NAME III Document
MD2/2 (version 29)
07/09/15

Input

D. J. Thomson, A. R. Jones, B. J. Devenish, M. C. Hort,
H. N. Webster, Eike H. Müller, E. Meneguz and F. M. Beckett

1 Introduction

Here we describe the input data needed by the model with the exception of some of the data, such as gridded met data, which is needed by specific met and flow modules. Some of this data is input via command line arguments while the bulk of the data is presented to the model via the 'main set of headed input files'.

The command line arguments are described in §2. In §3 the concept of headed input files is introduced, which is discussed further in the appendix. The 'main set of headed input files' is introduced in §4 and examples of data blocks within the files are given in §5. In §6 the formats used in these data blocks for inputting certain data items (e.g. times) are defined. Finally the various types of data block are described in §7.

2 Command line arguments

A number of optional arguments can be placed on the command line following the name of the executable file. These are case insensitive with the exception of file names and parts of file names — the latter may of may not be case sensitive, depending on the operating system. Possible arguments are:

- `-InputFile=`*file*
  Here *file* is the main input file. If this is the first command line option the '`-InputFile=`' can be omitted. If not specified the user is asked for the main input file name.
- `-LogFolder=`*folder*
  Here *folder* is the log folder (for keeping the log and error files in). If not specified the log folder is the folder containing the main input file.
- `-RestartFolder=`*folder*
  Here *folder* is the restart folder (for keeping the restart data in). If not specified the restart folder is the folder containing the main input file. This can be specified

whether or not the run is restartable (although it will have no effect if the run is not restartable).

The restart data consist of restart files (these contain data on the model state from which to restart), restart catalogues (containing information on the available restart files), and restart catalogue validity files (containing information on the validity of the restart catalogues). This somewhat complicated collection of data is needed to allow flexibility and to help prevent corruption of the restart data.

- `-Restart=`*suffix*
  If present this indicates that the run is to restart from a restart file. Otherwise the run starts from the beginning. Here *suffix* is the part of the restart file name between 'Restart_' and '.dat' (this part characterises the point in the run at which the restart file was created). If =*suffix* is omitted the latest restart file is used. If starting from the begining (i.e. -Restart absent) then restart catalogue validity files should not be present (if there are any restart catalogue validity files, the run stops to prevent accidental overwriting). If restarting from the latest restart file (i.e. -Restart present but =*suffix* absent) and there are no restart files, the run starts at the beginning.

- `-RunTo=`*time or case*
  If present, this causes the run to be suspended at the indicated model time or after the indicated case. The format for *time* follows that for all input times as described below, but with spaces replaced by under-scores.

- `-RunFor=`*time interval or number of cases*
  If present, this causes the run to be suspended after the indicated model time interval or after the indicated number of cases. The format for *time interval* follows that for all input time intervals as described below, but with spaces replaced by under-scores.

- `-ClosePromptLevel=`*n*
  Here *n* indicates the error level at or above which the user is prompted to close the window used by the model. Below this value the window closes automatically. The error levels are as follows:

  0 indicates no errors or warnings,

  1 indicates some warnings but no errors,

  2 indicates some non-fatal errors but no fatal errors,

  3 indicates fatal errors but no unexpected fatal errors,

  4 indicates unexpected fatal errors (unexpected fatal errors are errors which should not occur and which imply errors in the code).

  Note ClosePromptLevel does not apply to fatal errors which occur before the command line processing is complete (here the user is always prompted to close the window) or to untrapped fatal errors which are handled directly by the compiler. If not specified, the value is 0 (and so the user is always prompted to close the window). ClosePromptLevel is only effective on Windows systems.

3 Headed input files

Much of the input to NAME III is in the form of NAME III 'headed' input files. These files contain various 'blocks' of data, each block consisting of a number of rows of comma separated variables with headers which indicate the nature of the data in each column. Headed input files come in ordered 'sets' of one or more files. If there is more than one file in the set, the effect is, with one exception, the same as if the input files were concatenated following their order and with a blank line added between files. The exception is that sometimes it is convenient to give the names of the second and subsequent files (if any) in a block in the first file — for obvious reasons this block has to be in the first file.

A description of the general format of headed input files is given in the appendix. However it is recommended that the reader looks first at the examples in section §5 before reading this.

4 The main set of headed input files

A single set of headed input files, known as the 'main set'[1], is used to provide much of the information defining the run. The first file in this set is known as the main input file and the name of this file must be input on the command line or in response to a prompt when the code runs. If there are other files in the set, a block giving the names of the other files must be included in the first file.

The ability to use multiple files provides flexibility. As an example one could put everything in a single file, or one could use a separate file for information on the various contaminant species — this could be a reference file which is used for many different model runs, or one could have such a reference file but also have another file with further species needed for a particular run.

5 Examples of data blocks in the main set of headed input files

To orientate the reader we give some examples of (simplified) blocks which might be used in the main set of input files. Here is a simple example of a block:

```
Species:
Name,            Type, Molecular weight
SO2,             Gas,  22.1
NO2,             Gas,  23.0
Passive Tracer, Gas,
```

[1]In fact at present the main set of headed input files is the only set used by NAME III. However it is intended to extend the use of headed files in the future.

The block keyword ‘Species:’ indicates that this is a block which will defines the properties of some contaminant species, the following line contains a number of column keywords indicating what information will be given, and then the last three lines give the properties of three species. The columns can be in any order, so that the following block would have the same effect:

```
Species:
Molecular weight, Name,           Type
22.1,             SO2,            Gas
23.0,             NO2,            Gas
    ,             Passive Tracer, Gas
```

Note that no value is given for molecular weight of the passive tracer — this will take a default value. If the passive tracer was the only contaminant of interest one could omit the molecular weight column and use the following block:

```
Species:
Name,           Type
Passive Tracer, Gas
```

Equally extra columns can be added if desired to specify other properties of the material — e.g. deposition properties, radioactive half-life, and/or radioactive decay products.

Note a blank entry does not always imply that a default value is used. For example information on radio-active decay products is irrelevant if the material is radioactively stable — here a blank entry will reflect this irrelevance rather than implying that a default value is used. In general a blank entry means simply missing data, and, depending on the situation, the model may respond to this in a variety of ways (including giving an error message).

Here is a second more complicated example:

```
Vertical coord systems:
Name,         Type,                Scale, Eta Definition
ZCoord1,      Height above ground, 1.0,
Flight-level,                    ,    ,
ZCoord2,      Pressure based eta,     ,   Global UM
```

Here three vertical coord systems are defined. The first is named ZCoord1 and is defined as a coordinate based on height above ground. The scale of 1.0 (which is actually the default, so that a value for scale is in fact unnecessary here) implies that the coordinate is expressed in metres without any scale factor. For the second coord system, only the name is given — the model knows what ‘flight-level’ is so that no other information is needed. The third example is a more complex coord system with a pressure based eta coordinate. Eta values are defined using arrays giving the values of η, A and B (in fact only two of these three quantities are needed), and these arrays will generally be too large for putting all the values on a line to be feasable. Instead the information defining η is referred to by name ('Global UM') and this information is given in a separate block which might be something like this:

```
Eta,    A,      ! Notes
1.0000, 0.0,    Ground
0.9988, 0.0,    10m level
0.9970, 0.0,
0.9750, 0.0,
0.9304, 0.0,
0.8800, 0.0,
0.8274, 200.0,
0.7750, 881.3,
.       .
.       .
.       .
```

This is an example of a named block, with the block keyword being followed by the name 'Global UM'. As noted above only two of η, A and B are needed, and here we have chosen to use η and A. We have also added a column called '! notes' which the model won't recognise. Such unrecognised columns are allowed in any block. However, unless the unrecognised column keyword begins with '!', this will result in a warning message. Named blocks are generally used for the input of long arrays which cannot be input conveniently on a line.

6 Format for certain data items in the main set of headed input files

Certain types of input variable occur a number of times. Here we describe the format of such variables.

6.1 Switches

The names of (i.e. column keywords for) the switch variables generally end in '?'. Their values may be entered as 'yes', 'no' or anything begining with 'y', 'Y', 'n' or 'N' (possibly preceeded by blanks), or anything accepted by a Fortran list-directed logical read, i.e. true, false, or anything beginning with 't', 'T', '.t', '.T', 'f', 'F', '.f' or '.F' (again possibly preceeded by blanks).

6.2 Times

Times that are specified absolutely must be given in the form

`1/5/2001 15:30{:20{.34}} {UTC {±01:00}}` while times that are specified relatively or which indicate time intervals can be given in the form

```
{-}{5d}15:30{:20{.34}}
```

(non-descriptive format) or

```
{-}{5day}{15hr}{30min}{20{.34}sec}
```

(descriptive format). Here {...} indicates optional components; if omitted, the numerical parts of these optional components are taken as zero. $\pm$ indicates a sign, $+$ or $-$. For descriptive format, at least one of the optional components corresponding to 'day', 'hr', 'min' and 'sec' must be present.

