOCs and PM in reduced “pemtims”
preproc	does all the above steps
mdl_list	generates the list of speciated “pemtims”
FARM	generates all files needed by FARM
purge / clean	deletes all target files

FARM_line_sources

The command drives all the processing steps needed to generate the emission files for line sources related to FARM.

Syntax

```
. FARM_line_sources   < casedir >
                      < speciated / base / reduced / preproc |
                        modulated D1 D2 / FARM D1 D2 |
                        purge | clean >
```

where:

speciated	speciates the agregated species in “client files”
base	generates base “pemtims” from speciated “client files”
reduced	selects species from base “pemtims”
preproc	does all the steps listed above
modulated D1 D2	modulates “pemtims” for time period between D1 & D2 (and ending date of the target time period, specified as "DD MM YYYY")
FARM D1 D2	generates all files needed by FARM, for time interval between D1 & D2 (same as 'modulated')
purge	deletes intermediate files of last time period that has been run
clean	deletes all target files

FARM_area_sources

The command drives all the processing steps needed to generate the emission files for area sources related to FARM.

Syntax

```
. FARM_area_sources   < casedir >
                      < speciated / base / reduced / preproc |
                        modulated D1 D2 / FARM D1 D2 |
                        purge | clean >
```

where:

speciated	speciates the agregated species in “client files”
base	generates base “pemtims” from speciated “client files”
reduced	selects species from base “pemtims”
preproc	does all the steps listed above
modulated D1 D2	modulates “pemtims” for time period between D1 & D2 (and ending date of the target time period, specified as "DD MM YYYY")
FARM D1 D2	generates all files needed by FARM, for time interval between D1 & D2 (same as 'modulated')
purge	deletes intermediate files of last time period that has been run
clean	deletes all target files

4.2.3 UTILITY COMMANDS

touch_case_FARM

The command allows to consider all the emission files of a given FARM *case* as being “updated”, by changing their modification time according to the sequence of the processing chain (see Figures 9-11 for reference).

It can be useful during case setup and debug, when e.g. ancillary files used for emissions disaggregation (databases of profiles, mapping tables, etc.) may be progressively adjusted: after such an adjustment, the user can delete the emission files that need to be re-generated, then issue a `touch_case_FARM` to consider all the remaining emission files until a certain stage as “consolidated”, and finally re-issue the EMMA command(s) of interest to re-generate only the emissions files.

The command of course has to be employed carefully, since it forces the system to consider as “updated” also what in reality is not.

Syntax

```
. touch_case_FARM  < casedir >
                   < speciated / base / reduced / preproc /
                     modulated / FARM >
```

where:

speciated	until speciated client files
base	until base “pemtims”
reduced	until reduced species “pemtims”
preproc	all the “pemtims” of the-preprocessing phase
modulated	until modulated “pemtims”
FARM	all the *case*, until FARM input files

copy_case

The command creates a copy of an existing *case*, creating a new directories tree and filling the directories with copies of the base files of the originating *case* ("client files", classification and mapping tables, cartographic files, thematic layers, time modulation data, species split data).

Syntax

```
. copy_case < existing_casedir > < new_casedir >
```

where:

existing_casedir	directory of an existing and valid EMMA *case*
new_casedir	directory of the new EMMA *case* to be created

ANNEX A – OTHER FILES FORMATS

A.1. "PEMTIM"

PEMTIM is the standard file format used by most Emission Manager modules to store their input/output emissions data (notable exceptions are base emission inventory data, stored in "client files", and model-ready inputs, that can be stored in model-specific formats).
An example of such file (ASCII format) is given below. It contains emissions referred to 4 sources: 1 rectangular and 1 triangular (both non-modulated), 1 line, 1 point (with multiple time steps).

```
         801
           3           5
MCG.
           1
   161.0000000       2307.000000
           10  6 86 13 12 11
        1#SRF     #RECT    #R  6  6C  1     #       2      2      2      0      0
        1#
#       1# 24# 0# 0#
         #  10100.#  10100.#   10.#        0#    3#  0.20E+02#
         #   2000.#   2000.#   20.#  0.00E+00#   0.00#   0.00# 0.00E+00#
  1#CO      # .260E+11# 901#
  2#HC      # .115E+10# 919#
  3#NOX     # .6348+12# 903#
  4#PA      # .000E+00# 920#
  5#CO2     # .234E+17# 905#
  6#SO2     # .204E+10# 906#
        2#DISTRICT#Lorient1#RECT            #     11     15     15   -999   -999
        1#
#       1# 24# 0# 0#
         #  11377.#  13950.#    0.#        1#   31#   .20E+02#
         #  11212.#  14044.#    0.#         2         0         0         0
         #  11210.#  13949.#    0.#
         #      0.#      0.#    0.#   .00E+00#    .00#    .00#  .00E+00#
  1#CO      # .150E+10# 901#
  2#HC      # .045E+10# 919#
  3#NOX     # .112E+09# 903#
  4#PA      # .000E+00# 920#
  5#CO2     # .000E+00# 905#
  6#SO2     # .104E+09# 906#
        3#Transpor#Total   #TRAFIC         #      7   -999   -999   -999   -999
        2#
#       1# 14#55#35#
         #   4001.#   1100.#    0.#        0#    2# -0.99E+02#
```

