

DRAFT-REPORT

*TNO-report*

## **LOTOS-EUROS: User Manual**

TNO Built Environment and Geosciences

Date

Laan van Westenenk 501  
P.O. Box 342  
7300 AH Apeldoorn  
The Netherlands

T +31 55 549 34 93  
F +31 55 541 98 37  
www.tno.nl

Author(s)

**M. Schaap, G.A.C. Boersen, R.M.A. Timmermans, F.  
Sauter, Twan van Noie, P.J.H. Bultjes**

Order no.

Keywords

**LOTOS-EUROS, manual**

This draft report is supplied for internal use only and the results may not be made known to third parties as having been produced by TNO.

In this respect attention is drawn to article 8 (liability) and article 4 (rights concerning results) of the Standard Conditions for Research instructions given to TNO.

Intended for

**RIVM-MNP**

All rights reserved.

No part of this publication may be reproduced and/or published by print, photoprint, microfilm or any other means without the previous written consent of TNO.

In case this report was drafted on instructions, the rights and obligations of contracting parties are subject to either the Standard Conditions for Research Instructions given to TNO, or the relevant agreement concluded between the contracting parties.

Submitting the report for inspection to parties who have a direct interest is permitted.

© 2001 TNO



---

<b>1.</b>	<b>Introduction .....</b>	<b>5</b>
<b>2.</b>	<b>Modify and building a new model.....</b>	<b>Error! Bookmark not defined.</b>
2.1	Modify the source code	
2.2	makefile	
2.3	compiler options	
2.4	platform dependencies	
2.5	remarks .....	8
<b>3.</b>	<b>Perform a simulation.....</b>	<b>11</b>
3.1	The run control file .....	11
3.1.1	Run set-up.....	11
3.1.2	Input set-up .....	15
3.1.3	Output set-up .....	17
3.1.4	Loose ends .....	18
<b>4.</b>	<b>Input files.....</b>	<b>Error! Bookmark not defined.</b>
4.1	Land use.....	<b>Error! Bookmark not defined.</b>
4.2	Emissions.....	26
4.3	Meteorology	
4.4	Boundary conditions	
4.5	time-profiles and voc split table	
4.6	scenario handling.....	<b>Error! Bookmark not defined.</b>
<b>5.</b>	<b>Output and GrADS .....</b>	<b>27</b>
5.1	Output files .....	27
5.2	GrADS.....	28
5.3	Naverbrander .....	<b>Error! Bookmark not defined.</b>
<b>6.</b>	<b>Final remarks.....</b>	<b>30</b>



## 1. Introduction

At TNO, MNP and RIVM the air quality model LOTOS-EUROS is developed and used. The LOTOS-EUROS model is an eulerian 3D chemistry transport model for Europe. The model is described in detail in its documentation (Schaap et al., 2005). Nowadays, the model is to be used by a larger user community, of which not everyone is a modeling specialist. To support the use of the model a user manual has been written. The manual aims to provide the necessary information to independently install and run the model in its standard configurations. In addition, the steps needed to recompile the source code are described.

### *Applicability of the model*

The application of the model within its European domain can be performed in different configurations (components, resolution etc). The following components can be simulated with the model:

- Oxidants
- Secondary inorganic aerosol
- Secondary organic aerosol
- Primary aerosol
- Sea salt and dust aerosol
- Base cat-ions
- Heavy metals
- POP's

Application of the model is limited to Europe (see Figure 2.1). The boundaries of the master domain are 35 and 70 North and 10 West and 60 East. The projection is normal longitude-latitude and the standard grid resolution is 0.50 longitude x 0.25 latitude, approximately 25x25 km. By means of a control file the actual domain and resolution for a simulation can be set as long as it falls within the master domain as specified above.

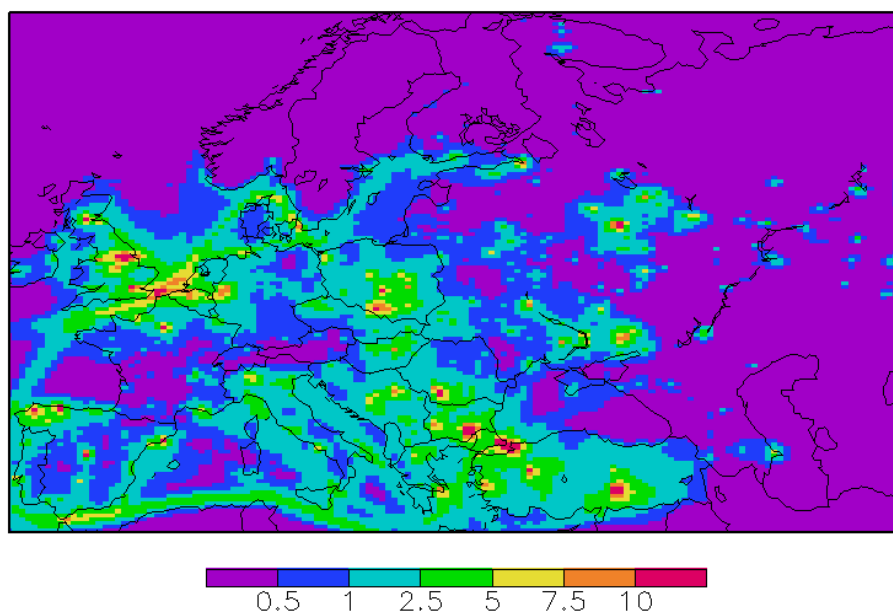


Figure 1.2 The domain of the LOTOS-EUROS modelling system. The example shows the average sulphur dioxide concentration ( $\mu\text{g}/\text{m}^3$ ) modelled for July, 1997.