Extra spaces are permitted except within 'UTC', 'day', 'hr', 'min', 'sec' and within the numerical components (a numerical component being a consecutive sequence of the digits 0 to 9, but not including any sign, decimal point, colon etc). The only compulsory space is that after the year (for absolute times). Leading zeros are not significant in any of the numerical components except after the decimal point, or except where the value is zero when a zero must be present. There is no set number of digits for each numerical component so that, for example, a time interval of 1000s could be indicated by '0:0:1000' as well as by '0:16:40'. However values can only lie outside their appropriate range (e.g. minutes in the range 0 to 59) if the time is not absolute and the value is the most significant non-zero element of the time. 'UTC', 'd', 'day', 'hr', 'min' and 'sec' are case insensitive.

The value following 'UTC' indicates the time zone in which the time is expressed, specified as hours and minutes ahead of UTC (so that time zones to the east of the Greenwich meridian will tend to have a positive value and time zones to the west will tend to have a negative value). The sign associated with a time zone applies to the whole time zone offset (hours and minutes). The time '1/5/2001 15:30 UTC+01:00' can be thought of as 1/5/2001 15:30 expressed in the time code 'UTC + 1 hour'. If there is nothing after UTC (or if UTC itself is absent) then the time must be expressed in UTC.

For times other than absolute times, the optional number followed by 'd' indicates the number of days. Any minus sign associated with a time other than an absolute time applies to the whole number, so that, e.g. –1:30 is one and a half hours before the reference time or a time interval of minus one and a half hours, and '–1d12:00' or '– 1day 12hr 0min' is minus one and a half days.

It is useful to be able to input positive and negative infinite times for some inputs and these are indicated by 'infinity' and '–infinity' (with 'infinity' being case insensitive and with optional spaces allowed between '–' and 'infinity').

6.3 Names of files and folders

Files and folders can be specified with full paths or relative to the current folder. Paths can be specified using '\' or '/' or a mixture of the two — the path will be converted to the right form for the operating system. When specifying folders (but not files), the folder can have a trailing '\' or '/' or not and the current folder can be indicated by '.\', './' or '.' (but not in general by a blank which may mean 'folder not specified').

7 Block types in the main set of headed input files

Here we describe the block types used in the main set of headed input files. Note that for an unnamed block, the block keyword and description are usually plural and the column keywords singular. For named blocks this is reversed.

Main Options:

Block keyword:	`Main Options:`
Description:	Gives the main options for the computation. Unnamed block.
Column keywords:	Definition:
`Absolute or Relative Time?`	Set to 'Gregorian' or 'Absolute' for the Gregorian calendar or to 'Relative' for a relative time frame. 360 day years can be invoked by setting this to '360-day years'.
`Backwards?`	Reverses direction of time. Default = 'No'.
`Fixed Met?`	Met does/does not change with time.
`Flat Earth?`	Set to 'Yes' for a simulation on a flat earth, 'No' otherwise.
`Run Name`	Name of run.
`Random Seed`	Seed of random number generator (RNG). Can be 'Fixed' (fortran compiler standard RNG with fixed seed), 'Random' (fortran compiler standard RND with random seed), 'Fixed (Parallel)' (parallel RNG with fixed seed) or 'Random (Parallel)' (parallel RNG with random seed). THIS HAS TO BE SET TO 'Fixed (Parallel)' OR 'Random (Parallel)' WHEN RUNNING IN PARALLEL. If the parallel RNG is used, the random numbers are specific for each particle/puff in the particle/puff stack.
`Run-To File`	Name of 'run-to' file. If at any point during the run this file is created, then the file is read and, if the file is empty or contains anything other than a time, the run is suspended immediately, while, if the file contains a time, the run is suspended once the time is passed.
`Max # Sources`	Maximal allowed number of sources. Default = 100.
`Same Results With/Without Update on Demand?`	Can be set to 'Yes' to indicate that the same results are required with or without update on demand. Default = 'No'
`Max # Field Reqs`	Maximal allowed number of field requirements. Default = 650.
`Max # Field Output Groups`	Maximal allowed number of output field groups. Default = 100.

Restart File Options:

Block keyword:	`Restart Options:`
Description:	Gives the options for writing restart files. Unnamed block.
Column keywords:	Definition:
`# Cases Between Writes`	Number of cases between writing restart files.
`Time Between Writes`	Time interval between writing restart files.
`Delete Old Files?`	Deletes old restart files once a new one has been written.
`Write On Suspend?`	Causes a restart file to be written when the run is suspended.

Only one of *# Cases Between Writes* and *Time Between Writes* can be used.

Multiple Case Options:

Block keyword:	`Multiple Case Options:`
Description:	Gives the options for computing multiple cases in a single run. Named block.
Column keywords:	Definition:
`Dispersion Options Ensemble Size`	Number of sets of dispersion options in ensemble. Default = 1.
`Met Ensemble Size`	Number of met cases in ensemble. Default = 1.

Output Options:

Block keyword:	`Output Options:`
Description:	Gives information on output options. Unnamed block.
Column keywords:	Definition:
`Folder`	Directory to which output will be written.
`Seconds?`	Write seconds in output time fields. Default = 'No'.
`Time Decimal Places`	Number of decimal places for seconds in output time fields.
`Pre 6.5 Format?`	Output of fields will use the pre-6.5 format (transitional only).

OpenMP Options:

Block keyword:	`OpenMP Options:`
Description:	Controls the parallelisation with OpenMP. Unnamed block.

Column keywords:	Definition:
`Use OpenMP?`	Enable parallelisation with OpenMP. If this is set to 'Yes' the following options are used, otherwise they will be ignored and set to their default values. Default = 'No'.
`Threads`	Number of threads in all parallelised for loops. The number of threads for individual loops can be specified further with the following four options. Default = 1.
`Particle Threads`	Number of threads in the particle loop. Default = 1.
`Particle Update Threads`	Number of threads in the particle update loop. Default = 1.
`Chemistry Threads`	Number of threads in the chemistry loop. Default = 1.
`Output Group Threads`	Number of threads in the output group loop. Default = 1.
`Output Process Threads`	Number of threads used for output processing. Default = 1.
`Parallel MetRead`	Read NWP MetData with separate IO thread. If this is option is set to 'Yes' and `Parallel MetProcess` is not specified, the latter is automatically set to 'Yes', i.e. the data is read *and* processed by the IO thread. Default = 'No'.
`Parallel MetProcess`	Process NWP MetData with separate IO thread. If this is set to 'Yes' and `Parallel MetRead` is 'No' (or not specified), then the latter is automatically set to 'Yes'. Default = 'No'.

These options are only effective if the code has been compiled in parallel mode with OpenMP support. Note that the scheduling strategy of the chemistry loop is controlled by the environment variable `OMP_SCHEDULE`, which can be set before the start of a NAME run.

When running the code in parallel it is crucial to use the correct random number generator as otherwise correctness of results can not be guaranteed and use of the wrong random number generator has a negative impact on the parallel scalability. To do this, set `Random Seed` in the Main Options block to either 'Fixed (Parallel)' or 'Random (Parallel)'.

Timer Options:

Block keyword:	`Timer Options:`
Description:	Controls the behaviour of timer output. Unnamed block.

Column keywords:	Definition:
`Use Timers?`	Print out timer information. Default = 'No'.
`Summary Only?`	Only print out summary information at the end of the run. Default = 'Yes'.

Note that setting `Summary Only?` to 'No' can potentially create very large timing files.

Input Files:

Block keyword:	`Input Files:`
Description:	Gives the names of the second and subsequent headed input files (if any). Unnamed block.
Column keywords:	Definition:
`File Names`	Name(s) of additional input file(s).

Array:

Block keyword:	`Array:`
Description:	Gives the values of a named array. Named block.
Column keywords:	Definition:
`Array Values`	Values of the array.

Message Controls:

Block keyword:	`Message Controls:`
Description:	Gives information on the way in which various messages are to be controlled. Unnamed block.
Column keywords:	Definition:
`Name`	Name of message.
`Action`	Action to be taken when the message is generated. Options are 'Default' or blank (same as if no message control was in place), 'Message', 'Warning', 'Error', 'Fatal Error', or 'Unexpected Fatal Error'.
`# of Messages Before Failure`	After this many messages any further message is escalated to a fatal error. If not specified, messages will never escalate to a fatal error.
`# of Messages Before Suppression`	After this many messages any further messages are suppressed. If not specified, messages will never be suppressed.

The messages for which message controls can be invoked are as follows:

Name of message:	Default behaviour:	Description:
`No flow`	Fatal error	There are no valid flow modules with the flow attribute.
`No flow for particle/puff`	Error	Particles/puffs are being lost due to lack of a suitable flow module.
`Inconsistent NWP header`	Error	An NWP field in a Name II or PP format met file has an inconsistent header.
`Inconsistent validity time`	Error	An NWP field in a Name II or PP format met file has an inconsistent validity time.
`Missing NWP field`	Warning	An NWP field is missing from the met file.
`Inconsistent radar met header`	Error	A field in a radar met file has an inconsistent header.
`Inconsistent radar validity time`	Error	A field in a radar met file has an inconsistent validity time.
`Missing radar met field`	Warning	A field is missing from the radar met file.
`No time zone in single site met file`	Warning	Time zone not specified in single site met file.
`Zero source strength`	Warning	All source strengths for a source are zero and the source will be ignored.