Header

Rectangular source — 1st time step

Triangular source — 1st time step

Line Source

```
          #   8001.#  29100.#    0.#          0          0          0          0
          #      0.#      0.#    0.#  0.00E+00#   0.00#   0.00# 0.00E+00#
  1#CO      #0.400E+13# 901#
  2#HC      #0.100E+13# 919#
  3#NOX     #0.440E+14# 903#
  4#PA      #0.000E+00# 920#
  5#CO2     #0.191E+17# 905#
  6#SO2     #0.108E+15# 906#
#        2#598#59# 0#
          #   8001.#   1200.#    0.#         0#    2# -0.99E+02#
          #  12001.#  29200.#    0.#          0          0          0          0
          #      0.#      0.#    0.#  0.00E+00#   0.00#   0.00# 0.00E+00#
  1#CO      #0.338E+15# 901#
  2#HC      #0.520E+14# 919#
  3#NOX     #0.117E+16# 903#
  4#PA      #0.000E+00# 920#
  5#CO2     #0.359E+18# 905#
  6#SO2     #0.462E+15# 906#
        4#CombIn1 #3M      #PTS              #      3      1      3   -999   -999
        3#
#        1# 18# 0# 0#
          #   1050.#   1060.#  900.#         0#    0#  0.25E+01#
          #      0.#      0.#    0.#  0.12E+03#   0.00#   0.00# 0.13E+02#
  1#CO      #0.200E+11# 901#
  2#HC      #0.250E+14# 919#
  3#NOX     #0.300E+11# 903#
  4#PA      #0.350E+09# 920#
  5#CO2     #0.400E+10# 905#
  6#SO2     #0.450E+10# 906#
#        2# 18#35# 0#
          #   1050.#   1060.#  900.#         0#    0#  0.25E+01#
          #      0.#      0.#    0.#  0.12E+03#   0.00#   0.00# 0.13E+02#
  1#CO      #0.202E+11# 901#
  2#HC      #0.252E+14# 919#
  3#NOX     #0.302E+11# 903#
  4#PA      #0.352E+09# 920#
  5#CO2     #0.402E+10# 905#
  6#SO2     #0.452E+10# 906#
#        3# 15#50#15#
          #   1050.#   1060.#  900.#         0#    0#  0.23E+01#
          #      0.#      0.#    0.#  0.12E+03#   0.00#   0.00# 0.13E+02#
  1#CO      #0.203E+11# 901#
  2#HC      #0.253E+14# 919#
  3#NOX     #0.303E+11# 903#
  4#PA      #0.353E+09# 920#
  5#CO2     #0.403E+10# 905#
  6#SO2     #0.453E+10# 906#
```

2nd time step

1er time step

2nd time step

3ième time step

Point source

The file is fixed-format, and meaning of data fields is outlined in the following tables.

File header

PEMTIM version code	
Number of sources	Number of classification levels
Unit of measurement	
Type of emission (1=mass; 2=activity)	
Absolute origin of coordinates along X (km)	Absolute origin of coordinates along Y (km)
File beginning date and hour (DD MM YY HH MM SS)	

Source records

Source progressive #	Name1 (A8)	Name2 (A8)	Name3 (A16)	Values of the category classification codes			
Number of time intervals							
Number of current time interval	Current time interval: # of hours (HH)	Current time interval: # of minutes (MM)	Current time interval: # of seconds (SS)				
X coord. of P1 (m)	Y coord. of P1 (m)	Source height from ground P1 (m)	# of poly-object to which the source belongs (0 otherwise)	Source geometry code (0,1,2,3,30 and 31, see below)	Temperature (°C)		**(1)**
X coord. of P2 (m)	Y coord. of P2 (m)	Source height from ground P2 (m)	# of preceding object	# of following object	3rd neighbourhood	4th neighbourhood	**(2)**
X coord. of P3 (m)	Y coord. of P3 (m)	Source height from ground P3 (m)					**(3)**
Source X-dimension (m)	Source Y-dimension (m)	Source Z-dimension(m)	Source diameter(m)	Horizontal speed along x (m/s)	Horizontal speed along y (m/s)	Emission vertical speed (m/s)	**(4)**
Progressive number of substance 1	Name of substance 1	Emission of substance 1 during current time interval	Code of substance 1				
...	...	...	...				
Progressive number of substance n	Name of substance n	Emission of substance n during current time interval	Code of substance n				

The presence of records marked with (1)-(4) depends on **source geometry**, according to the following table.

