
EMEP/MSC-W Model Unofficial User's Guide

Release 5.0

<https://github.com/metno/emep-ctm>

Mar 18, 2024

CONTENTS:

1	Welcome to EMEP	1
1.1	Licenses and Caveats	1
1.2	Computer Information	2
1.3	Getting Started	2
1.4	Model code	3
2	Input files	5
2.1	NetCDF files	7
2.2	ASCII files	10
3	Output files	17
3.1	Output parameters NetCDF files	18
3.2	Emission outputs	19
3.3	Add your own fields	20
3.4	ASCII outputs: sites and sondes	21
4	Setting the input parameters	23
4.1	config_emep.nml	23
4.2	Base run	24
4.3	Source Receptor (SR) Runs	25
4.4	Separate hourly outputs	26
4.5	Using and combining gridded emissions	26
4.6	Nesting	27
4.7	config: Europe or Global?	32
4.8	config: SOILNOX	33
4.9	Country Variable (CV) format	33
4.10	Masks	36
4.11	Other less used options	37
4.12	Defining own sectors	38
4.13	Local Fractions (under development)	39
4.14	Local Fractions for tracing natural emissions (under development)	40
4.15	Local Fractions for Source Receptor style runs	41
4.16	Local Fractions for Sensibilities with full chemistry	42
5	Technical	45
5.1	Time steps	45
6	References	47

WELCOME TO EMEP

This guide gives a brief documentation of the EMEP/MSC-W model version rv4.45. It is intended primarily as a guide on how to run the model, and to help users wishing to understand or change the model in terms of domains, outputs, chemistry, etc.

The main documentation for the EMEP/MSC-W model is an article published in Atmospheric Chemistry and Physics in 2012. This article will be referred to as Simpson et al. (2012) in this manual.

- Simpson, D., Benedictow, A., Berge, H., Bergström, R., Emberson, L.D., Fagerli, H., Flechard, C.R., Hayman, G.D., Gauss, M., Jonson, J.E., Jenkin, M.W., Nyíri, Á., Richter, C., Semeena, V.S., Tsyro, S., Tuovinen, J.-P., Valdebenito, Á., and Wind, P.: The EMEP MSC-W chemical transport model – technical description. Atmospheric Chemistry and Physics, 12, 7825-7865, 2012.

<http://www.atmos-chem-phys.net/12/7825/2012/acp-12-7825-2012.html>

The model source code is available from the Open Source EMEP/MSC-W model github page:

<https://github.com/metno/emep-ctm>

1.1 Licenses and Caveats

The EMEP code is provided under the GNU General Public License version 3 (<http://fsf.org> and/or <http://www.gnu.org/copyleft/gpl.html>).

Each code module is prefaced with something like:

```
! <EXAMPLE_CODE.f90 - A component of the EMEP MSC-W Eulerian
!   Chemical transport Model>
! *****!
!*
!* Copyright (C) 2007-2022 met.no
!*
!* Contact information:
!* Norwegian Meteorological Institute
!* Box 43 Blindern
!* 0313 OSLO
!* NORWAY
!* email: emep.mscw@met.no
!*
!* This program is free software: you can redistribute it and/or modify
!* it under the terms of the GNU General Public License as published by
!* the Free Software Foundation, either version 3 of the License, or
!* (at your option) any later version.
```

(continues on next page)

(continued from previous page)

```
!*
!*   This program is distributed in the hope that it will be useful,
!*   but WITHOUT ANY WARRANTY; without even the implied warranty of
!*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
!*   GNU General Public License for more details.
!*
!*   You should have received a copy of the GNU General Public License
!*   along with this program.  If not, see <http://www.gnu.org/licenses/>.
!*****!
```

And a copy of the license file, **gpl.txt**, is provided with the model code source files.

It is important to note that the code is provided “as it is”, and EMEP/MSC-W has limited resources with which to support usage of the code.

1.2 Computer Information

To compile the EMEP/MSC-W model you need:

- Fortran 95 compiler
- NetCDF Library (> 4.1.3)
- MPI Library (> 1.0)

It is necessary to compile with double precision real (8 bytes real). The program has been used on computers ranging from a Linux laptop to supercomputers (Itanium2 cluster, Intel Xeon cluster, Cray XT4, IBM power5+). It is compatible with all compilers tested so far: Intel, PGI, gfortran, XL Fortran. A Makefile is included, the path to NetCDF (INCL and LLIB) have to be adapted to your machine, and the Fortran compiler (F90) and flags (F90FLAGS) to the compiler you are using. Then the command **make** should compile the model and produce the executable **emepc**.

The code has been tested with 1 to 1024 CPUs, and scales well (for large grids). If only one CPU is used 1-2 GB memory is required. If more than one, for example 64 CPUs are used, 200 MB of memory per CPU is enough (in the case of a 132 X 159 grid size). For runs on more than 32 CPUs, a fast interconnect is recommended (infiniband for example), for smaller runs, gigabit Ethernet is sufficient. It takes ~5 hours on 64*Xeon X5355 (2.66GHz) for 1 year simulation.

When downloading input data in order to do a “base run” please make sure that there are 35 Gb disc space available, especially due to large meteorology input files. The model can be run for shorter periods, users can download meteorology for only the period they are interested in, plus one day.

1.3 Getting Started

This is what you need to do before you can do a “base run” with the EMEP/MSC-W model:

- Read the EMEP/MSC-W model User Guide
- Download input data, description and downloading instructions in Input files section.
- Download the EMEP/MSC-W model source code, description and downloading instructions Model code section.
- Follow the instructions for “Submitting a Run” description in the Base run section.
- Download some model results for comparison, description in and downloading instructions the Output files section.

1.4 Model code

The latest release Open Source of the EMEP/MS-CW model is [version 5.0](#). This and previous releases can be found on the [releases](#) section of the EMEP/MS-CW Open Source [github](#) page.

Although the source code is available for download as a single compressed file on the release page, the preferred retrieval method is via the [catalog](#) tool as follows:

```
# download the catalog tool
wget https://raw.githubusercontent.com/metno/emep-ctm/tools/catalog.py

# make it executable and run it
chmod +x catalog.py

# download the source code for 5.0 release
catalog.py -R 5.0 --source
```

The model source code, makefiles, and a copy of the license file will be placed under the directory `EMEP_MS-CW_model.rv4.45.OpenSource/source/`. An overview of the files is given in [Table 1.1](#).

Table 1.1: Model source files

Type	Filename
modules files	*.f90
include files	*.inc
namelist	config_emep.nml
makefiles	Makefile and Makefile.SRCS
dependency file	dependencies
a copy of the license	gpl.txt

In addition there is a run script called `modrun.sh`, which will be placed in the `EMEP_MS-CW_model.rv4.45.OpenSource` directory. The run script, `modrun.sh`, can easily be modified to work on your computer system.

INPUT FILES

This chapter provides an overview on the necessary input files to run the EMEP/MSC-W model. A complete set of input files is provided as part of the EMEP/MSC-W Open Source release to allow model runs for the meteorological year 2015. [Table 2.1](#) lists the input files.

In the latest release, meteorology is provided for 2 different model domains and resolutions:

- *EECCA* domain with a horizontal resolution of 50x50 km² (at 60°N), on polar stereographic projection, and 20 vertical levels;
- *EMEP01* domain with a 0.1x0.1 degrees on long-lat projection, and 34 vertical levels.

Download the input via the catalog tool ([Section 1.4](#)) as follows:

```
# download 2015 meteorology for the EECCA domain
catalog.py -Y 2015 -m --met-domain EECCA

# download 2015 meteorology for the EMEP01 domain
catalog.py -Y 2015 -m --met-domain EMEP01

# download other input files
catalog.py --input
```

The meteorology files will be placed under `EMEP_MSC-W_model.rv4.45.OpenSource/meteo2015/`, and the remaining input files will be placed under `EMEP_MSC-W_model.rv4.45.OpenSource/input/`

This are all input files needed to run the EMEP/MSC-W model, except the aircraft emissions (`AircraftEmis_FL.nc`), and forest fire emissions (`FINN_ForestFireEmis_2015.nc`). See sections [Section 2.1.9](#) and [Section 2.1.12](#) for details about these emissions data.

IMPORTANT:

The input data available in the EMEP/MSC-W Open Source Web site should be appropriately acknowledged when used for model runs. If nothing else is specified according to references further in this chapter, please acknowledge EMEP/MSC-W in any use of these data.

Table 2.1: List of input data files

Data	Name	Format
Meteorology data	meteoYYYY/GRID	
Meteorology	meteoYYYYMMDD.nc (365+1 files)	netCDF ¹
Degree-day factor	DegreeDayFactors.nc	netCDF
Other Input files	input/	
Global Ozone	GLOBAL_O3.nc	netCDF
New Global Ozone	Logan_P.nc	netCDF ²

continues on next page

Table 2.1 – continued from previous page

Data	Name	Format
BVOC emissions	EMEP_EuroBVOC.nc	netCDF
Landuse	glc2000xCLMf18.nc and Landuse_PS_5km_LC.nc	netCDF
N depositions	AnnualNdep_PS50x_EECA2005_2010.nc	netCDF
Road dust	RoadMap.nc and AVG_SMI_2005_2010.nc	netCDF ³
Aircraft emissions	AircraftEmis_FL.nc	netCDF ^{Page 6, 3}
Surface Pressure	SurfacePressure.nc	netCDF ³
Forest Fire	FINN_ForestFireEmis_YYYY.nc	netCDF ³
Dust files	Soil_Tegen.nc	netCDF ³
	SoilTypes_IFS.nc	netCDF ³
Emissions	EECCA/emislist.POLL (7 files, EMEP 50km PS grid)	ASCII ⁴
	EMEP01/GNFRemis_EMEP01_2015.nc (regional, 0.1 × 0.1 lon-lat)	netCDF ³
Vertical level distribution	Vertical_levels20_EC.txt (for 20lev runs)	ASCII
Time factors for monthly emissions	MonthlyFac.POLL (7 files)	ASCII ⁴
Time factors for daily emissions	DailyFac.POLL (7 files)	ASCII ⁴
Time factors for hourly emissions	HourlyFacs.INERIS	ASCII
Natural SO ₂	DMS.nc	netCDF
Volcanoes	columnsource_emission.csv	ASCII
	columnsource_location.csv	ASCII
Lightning emissions	lt21-nox.datMM (12 files)	ASCII ¹
Emissions speciation	emissplit.defaults.POLL	ASCII ⁴
	emissplit.specials.POLL	ASCII ^{4,3}
Emission factors for scenario runs	femis.dat	ASCII
Photo-dissociation rates	jclear.SEASON (4 files)	ASCII ⁵
	jcl1.SEASON (4 files)	ASCII ⁵
	jcl3.SEASON (4 files)	ASCII ⁵
Landuse definitions	Inputs_LandDefs.csv	ASCII
Stomatal conductance	Inputs_D03SE.csv	ASCII
Sites locations for surface output	sites.dat	ASCII
Sondes locations for vertical output	sondes.dat	ASCII

¹ YYYY: year, MM: month, DD: day.

² New O₃ boundary condition data in 30 levels. Can be used with NewLogan=.true. in BoundaryConditions_mod.f90.

³ Optional, in most cases.

⁴ POLL: pollutant type (NH₃, CO, NO_x, SO_x, NMVOC, PM_{2.5} and PM_{co}).

⁵ SEASON: seasonal files (jan, apr, jul, oct).

2.1 NetCDF files

2.1.1 Meteorology

The daily meteorological input data (meteoYYYYMMDD.nc, where YYYY is year, MM is month and DD is day) used for the EMEP/MSC-W Model are based on forecast experiment runs with the Integrated Forecast System (IFS), a global operational forecasting model from the European Centre for Medium-Range Weather Forecasts (ECMWF).

The IFS forecasts has been run by MSC-W as independent experiments on the HPCs at ECMWF with special requests on some output parameters. The meteorological fields are retrieved on a $0.1^\circ \times 0.1^\circ$ longitude latitude coordinates and interpolated to $50 \times 50 km^2$ polar-stereographic grid projection. Vertically, the fields on 60 eta (η) levels from the IFS model are interpolated onto the 37 EMEP sigma (σ) levels. The meteorology is prepared into 37 sigma levels since the model is under test for a finer vertical resolution.

The open source code is released with 20 sigma levels and to make the model read the meteorology properly, a description of the 20 vertical sigma levels is needed. This is provided in an ASCII file called `Vertical_levels.txt` together with the other input data (Table 2.1). The version of the IFS model used for preparing these fields, Cycle 38r2, is documented in <http://www.ecmwf.int/research/ifsdocs/index.html>. Previous years are based on Cycle 36r1 with a resolution of $0.2^\circ \times 0.2^\circ$ on a spherical grid. Meteorological fields currently used for EMEP/MSC-W Model runs are given in Table 2.2. Some verification and description of these meteorological fields are given in Chapter 2 of the EMEP Status Report 1/2016.

Acknowledgement:

ECMWF, met.no

Table 2.2: Input meteorological data used in the EMEP/MSC-W Model

Parameter	Unit	Description
3D fields		for 37 σ
u, v	m/s	Horizontal wind velocity components
q	kg/kg	Specific humidity
θ	K	Potential temperature
CW	kg/kg	Cloud water
CL	%	3D Cloud cover
$cnvuf$	kg/sm^2	Convective updraft flux
$cnvdf$	kg/sm^2	Convective downdraft flux
PR	mm	Precipitation
2D fields		for surface
PS	hPa	Surface pressure
$T2$	K	Temperature at 2m height
$Rh2$	%	Relative humidity at 2m height
SH	W/m^2	Surface flux of sensible heat
LH	W/m^2	Surface flux of latent heat
τ	N/m^2	Surface stress
SST	K	Sea surface temperature
SWC	m^3/m^3	Soil water content
$lspr$	m	Large scale precipitation
cpr	m	Convective precipitation
$sdepth$	m	Snow depth
ice	%	Fraction of ice
$SMI1$		Soil moisture index level 1
$SMI3$		Soil moisture index level 3
$u10, v10$	m/s	Wind at 10m height

2.1.2 Gridded emissions

Gridded emissions in NetCDF can be used in conjunction with sector definitions. See section [Defining own sectors and Country Variable \(CV\) format](#)

The main advantage of NetCDF emissions is that all the information about the data (projection, units) is given in the same file. This allows the code to reproject the emissions to any grid projection on the fly. It is easy to visualize the emissions of one country with simple tools, like `ncview`. The data is simple to interpret and it is possible to add new countries to an existing file (with appropriate tools).