Eta Definition:

Block keyword:	`Eta Definition:`
Description:	Gives the arrays needed to define an eta coord system. Named block.

Column keywords:	Definition:
`Eta`	Eta at given vertical level
`A`	parameter A at a given vertical level
`B`	parameter B at a given vertical level

At least two parameters out of the set `Eta`, `A` and `B` need to be specified, for details see the model document on coordinate systems.

Horizontal Coordinate Systems:

Block keyword:	`Horizontal Coordinate Systems:`
Description:	Gives the information defining a horizontal coordinate system. Unnamed block.

Column keywords:	Definition:
`Name`	Name of coordinate system (see also note below).
`Type`	Type of coordinate system: 1 - Coord system based on a latitude-longitude system with arbitrary orientation, i.e. arbitrary location for the latitude-longitude system's north pole and for the 'third' Euler angle representing a rotation about the system's north pole. 2 - Cartesian coord system using a polar stereographic (PS) projection. 3 - Polar coord system using a polar stereographic (PS) projection. 4 - Cartesian coord system using a transverse Mercator (TM) projection. 5 - polar coord system using a transverse Mercator (TM) projection.
`Pole Long`	Longitude of pole.
`Pole Lat`	Latitude of pole.
`Angle`	For lat-long coordinate systems: 'Third' Euler angle of coordinate system rotation. For other systems, see note below.
`X-Origin`	Offset of the coordinate system in the x direction.
`Y-Origin`	Offset of the coordinate system in the y direction.
`X-Unit`	x units of coordinate system.
`Y-Unit`	y units of coordinate system.
`Scale Factor`	Only used for coordinate systems 4 and 5 (TM projection); scale adjustment used by the TM projection.

For coord systems of type 2 (PS cartesian) 'Angle' is the angle between the x axis and the $\pi/2$ longitude line of a type 1 coord system with the same pole but with a zero value of 'Angle' (in radians).

For coord systems of type 3 (PS polar) 'Angle' is the angle between the x axis of the underlying Cartesian coord system and the $\pi/2$ longitude line of a type 1 coord system with the same pole but with a zero value of 'Angle' (in radians).

For coord systems of type 4 and 5 (TM partesian/polar) 'Angle' is not used but is always set to 0. The coord system is always east-north for type 4 and 5.

In each case a positive value means that, standing at the north pole or tangent point and looking down, the system is rotated anticlockwise relative to its orientation for a zero value.

Certain horizontal coordinate systems can be specified by just giving their names: 'Lat-Long', 'EMEP 50km Grid', 'EMEP 150km Grid', 'UK National Grid (m)' and 'UK National Grid (100m)'.

Vertical Coordinate Systems:

Block keyword:	`Vertical Coordinate Systems:`
Description:	Gives the information defining a vertical coordinate system. Unnamed block.

Column keywords:	Definition:
`Name`	Name of coordinate system.
`Type`	Type of coordinate system: 1 - height above ground based (metres) 2 - height above sea based (metres) 3 - pressure based (Pa) 4 - pressure based, converted to height above dea level using the ICAO standard atmosphere (metres) 5 - pressure based eta coord (dimensionless) 6 - height based eta coord (dimensionless).
`Unit`	Scale factor for unit of coordinate system relative to m or Pa. Not used for type 5 and 6 coordinate types.
`Eta Definition`	Definition of eta coordinate system, as described in the `Eta Definition:` input block (see above).
`Model Top Height`	Height above sea level of the model top (in metres). Corresponds to eta=1.
`Interface Height`	Height above sea level of the linear/quadratic interface (in metres); i.e. the lowest constant-height eta level.

Certain vertical coordinate systems can be specified by just giving their names: ‘m agl’, ‘m asl’, ‘FL’ and ‘Pa’.

Locations:

Block keyword:	`Locations:`
Description:	Defines a set of locations. Named block.

Column keywords:	Definition:
`Name`	Name of location.
`H-Coord`	Coordinate system to be used for locations.
`X`	X coordinate of location in given coordinate system.
`Y`	Y coordinate of location in given coordinate system.

Horizontal Grids:

Block keyword:	`Horizontal Grids:`
Description:	Gives information defining horizontal grids. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the grid.
`H-Coord`	Name of the horizontal coord system used to define the grid.
`nX`	Number of grid points in the X-direction.
`nY`	Number of grid points in the Y-direction.
`dX`	Grid spacing in the X-direction.
`dY`	Grid spacing in the Y-direction.
`X Min`	X-coord of first grid point.
`Y Min`	Y-coord of first grid point.
`X Max`	X-coord of last grid point.
`Y Max`	Y-coord of last grid point.
`X Centre`	X-coord of grid centre.
`Y Centre`	Y-coord of grid centre.
`X Range`	X Max – X Min.
`Y Range`	Y Max – Y Min.
`X-Array`	Name of the array of giving the X-coords of the grid.
`Y-Array`	Name of the array of giving the Y-coords of the grid.
`Wrap?`	Indicates whether the grid wraps round the earth in the x direction.
`Set of locations`	Name of a set of locations.
`Location of centre`	Name of location of grid centre in the set of locations.

Note that, in addition to specifying *Name*, the following options are allowed:

(i) *H-Coord*, *nX*, *nY*, two of *X Min*, *X Max*, *X Centre*, *dX* and *X Range* (but not *dX* and *X Range*), two of *Y Min*, *Y Max*, *Y Centre*, *dY* and *Y Range* (but not *dY* and *Y Range*), and possibly *Wrap*;

(ii) *H-Coord*, *nX*, *nY*, one of *dX* and *X Range*, one of *dY* and *Y Range*, *Set of locations*, *Location of centre* and possibly *Wrap*;

(iii) *H-Coord*, *X-Array*, *Y-Array* and possibly *Wrap*;

(iv) *Set of locations*, *dX* and *dY*.

Option (iii) defines a grid with irregular spacing while option (iv) defines an unstructured grid. Note also that *X Min* and *X Max* are the centre points of the first and last grid points not the smallest and largest X; one can have *X Max* $<$ *X Min* which has the same effect as $dX < 0$. This also means that *X range* is the distance between the centre of the first grid box and the centre of the last grid box.

Vertical Grids:

Block keyword:	`Vertical Grids:`
Description:	Gives information defining vertical grids. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the grid.
`Z-Coord`	Name of the vertical coord system used to define the grid.
`nZ`	Number of grid points.
`dZ`	Grid spacing.
`Z0`	Coord of grid point with the smallest coord value.
`Z-Array`	Name of the array of giving the coords of the grid, or the (case insensitive) phrase *Use Eta Levels* which, when an eta coord system is used, implies that the levels used in defining the coord system are also the grid levels.
`Av Z-Array`	Name of the array of giving the depth (dZ) of an explicit averaging region associated with each grid point. If *Av Z-Array* = 'Z On Boundaries' then the Z-Array is interpreted as defining the boundaries of the averaging regions from which the mid points and averaging depths are calculated.
`Index-Array`	For some purposes one may want to attach indices to the levels which are not necessarily 1, 2, 3, etc. In this case *Index-Array* can be used to give the name of the array which holds the desired indices.

If *Index-Array* is specified the grid is known as an 'indexed grid'.

Note that, in addition to specifying *Name*, *Z-Coord* and possibly *Index-Array*, the following options are allowed:

(i) *nZ*, *Z0* and *dZ*;

(ii) *Z-Array* = 'Use Eta Levels' (with *Z-Coord* a pressure-based eta coord system);

(iii) *Z-Array* $\neq$ 'Use Eta Levels';

(iv) *Z-Array* and *Av Z-Array* = 'Z On Boundaries';

(v) *Z-Array* and *Av Z-Array* $\neq$ 'Z On Boundaries'.

Specifying *Z-Array* enables a grid with irregular spacing to be defined. If *Z-Array* is specified to be *Use Eta Levels* then the coord system must be an eta coord system (and, in the unlikely event that there is an array called *Use Eta Levels*, it is not used).

Temporal Grids:

Block keyword:	`Temporal Grids:`
Description:	Gives information defining temporal grids. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the grid.
`nT`	Number of grid points.
`dT`	Grid spacing.
`T0`	Coord of grid point with the smallest coord value.
`T-Array`	Name of the array of giving the coords of the grid.

Note that either nT, dT and *T0* or *T-Array* should be specified (and not both). Specifying *T-Array* enables a grid with irregular spacing to be defined.

Domains:

Block keyword:	`Domains:`
Description:	Gives information defining a domain. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the domain.
`H Unbounded?`	Domain is horizontally unbounded (both in X and Y direction)
`X Unbounded?`	Domain is horizontally unbounded in X direction.
`Y Unbounded?`	Domain is horizontally unbounded in Y direction.
`H-Coord`	Name of the horizontal coordinate system used.
`X Min`	X-coord of domain start.
`Y Min`	Y-coord of domain start.
`X Max`	X-coord of domain end.
`Y Max`	Y-coord of domain end.
`X Centre`	X-coord of domain centre.
`Y Centre`	Y-coord of domain centre.
`X Range`	X extent of domain.
`Y Range`	Y extent of domain.
`Set of locations`	Name of a set of locations.
`Location of centre`	Name of location of domain centre in the set of locations.
`Z Unbounded?`	Domain is vertically unbounded.
`Z-Coord`	Name of the vertical coord system used.
`Z Max`	Maximum height of domain in the vertical coord system.
`T Unbounded?`	Domain is unbounded in time.
`Start Time`	Domain valid from this time.
`End Time`	Domain not valid after this time.
`Duration`	End Time – Start Time.
`Max Travel Time`	Maximum life time of particles in this domain.