Geometry code	Source geometry	Presence of record			
		(1)	(2)	(3)	(4)
0	point	✓			✓
1	box	✓			✓
2	line	✓	✓		✓
3	rectangular (P1, DX, DY)	✓			✓
30	rectangular (P1, P2, DY)	✓	✓	✓	✓
31	triangular	✓	✓	✓	✓

For completeness, the following table gives the names of the internal variables used by PEMTIM access routines, corresponding to header and data fields listed above (this can be useful when interpreting error messages given by some EMGR modules).

```
NUVSOU
          NUMSOU       NCASOU
UNINAM
           IUNITE
   XABSOR         YABSOR
            IJOORI IMOORI IANORI IHEORI IMIORI ISEORI
   (ISOU) # NAMUSR # NAMCOU #     NAMSOU     #      ICASOU ICASOU ICASOU ICASOU ICASOU
  NUMPER #
#  (IPER) # IDHSOU # IDMSOU # IDSSOU #
          # XEMSOU # YEMSOU # HEMSOU #  IOBSOU # ITYSOU #  TEMPER  #                    (1)
          # XEFSOU # YEFSOU # HEFSOU #         IN1SOU    IN2SOU    IN3SOU    IN4SOU   (2)
          # XETSOU # YETSOU # HETSOU #                                                  (3)
          # DXESOU # DYESOU # DZESOU #   DIAMOU # UEMSOU # VEMSOU #  WVERTI #          (4)
(IESP)# NAMSPE # QESPEC # INDSPE #
(IESP)# NAMSPE # QESPEC # INDSPE #
(IESP)# NAMSPE # QESPEC # INDSPE #
(IESP)# NAMSPE # QESPEC # INDSPE #
……
```

The same information is stored in the **binary version** of PEMTIM.

A.2. PEMSPE

PEMSPE is the standard file format used by most Emission Manager modules to store the reference list of species/substances and some related parameters.
An example of such file is given below. It is an ASCII file containing a header and data records, one for each species. Comment lines begin with an asterisk ("*") in the first column.

```
*********************************************************************
* NLAACT* = ARBITRARY VALUE
*-------------------------------------------------------------------
        1
*-------------------------------------------------------------------
* NDPUTI * = TOTAL NUMBER OF SUBSTANCES
*-------------------------------------------------------------------
        6
*********************************************************************
*-------------------------------------------------------------------
*IP* NAMSPE * IND*IL*  VDEP * RWASH   *  TDECAY  *    DENS * DIAM * IT*
*  *  (8.C) *    *  *   M/S * SEC-1   *  SEC     *  KG/M3  * MIC. * --*
*-------------------------------------------------------------------
  1*CO      * 919* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
  2*CO2     * 920* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
  3*COVNM   * 921* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
  4*NH3     * 922* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
  5*NOx     * 923* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
  6*SO2     * 924* 0* .00000*  .00E+00*  .000E+00* .999E+03*    0.*  0*
* ----------------------------
```

Description of parameters:

In species records the meaning of data columns is:

- NLAACT: arbitrary value (a value should be defined but it's not used)
- NDPUTI: total number of defined substances
- IP: species index (from 1 to NDPUTI, following the integer increasing numbers)
- NAMSPE: species name (maximum 8 characters)
- IND: species numerical code;
- IL: flag indicating the activation of a link with another table (e.g.: gas parameters, toxicity, etc.), using IND field as a link key:
  - 1 = link to radioactive elements table
  - 0 = no link with another table
  - 2 = link to heavy gases table
- DIAM: diameter (μm), when the entry is a particles dimensional class
- IT: species type:
  - 0 = gas;
  - 1 = aerosol;
  - 2 = heavy gas;
  - 3 = class of drops;
  - <0 = ignored (→ gas).
- DENS: density of the effluent (Kg m^{-3}):
  - gas density for gases;
  - density of dry fraction for particulates;

  - density of liquid fraction for aerosols;
  - < 0: ignored.
- VDEP: species dry deposition velocity (m s^{-1});
- RWASH: washout coefficient (sec^{-1});
- TDECAY: decay time (half-life; sec); when < 0: ignored; when IL = 1: ignored.