#### *The modelling system*

In Figure 2.2 a schematic view of the modelling system is given. The LOTOS-EUROS model is in the core of the system. To perform a simulation one needs to set the available run parameters to specify the exact simulation to perform. Within the steering file one can set the datasets for the three most important inputs to the model: meteorology, land-use and emissions. In practise, the first two are predefined data sets which can be selected and used, but not altered, by the user. The emissions on the other hand are to be changed on a regular basis and the way to do that is described in this manual. Using the input data sets and the run control file the model is able to execute the simulation.

Common output are files with concentration fields and deposition fields. The data can be viewed in GrADS. Further, a post-processor is available to perform a basic analysis on the data.

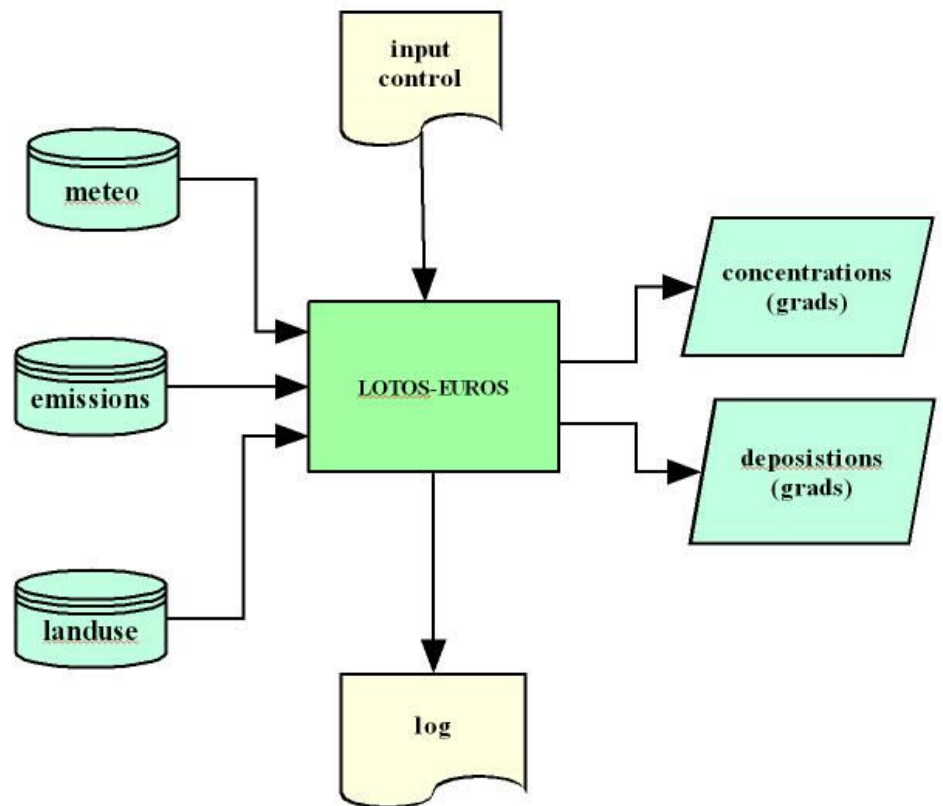


Figure 2.2. A schematic representation of the modelling system. ACTION: rename input control to simulation or run control

In the following chapters is described how to:

1. compile the source code
2. set the run parameters and perform a simulation
3. define the emissions
4. handle the output

## 2. Modify and building a new model

First condition to manage source code upgrades is knowledge about programming and structuring computer models. Maintaining the LE model it will be necessary to modify sources and even model frames by adding/deleting procedures and/or functions. To change the LE FORTRAN 90 sources a text editor must be available (for instance VI editor) and last but not least a Fortran 90 compiler and linker. The most important fact is knowledge about the LE model structure laid down in the makefile (look makefile description). Changing source codes implies rebuilding the LE model by running the makefile.

It is important to be aware about the consequences of modifying and running the makefile.

Before starting a new compilation it is wise to run first the makefile option to clean the old modules and object files by typing :

[Make clean](#)

The next step to recompile the modified code can be done by typing :

[Make model](#)

The LE model will be compiled and if no errors are detected a new executable will be build. If any error occurs during compilation, correct the source code and rerun both above commands. Each time the makefile is run successfully a new executable will be build and overwrite the existing model.

### 2.1 Makefile

The makefile is an ascii script describing the compiler how to compile the sources and to invoke the linker subsequently how to (re)build a new executable of the model.

In general the makefile contains all relevant commands to produce new executable code regarding all specific compiler options and assembling conditions.

It is recommended to read the compiler guide attentive in order to handle these options in a smart way regarding the compatibility and conformance to the standards. Watch the default setting of global data types on different platforms.

Special point of interest is the use off the debug options during development phase. The use of the debugger is a handy feature to inspect step by step the behaviour of your code during runtime. To use the debugger a few compiler options must be passed to the makefile.

Don `t forget to use the makefile clean option each time a new executable code must be made.

#### *Compiler options*

The user guide of each FORTRAN compiler describes in extenso all compiler options in conjunction with the assembler and linker system tools. Compilers may have additional internal components like

an optimizer, debugger or profiler. The use of these tools depends on specific user requirements and behaviour of the runtime model. Notice in this case the use of special user (dynamic link) libraries.