To specify the horizontal extent, the following options are allowed:

(i) *HUnbounded?* = ‘Yes’;

(ii) *HUnbounded?* = ‘No’ with the following specified: *H-Coord*, two of *X Min*, *X Max*, *X Centre*, dX and *X Range*, and two of *Y Min*, *Y Max*, *Y Centre*, dY and *Y Range*;

(iii) *HUnbounded?* = 'No' with the following specified: *H-Coord*, *X Range*, *Y Range*, *Set of locations* and *Location of centre*.

Note also that *X Min* and *X Max* are the start and end of the domain in the direction of increasing x; one can have *X Max* < *X Min* if the X coord is cyclic to indicate the domain occupies the union of $\{x \leq X\ Max\}$ and $\{X\ Min \leq x\}$.

X and Y direction of the domain can be set to 'Unbounded' separately by using the fields `X Unbounded?` and `Y Unbounded?`. Setting `H Unbounded?` to 'Yes' has the same effect as setting both `X Unbounded?` and `Y Unbounded?` to 'Yes'.

To specify the temporal extent, the following options are allowed:

(i) *TUnbounded?* = 'Yes';

(ii) *TUnbounded?* = 'No' with the following specified: two of *Start Time*, *End Time* and *Duration*.

Prototype Met Module Instances:

Block keyword:	`Prototype Met Module Instances:`
Description:	Specifies instances of the prototype met module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of prototype met module instance.
`H-Coord`	Horizontal grid.
`Z-Coord`	Vertical grid.
`Met File`	Name of file containing MetData for this module
`Update On Demand?`	Update module on demand? Default = 'No'.

Single Site Met Module Instances:

Block keyword:	`Single Site Met Module Instances:`
Description:	Specifies instances of the single site met module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of single site met module instance.
`H-Coord`	Horizontal grid.
`Long`	Longitude of measurement.
`Lat`	Latitude of measurement.
`Height`	Height of measurement.
`z0`	Roughness length.
`z0D`	Roughness length in dispersion area.
`Representative?`	Indicates that no distrinction is made between MetSite and dispersion area properties. 'Yes': one roughness lengths required. 'No': two roughness lengths required.
`Met File`	Name of single site met file.
`Ignore Fixed Met Time?`	Not fully implemented yet.
`Mesoscale SigU`	Standard deviation of horizontal velocity for unresolved mesoscale motions. Default = 0.5.
`Mesoscale TauU`	Horizontal Lagrangian timescale for unresolved mesoscale motions. Default = 7200.
`Free Trop SigU`	Standard deviation of horizontal velocity for free tropospheric turbulence. Default = 0.25.
`Free Trop SigW`	Standard deviation of vertical velocity for free tropospheric turbulence. Default = 0.1.
`Free Trop TauU`	Horizontal Lagrangian timescale for free tropospheric turbulence. Default = 300.
`Free Trop TauW`	Vertical Lagrangian timescale for free tropospheric turbulence. Default = 100.
`Update On Demand?`	Update module on demand? Default = 'No'.

NWP Met Definitions:

Block keyword:	`NWP Met Definitions:`
Description:	Gives information on the nature of a set of NWP meteorological data. Unnamed block.

Column keywords:	Definition:
`Name`	Name of NWP Met definition.
`Binary Format`	Binary format of met data i.e., BIG_ENDIAN or NATIVE.
`File Type`	The file format. Can be 'Name II', 'PP', 'GRIB' or 'NetCDF'. Using GRIB data requires the code to be compiled with GRIB API support. Using NetCDF data requires the code to be compiled with NetCDF support.
`dT`	Time interval of met data.
`T0`	Reference time for met data, defined so that the met data times have the form $T0 + n \times dT$. Default = '1/1/2000 00:00'.
`Day Per File`	One met file per day or per time interval.
`Prefix`	Prefix to met files.
`Suffix`	Suffix to met files.
`Next Heat Flux`	Indicates that, for heat flux and stress, interpolation in time should not be performed and that the future (next) value should be used instead. This is appropriate if the fluxes are derived from accumulated transfers.
`Next Precipitation`	Indicates that, for precipitation, interpolation in time should not be performed and that the future (next) value should be used instead. This is appropriate if precipitation is derived from accumulations. Default = 'No'.
`Mesoscale SigU`	Default value of the standard deviation of horizontal velocity for unresolved mesoscale motions.
`Mesoscale TauU`	Default value of the horizontal Lagrangian timescale for unresolved mesoscale motions.
`Met File Structure Definition`	Name of the Met File Structure Definition giving the format of met data.
`Z-Coord - W`	Vertical coordinate of which W is the rate of change.
`Z-Coord - Cloud Height`	Vertical coordinate of used for cloud heights.
`Z-Grid`	Name of main vertical grid.
`Z-Grid - UV`	Name of vertical grid for U and V.
`Z-Grid - W`	Name of vertical grid for W.
`Z-Grid - P`	Name of vertical grid for pressure.
`H-Grid`	Name of main horizontal grid.
`H-Grid - U`	Name of horizontal grid for U.
`H-Grid - V`	Name of horizontal grid for V.
`Topography File`	Name of topography file.

Ancillary Met Definitions:

Block keyword:	`Ancillary Met Definitions:`
Description:	Gives information on the nature of a set of met ancillary data. Unnamed block.

Column keywords:	Definition:
`Name`	Name of ancillary met definition.
`Binary Format`	Binary format of ancillary met data i.e., BIG_ENDIAN or NATIVE.
`File Type`	The file format. Can be 'Name II', 'PP' or 'GRIB Edition 1'. Using GRIB data requires the code to be compiled with GRIB API support.
`dT`	Time interval of ancillary met data. dT can be given as 'infinite' or 'monthly'. The former is used when the ancillary data does not change in time and the latter when fixed values are used for each of the 12 months of the year.
`T0`	Reference time for ancillary met data, defined so that the met data times have the form $T0 + n{\times}dT$. Default = '1/1/2000 00:00'.
`Prefix`	Prefix to ancillary met files.
`Suffix`	Suffix to ancillary met files.
`Next Heat Flux`	Indicates that, for heat flux and stress, interpolation in time should not be performed and that the future (next) value should be used instead. This is appropriate if the fluxes are derived from accumulated transfers.
`Next Precipitation`	Indicates that, for precipitation, interpolation in time should not be performed and that the future (next) value should be used instead. This is appropriate if precipitation is derived from accumulations. Default = 'No'.
`Met File Structure Definition`	Name of the ancillary Met File Structure Definition giving the format of ancillary met data.
`H-Grid`	Name of main horizontal grid.

Radar Met Definitions:

Block keyword:	`Radar Met Definitions:`
Description:	Gives information on the nature of a set of radar met data. Unnamed block.

Column keywords:	Definition:
`Name`	Name of radar met definition.
`Binary Format`	Binary format of radar met data, i.e., BIG_ENDIAN or NATIVE.
`File Type`	The file format. Currently only 'Nimrod' is supported.
`dT`	Time interval of radar met data.
`T0`	Reference time for radar met data, defined so that the data times have the form $T0 + n{\times}dT$. Default = '1/1/2000 00:00'.
`Day Per File`	One met file per day or per time interval.
`Prefix`	Prefix to radar met files.
`Suffix`	Suffix to radar met files.
`Met File Structure Definition`	Name of the Met File Structure Definition giving the format of radar met data.
`Next Precipitation`	Indicates that, for precipitation, interpolation in time should not be performed and that the future (next) value should be used instead. This is appropriate if precipitation is derived from accumulations. Default = 'No'.
`H-Grid`	Name of horizontal grid for radar met fields.

NWP Met File Structure Definition:

Block keyword:	`NWP Met File Structure Definition:`
Description:	Named block.

Column keywords:	Definition:
`Field Name`	Name of NWP field.
`Lowest Level`	Level number of lowest level of field.
`Highest Level`	Level number of highest level of field or 'Top' for top level.
`Field Code`	Code for field. For Name II and PP met files this is the 'stash' code. For GRIB met files this is the 'GRIB parameter code'.
`3-d?`	Indicates the field is part of a 3-d field.
`Field Qualifiers`	Specifies additional qualifiers for the input field. Currently the only recognised qualifiers are *Dynamic* (large-scale) or *Total* for use with cloud fields.
`NC Field Name`	NetCDF name of met variable in NWP field (required only when reading NetCDF met files).

Ancillary Met File Structure Definition:

Block keyword:	`Ancillary Met File Structure Definition:`
Description:	Named block.

Column keywords:	Definition:
`Field Name`	Name of Ancillary field.
`Lowest Level`	Level number of lowest level of field.
`Highest Level`	Level number of highest level of field or 'Top' for top level.
`Field Code`	Code for field. For Name II and PP met files this is the 'stash' code. For GRIB met files this is the 'GRIB parameter code'.
`3-d?`	Indicates the field is part of a 3-d field.