Rules for PEMPSE files:

- IPs are ascending integers.
- Any number of comment lines can be added provided they all start with '*'.
- Only those parameters that have a significant impact on transport and diffusion of the pollutants are defined here. Others properties like conductivity, viscosity, dose conversion factors e.t.c. can be defined in an external file and linked by using the IL and IND flags (see example of flag in PEMRAD file).

A.3. STACKS DEFAULT PARAMETERS FILE (GSPDEF.TXT)

Used by INEMOS. It is an ASCII file delimited by semicolon (CSV) specifying, by emission category, the set of default parameters for point sources. The file has a header, containing the fields names, followed by data records.
An example of such file is given below.

```
IDDEF;IDCLA1;IDCLA2;IDCLA3;IDCLA4;IDCLA5;HEIGHT;DIAMET;TEMPER;SPEED
1;0;0;0;0;0;167.08;5.78;416;22.64
2;1;1;0;0;0;167.08;5.78;416;22.64
3;1;4;6;0;0;15;1;353;5
4;3;1;0;0;0;80;3.7;413;6
5;4;4;7;0;0;20;1.2;343;6.5
```

The meaning of data fields is:

- IDDEF: numerical code of the parameters set
- IDCLA1: 1st level code of the emission category
- IDCLA2: 2nd level code of the emission category
- IDCLA3: 3rd level code of the emission category
- IDCLA4: 4th level code of the emission category
- IDCLA5: 5th level code of the emission category
- HEIGHT: default stack height (m above terrain level)
- DIAMET: default stack diameter (m)
- TEMPER: default gas exit temperature (K)
- SPEED: default exit velocity (m s^{-1})

A.4. NEW STACKS DEFAULT PARAMETERS FILE (GSPDEF.TXT)

It is an ASCII file delimited by semicolon (CSV) specifying, by emission category, the validity range of stack parameters and the set of default parameters. It is used by TRLGSP to check the input data and fill the missing values with default values, and by TRLCRS, PEM2GRID and MODGRID to assign default source heights. In case of MODGRID, two versions of the file can be used: one specifying for each category the min and max height of the emission layer, and one specifying a full arbitrary vertical profile.
The file has a header, containing the fields names, followed by data records.
An example of such file is given below.

```
IDCLA1;IDCLA2;IDCLA3;IDCLA4;IDCLA5;HEIGHT_MIN;HEIGHT_MAX;HEIGHT;DIAMET_MIN;DIAME
T_MAX;DIAMET;TEMPER_MIN;TEMPER_MAX;TEMPER;SPEED_MIN;SPEED_MAX;SPEED;Z1;Z2;Z3;Z4;
Z5;Z6;Z7;W1;W2;W3;W4;W5;W6;W7
1;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;0;0;0.25;51;45.3;
3.25;0.2
2;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;100;0;0;0;0;0;0
3;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;6;16;75;3;0;0;0
4;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;5;15;70;10;0;0;0
5;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;2;8;60;30;0;0;0
6;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;100;0;0;0;0;0;0
7;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;100;0;0;0;0;0;0
8;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;0;100;0;0;0;0;0
9;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;0;0;41;57;2;0;0
10;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;100;0;0;0;0;0;0
11;0;0;0;0;-
9;350;60;0.1;8;2;273;600;400;0;40;5;20;92;184;324;522;781;1106;100;0;0;0;0;0;0
```

The meaning of data fields is:

- IDCLA1: 1st level code of the emission category
- IDCLA2: 2nd level code of the emission category
- IDCLA3: 3rd level code of the emission category
- IDCLA4: 4th level code of the emission category
- IDCLA5: 5th level code of the emission category
- HEIGHT_MIN, HEIGHT_MAX: validity range for stack height (m above terrain level)
- HEIGHT: default stack height (m above terrain level)
- DIAMET_MIN, DIAMET_MAX: validity range for stack diameter (m)
- DIAMET: default stack diameter (m)
- TEMPER_MIN, TEMPER_MAX: validity range for gas exit temperature (K)
- TEMPER: default gas exit temperature (K)