## 2.2 Platform dependencies

In general the LE model will run on several different platforms like Linux, Unix and Windows.

Usually the LE model should be compiled, linked and run on one and the same platform in order to be sure the executable will run under system control like specific system libraries and functions.

Each platform has its own specific system characteristics that are important to determine your system is suitable to run the LE model.

In the matrix list below the minimum platform requirements are shown followed by a recommended quantity.

	minimum	recommended
Unix	HPUx 9.2	
Linux	RED Hat	
Windows	Xp	
CPU	1.2 GHz	> 2.5 GHz
Hard disk storage	20 GB	> 100 GB
RAM memory	2 G	3 G
printer	Standard A4	Colour A4
Swap area	2G	4G

Some additional remarks to this matrix.

The platform may support or can be part of a LAN (Local Area Network) structure. In that case it is useful to look-ahead to the consecutive network load because the LE model is highly i/o intensive. The most recommended environment is to run the LE model in a stand alone mode. This means only one run at a time. In practice it is not opportune to run in a multi task mode while the system has no dual processor.

The minimum hard disk capacity is based on the storage requirement of a single model run of one year ( 8760 hours), 10 pollutants, 2 layer output and a domain size of 140 x 140 grid cells ( scale 0.5 x 0.25). In this case the size of one floating point number is 4 bytes.

Regarding the 4 byte floating point number there is an important phenomenon to pay attention on.

This phenomenon is called big- and little endian storage. The adjectives big-endian and little-endian refer to which [bytes](#) are most significant in multi-byte [data types](#) and describe the order in which a sequence of bytes is stored in a computer's memory. Many [mainframe](#) computers, particularly IBM mainframes, use a big-endian architecture. Most modern computers, including [PCs](#), use the little-endian system. The [PowerPC](#) system is bi-endian because it can understand both systems. Converting data between the two systems is sometimes referred to as the NUXI prob-

lem. Imagine the word UNIX stored in two 2-byte words. In a Big-Endian systems, it would be stored as UNIX. In a little-endian system, it would be stored as NUXI. Using Grads it may be necessary to use the optional parameter ‘byteswapped’ in order to handle the different endian structures on different platforms.

### 3. Perform a simulation

Once the code has been compiled, the executable will be available in the run directory. To start a simulation one has to execute the executable and supply the code with a control file. In short, you can run the model typing:

```
./lotos-euros.exe
```

The first thing the model does is to prompt you for a control file. In our case, the control file 'simulation.rc' has been made and we give this in.

```
simulation.rc
```

The control file is described in detail in the next section. Now the model will run until the end of the simulation is reached.

#### 3.1 The control file

The control file gives the user the opportunity to control a number of parameters from outside model. Hence, you don't have to go into the code to run a standard simulation. The control file can be divided into several parts:

```
Run set-up  
Input set-up  
Output set-up  
Loose ends
```

Below we discuss all the options in the control file. In the file a header line in capital letters describes the options. The option is set at the line or lines below the header. It is important to keep in mind that the code checks the header precisely. Changes in the header or small mistakes in the setting of the options are being checked for in this way.

##### 3.1.1 Run set-up

###### *Project and simulation identification*

The first part of the control file is used to set the name of the simulation, the species for which the model is ran, the domain size and simulation period.

```
RUN DESCRIPTION  
'test'
```

Here the description of the simulation can be entered. It will be used in the CTL file for grads.

```
PROJECT FOR WHICH TO RUN WILL BE DONE (max 10 characters)
v1
RUN ID (max 8 characters)
test
```

With these names the output structure is made. The model will create the directory <output\_DIR>/v1/test

### *Component selection*

LOTOS-EUROS describes the distribution of oxidants, aerosols and POP's over Europe. The simulations for these components are often coupled but this is not always the case. For example, one may be interested in ozone but not in aerosols. Therefore, LOTOS-EUROS has the ability to perform simulations in different set-ups. The following options are available (yes/no):

#### Oxidants

To calculate ozone and other oxidant levels over Europe a gas phase chemistry scheme must be chosen. LOTOS-EUROS includes the condensed CBM-IV mechanism from LOTOS and the CB99 mechanism from EUROS. These schemes describe photochemistry using 29 or 40 tracers, respectively. The only aerosol species calculated in these schemes is sulphate.

#### Secondary inorganic aerosol

The option to calculate SIA invokes a call to the aerosol equilibrium module, which describes the equilibrium between ammonium nitrate and its gaseous counterparts, ammonia and nitric acid. SIA calculations can only be performed in combination with the full oxidant scheme.

#### Secondary organic aerosol

This option invokes a call to the aerosol equilibrium module, which describes the formation of secondary organic aerosol (SOA). SOA calculations can only be performed in combination with the full oxidant scheme.

#### Primary aerosol

This option enables to switch on/off the calculations for primary aerosol components. At the moment, the primary components include primary PM<sub>2.5</sub>, PM<sub>10-2.5</sub>, Black Carbon (BC) and coarse and fine mode sea salt. The calculations for the primary components can be performed stand alone.

#### Sea salt and dust aerosol

This option enables to switch on/off the calculations for coarse and fine mode sea salt and dust. The calculations for the primary components can be performed stand alone.