2.1.3 Global Ozone

Initial concentration of ozone are required in order to initialize the model runs. Boundary conditions along the sides of the model domain and at the top of the domain are then required as the model is running.

The `Logan_P.nc` file contains monthly averaged fields in NetCDF format. The initial and background concentrations are based on the Logan (1998) climatology. The Logan climatology is scaled on run time according to the Mace Head measurements as described in Simpson *et al.* (2003). For a number of other species, background/initial conditions are set within the model using functions based on observations (Simpson *et al.*, 2003 and Fagerli *et al.*, 2004).

2.1.4 BVOC emissions

Biogenic emissions of isoprene and monoterpene are calculated in the model as a function of temperature and solar radiation, using the landuse datasets. The light and temperature dependencies follow the ideas proposed in Guenther *et al.* (1993,1995), the first step in the emission processing is to define 'standard' emission potentials, which give the emissions of particular land-covers at standard environmental conditions (30°C and photosynthetically active radiation of 1000 $\mu\text{mole}/\text{m}^2/\text{s}$).

European forests are treated in most detail. For these, BVOC emission potentials have been created from the map of forest species generated by Koeble and Seufert (2001). This work provided maps for 115 tree species in 30 European countries, based upon a compilation of data from the ICP-forest network. The emission potentials for each species are as given in Simpson *et al.*, 2012, and have been aggregated into the four default forest classes used by EMEP over Europe (DF, CF, NF, BF). The NetCDF file `EMEP_EuroBVOC.nc` provides the aggregated emission potentials for these 4 categories. These emission potentials have unit $\mu\text{g}/\text{m}^2/\text{h}$, and refer to emissions per area of the appropriate forest category.

On the global scale, new landcover maps were created as a combination of GLC2000 and Community Land Model (CLM) data as described in Simpson *et al.*, 2017. The default emission potentials are given for these extra CLM categories, and for any non-forest land-cover on Europe in the file `Inputs_LandDefs.csv`. The underlying emission potentials, land-cover data bases, and model coding have however changed substantially since model version v.2011-06. The new approach is documented in Simpson *et al.*, 2012 and Simpson *et al.* 2017.

2.1.5 Landuse

Landuse data are required for modelling boundary layer processes (i.e. dry deposition, turbulent diffusion). The EMEP/MSC-W model can accept landuse data from any data set covering the whole of the domain, providing reasonable resolution of the vegetation categories. Gridded data sets providing these landuse categories across the EMEP domain have been created based on the data from the Stockholm Environment Institute at York (SEI-Y) and from the Coordinating Center for Effects (CCE). 16 basic landuse classes have been identified for use in the deposition module in the model, and three additional "fake" landuse classes are used for providing results for integrated assessment modeling and effects work.

There are two NetCDF files included, one file `Landuse_PS_5km_LC.nc` on 5 km resolution over the EMEP domain, and a global `LanduseGLC.nc` which combines data from GLC2000 with the Community Land Model (CLM). The different landuse types are described in Simpson et al (2012) and Simpson et al. (2017).

2.1.6 Degree-day factor

Domestic combustion which contribute to a large part of GNFR C (SNAP 2), varies on the daily mean temperature. The variation is based on the heating degree-day concept. These degree days are pre-calculated for each day and stored in the file `DegreeDayFactors.nc`. See Simpson et al. (2012) section 6.1.2.

2.1.7 NO_x depositions

Areas with high NO deposition loads have greater soil-NO emissions. To include this in the model, a NetCDF file where pre-calculated N-depositions are included. The file made by the results from the EMEP/MSC-W model runs over a 5-year period.

2.1.8 Road Dust

Road traffic produces dust. These emissions are handled in the EMEP/MSC-W model in the `Emissions_mod.f90` module. To include road dust, set `USE_ROADDUST=.true.` in `config_emep.nml`. There are two files included in input data, `RoadMap.nc` and `AVG_SMI_2005-2010.nc`. `RoadMap.nc` include gridded roads and PM emissions over Europe, `AVG_SMI_2005-2010.nc` are global.

2.1.9 Aircraft emissions

In the EMEP/MSC-W model aircraft emissions are 'OFF' by default. They can be switched 'ON' by setting `USE_AIRCRAFT_EMIS=.true.` in `config_emep.nml`. When using aircraft emissions there are two options: Either the dataset provided by the EU-Framework Programme 6 Integrated Project QUANTIFY (<http://www.pa.op.dlr.de/quantify>), or the more recent CAMS-GLOB-AIR dataset which can be downloaded from the Atmosphere Data Store (<https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-global-emission-inventories?tab=form>). However, before using these data a protocol has to be signed, which is why the data file can not be provided directly on the EMEP MSC-W Open Source website. Since rv4.39, CAMS-GLOB-AIR has been the default dataset. Registering at the Atmosphere Data Store is straightforward (<https://ads.atmosphere.copernicus.eu/user/register?destination=/cdsapp>) (If you rather want to use older QUANTIFY dataset go to <http://www.pa.op.dlr.de/quantify>, click on 'QUANTIFY emission inventories and scenarios', and then on 'Register'. That page will provide information about the registration process and the protocol that has to be signed. Once you are registered, click 'Login' and provide user name and password. On the new page, search for 'Emissions for EMEP', which links directly to the `Readme` file and the emission data file in NetCDF format. Download the emission data file and place it in the input folder.)

2.1.10 Natural SO₂

Natural SO₂ emissions (dimethylsulfide (DMS) from sea) are provided as monthly gridded files. The values are computed taking into account sea surface temperature and wind speed. Surface water concentrations of DMS (needed for the flux calculation) are taken from SOLAS (Surface Ocean Lower Atmosphere Study) and were downloaded from https://www.bodc.ac.uk/solas_integration/implementation_products/group1/dms/.

2.1.11 Surface Pressure

If `USE_AIRCRAFT_EMIS=.true.` in `config_emep.nml`, then in addition to the Aircraft Emission file, there will be need for a `SurfacePressure.nc` file, which is already in the `/input` folder. The NetCDF file consists of surface pressure fields for each of the months in 2008 called `surface_pressure`, and one field for the whole year called `surface_pressure_year`. All fields are given in Pa.

2.1.12 Forest Fire

Since model version rv3.9 (November 2011), daily emissions from forest and vegetation fires are taken from the “Fire INventory from NCAR version 1.0” (FINNv1, Wiedinmyer et al. 2011). Data are available from 2005, with daily resolution, on a fine $1\text{ km} \times 1\text{ km}$ grid. We store these data on a slightly coarser grid ($0.2^\circ \times 0.2^\circ$) globally for access by the EMEP MSC-W model. To include forest fire emissions set `USE_FOREST_FIRES=.true.` in `config_emep.nml` and download the 2012 GEOS-chem daily data <http://bai.acom.ucar.edu/Data/fire/>. The data needs to be stored with units mole/day in a NetCDF file called `FINN_ForestFireEmis_2015.nc` compatible with the `ForestFire_mod.f90` module.

2.1.13 Dust files

The annual ASCII data for sand and clay fractions as well as the monthly data for boundary and initial conditions for dust from Sahara are replaced with a single NetCDF file `Soil_Tegen.nc` since 2013. This covers data for a global domain in 0.5×0.5 degree resolution.

The variables ‘sand’ and ‘clay’ gives the fraction (in %) of sand and clay in the soil for each grid cell over land.

The files are used by the module `DustProd_mod.f90`, which calculates windblown dust emissions from soil erosion. Note that the parametrization is still in the development and testing phase, and is by default ‘turned off’. To include it in the model calculations, set `USE_DUST=.true.` in `config_emep.nml`. The user is recommended to read carefully documentation and comments in the module `DustProd_mod.f90`.

There is also a possibility to include boundary and initial conditions for dust from Sahara. The input file gives monthly dust mixing ratios (MM - month, e.g. 01, 02, 03,...) for fine and coarse dust from Sahara. The files are based on calculations from a global CTM at the University of Oslo for 2000. To include Saharan dust, set `USE_SAHARA=.true.` in `config_emep.nml`.

Another source for dust is an arid surface. This is accounted for by soilmoisture calculations in `DustProd_mod.f90`. Together with Soil Water Index from the meteorology files and permanent wilting point (pwp) from `SoilTypes_IFS.nc`. This file is global and NetCDF. See Simpson et al. (2012) section 6.10.

2.2 ASCII files

2.2.1 Volcanoes

Emissions from volcanic passive degassing of SO_2 are included for the active Italian volcanoes, Etna, Vulcano and Stromboli, and based upon the officially submitted data. To consider these volcanic emissions, we need to feed the locations and heights of volcanoes into the model. The input file `columnsource_location.csv` contains the geographical coordinates (latitudes and longitudes) and the heights (in meters) of the included volcanoes, while `columnsource_emission.csv` contains the emission parameters.

Since 2010 the EMEP/MSC-W model has also been used to model the transport of ash and SO_2 from volcanic eruptions. In addition to data for passive degassing of SO_2 , the above two input files also contain locations and emission parameters for two recent eruptions of Icelandic volcanoes (Eyjafjallajökull in 2010 and Grimsvötn in 2011). In order to include emissions from these eruptions one needs to set `USE_ASH=.true.` in `config_emep.nml`.

2.2.2 Gridded emissions

The official emission input for the EMEP/MSC-W model consists of gridded annual national emissions based on emission data reported every year to EMEP/MSC-W (until 2005) and to CEIP (from 2006) by each participating country. More details about the emission input with references can be found in Chapter 4 of the EMEP Status Report 1/2003 Part I (Simpson et al., 2003).

Since 2015 different formats of gridded emissions can be used and mixed (with some restrictions) in the EMEP model under one common framework. The new emission system is described in [Section 2.1.2](#). Here we focus only on the “old style” ASCII emission format.

Seven gridded emission input files (`emislist.poll`) are available in ASCII format for the following compounds: CO, NH₃, NO_x, PM_{2.5}, PM_{co}, SO_x and VOC.

The gridded ASCII emission files contain 16 columns where the first column represents the country code (http://www.emep.int/grid/country_numbers.txt), the second and the third columns are the *i* and *j* indices of the EMEP grid, the fourth and fifth columns are the total emissions from low and high sources, and the last 11 columns contain emissions from 10 anthropogenic SNAP sectors (http://reports.eea.eu.int/technical_report_2001_3/en) and 1 source-sector called “Other sources and sinks”, which include natural and biogenic emission sources. The data are given with the *M_g*.

Acknowledgement:

EMEP

2.2.3 Time factors for emissions

Monthly and daily time factors for emission can be specified for the 7 compounds (CO, NH₃, NO_x, PM_{2.5}, PM_{co}, SO_x and VOC). There is one file available per compound in ASCII format.

The first two columns in the files represent the country code (http://www.emep.int/grid/country_numbers.txt), the second column represents an index that can be referenced by the sector definition (originally this index corresponded to a SNAP sector).

In the monthly files (`MonthlyFacFile`), the 12 consecutive columns represent the time factors corresponding to the months of the year. These time factors are interpolated according to whether the “current” simulation day is in the first or second half of the month. For days in the first half of the month, the monthly factor is a combination of the factor of the previous month and the current month. For days in the second half of the month, the monthly factor combines the factor of the current and the next month (for details, see the source code).

In the daily files (`DailyFacFile`) there are 7 consecutive columns representing the time factor for each day of the week. The monthly timefactors are normalized in such a way that when combined with the daily timefactors, the total emission stays the same.

The file defined in `HourlyFacFile` includes factors for each of the eleven SNAP sectors for every hour (the columns) for each day of the week, see Simpson et al. (2012) section 6.1.2. An additional file defined in `HourlyFacSpecialsFile` can be created by the user with modified hourly factors to be used for specific countries. The format is the same as for the default factors, except for an additional first column specifying the country code number.

2.2.4 Emission heights

In previous versions the emission height distributions was given in a separate file. Now it is part of the code, and can also be modified by the users using `config_emep.nml` setting (see section “defining own sectors”).

A set of vertical distribution for different sectors are predefined in the model. The release heights are defined as a set of fractions released into predefined layers.

The release height definitions are independent of the layers used by the model.

There are 8 predefined release heights distributions. Those can also be defined through the `config_emep.nml` setting. The following will give exactly the same distributions as the predefined. You can then modify the values, or add new defined distributions.

Listing 2.1: Default definition of emission height distributions

```
Emis_Zlevels(1:)20.0, 50.0, 92.0, 184.0, 324.0, 522.0, 781.0, 1106.0,
Emis_h(1:,1) = 0.000, 0.000, 0.000, 0.003, 0.147, 0.400, 0.300, 0.150,
Emis_h(1:,2) = 1.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
Emis_h(1:,3) = 0.060, 0.067, 0.093, 0.750, 0.030, 0.000, 0.000, 0.000,
Emis_h(1:,4) = 0.050, 0.063, 0.087, 0.700, 0.100, 0.000, 0.000, 0.000,
Emis_h(1:,5) = 0.020, 0.034, 0.046, 0.600, 0.300, 0.000, 0.000, 0.000,
Emis_h(1:,6) = 0.000, 0.000, 0.000, 0.410, 0.570, 0.020, 0.000, 0.000,
Emis_h(1:,7) = 0.200, 0.300, 0.020, 0.044, 0.066, 0.094, 0.123, 0.153,
Emis_h(1:,8) = 0.200, 0.800, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
```

The `Emis_Zlevels` defines the height of the layer boundaries for emissions in meters. (Standard atmosphere is assumed to transform those in Pressure by the model). The first layers is from surface to 20 meters, the second layer from 20 to 50 m... until the eighth and last layer which runs from 781 to 1106 meters.

For example sectors defined with the height index “1”, will release nothing in the three lowest layers, 0.3% into the fourth layer, 14.7% into the fifth layer etc.

The layers defined in `Emis_h` are independent from the layers used in the model run and do not need to be adapted if the number of model layers is modified. The actual resulting distribution of emissions into model layers is computed by the model and will be shown in the standard output.