- SPEED_MIN, SPEED_MAX: validity range for exit velocity (m s^{-1})
- SPEED: default exit velocity (m s^{-1})
- Z1 … Zn: (optional – only used by MODGRID) emission layers top heights (m above orography)
- W1 … Wn: (optional – only used by MODGRID) percentage of mass emitted in each layer; for a given record, the sum of all percentages can be lower than 100: in such cases, the complement of the sum to 100 corresponds to the percentage of the mass that is emitted above the target grid, and so it will not be included in the output gridded emissions

Note: when (Z1…Zn) and (W1…Wn) are not present, MODGRID vertically distributes the emissions according to the layers defined through HEIGHT_MIN and HEIGHT_MAX; otherwise, when (Z1…Zn) and (W1…Wn) profiles parameters are present, MODGRID vertically distributes the emissions according to such defined profiles, disregarding HEIGHT_MIN and HEIGHT_MAX.

A.5. INEMOS INITIALIZATION FILE

It is an ASCII file used to configure INEMOS: how to interpret the files coming from the emissions monitoring systems and which operation to do on them.
An example of such file is given below, with in-line comments (comment lines are marked by an exclamation mark in the first column).

```
!=============================================================================
! INEMOS 1.0 (INgestion of Emissions MOnitoring System data)
! intialization file
!=============================================================================
!
! EMS input file(s) format flag
 1
!
!-----------------------------------------------------------------------------
! Control of missing/invalid data
!-----------------------------------------------------------------------------
!
! Number (-1: use name; 0: no column) and name of column
! containing overall plant status flags
 -1, Plant status
!
! list of values (comma delimited) marking plant stopped/out of order
! in plant status column (list can be empty)
 *, AC, SP, MN, FSF, FSG, MN
!
! list of values (comma delimited) marking missing/invalid data
! in emissions and stack exit parameters validity columns
! (list can be empty)

!
! list of values (comma delimited) marking missing/invalid data
! in emissions and stack exit parameters data columns
! (list can be empty)
 N.A., Tar., F.S.
!
!-----------------------------------------------------------------------------
! EMS data to be processed
!-----------------------------------------------------------------------------
```

```
!
! # of emitted species in EMS files to be processed
 2
!
! Species to be processed; each row (comma delimited):
!   species name,
!   number (-1: use name) and name of column containing data values,
!   number (-1: use name; 0: no column) and name of column
!     containing data validity flags
!
CO, -1, CO instrumental, 0, null
NOX, -1, NOx instrumental (as NO2), 0, null
!
! Source data assignment mode:
!   1 = species emissions in [kg/h] and exit speed in [m/s]
!   2 = species emissions in [mg/Nm3] + dry volumetric flow [Nm3/h]
!   3 = as 2, but with non-normalized volumetric flow [m3/h]
!   4 = as 2, but with correction of humid volumetric flow
!         with O2 [%] and H2O [%]
!   5 = as 4, but with non-normalized volumetric flow [m3/h]
 2
!
! Stack exit parameters be processed; each row (comma delimited):
!   parameter usage flag (0 = no; 1 = yes),
!   number (-1: use name) and name of stack exit parameter column,
!   number (-1: use name) and name of column containing
!     data validity flags,
!   A and B parameters of units conversion formula y = A * x + B
!
1, -1, Smoke temperature, 0, null, 1., 273.   ! exit temperature [K]
0, 0, null, 0, null, 1., 0.                   ! exit speed [m/s]
1, -1, Dry smoke flow, 0, null, 1., 0.        ! volumetric flow [Nm3/h]
0, 0, null, 0, null, 1., 0.                   ! humid O2 [%]
0, 0, null, 0, null, 1., 0.                   ! reference O2 [%]
0, 0, null, 0, null, 1., 0.                   ! H2O (humidity) [%]
!
! Species emissions conversion factor:
! from user units to: [kg/h] (mode 1),
! or [mg/Nm3] (mode 2,4), or [mg/m3] (mode 3,5)
  1.
!
!-------------------------------------------------------------------------
! Treatment of missing/invalid data
!-------------------------------------------------------------------------
!
! Flag for treatment of missing/invalid data
!   0 = put zero
!   1 = persistence (up to a given time interval)
!   2 = interpolation (up to a given time interval)
!   3 = use of default value in " client file"
!
 2     ! treatment of species emissions
 3     ! treatment of stack exit parameters
!
! Maximum time interval (hr) for missing/invalid data filling
! using persistence/interpolation
 3
!
! Name of pivot species
NOX
!
! Echo of processed values (0 = no, 1 = yes)
 1
```

Assumptions and conventions:

- column names can contain blanks
- column names and species names are case-insensitive