Note *Field Name*, *Lowest Level* and *Highest Level* refer to the field as it is regarded by NAME III. *Field Code* and *3-d?* refer to the field as it is regarded within the NWP model generating it.

Radar Met File Structure Definition:

Block keyword:	`Radar Met File Structure Definition:`
Description:	Named block.

Column keywords:	Definition:
`Field Name`	Name of radar met field.
`Lowest Level`	Level number of lowest level of field.
`Highest Level`	Level number of highest level of field or 'Top' for top level.
`Field Code`	Code for field.
`3-d?`	Indicates the field is part of a 3-d field.
`Field Qualifiers`	Specifies additional qualifiers for the input field.

Note that currently only the field name '*precipitation rate (mm/hr)*' is recognised (and '*dummy*' for skipping over fields), and it is not possible to read any multi-level fields.

However an input file structure similar to that for reading NWP met and ancillary met has been retained.

NWP Met Module Instances:

Block keyword:	`NWP Met Module Instances:`
Description:	Specifies instances of the NWP met module to be used. Unnamed block.
Column keywords:	Definition:
`Name`	Name of NWP met module.
`Min B L Depth`	Minimum boundary layer depth.
`Max B L Depth`	Maximum boundary layer depth.
`Use NWP BL Depth?`	Use boundary layer depth from NWP data.
`Mesoscale SigU`	Standard deviation of horizontal velocity for unresolved mesoscale motions. Default value in met definition file.
`Mesoscale TauU`	Horizontal Lagrangian timescale for unresolved mesoscale motions. Default value in met definition file.
`Free Trop SigU`	Standard deviation of horizontal velocity for free tropospheric turbulence. Default = 0.25.
`Free Trop SigW`	Standard deviation of vertical velocity for free tropospheric turbulence. Default = 0.1.
`Free Trop TauU`	Horizontal Lagrangian timescale for free tropospheric turbulence. Default = 300.
`Free Trop TauW`	Vertical Lagrangian timescale for free tropospheric turbulence. Default = 100.
`Restore Met Script`	Script used to restore met data (possibly with path). If blank, met data is not restored.
`Delete Met?`	Delete met data when finished.
`Met Folder`	Folder containing met data for non-met-ensemble runs.
`Ensemble Met Folder`	Folder containing met data for met-ensemble runs where all met cases are in the same file.
`Met Folder Stem`	The stem for a collection of met folders, each met folder containing a single met case from an ensemble. The met folder names are constructed by adding integers 1, 2, ..., Met Ensemble Size to the end of the met folder stem.
`Met Folders`	Name of an array of met folder names, each met folder containing a single met case from an ensemble.
`Topography Folder`	Folder containing topography data.
`Met Definition Name`	Name of NWP Met Definition characterising met data.
`Update On Demand?`	Update module on demand? Default = ‘No’.

Ancillary Met Module Instances:

Block keyword:	`Ancillary Met Module Instances:`
Description:	Specifies instances of the ancillary met module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of ancillary met module.
`Met Folder`	Folder containing ancillary met data for non-met-ensemble runs.
`Ensemble Met Folder`	Folder containing ancillary met data for met-ensemble runs where all met cases are in the same file. This can be used if different ancillary met data is used for different cases in the run.
`Met Folder Stem`	The stem for a collection of met folders, each met folder containing a single met case from an ensemble. The met folder names are constructed by adding integers 1, 2, ..., Met Ensemble Size to the end of the met folder stem.
`Met Folders`	Name of an array of met folder names, each met folder containing a single met case from an ensemble.
`Met Definition Name`	Name of Ancillary Met Definition characterising ancillary met data.
`Update On Demand?`	Update module on demand? Default = 'No'.

Radar Met Module Instances:

Block keyword:	`Radar Met Module Instances:`
Description:	Specifies instances of the radar met module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of radar met module.
`Restore Met Script`	Script used to restore met data (possibly with path). If blank, met data is not restored.
`Delete Met?`	Delete met data when finished.
`Met Folder`	Folder containing radar met data files.
`Met Definition Name`	Name of Radar Met Definition characterising met data.
`Update On Demand?`	Update module on demand? Default = 'No'.

Prototype Flow Module Instances:

Block keyword:	`Prototype Flow Module Instances:`
Description:	Specifies instances of the prototype flow module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of prototype flow module.
`Met Module`	Name of met module to use.
`Met`	Name of met module instance to use.
`H-Coord`	Name of horizontal coord system used.
`Z-Coord`	Name of vertical coord system used.
`Domain`	Name of the flow domain.
`Update On Demand?`	Update module on demand? Default = 'No'.

Single Site Flow Module Instances:

Block keyword:	`Single Site Flow Module Instances:`
Description:	Specifies instances of the single site flow module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of single site flow module instance.
`Met Module`	Name of met module to use.
`Met`	Name of met module instance to use.
`Domain`	Name of the flow domain.
`Update On Demand?`	Update module on demand? Default = 'No'.

NWP Flow Module Instances:

Block keyword:	`NWP Flow Module Instances:`
Description:	Specifies instances of the NWP flow module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of NWP flow module instance.
`Met Module`	Name of met module to use.
`Met`	Name of met module instance to use.
`Domain`	Name of the flow domain.
`Update On Demand?`	Update module on demand? Default = 'No'.

Building Flow Module Instances:

Block keyword:	`Building Flow Module Instances:`
Description:	Specifies instances of the building flow module to be used and the parameters required by the module. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the building flow module instance.
`H-Coord`	Name of horizontal coord system used to define the grid.
`Z-Coord`	Name of vertical coord system used to define the grid.
`Domain`	Name of the flow domain.
`Length`	Length of building.
`Width`	Width of building.
`Height`	Height of building.
`X`	X coordinate of centre of building.
`Y`	Y coordinate of centre of building.
`Angle`	Acute angle of building length to X axis.
`Update Flow Subset`	List of flow modules which Buildings Module may access.
`Update On Demand?`	Update module on demand? Default = 'No'.

Radar Flow Module Instances:

Block keyword:	`Radar Flow Module Instances:`

Description:	Specifies instances of the radar flow module to be used. Unnamed block.

Column keywords:	Definition:
`Name`	Name of radar flow module instance.
`Met Module`	Name of met module to use.
`Met`	Name of met module instance to use.
`Domain`	Name of the flow domain.
`Update On Demand?`	Update module on demand? Default = 'No'.

LINCOM Flow Module Instances:

Block keyword:	`LINCOM Flow Module Instances:`

Description:	Specifies LINCOM. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the LINCOM flow module instance.
`Terrain File`	Name of terrain file.
`Roughness File`	Name of surface roughness file.
`Domain`	Name of the flow domain.
`Terrain Pertubation?`	Calculate flow pertubation caused by terrain.
`Roughness Perturbation?`	Calculate flow pertubation caused by roughness variation.
`H-Grid`	Name of horizontal coord system used to define the grid.
`Z-Grid`	Name of vertical coord system used to define the grid.
`X Comp Grid`	Number of x grid points in LINCOM computational grid.
`Y Comp Grid`	Number of y grid points in LINCOM computational grid.
`Lincom Executable`	Name of LINCOM executable file.
`Update Flow Subset`	List of flow modules which LINCOM Module may access.
`Update On Demand?`	Update module on demand? Default = 'No'.

Flow Order:

Block keyword:	`Flow Order:`

Description:	Defines a priority order for the flow module instances. Named block.

Column keywords:	Definition:
`Flow Module`	Name of flow module.
`Flow`	Name of flow module instance.

Flow Subset:

Block keyword:	`Flow Subset:`
Description:	Defines a subset of flows module instances. Named block.
Column keywords:	Definition:
`Flow Module`	Name of flow module.
`Flow`	Name of flow module instance.

Flow Attributes:

Block keyword:	`Flow Attributes:`
Description:	Defines the priority order of the flow module instances to be used for each flow module attribute. Unnamed block.
Column keywords:	Definition:
`Name`	Name of flow attribute. Options are Update, Convert, Flow, Cloud, Rain, Surface, Soil and Plant.
`Flow Order`	Name of priority order for the attribute.

Surface and soil attributes are required for dust and sea salt (met dependent source types) runs. Surface and plant attributes are required for runs using the land use dependent dry deposition scheme.

Species:

Block keyword:	`Species:`
Description:	Gives information on species. Unnamed block.