2.2.5 Emission factor for scenario runs

Scenario run in the case of the EMEP/MSC-W model means a run to test the impact of one or more pollutants from a particular country.

Emission factors are applied to specified countries and emission sectors and can be set by changing the ASCII file `femis.dat`. This file can be changed by the users according to their needs. See the [Source Receptor \(SR\) Runs](#) section for details.

2.2.6 Vertical_levels.txt

Defines the vertical model layers. The numbers in Vertical_levels.txt correspond to the “A” and “B” coefficients of the hybrid (eta) coordinates ($P=A+B*P_{surf}$).

Close to the surface, A should be small, and higher up we should use pressure levels. Then there is a gradual transition from surface to pressure levels.

If the file is not provided, the meteorological vertical levels are used. In principle the model levels can be completely different, but it is more sensible to define layer boundaries that match the meteorological levels.

NB: it is important not to define a lowest layer thinner than about 45 meters; the deposition scheme will fail if the middle of the lowest layer is smaller than the highest defined vegetation.

2.2.7 Chemical speciation of emissions

Many of the emission files give emissions of a group of compounds, e.g. NO_x includes $NO+NO_2$, and VOC can include many compounds. The information needed to retrieve emissions of individual compounds from these the gridded files is given in files labelled `emissplit.defaults.POLL` or `emissplit.specials.POLL`, where POLL can be NO_x , VOC, etc.

The defaults file give the emission split for each sector split index (one per row, with second index being the sector split index), which is applied to all countries by default. For VOC this split was derived from the UK inventory of Passant (2002), as part of the chemical comparison project of Hayman *et al.* (2011).

The specials files are in general optional, and can be used to specify speciation for particular countries or sectors. The 1st column specifies the country code of interest, the second the sector index.

If forest fires are used, then the file `emissplit.specials.voc` is required (not optional), and the country-code 101 used to specify the VOC speciation of forest fires in this file.

2.2.8 Lightning emissions

Emissions of NO_x from lightning are included in the model as monthly averages on T21 ($5.65^\circ \times 5.65^\circ$) resolution (Köhler *et al.*, 1995). The lightning emissions are defined on a 64×32 grid with 17 vertical levels, with global coverage, and are provided as 12 ASCII files `lightningMM.dat`.

2.2.9 Landuse definitions

For the vegetative landuse categories where stomatal modelling is undertaken, the start and end of the growing season (SGS, EGS, in days) must be specified. The calculation of SGS and EGS with respect to latitude is done in the module `LandDefs_mod.f90`. The parameters needed to specify the development of the leaf area index (LAI) within the growing season are given in the ASCII file `Inputs_LandDefs.csv`. For more information, see chapter 5 of the EMEP Status Report 1/2003 Part I (Simpson *et al.*, 2003).

The file, designed to be opened with excel or gnumeric, contains a header briefly explaining the contents of the 14 columns. The first three columns are representing the landuse name, code (which are consistent with those in Landuse . Input file) and type (grouping of the landuse classes). The fourth column (PFT) gives a plant-functional type code (for future use), the fifth gives the maximum height of vegetation (m), the sixth indicates albedo (%) and the seventh indicates possible source of NH_x (0 off/1 on, currently not used). Columns 8 to 11 define the growing season (day number), column 12 and 13 lists the LAI minimum and maximum (m^2/m^2) and columns 14 and 15 defines the length of the LAI increase and decline periods (no. of days). Finally, the last four columns give default values of foliar biomass and biogenic VOC emission potentials. See Simpson *et al.*, (2012) for details.

2.2.10 Stomatal conductance

Parameters for the stomatal conductance model, deposition of O₃ and stomatal exchange (DO3SE) must be specified. That are based upon the ideas in Emberson *et al.*, 2000, and are discussed in Simpson and Emberson, 2006 and Tuovinen *et al.* 2004.

The ASCII file `Inputs_DO3SE.csv` provides land-phenology data of each landuse type for stomatal conductance calculations. The data are summarised in Table 5.1 in Chapter 5 of the EMEP Status Report 1/2003 Part I (Simpson *et al.*, 2003).

The file contains a **header** with the contents of the file, with different factors needed for each of the landuse classes used in the EMEP/MSC-W model. The first two columns represent the landuse code (which are consistent with those in `Landuse.Input` file) and name. The next 22 values are different phenology factors.

2.2.11 Photo-dissociation rates

The photo-dissociation rates (J-values) are provided as lookup tables. The method is previously described in Jonson *et al.*, (2001). J-values are provided as clear sky, light cloud and dense cloud conditions, and the model interpolates between these according to cloudiness from the meteorological input data. In the lookup tables data are listed for every 10 degree latitude at an interval of 1 degree zenith angle at every model height.

For the two types of cloud conditions there are one ASCII file averaged for each season (SS); 01, 02, 03 and 04. For light cloud the four seasonal files are called `jc11kmSS.dat`, for dense cloud conditions the four seasonal files are called `jc13kmSS.dat`, and then for clear sky four files called `jc1clearSS.dat`. In addition there are two files for June called `jc11.jun` and `jc13.jun`.

Each file contains 18 columns. The first column is latitude of zenith angle and then the next 17 are the values for the model levels with the 1/s. For more details about these rates, please read Chapter 7.2 of the EMEP Status Report 1/2003 Part I (Simpson *et al.*, 2003).

2.2.12 Site and Sonde files

The model provides a possibility for extra output data of surface concentration for a set of specified measurement site locations and concentrations for the vertical column above a set of specified locations. These site and sonde locations are listed in the ASCII files `sites.dat` and `sondes.dat` files. These files can be changed by the user, and provide the outputs described in `sec-sitesonde`.

Two main options are available for the output of ASCII files for comparison with measurements or detailed model analysis. These are

sites

output of surface concentrations for a set of specified measurement site locations.

sondes

output of concentrations for the vertical column above a set of specified locations.

Both sites and sondes are specified and handled in similar ways, in the module `Sites_mod.f90`, so we treat them both together below. Locations are specified in the input files `sites.dat` and `sondes.dat`. The files start with a description of its content followed by a list of the stations. For example, a `sondes.dat` input file may look like this: ..
`literalinclude:: sites.dat`

caption

Site location definition (`sites.dat`) example.

Listing 2.2: Site location definition, LatLonHrel Coords example.

```
# "Sites: names, locations, relative elevations"
# "lev: vertical coordinate m above local topography"
: Units index
: Coords LatLonHrel
: DomainName NA
#
name ix iy lev #HEADERS
-   index index level #SKIP
#DATA:
DK0005R  54.74650  10.73616      10 ! Keldsnor (10m a.s.l, terrain 0m)
NO0002R  58.38853   8.25200     109 ! Birkenes_II (219m a.s.l, terrain 110m)
SE0013R  67.88333  21.06667     184 ! Esrange (475m a.s.l., terrain 291m)
```

Listing 2.3: Site location definition, LatLonZm Coords example.

```
# "Sites: names, locations, relative elevations"
# "lev: vertical coordinate m above local topography"
: Units index
: Coords LatLonZ
: DomainName NA
#
name ix iy lev #HEADERS
-   index index level #SKIP
#DATA:
DK0005R  54.74650  10.73616      10 ! Keldsnor, 10m a.s.l
NO0002R  58.38853   8.25200     219 ! Birkenes_II, 219m a.s.l
SE0013R  67.88333  21.06667     475 ! Esrange, 475m a.s.l.
```

The first line in each file is a header with file content. Then, the contents are described in more detail. Text strings after # are just clarifying comments. 'Area', e.g., is the domain to which the stations belong, e.g. 'Northern Hemisphere'.

Text after : is read in by the model:

Units

Either 'deg' (degrees) or 'index' (model grid indices).

Coords

Vertical coordinate system that is used in the model - see below.

The VertCoords system was changed in rv4.36, to allow several options. Specifying altitude rather than model coordinate is of course an obvious alternative to the model layer number that was previously used, but it is also easily mis-used and mis-understood too. For example, a site at 500m might need vertical profile and/or deposition correction if sitting on a 500m high plateau (hence it has a relative altitude of 0m, and should use iz=KMAX_MID), but if sitting on an isolated mountain in terrain of height 0m, the relative altitude would indeed to 500m and we should pick from some model level which represents that. Tricky! (Best might be to give station altitude and relative altitude in separate columns, so it is explicit at least.)

Briefly though, one can now set the "Coords" parameter (in the header of the sites.dat input file, used to be just "Lat-Long") to be one of:

1. LatLonKdown - same as older LatLong, with lat/lon coordinates, then k-number with ground level being e.g. 20

2. LatLonZm - give lat, long, then altitude in metres. The model will then compare that altitude with the model's topography, to estimate a relative altitude (Hrel). It then calculates which model layer corresponds to that altitude.
3. LatLonHrel - give lat, long, and then your own preferred relative altitude (Hrel). This relative altitude may be calculated by comparison to digital elevation model (see for example, Loibl, W.; Winiwarter, W.; Kopsca, A.; Zufger, J. & Baumann, R. Estimating the spatial distribution of ozone concentrations in complex terrain Atmos. Environ., 1994, 28, 2557-2566).
4. IJKdown - does everything in model's i, j and k coordinates

In principle (3) should produce the best results, with relative altitudes calculated compared to local topography (e.g. < 5km). Option (3) also allows the same Hrel to be used regardless of the underlying meteo resolution. One can also get Hrel from the TOAR database for some sites (Schultz, MG, et al 2017 Tropospheric Ozone Assessment Report: Database and metrics data of global surface ozone observations. Elem Sci Anth, 5: 58, DOI: <https://doi.org/10.1525/elementa.244>).

However, testing of these systems against e.g. diurnal profiles of ozone shows rather unpredictable results in some cases, and often just using the model's surface concentrations produces results which are as good as those based upon altitude. The new systems are very flexible though, and allow the user to explore different methodologies.

Both `sites.dat` and `sondes.dat` files are optional, but recommended. The species and meteorological data requested for site and sonde output are specified in `Config_module.f90` by the use of arrays. Only a few met fields are defined so far but more can be added into `Sites_mod.f90` as required.

The output files `sites_2015.csv` and `sondes_2015.csv` are comma separated files that can be read by excel. netcdf versions of these files are also provided, `sites_2015.nc` and `sondes_2015.nc`. If you include the whole year, or the 31st December, `sites_2016.csv` and `sondes_2016.csv` are also included in the output.

OUTPUT FILES

Output files from a model run are written out in either ASCII, or (for most data outputs) in NetCDF format. The different NetCDF files are named after the `runlabel1` parameter set in `modrun.sh`. The model output is written to the same directory as where the runscript where submitted, as described in [Section 4](#).

To check your model run, already prepared model result files can be downloaded using the catalog tool ([Section 1.4](#)) as follows:

```
# download the output
catalog.py -R 5.0 --output
```

Unpacked files are placed in an output directory with model run results for a whole year, and sometimes with a smaller test run for e.g. April.

Table 3.1: List of model output files

Output data files	Short description	Format
Base_hour.nc	Gridded hourly values of a selection of compounds	NetCDF
Base_day.nc	Gridded daily values of a selection of compounds	NetCDF
Base_month.nc	Gridded monthly values of a selection of compounds	NetCDF
Base_fullrun.nc	Gridded yearly values of a selection of compounds	NetCDF
sites_YYYY.nc	Surface daily values of a selection of stations and compounds	NetCDF ¹
sondes_YYYY.nc	Vertical daily values of a selection of stations and compounds	NetCDF ¹
sites_YYYY.csv	ASCII version of sites_YYYY.nc	ASCII
sondes_YYYY.csv	ASCII version of sondes_YYYY.nc	ASCII ²
Additional files		
RunLog.out	Summary log of runs, including total emissions of different air pollutants per country	ASCII
Timing.out	Timing log file	ASCII

¹ YYYY: year.

² Deprecated output.

3.1 Output parameters NetCDF files

Parameters to be written out into `Base_hour.nc`, `Base_day.nc`, `Base_month.nc` and `Base_year.nc` are defined in `My_Derived_mod.f90` and `Derived_ml.f90`. In `My_Derived_mod.f90`, the user can specify the output species (air concentrations, depositions, column values), units and temporal resolution of the outputs (daily, monthly, yearly).

The name of output parameter provides some information about data. The names start with TYPE of the parameter, namely SURF (surface air concentrations), DDEP (Dry deposition), WDEP (Wet deposition), COLUMN (Vertically integrated parameters), Area (Surface area) etc.

For surface air concentrations, the general name pattern is `SURF_UNITS_COMPONENT`. Here, UNITS can e.g. be “ug” ($\mu\text{g}/\text{m}^3$), “ugS” ($\mu\text{g}(S)/\text{m}^3$), “ugN” ($\mu\text{g}(N)/\text{m}^3$), or “ppb”. Note that the components are classified either as “SPEC” (species) or “GROUP”. The content of complex GROUP components can be found in `CM_ChemGroups_mod.f90`.

For column integrated parameters, the names are `COLUMN_COMPONENT_kNLAYERS`, where *NLAYERS* is the number of layers from model top included in the integration. The units for column outputs are “ugm2” ($\mu\text{g}/\text{m}^2$), “mcm2” (molec/m^2) or “e15mcm2” ($10^{15}\text{molec}/\text{m}^2$).

For dry depositions, given per 1m^2 of specified landuse, the names look like `DDEP_COMPONENT_m2LANDUSE`, where LANDUSE can be either a specific landuse type or a cell average. For wet depositions, the names are `WDEP_COMPONENT`. The units for dry and wet depositions are mg/m^2 , $\text{mg}(S)/\text{m}^2$ or $\text{mg}(N)/\text{m}^2$.

Surface concentrations, column integrated, wet and dry deposition outputs are defined by the user in `config_emep.nml` file. Surface concentrations and column integrated outputs are described in `OutputConcs_config` namelist, Dry and wet deposition outputs are described in `OutputDep_config` namelist.

`VG_COMPONENT_LANDUSE` are the dry deposition velocities on various landuse types, typically in cm/s .

Table 3.2 lists most of output parameters, providing additional explanation to the complex components. For a complete suit of currently selected output parameters, see provided output NetCDF files, or `My_Derived_mod.f90` module.