A.6. EMISSIONS MODIFICATION FACTORS FILE

Used by POLMOD, it is an ASCII file delimited by semicolon (CSV) specifying, by emission category and species, the linear transformation parameters to be used to modify emission values. The file has a header, containing the fields names, followed by data records.
An example of such file is given below.

```
ID_CLA1;ID_CLA2;ID_CLA3;ID_CLA4;ID_CLA5;NAME_PROFIL;POLLUANT;A;B
1;2;2;0;0;CSNAP_1;CO;2;0.0
1;2;2;0;0;CSNAP_1;NOx;2;0.0
2;3;4;0;0;CSNAP_1;CO;0.0;14.0
2;3;4;0;0;CSNAP_1;NOx;0.0;657.
11;0;0;0;0;CSNAP_9;CO;4.0;7.0E+13
11;0;0;0;0;CSNAP_9;NOx;3.0;68.0E+19
11;0;0;0;0;CSNAP_9;COV;2.0;0.0
11;2;0;0;0;CSNAP_9;CH4;1;0.0
11;15;15;0;0;CSNAP_TST;CO;2.5;0.0
11;15;15;0;0;CSNAP_TST;COV;5.3;7.0
0;0;0;0;0;CSNAP_1;SO2;0.5;0.
0;0;0;0;0;CSNAP_1;PA;0.25;0.
```

The file is made up of 9 fields separated by semi-colon ";":

- ID_CLA1: 1st level code of the emission category;
- ID_CLA2: 2nd level code of the emission category;
- ID_CLA3: 3rd level code of the emission category;
- ID_CLA4: 4th level code of the emission category;
- ID_CLA5: 5th level code of the emission category;
- NOM_PROFIL: mnemonic name given to the profile;
- POLLUANT: species name of the pollutant to be altered;
- A: factor A of "output = A * input + B" formula;
- B: factor B of "output = A * input + B" formula.

It should be noticed that when IDCLA1 to IDCLA5 = 0, emission from all categories of the specified pollutant will be modified according to Ax + B. Also, the pollutants defined in this way should not be occuring elsewhere in the file. The categories 0 0 0 0 0 must be defined at the end of the file.

In the example above:

- The emissions of NOx and CO for SNAP code "1 2 2" are multiplied by 2
- A constant value is added to the emissions of NOx and CO for SNAP code "2 3 4"
- All emissions of SO2 and PA are divided by a factor (2 for SO2 and 4 for PA).

ANNEX B – EXAMPLE BATCH FILES TO RUN EMGR MODULES

Examples of DOS/Windows batch files running Emission Manager modules are given here.

A line starting with # is a comment and is not taken into account in the execution of the file. At the end of the line a \ should be inserted except after the last command line.

A line starting with « echo » will show the text indicated on the screen when running the batch file but it will not influence the execution of the program.

TRLGSP

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#
#  Starts from an LPS client file and yields 2 output files
#  of format "pemtim" and "pemspe". The emissions covers a
#  period of 24 hours and doesn't take any time variation
#  of the emissions into account.
# ========================================================
echo "-------------------------------------------------"
echo "    Lancement trlgsp   "
echo "-------------------------------------------------"
trlgsp \
/SYMPASIM/JUN95/CLIENTS/gsp.csv \
/SYMPASIM/SNAP/snap_4.motobad \
/SYMPASIM/JUN95/CLIENTS/gspdef.new \
/SYMPASIM/JUN95/EMISSIONS/GSP/"pemtim".ini \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.ini \
1 \
542. 4970. 52 52 4. \
1995/06/26 \
00/00/00   \
1995/06/26 \
24/00/00   \
24/00/00   \
1 /SYMPASIM/JUN95/EMISSIONS/GSP/bilan \
0
```

TRLCRS