Column keywords:	Definition:
`Name`	Name to be used for the species.
`Category`	Name of user defined category, e.g. Radionuclide.
`Half Life`	Half life of radioactive species. Blank or 'stable' (case insensitive) indicates a stable species. Default = 'Stable'.
`Daughter`	Name of daughter product.
`Branching Ratio`	Fraction of activity of the parent which is available for decay to daughter product.
`Cloud Gamma Parameters`	Set of Cloud Gamma Parameters.
`Power Law Decay Exponent`	Exponent for power law decay.
`Power Law Decay Delay`	Travel time before power law decay starts.
`UV Loss Rate`	Rate of species destruction due to UV. Default = 'Stable'.
`Surface Resistance`	Surface resistance for deposition.
`Deposition velocity`	Deposition velocity.
`A rain - BC`	Scavenging coefficient (A) for below-cloud wet deposition (washout) by rain ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`B rain - BC`	Scavenging coefficient (B) for below-cloud wet deposition (washout) by rain ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`A snow - BC`	Scavenging coefficient (A) for below-cloud wet deposition (washout) by snow ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`B snow - BC`	Scavenging coefficient (B) for below-cloud wet deposition (washout) by snow ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`A rain - IC`	Scavenging coefficient (A) for in-cloud wet deposition (rainout) by rain ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`B rain - IC`	Scavenging coefficient (B) for in-cloud wet deposition (rainout) by rain ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`A snow - IC`	Scavenging coefficient (A) for in-cloud wet deposition (rainout) by snow ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`B snow - IC`	Scavenging coefficient (B) for in-cloud wet deposition (rainout) by snow ($\Lambda = Ar^B$, r = precipitation rate (mm/hr)).
`Molecular Weight`	Molecular weight. Defined for gasses only.
`Material unit`	Unit used to store material quantities within the model.
`Land use dependent dry dep`	Indicate whether or not to use the land use dependent dry deposition scheme. Land use dependent dry deposition schemes are available for O3, NO, SULPHUR-DIOXIDE, HYDROGEN, AMMONIA, CH4, HCHO, PAN, NO2, CO and HNO3. Default = 'No'.
`Mean aerosol diameter`	Mass mean diameter (in um) for aerosol species

At most one of *Surface Resistance*, *Deposition velocity*, *Mean aerosol diameter* and *Land use dependent dry dep* can be specified or set to true. If none of *Surface Resistance*, *Deposition velocity*, *Mean aerosol diameter* and *Land use dependent dry dep* are specified or set to true, there is no dry deposition for non-sedimenting species. Sedimenting releases (i.e., those heavy particles for which gravitational settling is represented - details on how this is done can be found under the Sources block keyword section) are, however, a special case. For sedimenting particles, the non-sedimentation component of the dry deposition velocity is given by the *Deposition velocity* (if it is specified). If, however, a deposition velocity is not specified, then the non-sedimentation component of the dry deposition velocity is calculated using a particle diameter (specified in the Sources block either as a

Scavenging coefficient	All particles	Sulphur dioxide	Ammonia	HNO_3	HONO	H_2S	All other gases
`A rain - BC`	8.4E-5	1.2E-4	1.2E-4	1.0E-4	1.0E-4	2.0E-5	
`B rain - BC`	0.79	0.79	0.79	0.79	0.79	0.79	
`A snow - BC`	8.0E-5	1.2E-4	1.2E-4	1.0E-4	1.0E-4	2.0E-5	
`B snow - BC`	0.305	0.305	0.305	0.305	0.305	0.305	
`A rain - IC`	3.36E-4	1.0E-4	3.5E-4	3.5E-4	3.5E-4	1.7E-5	
`B rain - IC`	0.79	0.79	0.79	0.79	0.79	0.79	
`A snow - IC`	5.2E-5	1.5E-5	3.5E-4	3.5E-4	3.5E-4	2.6E-6	
`B snow - IC`	0.79	0.79	0.79	0.79	0.79	0.79	

Table 1: Recommended scavenging coefficients for different species

diameter or as a particle size distribution). Any values specified for *Surface Resistance*, *Mean aerosol diameter* or *Land use dependent dry dep* are ignored for sedimenting particles and, unlike non-sedimenting species, dry deposition (or even the non-sedimenting component) cannot be turned off by leaving all of *Surface Resistance*, *Deposition velocity*, *Mean aerosol diameter* and *Land use dependent dry dep* blank.

To turn off wet deposition for a particular species, all of the scavenging coefficients (*A rain - BC*, *B rain - BC*, *A snow - BC*, *B snow - BC*, *A rain - IC*, *B rain - IC*, *A snow - IC* and *B snow - IC*) should be left blank. To turn off wet deposition for a particular process (e.g. rainout by snow), the appropriate *A* scavenging coefficient should be set to 0.0. Recommended scavenging coefficients for different species, based on scavenging coefficients used in other dispersion models and reported in the literature, are given in Table 1. Revisions may be required to these recommended values.

Species Uses:

Block keyword:	`Species Uses:`
Description:	Gives information on species uses. Unnamed block.
Column keywords:	Definition:
`Species`	Species name as used in the species block.
`On Particles?`	Indicates whether this species should be carried on particles.
`On Fields?`	Indicates whether this species should be carried on fields.
`Advect Field?`	Indicates whether the field should be advected for a species carried on fields.
`Particle Size Distribution For Fields`	Indicates which particle size distribution is to be used, if any.

Cloud Gamma Parameters:

Block keyword:	`Cloud Gamma Parameters:`
Description:	Gives information on cloud gamma parameters. Unnamed block.

Column keywords:	Definition:
`Photon Energy`	Energy of photon.
`Photon Intensity`	Frequency with which photons are released.
`Linear Attenuation Coefficient`	Linear attenuation coefficient μ
`B Build-up factor a`	Parameter a parametrising the Berger buildup factor $B(E_\gamma, \mu r)$
`B Build-up factor b`	Parameter b parametrising the Berger buildup factir $B(E_\gamma, \mu r)$
`Air kerma pu fluence`	Dose in air incident on a unit area.
`Adult effective dose pu air kerma`	Whole body dose to an adult per unit dose in air
`Adult thyroid dose pu air kerma`	Adult thyroid dose per unit air kerma.
`Adult lung dose pu air kerma`	Adult lung dose per unit air kerma.
`Adult bone surface dose pu air kerma`	Adult bone surface dose per unit air kerma.

Particle Mass Limits:

Block keyword:	`Particle Mass Limits:`
Description:	Gives information on the mass limit to be imposed on particles. Unnamed block.

Column keywords:	Definition:
`Species Name`	Name of species.
`Particle Mass Limit`	Upper limit on mass of species to be carried on one particle.

Sources:

Block keyword:	`Sources:`
Description:	Gives information defining sources. Unnamed block.

Column keywords:	Definition:
`Name`	Name of the source.
`Shape`	'Cuboid', 'Ellipsoid' or 'Cylindroid'. Default = 'Cuboid'.
`Set of Locations`	Name of set of potential source locations.
`Location`	Name source location.
`H-Coord`	Name horizontal coord system used to define the source.
`Z-Coord`	Name vertical coord system used to define the source.
`H-Grid`	Name of horizontal grid used to define source locations.
`Z-Grid`	Name of vertical grid used to define source locations.
`X`	X coordinate of source centre.
`Y`	Y coordinate of source centre.
`Z`	Z coordinate of source centre.
`dH-Metres?`	Specify dX and dY in metres or coord system H-Coord.
`dZ-Metres?`	Specify dZ in metres or coord system Z-Coord.
`dX`	Length of source.
`dY`	Width of source.
`dZ`	Height of rectangular source.
`Angle`	Angle of dX to the X axis (measured anticlockwise).
`Uniform Area?`	Makes the source uniform across its horizontal area as opposed to being uniform in terms of the area dXdY defined by the coord system (at least if TopHat is true and particles are used). Default = No.
`No Reflect?`	Deletes any part of the source below the ground instead of reflecting it (at least if particles are used). Default = No.
`Source Strength`	Species and strength of source for the species, the source strength being given as the total mass released or the release rate.
`Time Dependency`	Indicates which source time dependency to use, if any.
`Plume Rise?`	Indicate whether or not to use plume rise scheme. Default = 'No'.
`Temperature`	Temperature for buoyant release.
`Volume Flow Rate`	Volume flow rate of emission.
`Flow Velocity`	Velocity of emission.
`# Particles`	Lower limit on number of particles released or particle release rate.
`Max Age`	Maximum age of particles.
`Top Hat`	Indicate whether or not to use top hat distribution.
`Start Time`	Starting time of release.
`Stop Time`	End time of release.
`Particle Diameter`	Diameter of particulate.
`Particle Density`	Density of particulate.
`Particle Shape`	Shape parameter of particle.
`Sedimentation Scheme`	Scheme to calculate sedimentation, includes methods for non spherical particles.
`Particle size distribution`	Indicates which distribution of particulate sizes to use, if any.
`Met-dependent Source Type`	Indicates that source depends on met data. Recognised values are: blank for no met dependency (default), 'Dust', 'Sea Salt' and 'Iterative Plume Rise Model'.
`Source Groups`	Semicolon separated list of source groups which this source belongs to.
`Lagrangian-Eulerian Time`	Time (after release) at which particle mass is transferred to field.
`Z-Grid Iterative Plume Rise`	Vertical grid for use by the iterative plume-rise model.
`Dry Gas Mass Fraction`	Initial gas mass fraction for plume-rise model. Default = 0.03.
`Water Vapour Mass Fraction`	Initial moisture content for plume-rise model. Default = 0.0.
`Plume Rise Height`	Target rise height for plume-rise model.
`Summit Elevation`	Summit height for plume-rise model.
`Distal Fine Ash Fraction`	Fraction of ash that remains in the distal cloud. Default = 0.05.

For each source, the set of source groups this particular source belongs to can be set up. The output can then be restricted to one or more source groups.