Table 3.2: List of output parameters

Parameter name	Short description	Comments
Surface Concentrations		
<code>SURF_ppb_O3</code>	O_3 [ppb]	
<code>SURF_ugN_NO</code>	NO [$\mu\text{g}(N)/\text{m}^3$]	Available also in ppb
<code>SURF_ugN_NO2</code>	NO_2 [$\mu\text{g}(N)/\text{m}^3$]	Available also in ppb
<code>SURF_ugN_HNO3</code>	HNO_3 [$\mu\text{g}(N)/\text{m}^3$]	Available also in ppb
<code>SURF_ugN_NH3</code>	NH_3 [$\mu\text{g}(N)/\text{m}^3$]	Available also in ppb
<code>SURF_ugS_SO2</code>	SO_2 [$\mu\text{g}(S)/\text{m}^3$]	Available also in ppb
<code>SURF_ug_S04</code>	SO_4^{2-} [$\mu\text{g}/\text{m}^3$]	
<code>SURF_ug_NO3_F</code>	NO_3^- fine aerosol [$\mu\text{g}/\text{m}^3$]	As ammonium nitrate
<code>SURF_ug_NO3_C</code>	NO_3^- coarse aerosol [$\mu\text{g}/\text{m}^3$]	Associated with sea salt and mineral dust
<code>SURF_ug_TN03</code>	NO_3^- total [$\mu\text{g}/\text{m}^3$]	Sum of fine and coarse nitrate
<code>SURF_ug_NH4_F</code>	NH_4^+ fine aerosol [$\mu\text{g}/\text{m}^3$]	As ammonium sulphate and ammonium nitrate
<code>SURF_ug_SIA</code>	SIA [$\mu\text{g}/\text{m}^3$]	Secondary Inorganic Aerosol
<code>SURF_ug_SIA</code>	SIA [$\mu\text{g}/\text{m}^3$]	Secondary Inorganic Aerosol
<code>SURF_ug_ECFINE</code>	EC fine [$\mu\text{g}/\text{m}^3$]	Elemental carbon
<code>SURF_ug_ECCOARSE</code>	EC coarse [$\mu\text{g}/\text{m}^3$]	Elemental carbon
<code>SURF_ug_PM_OM25</code>	OM fine [$\mu\text{g}/\text{m}^3$]	Organic Matter fine aerosol
<code>SURF_ug_PM_OMCOARSE</code>	OM coarse [$\mu\text{g}/\text{m}^3$]	Organic Matter coarse aerosol
<code>SURF_ug_SEASALT_F</code>	Sea salt fine aerosol [$\mu\text{g}/\text{m}^3$]	
<code>SURF_ug_SEASALT_C</code>	Sea salt coarse aerosol [$\mu\text{g}/\text{m}^3$]	
<code>SURF_ug_SEASALT</code>	Sea salt [$\mu\text{g}/\text{m}^3$]	Sum of fine and coarse sea salt
<code>SURF_ug_DUST_ROAD_F</code>	Road dust fine aerosol [$\mu\text{g}/\text{m}^3$]	

continues on next page

Table 3.2 – continued from previous page

Parameter name	Short description	Comments
SURF_ug_DUST_ROAD_C	Road dust coarse aerosol [$\mu\text{g}/\text{m}^3$]	
SURF_ug_DUST_WB_F	Windblown dust fine [$\mu\text{g}/\text{m}^3$]	
SURF_ug_DUST_WB_C	Windblown dust coarse [$\mu\text{g}/\text{m}^3$]	
SURF_ug_DUST_SAH_F	Saharan dust fine [$\mu\text{g}/\text{m}^3$]	From Boundary conditions
SURF_ug_DUST_SAH_C	Saharan dust coarse [$\mu\text{g}/\text{m}^3$]	From Boundary conditions
SURF_ug_DUST_NAT_F	Natural dust fine [$\mu\text{g}/\text{m}^3$]	Windblown and Saharan
SURF_ug_DUST_NAT_C	Natural dust coarse [$\mu\text{g}/\text{m}^3$]	Windblown and Saharan
SURF_ug_DUST	Mineral dust [$\mu\text{g}/\text{m}^3$]	From all sources
SURF_ug_PM10	PM ₁₀ dry [$\mu\text{g}/\text{m}^3$]	
SURF_ug_PM10_rh50	PM ₁₀ wet [$\mu\text{g}/\text{m}^3$]	PM ₁₀ particle water at 50 %rh
SURF_ug_PM25	PM _{2.5} dry [$\mu\text{g}/\text{m}^3$]	Includes fine PM and 27% of coarse NO ₃ ⁻
SURF_ug_PM25_rh50	PM _{2.5} wet [$\mu\text{g}/\text{m}^3$]	PM _{2.5} particle water at 50 %rh
SURF_ug_PM25X	PM _{2.5} dry [$\mu\text{g}/\text{m}^3$]	Includes fine PM and 27% of coarse NO ₃ ⁻ , EC and OM
SURF_ug_PM25X_rh50	PM _{2.5} [$\mu\text{g}/\text{m}^3$]	As PM25X + particle water at 50 %rh
SURF_ug_PMFINE	Fine PM [$\mu\text{g}/\text{m}^3$]	Sum of all fine aerosols
SURF_ug_PPM25	Primary PPM25 [$\mu\text{g}/\text{m}^3$]	Anthropogenic emissions
SURF_ug_PPM_C	Primary coarse PM [$\mu\text{g}/\text{m}^3$]	Anthropogenic emissions
SURF_ug_PM25_FIRE	PM _{2.5} from forest fires [$\mu\text{g}/\text{m}^3$]	Sum of BC, OC and rest PM _{2.5}
Dry Depositions		
DDEP_SOX_m2Grid	Oxidized sulphur [$\text{mg}(\text{S})/\text{m}^2$]	For a grid cell landuse area weighted
DDEP_SOX_m2Conif	Oxidized sulphur [$\text{mg}(\text{S})/\text{m}^2$]	To coniferous forest
DDEP_NOX_m2Grid	Oxidized nitrogen [$\text{mg}(\text{N})/\text{m}^2$]	For a grid cell landuse area weighted
DDEP_NOX_m2Decid	Oxidized nitrogen [$\text{mg}(\text{N})/\text{m}^2$]	To deciduous forest
DDEP_RDN_m2Grid	Reduced nitrogen [$\text{mg}(\text{N})/\text{m}^2$]	For a grid cell landuse area weighted
DDEP_RDN_m2Seminat	Reduced nitrogen [$\text{mg}(\text{N})/\text{m}^2$]	To semi-natural
Wet Depositions		
WDEP_PREC	Precipitation [mm]	
WDEP_SOX	Oxidized sulphur [$\text{mg}(\text{S})/\text{m}^2$]	
WDEP_SS	Sea salt [mg/m^2]	
Others		
AOD	Aerosol Optical Depth at 550nm	Experimental
Area_Crops_Frac	Area fraction of crops	Available for several landuses
VG_NO3_F_Grid	Dry deposition velocity of fine NO ₃ ⁻	Grid cell average
Meteorological parameters		
USTAR_GRID	U^* grid averaged	Available for several landuses
T2m	Temperature at 2m [$^{\circ}\text{C}$]	
rh2m	Fractional relative humidity at 2m	

3.2 Emission outputs

Emis_mgm2_XX fields in the output, give all emissions used by the model (accumulated over the relevant period). Sec_Emis_mgm2_XX are “sector emissions”, i.e. includes only contributions from the files defined in emis_inputlist and Emis_sourceFiles. Sec_Emis_mgm2_XX do not include emissions such as volcanoes, forest fires, DMS, lightning, aircraft etc.

For hourly outputs of emissions set

HourlyEmisOut = T,

For daily outputs of emissions set

```
DailyEmisOut = T,
```

Detailed emissions by sectors can be obtained with the keyword `SecEmisOutWanted` for the wanted sectors. For example adding the lines:

```
SecEmisOutWanted(2) = T,  
SecEmisOutWanted(7) = T,
```

will give you the emissions for sector 2 and 7 for all components.

Totals per country and sectors (all), can be obtained in the log with:

```
SecEmisTotalsWanted = T,
```

To get emissions partitioned into splitted compounds (up to 18), the value `EmisSplit_OUT=.true.` must be set in `Config_module.f90`, and the code recompiled. (This parameter cannot be set in `config_emep.nml` for now)

3.3 Add your own fields

Most standard output can be outputted by adding lines and modifying the parameters in the `config_emep.nml` file.

The meteorological fields defined in the `met` array in the `MetFields_mod.f90` file, can be retrieved by using the 'MET2D' or 'MET3D' keywords. If a 3D array is requested with the 'MET2D' keyword, only the lowest level is written out.

If you want an array that does not fit in any category, or even make your own special field, you can get it in the output using the procedure shown below; this will however require that you write in the code and recompile. For instance in `config_emep.nml` `OutputMisc` define:

```
`` 'J(NO2)', 'USET', 'D3_J(NO2)', 'photorate', '1/s', -99, -99, F, 1.0, T, 'H', ``
```

- The first column (name) is the name as shown in the output
- The second column (class) must be 'USET'
- The strings of the first and third columns can be chosen freely, but if one of them starts with the two characters 'D3', it will be interpreted as a 3 dimensional field
- The fourth column can be any string
- The fifth column is the unit, as show in the output
- The sixth column (index) is an integer that can be used to characterize internal indices
- The seventh columns should be a negative integer
- The eighth column can be F or T, indicating wether the field must be divided by the time step (`dt_advec`)
- The ninth column (scale) is a scaling factor
- The tenth column, F or T, indicates if the field must be averaged (T) or accumulated (F)
- The eleventh (last) column indicates the periodicity of the output. 'H' -> every hour, 'YMH' -> every hour, month and at the end of the run (and other combinations are allowed).

In the code you must define the indice of your new ouput. The requested outputs strings are stored in `f_2d` and `f_3d`; for instance

```
photo_out_ix = find_index("D3_J(NO2)", f_3d(:)%subclass)
```


and the values of the field must be put into the d_2d or d_3d array, using this index, for instance:

```
if(photo_out_ix>0) d_3d(photo_out_ix,i,j,1:num_lev3d,IOU_INST) = rcphot(IDN02,  
↪lev3d(1:num_lev3d))
```

(for 2D output, write in d_2d and ommit the vertical index)

3.4 ASCII outputs: sites and sondes

As noted in `ec-sitessondes-input`, two main options are available for the output of ASCII files for comparison with measurements or detailed model analysis at specific sites. These are

sites

output of surface concentrations for a set of specified measurement site locations.

sondes

output of concentrations for the vertical column above a set of specified locations.

The output files `sites_2015.csv` and `sondes_2015.csv` are comma separated files that can be read by excel, python or fortran tools. If you include the whole year, or the 31st December, `sites_2016.csv` and `sondes_2016.csv` are also included in the output.

SETTING THE INPUT PARAMETERS

In this chapter we provide detailed information on how to configure the parameters of the regional EMEP/MSC-W model.

In general the parameters and pathes to the input files can all be set in the configuration file `config_emep.nml` (a fortran namelist).

4.1 `config_emep.nml`

The default parameter, constants and flags are defined in `Config_module.f90`, and they can be overwritten by `config_emep.nml` settings.

Here is an example of content of the most important parameters:

Listing 4.1: Basic namelist example; `config_emep.nml` extract.

```
&Model_config
  GRID = 'EECCA',
  iyr_trend = 2015,
  runlabel1 = 'Base',
  runlabel2 = 'Opensource_Setup_2022',
  startdate = 2015,01,01,00,
  enddate = 2015,01,10,24,
  spinup_enddate = 2015,01,02,06,

  DataPath(1) = '../input', ! define 'DataDir' keyword

  meteo = '../meteoYYYY/GRID/meteoYYYYMMDD.nc',
  DegreeDayFactorsFile = 'MetDir/DegreeDayFactors.nc',
!-----
  EmisDir = 'DataDir/GRID',
  emis_inputlist(1)%name= 'EmisDir/gridPOLL', !example of ASCII type
  emis_inputlist(1)%type = 'GNFR_CAMSsectors',
!-----Sub domain x0, x1, y0, y1
  RUNDOMAIN = 36, 100, 50, 150, ! EECCA sub-domain
&end
```

In the extract above, the model is run for the period 1 January to 10 January 2015 and the trend year used is 2015, with output staring at 6 UTC on January 2nd. The `startdate` must correspond to a time step on the input meteorology, `enddate` and `spinup_enddate` can be chosen freely. Output files will be stored with the name 'Base' and the meteorological correspond to the 'EECCA' grid.

Note the definition of the meteorological input (“meteo”): the keywords, ‘YYYY’, ‘MM’, ‘DD’ and ‘GRID’ will be replaced on the fly by respectively, year, month, day and grid (‘EECCA’ here).

It is possible to run the model on a smaller domain than the full regional model domain, as defined by indexes x and y . For the ‘EECCA’ grid $x = 1, \dots, 132$; $y = 1, \dots, 159$. To set a smaller domain, use RUNDOMAIN variable in the ModelConstants_config namelist to indicate the sub-domain indexes. In the config_emep extract above, RUNDOMAIN defines a subdomain with $x = 36, \dots, 100$; $y = 50, \dots, 150$. The indices refer always to the meteo grid (starting from 1). If the indices are outside the grid, only the portion within the full grid is taken into account.

4.2 Base run

To run the model you need access to the executable (emepctm) and the config_emep.nml files. The path to the other input files are defined by defaults or set in the config_emep.nml file.

To run the model you can either run interactively (mostly for short run), using something similar to (it depends on your system):

```
mpirun emepctm
```

For longer runs you should run the model as a batch job (the details will depend on your system). If the run was successful, a message

```
programme is finished
```

will be stated at the end of the standard output (screen or some appropriate file for batch jobs).

The model results will be written to the same directory. Please make sure there is enough disk place for the model results. For more information about the model result output files, see [Section 3](#).

If for some reason the model crashed, please check both the log and the error file for any clue of the crash. After fixing the problem the job can be submitted again.

The variables wanted in the output are specified in the OutputConcs, DDEP_ECOS, DDEP_WANTED, WDEP_WANTED and OutputVeg03 parameters for surface concentrations, depositions and other miscellaneous outputs.