#### Base cat-ions

This option enables to switch on/off the calculations for base cat-ions. At the moment, they include primary Na, Ca and Mg. The calculations for these components have to be performed in combination with the sea salt and dust calculations.

### Heavy metals

This option enables to switch on/off the calculations for heavy metal components, Cd and Pb, which can be performed stand alone.

### Sulphur-only

The sulphur-only option performs a simulation for SO<sub>2</sub> and SO<sub>4</sub> using predefined OH radical concentrations. Hence, the simulation comprises only 2 tracers and is very fast. The sulphur only option can not be performed together with oxidant calculations as it does not make any sense.

### POP's

LOTOS-EUROS also contains a module to perform calculations for PAH's and POP's. Simulations for POPs can not be combined with other components.

#### *Select meteorological data*

Set the Meteorological input to one of the following options:

METEO DATA (fub or ecmwf)

xxx

Fub: diagnostic meteorology of the Free University of Berlin

Ecmwf: Meteorological data from ECMWF

### Domain control

The domain size and resolution are controlled by four input parameters. First of all, it must be indicated if the zoom options will be used. If this is the case than the zooming factor has to be set. A factor of 2 2 will result in a simulation with twice the normal horizontal resolution: 0.25x0.125 in stead of 0.5x0.25. Presently, a factor of 8 is the maximum setting. Furthermore, it is possible to run the model at a coarse resolution of 1.0x0.5 (zoom-out) by indicating -2 -2.

GRID RESOLUTION (zoom or normal)

xxxx

ZOOMING FACTOR X AND Y direction

x y

COORDINATES OF LOWER LEFT CORNER

Xxx yy

NX, NY, NZ

x y z

The selected domain is defined by the location of the lower left corner (i.e. in the southwest) and the number of cells in the x and y direction. Note that the Nx and Ny are presently defined using the normal grid resolution of 0.5x0.25 degrees.

For example, a simulation over the Netherlands with a horizontal resolution of 0.25x0.125 may look like this:

---

```
GRID RESOLUTION (zoom or normal)
zoom
ZOOMING FACTOR X AND Y direction
2 2
COORDINATES OF LOWER LEFT CORNER
0 49
NX, NY, NZ
20 20 4
```

This domain ranges from 0-10 lon and 49-54 lat. During the simulation the nx and ny are defined as the zooming factor multiplied by nx (and ny), which gives 40x40 for this simulation.

*Start and end of simulation*

The starting data and ending date are inputted as given below. Note that the model has the hour 0-23 and does not have/recognise hour 24.

```
START DATE OF SIMULATION
yyy,m,d,h
END DATE OF SIMULATION
yyy,m,d,h
```

*Start from a saved model state*

The model has the option to save the model state at midnight every day. These files can be used as restart files to e.g. restart simulations after a power down period. Not using this options means that the model starts by interpolating the boundary conditions.

```
START FROM RESTART FILE REQUIRED
no
```

Using the restart files means that the options has to be set to yes and that the directory that contains the restart files should be given in the input specification part of the control file:

```
START FROM RESTART FILE REQUIRED
Yes
```

```
LOCATION OF RESTART FILE
<Directory in which the files are located>
```

---

```
RUN DESCRIPTION
'test'
PROJECT FOR WHICH TO RUN WILL BE DONE (max 10 characters)
v1
RUN ID (max 8 characters)
test
CHEMISTRY MODE (CBM4 or CB99 or none)
CBM4
SEC. INORG. AEROSOL COMPUTATIONS NEEDED?
yes
PRIM. AEROSOL COMPUTATIONS NEEDED?
no
SEA SALT AND DUST COMPUTATIONS NEEDED?
yes
BASE CATION COMPUTATIONS NEEDED?
no
SEC. ORG. AEROSOL COMPUTATIONS NEEDED?
no
POPS COMPUTATIONS NEEDED?
no
HM COMPUTATIONS NEEDED?
no
SULPHUR ONLY COMPUTATION?
no
METEO DATA (fub or ecmwf)
fub
GRID RESOLUTION (zoom or normal)
zoom
ZOOMING FACTOR X AND Y direction
-2 -2
COORDINATES OF LOWER LEFT CORNER
-10.0 35.0
NX, NY, NZ
100 140 4
START DATE OF SIMULATION
2000,1,1,0
END DATE OF SIMULATION
2000,12,31,23
START FROM RESTART FILE REQUIRED
no
USE OF DEPAC REQUIRED (deposition module)
Yes
```

---

Figure x. First part of control file

### 3.1.2 Input set-up

In this part of the set-up the input of the model is defined. It means that all the paths to the emissions, land use, boundary conditions, standard input and (for convenience) also the output are to be set here. Further, the files containing the land use and the emissions for a number of components have to be set here.