```
# ========================================================
cd /ADSODEV/EXEC43P
# ========================================================
#  Starts from the 2 Area source client files and yields 2 output
#  files of format "pemtim" and "pemspe". The emissions covers a
#  period of 24 hours and doesn't take any time variation of the
#  emissions into account.
# ========================================================
echo "-------------------------------------------------"
echo "    Lancement trlcrs   "
echo "-------------------------------------------------"
trlcrs \
/AIRPARIF/CRS/PARIS/CLIENTS/doncar.srf \
/AIRPARIF/CRS/PARIS/CLIENTS/sf_ok.txt \
/AIRPASIM/SNAP/snap_4 \
/AIRPARIF/CRS/PARIS/EMISSION/"pemtim".crs \
/AIRPARIF/CRS/PARIS/EMISSION/pemspe.crs \
/AIRPARIF/CRS/PARIS/CLIENTS/crs.txt \
/AIRPARIF/CRS/PARIS/CLIENTS/test.ldu \
1 0 \
585. 2414. \
31 31 1. \
1995/11/06 0/0/0 \
1995/11/06 24/0/0 \
24/0/0
```

TRLLIN

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#  Starts from an Line source client file and yields 2 output
#  files of format "pemtim" and "pemspe". The emissions covers a
#  period of 24 hours and doesn't take any time variation
#  of the emissions into account.
echo "-------------------------------------------------"
echo "    Lancement trllin   "
echo "-------------------------------------------------"
trllin \
/WORKPAN/ST_ETIENNE/EMISSION/emis_lin_p1.csv \
/WORKPAN/ST_ETIENNE/EMISSION/"pemtim"_lin_P1 \
/WORKPAN/ST_ETIENNE/EMISSION/pemspe_lin \
1995/01/01 19/0/0 \
1995/01/02 7/0/0 \
12/0/0 \
1 /WORKPAN/ST_ETIENNE/EMISSION/bilan_lin_P1 \
0
```

CRTEMP

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a « ”pemtim” » and “pemspe” file and applies
#   the coefficients as defined in the time modulation files
#   on the emissions in the “pemtim” file.
echo "-------------------------------------------------"
echo "    Lancement crtemp   "
echo "-------------------------------------------------"
crtemp \
/SYMPASIM/JUN95/EMISSIONS/SRF/”pemtim”.ini \
/SYMPASIM/JUN95/EMISSIONS/SRF/pemspe.ini \
/SYMPASIM/JUN95/CRTEMP/prf_par_gsp.csv \
/SYMPASIM/JUN95/CRTEMP/prf_dxt.csv \
/SYMPASIM/JUN95/CRTEMP/crt_m.csv \
/SYMPASIM/JUN95/CRTEMP/crt_j.csv \
/SYMPASIM/JUN95/CRTEMP/crt_hloc.csv \
/SYMPASIM/SNAP/snap_4.motobad \
0 0 1 1 \
95/06/26 \
1 /SYMPASIM/JUN95/EMISSIONS/SRF/”pemtim”.crt \
```

LIGREC

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a line source « ”pemtim” » file and distributes the
#   emissions on a grid net in order to create rectangular emissions.
#
echo "-------------------------------------------------"
echo "    Lancement ligrec   "
echo "-------------------------------------------------"
ligrec \
/SYMPASIM/JUN95/EMISSIONS/LIN/pemspe.traf \
/SYMPASIM/JUN95/EMISSIONS/LIN/”pemtim”00.txt \
/SYMPASIM/SNAP/snap_4.motobad \
542. 4970. 52 52 4000. \
/SYMPASIM/JUN95/EMISSIONS/LIN/”pemtim”00.rec \
1
```

CATPER

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from N "pemtim" files, all with the same number of sources
#   but with emissions covering different time periods. The N different
#   files are merged into one file where each source has got N time
#   intervals.
echo "-------------------------------------------------"
echo "    Lancement catper   "
echo "-------------------------------------------------"
catper \
/SYMPASIM/JUN95/EMISSIONS/LIN/pemspe.traf \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim".rec \
0 \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"00.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"01.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"02.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"03.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"04.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"05.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"06.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"07.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"08.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"09.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"10.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"11.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"12.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"13.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"14.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"15.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"16.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"17.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"18.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"19.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"20.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"21.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"22.rec \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim"23.rec
```

CATSOU

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from N « "pemtim" » files, each with an arbitrary number
#   of sources but starting at the same time. The files are merged
#   into one «"pemtim" » file containing the total of the sources
#   in the N input files
echo "-------------------------------------------------"
echo "    Lancement catsou   "
echo "-------------------------------------------------"
catsou \
/SYMPASIM/JUN95/EMISSIONS/pemspe.all \
/SYMPASIM/JUN95/EMISSIONS/"pemtim".all \
0 \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.red \
/SYMPASIM/JUN95/EMISSIONS/GSP/"pemtim".crt \
/SYMPASIM/JUN95/EMISSIONS/LIN/pemspe.red \
/SYMPASIM/JUN95/EMISSIONS/LIN/"pemtim".red \
/SYMPASIM/JUN95/EMISSIONS/SRF/pemspe.ini \
/SYMPASIM/JUN95/EMISSIONS/SRF/"pemtim".crt
```