The main output files are

- Base_fullrun.nc gives values averaged over the entire simulation. This file has always 1 record and the data is an average over the the period startdate to enddate.
- Base_month.nc average over each calendar month (12 records for a yearly run)
- Base_day.nc averages over 24 hours, starting at 06:00 UTC (each record gives average values between 06:00 and 06:00 the next day. The first time step is written out at 06:00 hr on 2Jan (i.e the first 6 hours are “lost”). The last record may also be averaged only until the end of the run, and not at 06:00.)
- Base_hour.nc, averaged over 1 hour.
- Base_hourInst.nc, instantaneous values every hour.

4.3 Source Receptor (SR) Runs

The EMEP/MSC-W model can be used to test the impact of reduced emission of one or more pollutants from a particular country or a number of countries. Such runs are called “Scenario runs”. They are used for source-receptor calculations.

Emission factors for reduced emissions of pollutants from different sectors and countries can be defined in the input file `femis.dat`. The path to this file can be set in `config_emep.nml`, `femisFile = /my/path/femis.dat`

Listing 4.2: `femis.dat` for a base run.

Name	sector	sox	nox	co	voc	nh3	pm25	pmco
27	0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

This base run example means that there are (1.0), no emission reductions of SO_x, NO_x, CO, VOC, NH₃, PM_{2.5} and PM_{co} from all sectors in the UK.

- The first column of the second line represents the country code. (27 is the code for UK.) The codes for all countries can be found in Fortran module `Country_mod.f90` or at (http://www.emep.int/grid/country_numbers.txt). The country code must be the same as in the emission files for the given country. Some countries and areas are divided into sub-areas in the emission files. In this case, one line for each sub-area has to be included into the `femis.dat` file. Countries and areas where emissions are given for sub-areas include the Russian Federation, Germany and all sea areas. “0” means all countries.
- The second column of the second line represents the sector and “0” means all sectors. Here one can write the appropriate sector code if the emission is reduced only from a specific sector. The description of each sector can also be found in the Fortran module `EmisDef_mod.f90`.
- The columns under the pollutant names show the emission factors for the given pollutants. For example, 0.7 would mean 70% of the original emission, thus 30% reduction.
- The first line is important, it must consist of two words followed by the names of the pollutants emitted (in any order).

An example of `femis.dat` file describing 50% reduced emission of NH₃ from sector 10 (the emission from agriculture) in the UK is shown in Listing 4.3 and Listing 4.4.

Listing 4.3: `femis.dat` for 50% NH₃ reduction from sector 10 over UK.

Name	sector	sox	nox	co	voc	nh3	pm25	pmco
27	10	1.0	1.0	1.0	1.0	0.5	1.0	1.0

Or alternative syntax using country ISO short names:

Listing 4.4: `femis.dat` for 50% NH₃ reduction from sector 10 over UK.

Name	sector	sox	nox	co	voc	nh3	pm25	pmco
Country UK	10	1.0	1.0	1.0	1.0	0.5	1.0	1.0

Instead of entire countries, reductions can also be specified by coordinates (and combined with country reductions). The line with coordinate corrections must start with the keyword `lonlat`. The coordinates are given in longitude latitude (min and max and the coordinates of the centre of the gridcells are tested. Gridcells are either entirely included or entirely reduced, never cut into smaller parts).

Listing 4.5: `femis.dat` example.

Name	7	sox	nox	co	voc	nh3	pm25	pmco
17	0	1.0	1.0	1.0	1.0	1.0	0.5	0.5

(continues on next page)

(continued from previous page)

```
lonlat 3.3 7.2 50.7 53.5 17 0 1.0 1.0 1.0 1.0 0.0 1.0 1.0
```

In Listing 4.5, country with code 17 (NL) will reduce PM_{2.5} and PM₁₀ emissions by half for all sectors. Emissions of NH₃ from country with code 17 only, will be removed from the rectangle with longitudes between 3.3 and 7.2 degrees East, and between 50.7 and 53.5 degrees North. Use zero (0) as country code to specify that emissions from all countries should be reduced.

4.4 Separate hourly outputs

The Base_hour.nc and Base_uEMEP_hour.nc files can become very large. It is possible to split them into one file per day by adding the keyword `HOURLYFILE_ending = 'JJJ.nc'`, in the configuration file. `JJJ`, will be automatically replaced by the corresponding day of the year (i.e. a number from 1 to 366 giving for instance Base_hour_001.nc). The full date as in Base_hour_20180101.nc can be obtained by defining `HOURLYFILE_ending = 'YYYYMMDD.nc'`,.

4.5 Using and combining gridded emissions

The gridded emission files are controlled via the config_emep.nml file. Each file is assigned as one set of values for emis_inputlist. Listing 4.6 line 1 includes an ASCII emission file, where the keyword POLL will be replaced by the model by all the emitted pollutants (according to the names defined in CM_EmisFiles.inc). An additional NetCDF emission is included in line 2.

Now all emissions from both ASCII file and NetCDF file will be used. In practice some countries might be counted twice. Therefore some new data can be included in the emis_inputlist, to specify which countries to keep or to avoid. Listing 4.6 lines 3–4 will include only 'NO', 'SE' and 'FI' from the first file (ASCII), and take all countries except 'NO', 'SE' and 'FI' from the second file (NetCDF).

Sets of countries can in principle be defined; for now only the set 'EUMACC2' is defined.

Listing 4.6: Mixed emission configuration example.

```
emis_inputlist(1)%name = '/MyPathToEmissions/emislist.POLL',
emis_inputlist(1)%type = 'GNFR_CAMSsectors',
emis_inputlist(2)%name = '/MyPathToEmissions/Emis_GLOB_05.nc',
emis_inputlist(1)%incl(1:) = 'NO','SE','FI',
emis_inputlist(2)%excl(1:) = 'NO','SE','FI',
emis_inputlist(1)%PollName(1:2) = 'voc','sox',
```

It is also possible to restrict the number of pollutants from each of the files. If not all pollutants from CM_EmisFiles.inc are to be read, one can specify a list of pollutants to be included using "PollName". For instance in the example above, the last line specifies that emissions will include only VOC and SOx from the file defined by emis_inputlist(1)%name. If PollName is not specified at all, all pollutants are included (therefore all pollutants from emis_inputlist(2)%name will be included). The specified pollutants must already be defined in CM_EmisFiles.inc. It is possible to disregard the "lonlat" reductions introduced by femis.dat for specific emissions. To do this use the "use_lonlat_femis" flag. Example: switch off emissions covering one region from Emis_GLOB_05.nc as specified by femis, and replace the emissions in that data using emislist.POLL. use_lonlat_femis = F will switch off the reductions, therefore the lonlat coordinate in femis.dat should match the domain covered by emislist.POLL.

Listing 4.7: Do not take into account the lines starting with lonlat in femis.dat for emis_inputlist(1)%name.

```
emis_inputlist(1)%name = '/MyPathToEmissions/emislist.POLL',
emis_inputlist(1)%use_lonlat_femis = F,
emis_inputlist(2)%name = '/MyPathToEmissions/Emis_GLOB_05.nc',
```

An alternative way of combining overlapping emissions, is to use a “mask” approach. This is typically used, when the emissions of one city is known in more details, but one wish to include default regional or global emissions elsewhere. The city emissions are used to set the mask, and that mask is then in turn used by the second emission sources to turn off emissions within the city.

Listing 4.8: Mixed emission using mask configuration example.

```
emis_inputlist(1)%name = '/MyPathToEmissions/emis_local.POLL',
emis_inputlist(1)%set_mask = T,
emis_inputlist(2)%name = '/MyPathToEmissions/Emis_GLOB_05.nc',
emis_inputlist(2)%use_mask = T,
```

There are some details one should take into account: the order of the “set_mask” and “use_mask” matter. The mask should be set before it is used; “before” meaning that the indice of “emis_inputlist” is lower. Also yearly emissions are treated before monthly emissions, therefore one should not use monthly emissions to mask yearly emissions. The mask is set for a given position if emissions at that point are larger than 1.0e-20. If emissions are zero at some point the mask will not be set for that point, and regional emissions will be included there. There is only one mask. Several emissions files can set and use the mask.

Masks can also be set from any field, see own section below.

4.6 Nesting

The model can be run in a large domain and all the concentrations of pollutants stored at fixed intervals (3 hours typically). Then we can define a smaller region within the large domain, and rerun the model in the smaller region, using the stored concentrations at the domain boundaries. It is then possible to make a simulation in a restricted region with fine resolution, but still taking account the effect of pollutants from outside the small region. This is called nesting. The large domain defines the Boundary Conditions (BC, which are only used at the boundaries of the small domain), and possibly the Initial Conditions (IC, which must be defined everywhere in the small domain, but only for the start date).

Depending on what is wanted different “Nesting modes” can be defined. The different options are controlled by the NEST_MODE_READ and NEST_MODE_SAVE variables in Model_config in config_emep.rml file. The mode options are:

NEST_MODE_READ

‘NONE’

do nothing (default).

‘START’

read at the start of run.

‘RESTART’

read at the start of run, but do not overwrite BC. Use this option when you restart a run

without changing the domain size (“checkpoint-restart”). The difference with “START” is only that BC are not overwritten. (This is important in the case of restarting exactly at the start of a month (at time 00:00, the 1st), because

the nest file contains the BC from preceding month, and if 'START' is used it will use those for the new month, i.e. overwrite new BC with old ones.)

'NHOURL'

read at given NEST_NHOURLREAD hourly intervals, if the file is found. NEST_NHOURLREAD is set in Model_config and should be an integer fraction of 24.

'MONTH'

read at start of each month.

NEST_MODE_SAVE

'NONE'

do nothing (default).

'END'

write at end of run.

'NHOURL'

write at given NEST_NHOURLSAVE hourly intervals. NEST_NHOURLSAVE is set in Model_config and should be an integer fraction of 24.

'MONTH'

write after each month (used for checkpoint/restart for instance).

The name of the file to write to is defined by NEST_template_write (also in Model_config). The name of the file to read to IC data from is defined by NEST_template_read_3D. The name of the file to read to BC data from is defined by NEST_template_read_BC; it can be the same file as NEST_template_read_3D. If for example NEST_NHOURLSAVE=3, but NEST_NHOURLREAD = 1, the data is interpolated in time to get a smooth transition between the 3-hourly values (recommended).

In addition, 3D outputs can be asked for at specific date. Example:

Listing 4.9: Write 3D at specific dates example.

```
&Model_config
[...]
  NEST_OUTDATE_NDUMP      = 2,           ! choose two dates
  NEST_outdate(1)= 2019,01,01,00,       ! save at end of the 2018 year
  NEST_outdate(2)= 2018,05,17,00,       ! save also 17th of May 2018
&end
```

4.6.1 Example write BCs

Listing 4.10 shows an example to write every 3 hours into daily Nest/BC files. Output file name is defined by NEST_template_write ('BC_YYYYMMDD.nc'), where 'YYYYMMDD' is replaced by corresponding date.

All advected model variables will be written out for the sub-domain defined by NEST_out_DOMAIN ($x = 60, \dots, 107; y = 11, \dots, 58$). If no NEST_out_DOMAIN is given, the entire model rundomain will be written out.

Listing 4.10: Write BCs configuration example.

```
&Model_config
[...]
  NEST_MODE_READ          = 'NONE',       ! do not read BC
  NEST_MODE_SAVE          = 'NHOURL',     ! write BCs
  NEST_NHOURLSAVE         = 3,           ! every 3 hours
  NEST_template_write     = 'BC_YYYYMMDD.nc' ! to your (daily) BC output file
```

(continues on next page)

(continued from previous page)

```

!----- Sub domain for write modes
NEST_out_DOMAIN      = 60,107,11,58,    ! istart,iend,jstart,jend
&end

```

4.6.2 Read BCs produced by a previous EMEP MSC-W model run

Listing 4.11 shows an example to read every 1 hours from the Nest/BC files created previously by running Listing 4.10. Please note that the model sub-domain for a nested run is set by RUNDOMAIN, as shown in Listing 4.1.

Listing 4.11: Read BCs configuration example.

```

&Model_config
[... ]
  NEST_MODE_READ      = 'NHOURL',        ! read external BC
  NEST_NHOURLREAD     = 1,                ! every hour
  NEST_template_read_BC = 'BC_YYYYMMDD.nc' ! your (daily) BC input file
  NEST_MODE_SAVE      = 'NONE',          ! do not write BCs
&end

```

Note that NEST_NHOURLREAD can (and should) be smaller than the value used for NEST_NHOURLSAVE. The values between saved dates will then be interpolated in time, giving a smoother transition.

4.6.3 Reduce the size of BC files

The size of the files obtained in a nesting configuration can be very large if the out_DOMAIN is large. If the inner domain is known in advance, only the part matching exactly the part needed to construct the BC of the small domain can be stored. Define NEST_MET_inner in &Model_config, which should be a link to any metadata of the inner grid; it will only be used to define the projection parameters of the inner grid (i.e. dates and other content do not matter).

Listing 4.12: Inner domain options for nested BC output example.

```
&Model_config
[...]
  NEST_MET_inner = 'inner_domain/wrfout_d03_2015-01-01_00:00:00',
  NEST_RUNDOMAIN_inner = 12, 136, 100, 300,
&end
```

You cannot use the implicit dates (“YYYY” etc.); you must put explicit numbers. Note that the file will have the same dimensions, but zeros are put into the unused parts. The NetCDF internal compression will take care of reducing the actual size, as measured by used disc space.

If a BC file has been created using the NEST_MET_inner method, it cannot be used for initializing concentrations at the start of the run. A separate file has to be created for initializations. This file can then be used by the inner grid by defining NEST_template_read_3D in config_emep.nml.

4.6.4 Read external BCs

So far only BC created by the model itself have been used. Reading external BCs, i.e. produced by other means (another model for example) is more involved. The chemical species may be different and the vertical levels also. The vertical axis and variables in the file need then to be mapped to the corresponding model variables.

Listing 4.13 shows an example to read every 3 hours from an external BC file. The model will read 3 variables from MyBC.nc: O₃, NO, and NO₂. The mapping between the MyBC.nc variables and the corresponding model variables is defined in the ExternalBICs_bc namelist.