---

```
LOCATION OF METEO
/linuxMA5b/lotos/meteo/
LOCATION OF EMISSIONS
/linuxMA5b/lotos/emissions/201205/
EMISSION FILE for oxidants
base_emis.txt
EMISSION FILE for heavy metals
emis-hm-cd-pb.txt
EMISSION FILE for shipping
shipping_rap.txt
EMISSION FILE for POPs
pop_emis.dat
EMISSION FILE for basecations
emis_basecation.txt
LOCATION OF EMISSION FACTORS
/linuxMA5b/lotos/emissions/201205/
LOCATION OF STANDARD INPUT
/user4/lotos/standard/
LOCATION OF LANDUSE DATA
/linuxMA5b/lotos/landuse/fine/
LAND USE FILE
landuse_russia.txt
LOCATION OF BOUNDARY CONDITIONS
/linuxMA5b/lotos/bound_cond/
TYPE OF BOUNDARY ( 1=tm3, 2=le-run, 3=logan)
3
LOCATION OF OUTPUT
/linuxMA1b/lotos/output/
LOCATION OF RESTART FILE
.
```

---

#### *Boundary conditions*

For the boundary conditions several options exist. For 1997 TM3 boundary conditions (1) are available. Further, climatologically derived data can be used. These include the logan database for ozone and the EMEP approach for other components. Finally, one can use boundary conditions as derived from LOTOS-EUROS itself. For this purpose one has to specify the directory with the data on the line below. The model opens the CTL file of the model output there and searches for the available data and matches this with the domain used. Species for which no modelled BC concentrations are available are treated following the climatological approach. The same applies to the model top. No need to say that the data used for boundary conditions should encompass the full model domain and 3 or 4 layers.

```
TYPE OF BOUNDARY ( 1=tm3, 2=le-run, 3=logan)
2
```

<Directory with the data files>

### 3.1.3 Output set-up

The output of modelled species can be fully specified in the control file. The starting date of the output the output can be set. Further the number of layers and the number of components can be set. For the output layers the valid layers range from 0 to nz, where 0 represents the output at measurement height. For the specification of the output the name of the species must be indicated. These are listed in Annex A. For a number of components a group name can be indicated. For instance, the sea salt tracer sodium (na) can be indicated as model output, which results in both fine and coarse mode sodium (na\_f and na\_c) output. Similarly nmvoc can be specified which will give all nmvoc species as output.

The emission output specifies the emission totals used in the simulation. The output can be arranged by country or snap sector or the combination of both. Station output does not work at the moment and are dummy options.

The switch SUPPRESS OUTPUT controls the output to the screen. Suppressing the screen output gives only a limited output to screen.

With the switch for OH output separate files with the hourly OH fields can be generated. It is these files that are needed in the simulations with only the sulphur cycle included.

---

```
GRADS OUTPUT REQUIRED
yes
START DATE OF THE OUTPUT
2000,1, 1, 0
OUPUT FOR LAYERS (number, followed by layer numbers)
1
0
END OUTPUT FOR LAYERS
OUTPUT FOR SPECIES (if < #species: specify)
4
o3
bc
pb
na
END OUTPUT FOR SPECIES
EMISSIONS OUTPUT (0=none, 1=land, 2=snap, 3=detailed)
2
HOURLY STATION OUTPUT ?
no
DAILY AVERAGE STATION OUTPUT?
no
```

---

```
FILE WITH STATIONS SPECIFICATIONS
emep_selected.txt
SUPPRESS PRINTED OUTPUT?
no
DEBUG OUTPUT?
no
OH OUTPUT NEEDED?
no
NUMBER OF (STRANG SPLIT) STEPS WITHIN ONE HOUR
3
MINIMUM AND MAXIMUM NUMBER OF ITERATION IN THE CHEMISTRY
10 15
```

---

### 3.1.4 Loose ends

Finally, two parameters concerning the numerics can be set in the control file. First the strangs split can be set. The strang split controls the actual time step in the model, i.e. it is the number of time steps within an hour. Within each timestep all the precesses are solved twice: ones in one order and than the other way around, ABCDDCBA.

```
NUMBER OF (STRANG SPLIT) STEPS WITHIN ONE HOUR
3
```

The last option is the minimum and maximum number of iterations in the chemistry. It is advised to keep the minimum number of iterations at 10 or more.

```
MINIMUM AND MAXIMUM NUMBER OF ITERATION IN THE CHEMISTRY
10 15
```

## **4. Emissions**

The emissions are to be specified from outside the model. Hence, a rather elaborate structure is set up to be as flexible as possible.

### **4.1 Definition of emission structure**

The anthropogenic emissions can be fully determined from outside the model code. This means that one has to provide the exact definition of the emission structure (sectors, stackheights, VOC split, etc). Moreover, the system is designed to be able to accommodate a combination of emission structures for different countries. For example, it is possible to define the Dutch emissions based on the LED system, whereas the rest of Europe follows the TNO-SNAP system.

The following steps have to be performed to control the emission input to the model:

1. define the country codes
2. define the emission distributions
3. apportion the emission distributions to be used for each country

For each emission distribution

4. define the emission categories
5. define time profiles (for each category)
6. define VOC-profiles
7. define scenario-profiles
8. define stack heights
9. define black carbon-profiles
10. Specify the emissions itself

Finally, the common format of the files and the parameters used in the code are briefly described.

#### **4.1.1 Controlling the emission distributions used**

##### **4.1.1.1 Definition of country codes**

To the define the countries for which emissions are to be inputted in the model a file is read. The file follows the common structure as given above, with the header, BEGIN statement, core and END statement. In the core it simply defines country codes (a3) and the name of the country.

Note that all other input must use the same countries and in the same order as specified here.

File: country\_code.txt

---

```
# country codes
# date: 2005-11-17
# reference: ...
# ....
# -----
BEGIN country codes
iso3  country
ALB  Albania
ARM  Armenia
AUT  Austria
.....
YUG  Yugoslavia
END country codes
```

---

#### 4.1.1.2 Define the emission distributions

Now introduce the emission distributions used. In the file below we introduce two distributions. We define the TNO-SNAP distribution and a new distribution YYYY.