ARQRAS

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a « ”pemtim” » and “pemspe” file, and distributes the
#   emissions on a grid net to create as many LPS and/or SRF files
#   as SNAP categories.
echo "    Lancement arqras   "
echo "-----------------------------------------------"
arqras \
/SYMPASIM/JUN95/EMISSIONS/GSP/”pemtim”.crt \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.red \
/ADSO/BLOCS/LYON2/ARCHIV/reluambis.wrk \
/SYMPASIM/JUN95/EMISSIONS/AIRQUAL/GSP/ \
1 \
/SYMPASIM/JUN95/EMISSIONS/AIRQUAL/GSP/rqliste.txt \
0 1 \
/SYMPASIM/JUN95/EMISSIONS/AIRQUAL/GSP/Liste_classe.txt
```

REDSPE

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a « ”pemtim” » and “pemspe” file having a number of
#   substances A and creates a new « ”pemtim” » having a number of
#   substances B where B < A
echo "-----------------------------------------------"
echo "    Lancement  redspe "
echo "-----------------------------------------------"
redspe \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.red \
/SYMPASIM/JUN95/EMISSIONS/GSP/”pemtim”.red \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.ini \
/SYMPASIM/JUN95/EMISSIONS/GSP/”pemtim”.ini
```

BILSOU

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a « ”pemtim” » and “pemspe” file and creates an
#   ASCII file giving the summary of emissions by substance and by
#   SNAP category.
echo "-----------------------------------------------"
echo "    Lancement  bilsou "
echo "-----------------------------------------------"
bilsou \
/SYMPASIM/JUN95/EMISSIONS/GSP/pemspe.red \
/SYMPASIM/JUN95/EMISSIONS/GSP/”pemtim”.red \
/SYMPASIM/SNAP/snap_4.motobad \
/SYMPASIM/JUN95/EMISSIONS/GSP/bilan.ini
```

PEMSOM

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#  Starts from a « ”pemtim” » and “pemspe” file including SNAP
#  classification and creates a new « ”pemtim” » file with the sum of
#  the emissions for each source and by deleting the SNAP classification
echo "-------------------------------------------------"
echo "    Lancement pemsom   "
echo "-------------------------------------------------"
pemsom \
/SYMPASIM/JUN95/EMISSIONS/”pemtim”.all \
/SYMPASIM/JUN95/EMISSIONS/pemspe.all \
/SYMPASIM/JUN95/EMISSIONS/”pemtim”.som
```

BINRAS

```
# ========================================================
cd /ADSO/EXECI64
# ========================================================
#   Starts from a « ”pemtim” » and “pemspe” file, distributes the
#   emissions on a grid net and creates a BIN file.
echo "-------------------------------------------------"
echo "    Lancement binras   "
echo "-------------------------------------------------"
binras \
/SYMPASIM/JUN95/EMISSIONS/”pemtim”.som \
/SYMPASIM/JUN95/EMISSIONS/pemspe.all \
/ADSO/BLOCS/LYON2/ARCHIV/reluambis.wrk \
1 \
15 15000. 7 40. \
/SYMPASIM/JUN95/EMISSIONS/UAM/BIN/emis \
1 \
/SYMPASIM/JUN95/EMISSIONS/LIN/BIN/rqliste.txt \
0 0 01/00/00
```

PEMMOD

```
# ========================================================
cd /ADSO/EXEC43P
# ========================================================
#   Starts from an ASCII « ”pemtim” » and “pemspe” file and transforms the
#   « ”pemtim” » to binary format or vice versa.
echo "-------------------------------------------------"
echo "    Lancement pemmod   "                          "
echo "-------------------------------------------------"
 pemmod \
/SYMPASIM/MAI98/EMISSIONS/SRF/”pemtim”.ini \
/SYMPASIM/MAI98/EMISSIONS/SRF/pemspe.ini \
1 \
/SYMPASIM/MAI98/EMISSIONS/SRF/”pemtim”.ini.bin \
0 0
```

POLMOD

```
# ============================================================
cd /ADSO/EXEC43P
# ============================================================
#   Starts from an « ”pemtim” » and “pemspe” file and alters the
#   emissions as a function of substance and SNAP category
echo "-------------------------------------------------"
echo "        Lancement polmod                          "
echo "-------------------------------------------------"
 polmod \
/SYMPASIM/MAI98/EMISSIONS/SRF/”pemtim”.crt \
/SYMPASIM/MAI98/EMISSIONS/SRF/pemspe.crt \
/SYMPASIM/MAI98/EMISSIONS/snap_modul.csv \
0 2 \
/SYMPASIM/MAI98/EMISSIONS/SRF/”pemtim”.mod \
/SYMPASIM/MAI98/EMISSIONS/SRF/bilan.mod
```