Note that if plume rise is indicated then a temperature must be specified. Furthermore, *Volume Flow Rate* and *Flow Velocity* cannot be specified simultaneously.

Multiple species may be released from a single source by having multiple Source Strength columns.

Specifying any of *Particle Density*, *Particle Diameter* and *Particle Size Distribution* implies a particulate release. It is not possible to specify *Particle Diameter* and *Particle Size Distribution* simultaneously.

If *Particle Shape* is given, a *Sedimentation Scheme* must also be given. Sedimentation Schemes available are White (1974), Wilson and Huang (1979) and Ganser (1993), which are identified by 'White', 'WH' and 'Ganser' respectively. Note the Wilson and Huang (1979) scheme is only valid for volcanic ash particles.

 # Particles should be given in the form

$$n$$

or

$$m \ / \ \{\mathrm{s}|\mathrm{sec}|\mathrm{min}|\mathrm{hr}|\mathrm{day}\}$$

where n is the total number of particles and m is the release rate. Source strength should be given in the form

$$\text{species-name } n \text{ unit}$$

or

$$\text{species-name } m \text{ unit} \ / \ \{\mathrm{s}|\mathrm{sec}|\mathrm{min}|\mathrm{hr}|\mathrm{day}\}$$

where n is the total mass released and m is the release rate. {...|...|...} indicates alternatives.

Particle Size Distribution:

Block keyword:	`Particle Size Distribution:`
Description:	Gives information on a particle size distribution. Named block.

Column keywords:	Definition:
`Diameter Range Boundary`	The maximum diameter of the particulate in a given range.
`Cumulative Fraction`	The fraction of particulates having diameters less than or equal to the specified diameter.
`Particle Density`	The density of particles in each size range.
`Particle Shape`	The shape of particles in each size range.
`Representative Diameter`	The representative diameter for particles in each size range. This is used to calculate the sedimentation velocity for scalar fields containing solid species. If it is not specified, the default value is the geometric mean of the size range.

Note that the first row must specify zero fraction of particulates having a diameter below the specified diameter, in effect, a minimum diameter. Correspondingly, the last row must specify that 100% of particulates are below the specified maximum, in effect, the absolute maximum. For example,

```
Particle size distribution: Volcano A
Diameter Range Boundary,    Cumulative Fraction
0.1,                        0.0
0.3,                        0.001
1.0,                        0.006
3.0,                        0.056
10.0,                       0.256
30.0,                       0.956
100.0,                      1.0
```

When the *Particle Shape* and *Particle Density* are entered via the `Particle Size Distribution` block then the last line of the *Particle Shape/Particle Density* column must be left blank. If the first line is blank then all subsequent lines must be blank.

Source Time Dependency:

Block keyword:	`Source Time Dependency:`
Description:	Gives information on a time dependent source. Named block.
Column keywords:	**Definition:**
`From 1`	Start time.
`To 1`	End time.

`From 2`
...
`From 5`
	A second to fifth (optional) start time.

`To 2`
...
`To 5`
	A second to fifth (optional) end time.

`Factor`	The factor by which the source strength is multiplied during the specified time period.

The time period over which the source strength is multiplied by the given factor can be specified in a number of ways. Formally this can be understood by considering each pair of start/end times as a set. For a given row, the source strength is multiplied by the time dependent factor if the release time occurs within the intersection of multiple sets of start/end times. If there are two or more rows then each row is a set of the set of start/end times. In this case, the source strength is multiplied by the relevant time dependent factor if the release time occurs in the union of the set of rows. Moreover, if the release time occurs within the intersection of these sets, then the source strength is multiplied by the product of the time dependent factors. The following examples illustrate the possibilities.

Example 1

```
Source time dependency: Power Station A
From 1     , To 1        , From 2     , To 2       , Factor
1/1/* 00:00, 31/3/* 23:59, */*/* 08:00, */*/* 10:00, 2.0
```

In the first example, the source strength is multiplied by 2.0 whenever the release time is between 08:00 and 10:00 and on any day between 1st January and 31st March. The 'wild card' character '$*$' in the 'year' field indicates that this is true for *any* year. A wild card can be specified in any field larger than 'seconds'. It must occur in all fields larger than the smallest one in which it is specified. For example, the combination $*/1/*$ will generate an error.

Example 2

```
Source time dependency: Power Station B
From 1          , To 1            , Factor
1/1/2001 00:00, 31/3/2001 23:59, 2.0
22:00           , 10:00           , 2.0
```

In this case, the source strength is multiplied by 2.0 if the release time is between 1/1/2001 and 31/3/2001 or between 22:00 and 10:00. If the release time is between 22:00 and 10:00 and 1/1/2001 and 31/3/2001, then the time dependent factors are multiplied together and the source strength is multiplied by 4.0.

Example 3

```
Source time dependency: Traffic
From 1    , To 1     , Factor
Mon 09:00, Fri 18:00, 2.0
```

In the final example, the source strength is multiplied by 2.0 if the release time is between Monday at 09:00 and Friday at 18:00.

When wild cards are present all parts of the time must lie in their appropriate range (e.g. minutes must lie in the range 0 and 59). The optional start/end times do not have to be nested but there is an efficiency saving if the most rapidly varying time period (within a slower varying one) is specified last. The maximum number of optional start/end times is four.

Output Requirements — Fields:

Block keyword:	`Output Requirements - Fields:`
Description:	Gives information on the output fields required. Unnamed block.
Column keywords:	Definition:
`Name`	Name by which to identify the output requirement (optional; if not specified a name will be assigned).
`Quantity`	Type of field required (e.g. 'Air Concentration', 'puff centres' or 'deposition rate').
`Decay deposition?`	Indicates whether time averaged or integrated deposition rate is to be calculated with allowance for decay of the deposited material. Default = yes.

`Species`	Species for which the field is required (if more than one species use semi-colon separated list).
`Source`	Source for which the field is required (blank indicates all sources).
`Source Group`	Source group for which the field is required (blank indicates all source groups).
`H-Grid`	Horizontal grid.
`Z-Grid`	Vertical grid.
`T-Grid`	Temporal grid.
`S-Grid`	Travel time grid.
`H-Coord`	Horizontal coordinate system.
`Z-Coord`	Vertical coordinate system.
`BL Average?`	Results are averaged over boundary layer depth. Default = 'No'.
`T Av Or Int`	'Av' indicates time averaging, 'Int' indicates time integrating, and 'No' or blank indicates no time averaging or integrating.
`Av Time`	Average / integration time.
`# Av Times`	Number of instantaneous results to be used to form average.
`Ensemble Av?`	Results are averaged over the ensemble of cases. Default = 'No'.
`Probabilities`	Semicolon separated list of values for which exceedence probabilities are required.
`Percentiles`	Semicolon separated list of percentiles required.
`P Time`	Time over which the percentile is calculated.
`P dT`	Time interval between values contributing to the percentile.
`Ensemble P?`	Probabilities and percentiles are over the ensemble of cases. Default = 'No'.
`Fluctuations?`	Indicates that concentration fluctuations are to be taken account of. Needed if *Quantity* = 'Sigma C'. In due course it will also allow fluctuations to be taken into account in calculating probabilities, percentiles and moments. Default = 'No'.
`Sync?`	Calculate output when particle/puffs are synchronised.
`X Scale`	Plot scale for graphical output. Default = '1.0'.
`Y Scale`	Plot scale for graphical output. Default = '1.0'.
`Across`	A string of characters indicating that the variables T (time), S (travel time), X, Y or Z are placed at the top of columns (as opposed to at the left of rows).
`Separate File`	A string of characters indicating that different values of T (time), S (travel time), X, Y or Z should be placed in seperate files. N indicates output starts afresh after a restart (possibly overwriting any previous files).
`Output Format`	A string of characters defining the output file format: I – Include grid point Indices. A – Align columns. Z – Include grid points with Zero values. F – Flush buffer after writing to file to keep file up to date. This should only be used where needed (e.g. output of particle numbers to monitor progress of run). It can slow down the run if used for large output files. 2 – Format as NAME II.
`Output Route`	A string of characters defining the output route: D – Numerical output to Disk. S – Numerical output to Screen. G – Graphical output to screen (Windows PC only).
`Output Group`	Output group name. Output with same group name is placed in same file.
`Particle Size Distribution`	Name of particle size distribution giving the particle size ranges to be used in calculating the field. Blank indicates no restriction by particle size.
`Semi-infinite approx?`	Indicates that cloud gamma doses are to be calculated using the semi-infinite cloud approximation. Must be 'No' unless Quantity is 'Adult Effective Cloud Gamma Dose', 'Adult Lung Cloud Gamma Dose', 'Adult Thyroid Cloud Gamma Dose' or 'Adult Bone Surface Cloud Gamma Dose'. Default = 'No'.
`Material unit`	Unit of output requirement. Can be specified for `Air Concentration`, `Dry Deposition Rate`, `Wet Deposition Rate`, `Deposition rate`, `Mixing Ratio` and `E Mixing Ratio`, see comment below.

Probabilities and percentiles are calculated after any averaging defined by *Av Time* etc and by *Ensemble Av?*. They can be taken over time or over the ensemble or both (although of course *Ensemble P?* can't be used with *Ensemble Av?*). Probabilities and percentiles can't both be requested in one output requirement.