File: emission\_distributions.txt

---

```
# emission distributions
# Ferd Sauter
# date: 2005-12-19
BEGIN emission distributions
name  comment
SNAP14  'SNAP level 1 + SNAP 71,72,73,74 for road transport'
YYYY    'new definition'
END emission distributions
```

---

#### 4.1.1.3 Apportionment of the emission distributions per country

Now combine the emission distributions used to each country. For this a simple two column file was constructed with the country code and distribution code in the columns. Below we illustrate a file with all countries using the TNO distribution except Armenia (ARM) which uses the YYYY distribution.

File: country\_distribution.txt

---

```
# emission distribution for each country
# Ferd Sauter
# date: 2005-12-19
BEGIN country emission distribution
iso3  emission_distribution
ALB  SNAP14
ARM  YYYY
AUT  SNAP14
.....
YUG  SNAP14
END country emission distribution
```

---

#### **4.1.2 Detailed specification of the emission distributions**

For each emission distribution one has to define a number of features, steps 4-9 in the abovementioned procedure. All file names involved in these steps contain the name of the emission distribution as specified in emission\_distributions.txt.

##### **4.1.2.1 Define the categories**

The categories must be numbered. The category names as defined in the file are used in all other files and should be consistent.

File: emis\_cat\_YYYY.txt

---

```
# emission categories
# Ferd Sauter
# date: 2005-11-29
BEGIN emission categories
code  name
1     Power_generation
2     "Residential,_commercial_and_other_combustion"
3     Industrial_combustion
4     Industrial_processes
5     Extraction_distribution_of_fossil_fuels
6     Solvent_use
7     Road_transport
8     Other_mobile_sources
9     Waste_treatment_and_disposal
10    Agriculture
71    Road_transport_gasoline
```

```

72   Road_transport_diesel
73   Road_transport_lpg
74   Road_transport_evaporation
END emission categories

```

---

#### 4.1.2.2 define time profiles (for each category)

Now generate a file formatted as below with for all categories the factors to be applied for the hour of the day, the day of the week and the month. All factors for each timescale should be on average 1.0 as they are multiplied with the average emission strength (annual total divided by number of hours) in the model.

File: time\_var\_emis\_YYYY.txt

---

```

#   TEMPORAL VARIATION OF EMISSIONS FOR SNAP14
#   Ferd Sauter
#   date: 2005-11-30
#   reference: CORINAIR SNAP CODES 1997
BEGIN monthly time profiles emissions
UNIT [-]
code  category_name      jan feb mar apr may jun jul aug sep oct nov dec
1     'Power_generation'   1.2 1.15 1.05 1 0.9 0.85 .....
2     'Residential,_commercial_and_other_combustion' 1.7 1.5 1.3 1 0.7 .....

74   'Road_transport_evaporation' 0.88 0.92 0.98 1.03 1.05 .....
END monthly time profiles emissions

BEGIN daily time profiles emissions
UNIT [-]
code  category_name mon   tue    wed    thu    fri    sat    sun
1     'Power_generation'  1.06  1.06  1.06  1.06  1.06  1.06  0.85
      0.85
.....
74   'Road_transport_evaporation' 1.02  1.06  1.08  1.1   1.14
      0.81  0.79
END daily time profiles emissions

BEGIN hourly time profiles emissions
UNIT [-]
code  category_name 1      2      3      4      5      6      7      8
      9      10     11     12     13     14     15     16     17     18
      19     20     21     22     23     24

```

1	'Power_generation'	0.79	0.72	0.72	0.71	0.74	0.8		
		0.92	1.08	1.19	1.22	1.21	1.21	1.17	1.15
		1.13	1.1	1.07	1.04	1.02	1.02	1.01	0.96
.....									
74	'Road_transport_evaporation'	0.19	0.09	0.06	0.05	0.09			
		0.22	0.86	1.84	1.86	1.41	1.24	1.2	1.32
		1.45	1.59	2.03	2.08	1.51	1.06	0.74	0.62
		0.44							

END hourly time profiles emissions

Some sources have an emission strength dependent on the ambient temperature. Per category a profile should be set. In this case the traffic emissions are a function of temperature with higher emissions under cold circumstances.

File: temperature\_var\_emis\_YYYY.txt

```
# temperature dependent emission factors for VOC and CO (SNAP14)
# Ferd Sauter
# date: 2005-12-5
BEGIN temperature dependent emission factors VOC
UNIT [-]
code category_name -30 -28 -26 -24 -22 -20 -18 -
16 -14 -12 -10 -8 -6 -4 -2 0 2 4
6 8 10 12 14 16 18 20 22 24
26 28 30 32 34 36 38 40
1 Power_generation 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
.....
71 Road_transport_gasoline 1.3 1.288 1.276 1.264 1.252
1.24 1.228 1.216 1.204 1.192 1.18 1.168 1.156 1.144
1.132 1.12 1.108 1.096 1.084 1.072 1.06 1.048 1.036
1.024 1.012 1 0.988 0.976 0.964 0.952 0.94 0.928
0.916 0.904 0.892 0.88
```

.....