Listing 4.13: External BCs configuration example.

```
&Model_config
[...]
  NEST_MODE_READ      = 'NHOURL',      ! read external BC
  NEST_NHOURLREAD     = 3,              ! every 3 hours
  NEST_template_read_BC = 'MyBC.nc'     ! from your BC input file
  NEST_MODE_SAVE      = 'NONE',        ! do not write BCs
  USE_EXTERNAL_BIC    = T,
  EXTERNAL_BIC_NAME= 'MyBICScenario', ! variable mapping, see ExternalBICs_bc
  TOP_BC              = T,              ! use top BC also from your BC file
  filename_eta        = 'filename.zaxis', ! text file containing the
                                      ! description of your BC file
&end
&ExternalBICs_bc
  description='MyBICScenario','Version name',3, ! name,version,size
  map_bc=! emep,external,frac,wanted,found,IXADV,
    'O3' , 'O3_VMR_inst' ,1.0,T,F,-1,
    'NO' , 'NO_VMR_inst' ,1.0,T,F,-1,
    'NO2', 'NO2_VMR_inst',1.0,T,F,-1,
&end
```

Vertical coordinate

In order to determine the vertical levels on the external BC file ('MyBC.nc' in Listing 4.13), the following checks will take place in the following order:

1. **η (eta) coordinate:**
If the variable *hyam* (hybrid a coefficient at layer midpoint) is found, η is calculated from *hyam* and *hybm* variables on the file.
2. **σ (sigma) coordinate:**
If the vertical level is indexed by variable *k*.
3. **η coordinate:**
If the file defined by variable *filename_eta* exist ('filename.zaxis' in Listing 4.13), η is derived from *vct* and *vctsize* variables on the file.
4. **pressure coordinate:**
If the vertical level is indexed by variable *lev*.

Independent of the coordinates of the BC file, the BC levels will be interpolated into EMEP model levels. If the BC file level structure is not recognized, and there is no *filename_eta* provided, the model will write an error message and stop execution.

An example of the *filename_eta* for EMEP model levels is given below. Here the *vct* variable describes the model level boundaries in hybrid eta coordinate:

Listing 4.14: *filename_eta* for EMEP model levels.

```

zaxistype = hybrid
size      : 20
name      : k
longname   : vertical sigma coordinates
units     : sigma_level
levels    : 0.0200 0.0600 0.1000 0.1425 0.1950 0.2635 0.3470
           0.4365 0.5215 0.5990 0.6695 0.7330 0.7895 0.8390
           0.8815 0.9170 0.9455 0.9670 0.9820 0.9940
vctsize    = 42
vct        =
           10000. 09600. 09200. 08800. 08350. 07750. 06980.
           06080. 05190. 04380. 03640. 02970. 02370. 01840.
           01380. 00990. 00670. 00420. 00240. 00120. 00000.
           0.0000 0.0400 0.0800 0.1200 0.1650 0.2250 0.3020
           0.3920 0.4810 0.5620 0.6360 0.7030 0.7630 0.8160
           0.8620 0.9010 0.9330 0.9580 0.9760 0.9880 1.0000

```

vct is the vertical coordinate table describing the hybrid a and b parameters at the layer interfaces in η coordinate system (*hyai* and *hybi*). They must respect the following constraints:

- $hyai_1 = 0$ and $hybi_1 = 1$;
- $hyai_0 = P_t$ and $hybi_0 = 0$.
- P_t is the pressure at top of the model domain.

In this file, the first 21 values in *vct* represent *hyai* and the remaining 21 represent *hybi* values in hPa.

Variable mapping

The variables to be used from the external boundary condition data are given in the `ExternalBICS_bc` namelist in the `config_emep.nml` file. In [Listing 4.13](#), `map_bc` maps 3 variables in the 'MyBC.nc' file to 3 model variables (O_3 , NO, and NO_2). On each line of `map_bc` contains the 6 elements:

1. Variable name in EMEP MSC-W model, e.g. ' O_3 '.
2. Variable name in the External BC data file, e.g. ' $O_3_VMR_inst$ '.
3. External BC component to EMEP component fraction, e.g. 1.0 .
4. Is this component wanted or not, set to `T` or `.true.` to read the variable.
5. Was the BC variable found on the file, will be set by the model on run time.
6. Index of the advected model variable, will be set by the model on run time.

The fraction is helpful, when one has to map a variable that is explicitly not in the model but a fraction of that variable can be mapped to a matching variable in the model.

Unit conversions are delegated to the `Units_mod.f90` module. The supported units are: $\mu g/m^3$, $\mu g(S)/m^3$, $\mu g(N)/m^3$, $\mu g(C)/m^3$, ppb, mixing ratio (mol/mol) and mass mixing ratio (kg/kg).

If the BC data has different units, either convert them into one of the above mentioned units in pre-processing or add the respective conversion factor in the module `Units_mod.f90`.

4.7 config: Europe or Global?

The EMEP model has traditionally been run on the EMEP grid covering Europe, and using meteorology from the ECMWF IFS model. In this environment, we typically set several configuration variables to make use of Euro-specific data. In other regions it is better to make use of the model's 'global' settings, which will ensure better handling of vegetation (eg LAI) changes, convection, and various emission settings.

Listing 4.15: Typical European/EMEP settings.

```
USES%DEGREEDAY_FACTORS = T,      ! though F is okay too
USES%PFT_MAPS = F,
USES%MonthlyNH3 = 'LOTOS', ! Better monthly profile, for Europe only!
USES%CONVECTION = F,
USES%SOILNOX_METHOD = 'NoFert', ! If using ECLIPSEv6
```

Listing 4.16: Typical non-European/global settings.

```
USES%DEGREEDAY_FACTORS = F
USES%PFT_MAPS = T,      ! PFT LAI tests
USES%MonthlyNH3 = '-', ! Better monthly profile, for Europe only!
USES%CONVECTION = T,
```

The `DEGREEDAY_FACTORS` setting triggers the use of degree-days in controlling residential combustion (GNFR C/SNAP2) emissions. This requires pre-processed files of heating degree days. Such files (`DegreeDayFactors.nc`) can be produced from any meteorology, but the difference in results even for Europe is not too significant. In other regions of the world emissions from residential combustion may not be as dependent on degree-days as in Europe, and so this setting should probably be false.

`PFT_MAPS=T` triggers the use of a global file which provides monthly variations in leaf area index (LAI) for different vegetation types. This controls deposition and biogenic VOC emission parameters. In Europe, a simpler latitude-dependent system is used (based upon DO3SE), and so `PFT_MAPS` should be set F.

MonthlyNH3='LOTOS' is also only relevant for European simulations; and indeed any non-European runs are better off with monthly emissions for that particular area. To disable the monthly NH3 profile from LOTOS-EUROS and instead use the monthly profile defined, for example, in the MonthlyFacFile, set MonthlyNH3='-' explicitly. If not set explicitly, MonthlyNH3='LOTOS' will be set automatically for European simulation runs.

CONVECTION is difficult. In principle, all models runs should use the T setting, but for Europe we find it degrades the model results too much and we use F. The problem is likely that the sub-grid processes behind convection are so complex and the parameterisation is very uncertain. Note also that in Config_module we have the default setting CONVECTION_FACTOR=0.33, which may be changed to allow more or less influence of this variable.

4.8 config: SOILNOX

Listing 4.17: Soil-NO settings.

```
USES%SOILNOX_METHOD = 'NoFert', ! If using ECLIPSEv6 or EMEP European
USES%SOILNOX_METHOD = 'Total',  ! If using ECLIPSEv5
```

The model makes use of global 0.5 degree data from the CAMS81 project, but the user needs to specify the data to be used from this system. The choice, between Total and NoFert depends on the anthropogenic emission inventory in use. If this anthropogenic inventory already includes fertilizer-induced soil NO emissions, then the NoFert data from the soil-NO system is used. If not, choose Total.

4.9 Country Variable (CV) format

In the CV format, emissions are organised in a number of files (Emis_sourceFiles(i_file)), each files containing a number of sources (Emis_sourceFiles(i_file)%source(j_source)). A source can be any 2D field. It can also be a 3D field, if the third dimension is the sector. The sources are associated to country through a country code or a country code. The file must have a 'lon' and a 'lat' variable, showing longitude and latitudes of each grid point. 'lon' and 'lat' must be 1D variables if the projection is 'lon lat', 2D otherwise.

The code directory contain a Python script that can create an emission file in this format: emissions_TXT2ncCV.py

The file and sources can be characterized by a set of variables. In general these variable can be set by, and in order of increasing priority:

1. Default value
2. Global attribute read in the netcdf file
3. Variable attribute read in the netcdf file
4. Value set for Emis_sourceFiles(i)%XXX in config_emep.nml
5. Value set for Emis_sourceFiles(i)%source(s)%XXX in config_emep.nml

Exceptions to the priority rule are:

- maskID cannot be set by attributes in the netcdf file
- the file and source 'factor' are on top of each other, not replaced
- boolean parameters (like apply_femis), are used as "and" (i.e. if any is false, the result is false)

List of file attributes (default in parenthesis):

- filename ('NOTSET') Name of the file (with path)

- projection ('lon lat') Only three categories 'lon lat', 'native' or any other (for example 'Lambert' or 'Stereographic' would give the same result). 'native' means that emissions are given in the same grid as the model grid and the data is not interpolated (use it if you can).
- grid_resolution (an approximate value is computed from the lon and lat, if no value is given). It does not need to be exact (cannot be exact on a sphere anyway!). This grid_resolution steers the interpolation algorithm; A large value will force the code to subdivide each emission gridcell in large number of pieces, that are assigned to the model grid. Larger values means smoother interpolation, but more cpu time.
- periodicity ('time') How often the values are updated. Can be 'yearly', 'monthly', 'hourly' or 'time'. 'hourly' or 'time' means that the time as defined in the netcdf is used to define when to fetch a new record. The timestamp must correspond to the end of the time period of validity. For 'yearly' monthly timefactors are applied, if a sector is defined. For 'monthly' and 'yearly', an hourly timefactor is applied if a sector is defined. For 'hourly' or 'time', no additional timefactors are applied.
- factor (1.0) multiplicative factor for all sources in the file
- units ('NOTSET') will be used as default for sources units if set.
- apply_femis (true) whether to apply the femis reductions to the sources of this file.
- include_in_local_fractions (true) whether to take sources from this file into account for the local fraction calculations
- countrycode (-1) will be used as default for sources country code if set. Use rather country_ISO if you can.
- country_ISO ('NOTSET') will be used as default for sources country ISO code (for example 'FR' for France) if set.
- sector (-1) will be used as default for sources sector if set.
- SECTORS_NAME or sectorsName ('GNFR_CAMS') . It must be set to one of the SECTORS%name defined ('GNFR_CAMS' for example).
- mask_ID ('NOTSET') the name of the mask, if you want to apply one.
- mask_ID_reverse ('NOTSET') the name of the mask, if you want to apply one in the complementary region.
- country_ISO_incl ('NOTSET') If used, only use emissions from the countries in the list (see example below)
- country_ISO_excl ('NOTSET') If used, do not take into account emissions from countries in the list (see example below)

List of source attributes:

- varname ('NOTSET') The name as used in the netcdf file
- species ('NOTSET') Either one of the emission group species, as defined in CM_EmisFile.inc (generally sox, nox, pm25, pmco, nh3, co, voc)
- factor (1.0) multiplicative factor. Can be used to change units to model definitions. Comes on top of the file multiplicative factors and possibly other factors.
- units ('mg/m2/h') Units *after* the factor multiplication.
- countrycode (-1) will be used as default for sources country code if set. Use rather country_ISO if you can.
- country_ISO ('N/A') the country code, as defined in Country_mod.f90 (for example 'FR' for France). 'N/A' is a valid code, but it does not correspond to any country.
- apply_femis (true) whether to apply the femis reductions to this source.
- include_in_local_fractions (true) whether to take this source into account for the local fraction calculations
- mask_ID ('NOTSET') the name of the mask, if you want to apply one.

- mask_ID_reverse ('NOTSET') the name of the mask, if you want to apply one in the complementary region.

If a dimension with name “sector” is found, it is assumed that the third dimension of emission variables represent the sector index.

The idea is that only variables that clearly are required in a specific context need to be set; if the value can be inferred from other information, the code should do it. Depending of the type of source, not all variables are used.

Note about species: These can be interpreted in one of three categories

1. emitted species (nox,sox,pm25 ...) with sector (1...11 (or 13)) (“sector species”)
2. individual species (SO2, NO, NO2, ...) with sector. The species MUST be one of the splitted species. These will be treated as one of the “sector species” from 1. (but not splitted of course). Careful with units, it follows the same rules as “sector species”; molecular weight for SO4 for example is considered “as SO2” and NO “as NO2”. You can correct the units using the factor attribute.
3. individual species (SO2, APINENE, O3 ...) without sector (<=0, or not specified). No timefactors, vertical release heights or splits are applied. In this case the emissions are summed up in setup_rcemis (not in EmisSet)

Example:

Listing 4.18: Read a field named “emis_tra” from a file
“/path/ECLIPSE_V6a_CLE_base_PM25.nc”.

```
Emis_sourceFiles(1)%filename = '/path/ECLIPSE_V6a_CLE_base_PM25.nc',
Emis_sourceFiles(1)%sectorsName = 'GNFR',
Emis_sourceFiles(1)%projection = 'lon lat',
Emis_sourceFiles(1)%periodicity = 'yearly',
Emis_sourceFiles(1)%source(1)%varname='emis_tra',
Emis_sourceFiles(1)%source(1)%species='pm25',
Emis_sourceFiles(1)%source(1)%sector=6,
Emis_sourceFiles(1)%source(1)%factor=1000000.0,!kt->kg
Emis_sourceFiles(1)%source(1)%units='kg',
```

If two CV emission files are wished to be combined in a single simulation, for example so that different emission scenarios can be applied inside and outside of the EMEP region (UNECE excl. NA), one can use the Emis_sourceFiles(1)%country_ISO_excl(1:) and Emis_sourceFiles(1)%country_ISO_incl(1:) options. The below dummy example applies a 2015 baseline scenario inside the EMEP region and 2050 MFR scenario in the rest of the world (ROW), based on 0.5 degree emission input data files.

Listing 4.19: Include or exclude countries from emission input files.

```
Emis_sourceFiles(1)%filename = '/path_to_global_0.5deg_2015_baseline.nc',
Emis_sourceFiles(2)%filename = '/path_to_global_0.5deg_2050_MFR.nc',

Emis_sourceFiles(1)%country_ISO_incl(1:48) = "AL", "AM", "AT", "AZ", "BY", "BE", "BA",
↪ "BG", "CY", "CZ", "DE",
                                         "DK", "EE", "FI", "FR", "GE", "GR", "HR",
↪ "HU", "IS", "IE", "IT",
                                         "LV", "LT", "LU", "MT", "NL", "NO", "PL",
↪ "PT", "MD", "RO", "SI",
                                         "ES", "SE", "CH", "MK", "TR", "SK", "UA",
↪ "RS", "ME", "GB", "KG",
                                         "RUSS_EURO", "RUSS_ASIA", "KZT", "FSUA",
```

(continues on next page)

(continued from previous page)

```
Emis_sourceFiles(2)%country_ISO_excl(1:48) = "AL", "AM", "AT", "AZ", "BY", "BE", "BA",
↪ "BG", "CY", "CZ", "DE",
                                         "DK", "EE", "FI", "FR", "GE", "GR", "HR",
↪ "HU", "IS", "IE", "IT",
                                         "LV", "LT", "LU", "MT", "NL", "NO", "PL",
↪ "PT", "MD", "RO", "SI",
                                         "ES", "SE", "CH", "MK", "TR", "SK", "UA",
↪ "RS", "ME", "GB", "KG",
                                         "RUSS_EURO", "RUSS_ASIA", "KZT", "FSUA",
```

4.10 Masks

Typically you have got fine scale emissions for a small region of interest, a city for instance. You may want to remove that area from the coarse scale emissions, and replace it with your own. The mask allows you to define a specific region (the mask). To define which gridcells to include in your local region, you must find a suitable variable that shows the region of interest. It could be for example the PM emissions in your local area.

A “mask” can be defined for instance with:

Listing 4.20: Define a cell-fraction mask, example. The mask value represents the fraction of the grid-cell area to be reduced.

```
EmisMask(1)%filename = '/mypath/myfile.nc', ! name of the netcdf file to read from
EmisMask(1)%cdfname = 'London_PM', ! name of the variable to read from the file
EmisMask(1)%type = 'CELL-FRACTION', ! the mask value is part of the reduction,
↪ default
EmisMask(1)%ID = 'LONDON', ! the name you give to that mask
EmisMask(1)%fac = 0.85, ! multiplicative factor to use. Default 0.0

! define species to be read from the emission file separately, as if reading from
↪ different files
Emis_sourceFiles(1:7)%filename=7*'path/Emis_CAMS_v5_1_with_Ref2_0_1_GNFR_CAMS_01005deg_
↪ 2018_CAMS2_40_U5.nc',
Emis_sourceFiles(1:7)%species='sox','nox','voc','nh3','co','pm25','pmco',
! apply mask to pm25 and pmco emissions
Emis_sourceFiles(6:7)%mask_ID =2*'London_PM',
```

Listing 4.21: Define a mask with lower and upper threshold, example

```
EmisMask(1)%filename = '/mypath/myfile.nc', ! name of the netcdf file to read from
EmisMask(1)%cdfname = 'London_PM', ! name of the variable to read from the file
EmisMask(1)%type = 'THRESHOLD',
EmisMask(1)%ID = 'LONDON', ! the name you give to that mask
EmisMask(1)%threshold = 1.0E-10, ! the mask is set at any point larger than the threshold
EmisMask(1)%threshold_max = 100, ! ... and smaller than the threshold_max value (default
↪ 1E60)
EmisMask(1)%fac = 0.85, ! multiplicative factor to use. Default 0.0
```

Several masks can be defined. Each mask is identified by their “ID”. If you want to include in the region also the gridcell which are zero, you can set the threshold slightly negative (-1.0E-10), to include the entire region covered by the variable (otherwise zero values would be defined equivalently to outside of region). To include for example all

values=23, but not 24, set `EmisMask(1)%threshold = 22.5` and `EmisMask(1)%threshold_max = 23.5`

A mask defines only a region. It is not directly related to any pollutant.

The masks defined here, will also be applied on files from `emis_inputlist` (old format), if `use_mask` is set (but the multiplicative factor is always 0.0 for old format emissions). It is however not possible to set masks by both systems simultaneously. In the old format only one mask can be used at a time. It will be the reunion of all masks produced by the system above (the ID is meaningless and cannot be specified in old format).

To be used with the Local Fractions (see below), one can also define a set of regions defined by integer numbers. For this one must define the ID with the keyword `NUMBER`:

Listing 4.22: Define a set of masks with integers, example

```
EmisMask(1)%filename = '/mypath/myfile.nc', !name of the netcdf file to read from
EmisMask(1)%cdfname = 'region_id', !name of the variable to read from the file. The
↪variable must be an integer!
EmisMask(1)%type = 'NUMBER',
EmisMask(1)%ID = 'specific-mask-name',
```

4.11 Other less used options

There are many internal settings that are set to a default value by the model. These default values can often be overridden by setting specific values for keywords in `config_emep.nml`.

The `RUNDOMAIN` is divided by the model into subdomains which are assigned to each processor. Normally this partitioning is done such that the X and Y direction are divided into approximately equal number of parts. For runs in lat lon projection containing poles the Y division is done into one or two parts, so that each processor has the same share of pole regions. The default partitioning can be overridden using the `DOMAIN_DECOM_MODE` parameter in `config_emep.nml`. Recognized values are: 'X*Y', 'XY', 'X*I', 'Y=I', 'X', 'I*Y', 'X=I', 'Y', '2*Y', 'X=2', 'X*2', 'Y=2'. See also in `Par_mod.f90` for details.

The main timestep parameter `dt_advec` can be set manually in `config_emep.nml` (in seconds, real number). It must be a factor of 3600.

`JUMPOVER29FEB` : if set to T, will not treat the 29th of February even for leap years. (Useful if that date is missing, for instance in Climate runs).

`NETCDF_DEFLATE_LEVEL`: The level of compression used in the NetCDF output files (integer). Negative values means netcdf3 format.

`END_OF_EMEDAY`: (integer) Hour when to start and end a "day". For daily averages gives for example averages from 06:00 to 06:00 next day, if `END_OF_EMEDAY = 6`.

`USES%SKIP_INCOMPLETE_OUTPUT` : if T, will skip daily/monthly/fullrun output for runs under 1/28/181 days.

4.12 Defining own sectors

Emissions can be assigned to a sector. A sector defines three properties:

1. Split into species
2. Emission height release distribution
3. Timefactors

The set of splits is defined in files “emissplit.defaults.POLL” and “emissplit.specials.POLL”. The release height distribution can be chosen among the 6 predefined values, or defined in config_emep.nml. The timefactors are defined in “MonthlyFac.POLL” “DailyFac.POLL” and “HourlyFac.INERIS”.

Which height/split/timefac is chosen for a given sector can be also be controlled by the user. GNFR_CAMS sectors are predefined (SNAP also, but we recommend to use GNFR_CAMS). The values for split, emission release height and timefactors can be defined through settings in the config_emep.nml settings. The following will reproduce the default settings:

Listing 4.23: Sector definitions equivalent to the predefined sectors for GNFR_CAMS.

```
SECTORS_ADD(1) = 'GNFR_CAMS', 'GNFR_A', 'sec01', 1, 1, 1, 'Public Power', 'ALL',
SECTORS_ADD(2) = 'GNFR_CAMS', 'GNFR_B', 'sec02', 3, 3, 2, 'Industry', 'ALL',
SECTORS_ADD(3) = 'GNFR_CAMS', 'GNFR_C', 'sec03', 2, 2, 3, 'OtherStationaryComb', 'ALL',
SECTORS_ADD(4) = 'GNFR_CAMS', 'GNFR_D', 'sec04', 5, 5, 4, 'Fugitive', 'ALL',
SECTORS_ADD(5) = 'GNFR_CAMS', 'GNFR_E', 'sec05', 6, 2, 5, 'Solvents', 'ALL',
SECTORS_ADD(6) = 'GNFR_CAMS', 'GNFR_F', 'sec06', 7, 2, 6, 'RoadTransport', 'ALL',
SECTORS_ADD(7) = 'GNFR_CAMS', 'GNFR_G', 'sec07', 8, 2, 7, 'Shipping', 'ALL',
SECTORS_ADD(8) = 'GNFR_CAMS', 'GNFR_H', 'sec08', 8, 7, 8, 'Aviation', 'ALL',
SECTORS_ADD(9) = 'GNFR_CAMS', 'GNFR_I', 'sec09', 8, 2, 9, 'Offroad', 'ALL',
SECTORS_ADD(10) = 'GNFR_CAMS', 'GNFR_J', 'sec10', 9, 6, 10, 'Waste', 'ALL',
SECTORS_ADD(11) = 'GNFR_CAMS', 'GNFR_K', 'sec11', 10, 2, 11, 'AgriLivestock', 'ALL',
SECTORS_ADD(12) = 'GNFR_CAMS', 'GNFR_L', 'sec12', 10, 2, 12, 'AgriOther', 'ALL',
SECTORS_ADD(13) = 'GNFR_CAMS', 'GNFR_M', 'sec13', 5, 5, 13, 'Other', 'ALL',
SECTORS_ADD(14) = 'GNFR_CAMS', 'GNFR_A1', 'sec14', 1, 1, 1, 'PublicPower_Point', 'ALL',
SECTORS_ADD(15) = 'GNFR_CAMS', 'GNFR_A2', 'sec15', 1, 3, 1, 'PublicPower_Area', 'ALL',
SECTORS_ADD(16) = 'GNFR_CAMS', 'GNFR_F1', 'sec16', 7, 2, 16,
↳ 'RoadTransportExhaustGasoline', 'ALL',
SECTORS_ADD(17) = 'GNFR_CAMS', 'GNFR_F2', 'sec17', 7, 2, 17, 'RoadTransportExhaustDiesel',
↳ 'ALL',
SECTORS_ADD(18) = 'GNFR_CAMS', 'GNFR_F3', 'sec18', 7, 2, 18, 'RoadTransportExhaustLPGgas',
↳ 'ALL',
SECTORS_ADD(19) = 'GNFR_CAMS', 'GNFR_F4', 'sec19', 7, 2, 19,
↳ 'RoadTransportNonExhaustOther', 'ALL',
```

The name in the first column (GNFR_CAMS) should match the sector name defined in the emission file, or be given in the config_emep.nml (see example below). The second name can be chosen by the user (it will be used if SecEmisOut-Wanted is T)), the third name is the variable name ending in the netcdf emission file (the start being the name of the pollutant) only implemented for “fraction” format, the first number refers to the index used for time factors, the second number the index in the Emis_h array, the third number is the index used in the split files. The long name in the second last column is a longer description, and the last column refers to the species to be included for this sector (if they exist in the emission file).

From version 4.45 and onwards, sector type or names are compulsory for each emission file. This can be given either in the config_emep.nml or in the NetCDF file.

Here is an example of how to define a new sector with a new height distribution, used by emissions given in a separate file.

Listing 4.24: Settings for defining the pm 2.5 emissions from the file MyEmis.nc with variable name 'pm25_MyCar', with emission released between 20 and 50 meters.

```
Emis_Zlevels(1:)20.0, 50.0, 92.0, 184.0, 324.0, 522.0, 781.0, 1106.0,
Emis_h(1:,1) = 0.000, 0.000, 0.000, 0.003, 0.147, 0.400, 0.300, 0.150,
Emis_h(1:,2) = 1.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,
Emis_h(1:,3) = 0.060, 0.067, 0.093, 0.750, 0.030, 0.000, 0.000, 0.000,
Emis_h(1:,4) = 0.050, 0.063, 0.087, 0.700, 0.100, 0.000, 0.000, 0.000,
Emis_h(1:,5) = 0.020, 0.034, 0.046, 0.600, 0.300, 0.000, 0.000, 0.000,
Emis_h(1:,6) = 0.000, 0.000, 0.000, 0.410, 0.570, 0.020, 0.000, 0.000,
Emis_h(1:,7) = 0.200, 0.300, 0.020, 0.044, 0.066, 0.094, 0.123, 0.153,
Emis_h(1:,8) = 0.000, 1.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000,

emis_inputlist(1)%name='GNFR.nc',
emis_inputlist(2)%name='MyEmis.nc',
emis_inputlist(2)%sector='MyNewSector',
SECTORS_ADD(1) = 'MyNewSector', 'MyTestSector', 'MyCar', 7, 8, 6, 'Special car exhaust
↪', 'pm25',
```

Note that if you define new splits, you must include defaults values in all the default files (even if they are overwritten by the specials).

4.13 Local Fractions (under development)

When chosen, the Local Fractions (used by the urban EMEP, uEMEP) will be outputted in separate files.

config_emep.nml settings:

Listing 4.25: Local Fractions flag example

```
USES%LocalFractions = T, ! T for computing Local Fractions, F otherwise
!Local Fractions frequency of output (separate file for each). Can be any of: YEAR,
↪MONTH, DAY, HOUR, HOUR_INST
lf_set%YEAR = T, !average value for full run in output
lf_set%Nvert = 14, !How many vertical level to include in treatment. Should be higher
↪than highest emissions

!NB: for now lf_src(1)%dist will be used for all sources
lf_src(1)%dist = 5, !how far the neighbors can be in each direction (NB: high cost for
↪large dist)

!Local Fractions pollutants and sectors to include:
lf_src(1)%species="pm25", ! any of EMIS_File: "sox ", "nox ", "co ", "voc ", "nh3 ",
↪"pm25", "pmco"
lf_src(1)%sector=0, !0 means sum of all sectors
lf_src(1)%make_fracsum=T,
lf_src(2)%species="pm25", ! any of EMIS_File: "sox ", "nox ", "co ", "voc ", "nh3 ",
↪"pm25", "pmco"
lf_src(2)%sector=0, !0 means sum of all sectors
```

(continues on next page)

(continued from previous page)

```
lf_src(2)%res=5, ! sources are squares with size 5x5 gridcells
lf_src(2)%make_fracsum=T,
lf_src(3)%species="nox ",
lf_src(3)%sector=0,
lf_src(4)%species="nox ",
lf_src(4)%sector=8,

! lf_set%DOMAIN = 370, 420, 270, 320, !which domain to include in output. Will save disk,
↪ but not CPU to reduce.
```

If one wants to include many species, sectors and res values, without writing one entry per source, one can use the following syntax:

Listing 4.26: Local Fractions sectors arrays example

```
USES%LocalFractions = T, ! T for computing Local Fractions
lf_species(1:2)%name = 'pm25', 'nox',
lf_species(1)%sectors(1:) = 0, 1, 2, 8,
lf_species(1)%res(1:) = 1, 4,
lf_species(2)%sectors(1:) = 0, 1, 2, 8,
lf_species(2)%res(1:) = 1, 4,
```

The corresponding lf_src values will then be added to the already defined lf_src ($3 \times 4 \times 2 = 24$ new sources in this example). You can not with this syntax combine different res for different sectors for the same species.