END temperature dependent emission factors VOC

BEGIN temperature dependent emission factors CO

```
UNIT [-]
code category_name -30 -28 -26 -24 -22 -20 -18 -
16 -14 -12 -10 -8 -6 -4 -2 0 2 4
6 8 10 12 14 16 18 20 22 24
26 28 30 32 34 36 38 40
```

---

```

1      Power_generation      1      1      1      1      1      1      1      1
      1      1      1      1      1      1      1      1      1
      1      1      1      1      1      1      1      1      1
      1      1      1      1      1      1      1      1      1

```

```

.....
71      Road_transport_gasoline      1.65  1.624  1.598  1.572  1.546
      1.52  1.494  1.468  1.442  1.416  1.39  1.364  1.338  1.312
      1.286  1.26  1.234  1.208  1.182  1.156  1.13  1.104  1.078
      1.052  1.026  1      0.974  0.948  0.92  0.896  0.87  0.844
      0.818  0.792  0.766  0.74

```

```

.....
END temperature dependent emission factors CO

```

---

#### 4.1.2.3 define VOC-profiles

The emission file gives the total emission of non methane hydrocarbons (NMVOC). In the model these species are to be lumped to the species treated in the chemistry mechanism. Hence, for every mechanism a file should be supplied. Note the unit of mole/Kg VOC is expected by the model.

File: voc\_profiles\_CBM4\_YYYY.txt, voc\_profiles\_CB99\_YYYY.txt

---

```

#      VOC profiles YYYY
#      source: TROTREP
#      Ferd Sauter
#      date: 2005-11-30
#      "reference: report Femke Brouwer, 2005"
BEGIN VOC profiles CBM4
UNIT [mol/kg VOC]
code  category_name OLE  PAR  TOL  XYL  FORM ALD  KET
      ACET  ETH  UNR
1      Power_generation      0.45 7.079 0.218 0.094 1.036 1.698 0 0 5.355 38
.....
74      Road_transport_evaporation 0.806 63.03 0.164 0.047 0.269 1.341 0 0 0
0.982
END VOC profiles CBM4

```

---

#### 4.1.2.4 define stack heights

For each category a stack height has to be supplied. This height (m) will be applied to all emissions allocated to point (p) sources in the emission file. For area sources this list is neglected.

File: stack\_height\_YYYY.txt

---

```
#      stack heights SNAP14
#      Ferd Sauter
#      date: 2005-11-30
#      reference: source code LOTOSEUROS v0.5
BEGIN stack heights for point sources
UNIT [m]
code  category_name stack_height
1     "Power_generation"    150
2     "Residential,_commercial_and_other_combustion"    50
.....
73    "Road_transport_lpg"   0
74    "Road_transport_evaporation"  0
END stack heights for point sources
```

---

#### 4.1.3 File formats

All files used in the definition of the emissions have a common format. The files have a header, a BEGIN and END statement and a core which contains the tabulated data.

Each file starts with a header with an arbitrary number of lines, each header line starts with a #

The strings BEGIN "id\_string" and END "id\_string" are obligatory. The column header (e.g. "code category\_name mon tue wed thu fri sat sun") is also obligatory.

Columns in these files are separated by white space. These files can be created and maintained using a spreadsheet (save as TAB separated).

Note that category names must be between double quotes if there are spaces in a name.

#### 4.1.4 Parameters used in the code

The parameters are used in the emission module.

ndistr	: number of distributions
distr_nam(1:ndistr)	: name of distribution
ncountry	: number of countries
country_code(1:ncountry)	: country code (e.g. 'ALB')

---

country2distr(1:ncountry) each country	: index of emission distribution used for each country
ncat(1:ndistr)	: number of emission categories
mcat	: maximal number of emission categories
cat_code(1:mcat,1:ndistr)	: code-number of each category
cat_nam(1:mcat,1:ndistr)	: name of each category
imonthdp(1:mcat,12,1:ndistr)	: time dependency factor month
idaydp(1:mcat,7,1:ndistr)	: time dependency factor day
ihourdp(1:mcat,0:24,1:ndistr)	: time dependency factor hour
nvoc	: number of VOC-species that are emitted
vocprof(1:mcat,1:nvoc,1:ndistr)	: distribution factor voc-split
hstack(1:mcat,1:ndistr)	: stack height
frac_bc(1:mcat,1:ncountry)	: fraction black carbon
emisfac(1:mcat,1:ncountry,1:nemis)	: scenario factors define scenario's

## 4.2 Emission file

The emission file is a very simple text file with a specific format. For each record one has to specify the:

```
Countrycode  Snapcode    Type  Lon   Lat   ..... .....
```

After the latitude the emissions per component are listed. These are now fixed per emission file. See emission routine.

## 5. Output and GrADS

### 5.1 Output files

#### *General*

In general each calculation run requires its own exclusive output directory. This directory will automatically be created by running the model. The location and name of the output files are derived from input parameters specified by the user in the Run Control file the RC file (see also run setup). During the run the model writes all the results in different types of files in that exclusive directory. The presence of the run control file is essential in order to be sure the results in this directory belong to the run definition that is defined in the copied Run Control file. It is highly recommended to examine this accuracy before analysing the calculation results.

Each model run will create a couple of standard log files and optionally a number of binary files which represent the calculation results. The log files ( \*.txt) are useful to examine the progress report. The optional file 'emission-output-table' reports in a quick view the total amount of the emissions of relevant pollutants on different levels. The most important files are the binary data files and their linked descriptor or control file, the so-called CTL file. All the binary output files are formatted in a Grads structure which means those files can easily be accessed by the Grads application (see below). Before starting a recalculation it is necessary to clean (remove all data manually ) the output directory first as it is not permitted to overwrite existing (binary) output files

#### *Quick overview of model output conventions.*

- All the output file names are supervised by the LE model and derived from the run control file.
- Output files will be stored on an unique directory derived from the run control file.
- Each calculation requires an empty output location to prevent interference with previous results
- Each output directory contains at least a copy of the run control file.
- The calculation results are divided in monthly parts of binary files
- The binary output data is structured in GRADs format which is partly derived from the output options specified in the run control file.
- Output files can be identified by using the year and month number which is part of the filename:

le\_conc\_YYYY\_MM.dat

le\_dep\_YYYY\_MM.dat

The concentration and deposition data are reported in different files.

```
emission_output_table  
le_conc_1999_00.ct1  
le_conc_1999_01.ct1  
le_conc_1999_01.dat  
le_conc_1999_02.ct1  
le_conc_1999_02.dat  
le_dep_1999_01.ct1  
le_dep_1999_01.dat  
le_dep_1999_02.ct1  
le_dep_1999_02.dat  
log_chem.txt  
log.txt  
run.rc
```

### Directory list of standard LE model output files

## 5.2 GrADS

The Grads application is a Grid Analysis and Display System to analyse gridded based binary data. Grads will run on several different platforms like Unix, Linux and Windows. Grads is public domain software and can be installed by a simple download from the website (<http://grads.iges.org/grads/grads.html>). Grads is a scriptable command line driven graphical user interface and therefore it is important to have knowledge about the quintessence and format of each command. All the Grads features are described in an extensively user manual which is part of the installation and the website (<http://grads.iges.org/grads/grads.html>). There is also a Grads Tutorial available for new users.

### Starting Grads.

Grads needs a control file to understand how the calculation data is structured and filed. This file is part of the standard output of the LE model and this script describes how the output data is organised. Normally this file has to be opened by Grads first to be able to access all the data in proper way. Concerning this CTL example the template options is the feature to inquire Grads the calculation results splitted into monthly parts. In effect Grads automatically opens the right file(s) to process the data correctly. Read the Grads documentation to understand the command structure in this example.

Following the data files the following files are supplied by the model:

le\_conc\_YYYY\_MM.ct1

le\_dep\_YYYY\_MM.ct1

Besides these, there is one template file which allows to open all the monthly files at once. This CTL file has MM = 00.

```
DSET ^le_conc_1999_%m2.dat  
OPTIONS TEMPLATE  
TITLE ozon for project test
```

```
UNDEF -9.99e-38
XDEF 70 LINEAR -9.750 0.500
YDEF 90 LINEAR 42.625 0.250
ZDEF 2 LINEAR 0 1
TDEF 8760 LINEAR 01Z01jan1999 1hr
VARS 3
no2 2 99 no2 [ppb]
no 2 99 no [ppb]
o3 2 99 o3 [ppb]
ENDVARS
```

Example of Grads ctl file

### Scripting language

GrADS allows to use scripts to make plots in a fast and reproducible way. A number of these scripts will be standardised and put in the library. These common scripts will be described in later versions of the manual.

Please note that there is also a visualisation tool that enables to screen the data quickly. To use this viewer type *run view* in the command line after starting GrADS.

### 5.3 Postprocessor

The postprocessor is nearly finished and a description and manual will be supplied in the near future.

## **6. Final remarks**

## VI quick reference card

### Movement

*By Character*

Arrow keys

*By Line*

**nG** to line *n*

**0**, **\$** first, last position on line

**+**, **-** first character on next, prev line

*By Screen*

**^F**, **^B** scroll forward, back one full screen

**^D**, **^U** scroll forward, back half a screen

**^E**, **^Y** show one more line at bottom, top

**L** go to the bottom of the screen

*Miscellaneous Movement*

**fm** forward to character *m*

**Fm** backward to character *m*

**tm** forward to character before *m*

**Tm** backward to character after *m*

**w** move to next word (stops at punctuation)

**W** move to next word (skips punctuation)

**b** move to previous word (stops at punctuation)

**B** move to previous word (skips punctuation)

**e** end of word (punctuation not part of word)

**E** end of word (punctuation part of word)

### Entering Text

**a** append after cursor

**A** or **\$a** append at end of line

**i** insert before cursor

**I** or **\_i** insert at beginning of line

**o** open line below cursor

**O** open line above cursor

**cm** change text (*m* is movement)

### Cut, Copy, Paste (Working w/Buffers)

**dm** delete (*m* is movement)

**dd** delete line

**D** delete to end of line

**x** delete char under cursor

**X** delete char before cursor

**ym** yank to buffer (*m* is movement)

**yy** or **Y** yank to buffer current line

**p** paste from buffer after cursor

**P** paste from buffer before cursor

**“bdd** cut line into named buffer *b* (a..z)

**“bp** paste from named buffer *b*

### Searching and Replacing

**/w** search forward for *w*

**?w** search backward for *w*

**/w/+n** search forward for *w* and move down *n* lines

**n,N** repeat search (forward, backward)

**:s/old/new** replace next occurrence of *old* with *new*

**:s/old/new/g** replace all occurrences on the line

**:x,ys/old/new/g** replace all occurrences from line *x* to *y*

**:%s/old/new/g** replace all occurrences in file

### Miscellaneous

**n>m** indent *n* lines (*m* is movement)

**n<m** un-indent left *n* lines (*m* is movement)

**.** repeat last command

**u** undo last command

**:rf** insert text from external file *f*

INSERT GRADS quick reference card, see pdf