Note that the files can be very large if hourly outputs and/or many neighbors are requested.

4.14 Local Fractions for tracing natural emissions (under development)

Natural emissions are assumed emitted from surface (in current version). So far only the DMS emissions can be tracked. To include DMS outputs:

Listing 4.27: Example for DMS output

```
lf_src(1)%name = 'DMS',
lf_src(1)%dist = 2, !will track over up to 2 gridcells in all directions
lf_src(1)%nhour = 1, ! will track separately emissions every 1 hour, and reset every 24
↳hours
lf_set%HOURL_INST = T, ! output instantaneous values every hour
```

4.15 Local Fractions for Source Receptor style runs

Local fractions can also be used to make traditional Source Receptor (or blame) matrices, in a single run. (the %res flag is then not used)

Listing 4.28: Local Fractions Country source receptor type example

```
USES%LocalFractions = T, ! T for computing Local Fractions
!Local Fractions frequency of output (separate file for each). Can be any of: YEAR,
↳MONTH, DAY, HOUR, HOURL_INST
lf_set%YEAR = T, !average value for full run in output
lf_set%Nvert = 14, !How many vertical level to include in treatment. Should be higher
↳than highest emissions

!Local Fractions pollutants and sectors to include:
lf_src(1)%species="pm25", ! any of EMIS_File: "sox ", "nox ", "co ", "voc ", "nh3 ",
↳"pm25", "pmco"
lf_src(1)%type='country', ! Means make country style SR
lf_src(1)%drydep=T, !means make country dry deposition maps too
lf_src(1)%wetdep=T, !means make country wet deposition maps too

lf_src(2)%species="pmco", ! any of EMIS_File: "sox ", "nox ", "co ", "voc ", "nh3 ",
↳"pm25", "pmco"
lf_src(2)%type='country', ! Means make country style SR
lf_src(2)%drydep=T, !means make country dry deposition maps too
lf_src(2)%wetdep=T, !means make country wet deposition maps too

! Specify which countries and sectors
lf_country%sector_list(1:8)=0,1,2,3,4,5,6,7,
lf_country%list(1:20)='FR','IT','DE','ES','NO','NL','SE','PL','AT','BE','BG','DK','FI',
↳'GR','HU','PT','RO','CH','TR','GB',
lf_country%group(1)%name='NORDIC', !any name given to the group (used as output name)
lf_country%group(1)%list(1:)='NO','DK','SE','FI', ! countries included in the group
```

Instead of defining countries in the emission files, one can define “source regions” in a separate netcdf file. Each region must have an integer value. The values to be included as a source region are then specified by a list or as the minimum and maximum value to include (defining both a list and an interval is allowed, but only one list and one interval). The maskfile should be defined too. For example to include masks 2,5,8,12 and all mask between 100 and 357 (included):

Listing 4.29: Local Fractions mask regions source receptor example

```
EmisMask(1)%filename='municip_mask/municip_mask_500m.nc',
EmisMask(1)%cdfname =region_id',
```

(continues on next page)

(continued from previous page)

```
EmisMask(1)%type      ='NUMBER',
EmisMask(1)%ID        ='municip_mask',

lf_country%mask_val(1:) = 2,5,8,12,
lf_country%mask_val_min = 100,
lf_country%mask_val_max = 357,
lf_country%sector_list(1:1)=0, ! sum of all sectors
lf_country%cellmask_name(1:1) = 'municip_mask',
lf_src(1)%species='pm25',
lf_src(1)%type='country',
mask2name(1)='Oslo',      !map id numbers to names (for use in output)
mask2name(2)='Eigersund',
```

If a value is within the min and max range, but does not appear in the mask file, it will not be taken into account (meaning it is ok to specify a range that covers all the masks values, even if some values are not defined on the mask)

(under development): use mask file with fraction of emissions in each gridcell:

Listing 4.30: Local Fractions using mask with fractions of gridcells

```
EmisMask(1)%filename = '/ec/res4/hpcperm/fan/Data/Masks/cameo_city_masks.nc',
EmisMask(1)%cdfname = 'TouHan',
EmisMask(1)%type = 'CELL-FRACTION',
EmisMask(1)%ID = 'ToHa',

EmisMask(2)%filename = '/ec/res4/hpcperm/fan/Data/Masks/cameo_city_masks.nc',
EmisMask(2)%cdfname = 'PorUtr',
EmisMask(2)%type = 'CELL-FRACTION',
EmisMask(2)%ID = 'PoUt',

lf_country%cellmask_name(1) = 'ToHa',
```

In this example two masks are defined (those can be used for traditional SR runs too), and only the mask with name “PoUt” is used as a “country” in the LF run.

4.16 Local Fractions for Sensibilities with full chemistry

The full chemistry can be included. This option is under development, and only limited options are available. The cpu cost is high, approximatively 20 times the cost without this option (independently of the number of sources tracked). To use this option the fortran code must be prepared with the script `utils/mk.LF_Chem`. Example of config settings:

Listing 4.31: Local Fractions Country source receptor type example

```
USES%LocalFractions = T, ! T for computing Local Fractions
lf_set%YEAR = T, !average value for full run in output
lf_set%Nvert = 14, !How many vertical level to include in treatment. Should be higher_
↳ than highest emissions
lf_set%full_chem = T, ! to indicate that all species must be included
lf_set%EmisDer_all = T, ! reduce voc, sox, nox, nh3 together (separately if F)

! Specify which countries and sectors
lf_country%sector_list(1:)=0,1,8,
```

(continues on next page)

(continued from previous page)

```

lf_country%list(1:20)='FR','IT','DE','ES','NO','NL','SE','PL','AT','BE','BG','DK','FI',
↳ 'GR','HU','PT','RO','CH','TR','GB',
lf_country%group(1)%name='NORDIC', !any name given to the group (used as output name)
lf_country%group(1)%list(1:)= 'NO','DK','SE','FI', ! countries included in the group

! Specify which species or group of species must be outputted
lf_spec_out(1)%name='O3', !single species
lf_spec_out(2)%name='NO', !single species
lf_spec_out(3)%name='NO2', !single species
lf_spec_out(4)%name='SIA', !group of species
lf_spec_out(4)%species(1:)= 'SO4','NO3_f','NO3_c','NH4_f', !species to include in the
↳ group
lf_spec_out(4)%species_fac(1:)=1.0, 1.0, 1.0, 1.0, !weights (default 1.0)
lf_spec_out(5)%name='NOx', !group of species
lf_spec_out(5)%species(1:)= 'NO','NO2', !species to include in the group

lf_set%MDA8 = T, !special: make AvgMDA8_6month. NB: requires that O3 is outputted too

```


5.1 Time steps

There are three types of timesteps in the model: `METSTEP`, `dt_advec` and `dt_chem`.

`METSTEP` is the time interval between two readings of the meteorological file. It is not set by the user, but uses whatever is used as meteorological input. It must be a divisor of 24 hours. Three hours is standard, but other intervals are used too.

`dt_advec` is the “time splitting” interval, i.e. the time between two sequences of advection-chemistry/emissions-deposition. (Note that the name is misleading, the advection can have a smaller internal timestep if it is required by the Courant number). The value of `dt_advec` must be an entire fraction of one hour (units seconds). It is determined by the model using the grid resolution. For a grid resolution larger than 61 km it is 1800s, or 1200s if larger than 21km, or 900s if larger than 11km, or 600s if larger than 6km, or 300s if larger than 2km or 100s if smaller than 2km. Those value can be overridden by defining it in `config_emep.nml`. You can check the value in the standard output (“advection time step (`dt_advec`) set to: 1200 seconds”).

The values of `dt_chem` are the internal Chemistry timesteps. It is variable: usually the first 5 steps last 20 seconds, then followed by 10 larger timesteps, so that the sum of all `dt_chem` timesteps is exactly equal to `dt_advec`. They are printed out in standard output. Note that emissions are included as a source in the Chemistry, and thus have the same timesteps.

REFERENCES

Andersson-Sköld and Simpson, 1997

Y. Andersson-Sköld and D. Simpson. Comparison of the chemical schemes of the EMEP MSC-W and the IVL photochemical trajectory models *Atmos. Environ.*, 1999, 33, 1111-1129

Emberson *et al.*, 2000

L. Emberson, D. Simpson, J.-P. Tuovinen, M.R. Ashmore and H.M. Cambridge. Towards a model of ozone deposition and stomatal uptake over Europe. EMEP/MS-CW Note 6/2000.

Fagerli *et al.*, 2004

H. Fagerli, D. Simpson and S. Tsyro. Unified EMEP model: Updates. In *EMEP Status Report 1/2004, Transboundary acidification, eutrophication and ground level ozone in Europe. Status Report 1/2004*, pages 11–18. The Norwegian Meteorological Institute, Oslo, Norway, 2004.

Gardner *et al.*, 1997

R.M. Gardner, K. Adams, T. Cook, F. Deidewig, S. Ernedal, R. Falk, E. Fleuti, E. Herms, C.E. Johnson, M. Lecht, D.S. Lee, M. Leech, D. Lister, B. Masse, M. Metcalfe, P. Newton, A. Schmitt, C. Vandenberg and R. Van Drimmelen. The ANCAT/EC global inventory of NO_x emissions from aircraft. *Atmos. Environ.*, 31 (12): 1751-1766, 1997.

Jonson *et al.*, 2001

J.E. Jonson, J.K. Sundet and L. Tarrasón. Model calculations of present and future levels of ozone and ozone precursors with a global and a regional model. *Atmos. Environ.*, 35: 525-537, 2001.

Köhler *et al.*, 1995

I. Köhler, R. Sausen and G. Klenner. NO_x production from lightning. The impact of NO_x emissions from aircraft upon the atmosphere at flight altitudes 8-15 km (AERONOX), edited by U. Schumann, final report to the Commission of the European Communities, Deutscher Luft und Raumfahrt, Oberpfaffenhofen, Germany, 343–345, 1995.

Logan, 1998

J. A. Logan. An analysis of ozonesonde data for the troposphere: Recommendations for testing 3-D models and development of a gridded climatology for tropospheric ozone. *J. Geophys. Res.*, 10 (D13):16, 115–116, 149, 1998.

Passant, 2002

Passant NR., Speciation of UK emissions of non-methane volatile organic compounds. AEA Technology Report ENV-0545, Culham, Abingdon, United Kingdom, 2002.

Simpson *et al.*, 2003

D. Simpson, H. Fagerli, J.E. Jonson, S. Tsyro, P. Wind and J.-P. Tuovinen. *The EMEP Unified Eulerian Model. Model Description. EMEP MSC-W Report 1/2003*. The Norwegian Meteorological Institute, Oslo, Norway, 2003.

Simpson and Emberson, 2006

D. Simpson and L. Emberson. Ozone fluxes – Updates. In *Transboundary Acidification, Eutrophication and*

Ground Level Ozone in Europe from 1990 to 2004 in support for the review of the Gothenburg Protocol. EMEP Status Report 1/2006, pages 63–79. The Norwegian Meteorological Institute, Oslo, Norway, 2006.

Simpson *et al.*, 2012

Simpson, D., Benedictow, A., Berge, H., Bergström, R., Emberson, L. D., Fagerli, H., Flechard, C. R., Hayman, G. D., Gauss, M., Jonson, J. E., Jenkin, M. E., Nyri, A., Richter, C., Semeena, V. S., Tsyro, S., Tuovinen, J.-P., Valdebenito, Á., and Wind, P. The EMEP MSC-W chemical transport model – technical description *Atmos. Chem. Physics*, 2012, 12, 7825-7865

Simpson *et al.*, 2017

Simpson, D., Bergström, R., Imhof, H. and Wind, P., Updates to the EMEP/MSC-W model, 2016-2017, in EMEP Status Report 1/2017, pp115-122, The Norwegian Meteorological Institute, Oslo, Norway, www.emep.int, 2017

Simpson *et al.*, 2020a

Simpson, D., Bergström, R., Tsyro, S. and Wind, P., Updates to the EMEP/MSC-W model, 2019-2020, in EMEP Status Report 1/2020, pp155-165, The Norwegian Meteorological Institute, Oslo, Norway, www.emep.int, 2020

Simpson *et al.*, 2020b

Simpson, D., Bergström, R., Briolat, A., Imhof, H., Johansson, J. Priestley, M., Valdebenito, A., GenChem v1.0 - a chemical pre-processing and testing system for atmospheric modelling, *Geosci. Model Devel.*, <https://doi.org/10.5194/gmd-2020-147> 2020

Tuovinen *et al.*, 2004

Tuovinen, J.-P.; Ashmore, M.; Emberson, L. & Simpson, D. Testing and improving the EMEP ozone deposition module *Atmos. Environ.*, 2004, 38, 2373-2385

Wiedinmyer *et al.*, 2011

Wiedinmyer, C., Akagi, S. K., Yokelson, R. J., Emmons, L. K., Al-Saadi, J. A., Orlando, J. J., and Soja, A. J.: The Fire INventory from NCAR (FINN): a high resolution global model to estimate the emissions from open burning, *Geosci. Model Dev.*, 4, 625–641, doi:10.5194/gmd-4-625-2011, 2011

Wind *et al.*, 2020

Wind, P., Rolstad Denby, B. and Gauss, M., Local fractions – a method for the calculation of local source contributions to air pollution, illustrated by examples using the EMEP MSC-W model (rv4_33), *Geosci. Model Devel.*, 13, 3, 1623-1634, <https://www.geosci-model-dev.net/13/1623/2020/>, 2020