Possible values for *Quantity* are as follows:

Quantity:	Description:
`Air Concentration`	Air concentration from particles or puffs
`Mixing Ratio`	Mixing ratio output from particles or puffs
`Dry Deposition Rate`	
`Wet Deposition Rate`	
`Deposition Rate`	
`# Particles`	
`# Puffs`	
`# Particle Steps`	
`# Puff Steps`	
`Mass`	
`Mean Z`	
`Sigma Z`	
`X Stats`	
`Mean Travel Time`	
`Puff Centres`	
`Sigma C`	
`Chemistry Field`	
`Eulerian Concentration`	Air concentration on Eulerian field
`E Mixing Ratio`	Mixing ratio output from Eulerian field
`Concentration`	Combined air concentration from particles and field
`Sigma WW`	
`Tau WW`	
`Mean Flow U`	
`Mean Flow V`	
`Mean Flow W`	
`Temperature (K)`	
`Potential Temperature (K)`	
`Specific Humidity`	
`Pressure (Pa)`	
`Density`	
`Topography`	
`u-star`	
`Sensible Heat Flux`	
`Boundary Layer Depth`	
`Wind Speed`	
`Wind Direction (degrees)`	
`Precipitation Rate (mm/hr)`	
`Temperature (C)`	
`Cloud Amount (oktas)`	
`Relative Humidity (%)`	
`Pasquill Stability`	
`# Particles By Species`	
`Progress (%)`	
`Clock Time`	
`X`	X coord at output location in user specified coord system
`Y`	Y coord at output location in user specified coord system
`Sigma VV`	

`Mesoscale Sigma VV`	
`Cloud Water (kg/kg)`	
`Cloud Ice (kg/kg)`	
`3d Cloud (Fraction)`	
`Roughness Length`	
`Sea Level Pressure (Pa)`	
`Photon Flux`	
`Adult Effective Cloud Gamma Dose`	
`Adult Lung Cloud Gamma Dose`	
`Adult Thyroid Cloud Gamma Dose`	
`Adult Bone Surface Cloud Gamma Dose`	
`Area at risk`	
`Land Use Fractions`	
`Canopy Water`	
`Leaf Area Index`	
`Canopy Height`	
`Stomatal Conductance`	
`Soil Moisture`	
`Land Fraction`	
`Convective Cloud Base`	Convective cloud base in m agl
`Convective Cloud Top`	Convective cloud top in m agl
`Eulerian Total Deposition Rate`	Total deposition rate from Eulerian field
`Eulerian Dry Deposition Rate`	Dry deposition rate from Eulerian field
`Eulerian Wet Deposition Rate`	Wet deposition rate from Eulerian field
`Original Source Strength`	(see iterative plume rise scheme)
`Revised Source Strength`	(see iterative plume rise scheme)

For particle concentrations, deposition rates and mixing ratios the unit in the field requirement can differ from the one of the corresponding source species. In the first two cases conversions between mass units or between activity units are allowed, see Tabs. 4 and 5 for a list of supported units. In addition, units not within these tables are allowed, as long as they agree between the species and the output requirement. If the Material unit of an output requirement is left blank, the unit of the source species will be used. For volumetric mixing ratios (relative to dry air) the values `ppm` and `ppb` are allowed, as long as the source species is given in mass units. Mixing ratios can only be calculated if both a vertical and a horizontal grid are defined, vertical averages/integrals are not allowed. This applies to both *Mixing Ratio* and *E Mixing Ratio*. Vertically integrated air concentrations can be output in Dobson units (`DU`) if the source species is given in mass units.

For more details see the model document on unit conversion.

symbol	name	ρ
`t`	tonne	10^6
`kg`	kilogram	10^3
`g`	gram	1
`mg`	milligram	10^{-3}
`mcg`	microgram	10^{-6}
`lb`	pound	453.59237
`oz`	ounce	28.349523125

Table 4: Mass units and conversion factors. The reference unit is 1 gram (`g`).

symbol	name	ρ
`Bq`	Becquerel	1
`mBq`	milliBecquerel	10^{-3}
`mcBq`	microBecquerel	10^{-6}
`Ci`	Curie	3.7×10^{10}

Table 5: Activity units and conversion factors. The reference unit is 1 Becquerel (`Bq`).

Output Requirements — Sets of Particle/Puff Information:

Block keyword:	`Output Requirements - Sets of Particle/Puff Information:`
Description:	Gives information on single particle or puff. Unnamed block.
Column keywords:	Definition:
`Output Name`	Name by which to identify the output requirement
`Particles?`	Include particle information in output?
`Puffs?`	Include puff information in output?
`First Particle`	First particle to be included. Default = 0.
`Last Particle`	Last particle to be included. Default = 0.
`First Puff`	First puff to be included. Default = 0.
`Last Puff`	Last puff to be included. Default = 0.
`Source`	If source name given output is restricted to this source.
`Met?`	Include information on met?
`Mass?`	Include information on mass
`Plume Rise?`	Include information on plume rise?
`Dispersion Scheme?`	Include information on dispersion scheme?
`Puff Family?`	Include information on puff family (puffs only).
`H-Coord`	Horizontal coordinate system.
`Z-Coord`	Vertical coordinate system.
`T-Grid`	Temporal grid.
`Sync?`	Calculate output when particle/puffs are synchronised.
`Screen?`	Real time numerical output on screen.
`Disk?`	Output to file.
`Output Format`	A string of characters defining the output file format: T – Separate Times in separate files. P – Separate Particles in separate files. F – Flush buffer after each sync time interval to keep file up to date. This should only be used where needed. It can slow down the run if used for large output files.
`Output Route`	A string of characters defining the output route: D – Numerical output to Disk. S – Numerical output to Screen. G – Graphical output to Screen (Windows PC only).

Output Requirements — Pdfs:

Block keyword:	`Output Requirements - Pdfs:`

Description:	Unnamed block.

Column keywords:	Definition:
`Name`	Name by which to identify the output requirement
`Species`	Species for which the pdf is required.
`Source`	Source for which the pdf is required (blank indicates all sources).
`Source Group`	Source group for which the pdf is required (blank indicates all source groups).
`Pdf Type`	Type of pdf. 1 = exceedence probabilities, 2 = percentiles.
`H-Grid`	Horizontal grid.
`Z-Grid`	Vertical grid.
`T-Grid`	Temporal grid.
`T Av Or Int`	Results averaged or integrated over Average Time.
`Average Time`	Average / integration time.
`Screen?`	Real time numerical output on screen.
`Disk?`	Output to file.
`Stat?`	Indicates results are statistically processed. Default = 'No'.

Chemistry Options:

Block keyword:	`Chemistry Options:`

Description:	Gives options for chemistry. Unnamed block.

Column keywords:	Definition:
`Chemistry Folder`	Folder containing background chemistry fields. Default = `../Resources/Stochem/`.

Sets of Dispersion Options:

Block keyword:	`Sets of Dispersion Options:`

Description:	Gives information on dispersion options. Unnamed block.

Column keywords:	Definition:
`Max # Particles`	Maximum number of particles that can be used.
`Max # Full Particles`	Maximum number of full particles that can be used. Full particles are particles that have any options other than the most straighforward (e.g. velocity memory or plume rise).
`Max # Puffs`	Maximum number of puffs that can be used.
`Max # Original Puffs`	Maximum number of original puffs that can be used. Original puffs are those released at the source and not created by puff splitting.
`Particle Ceiling`	When the number of particles reaches Particle Ceiling, the mass per particle for particles being released or split is increased by a factor Particle Factor and the release or splitting rate is correspondingly reduced. This helps avoid Max # Particles being reached. Default = Max # Particles, i.e. no reduction in release rate (until the particle limit is reached).
`Particle Factor`	See Particle Ceiling. Default = 1, i.e. no reduction in release rate (until the particle limit is reached).

`Skew Time`	Travel time over which the model allows for skewness in the velocity variance profile.
`Velocity Memory Time`	Travel time for which the more expensive dispersion scheme with velocity memory is used. Set to 00:00 to use the cheap scheme for the entire run.
`Inhomogeneous Time`	Travel time for which height dependent inhomogenous velocity variance profile is used. Set to 00:00 to use the homogenous profile for the entire run.
`Mesoscale Velocity Memory Time`	Travel time for which the more expensive dispersion scheme with velocity memory is used for the unresolved mesoscale motions component of the velocity field. Set to 00:00 to use the cheap scheme for the entire run.
`Damping?`	Damps the spread with the diffusive (i.e. no velocity memory) model to make it more like the velocity memory model. Applies to both turbulence and unresolved mesoscale motions. Default = 'Yes'.
`Puff Time`	Travel time over which puffs are used. Set to 00:00 to use particles for the entire run.
`Sync Time`	Time interval at which particles are synchronised.
`Computational Domain`	Domain of interest.
`Puff Interval`	Interval between puff releases.
`DeltaOpt`	
`Time of Fixed Met`	Time of met to use for a fixed met run.
`Deep Convection?`	Switch a deep convection scheme on/off. Acceptable answers: 'No', 'Old' and 'New'.
`Radioactive Decay?`	Switch radioactive decay on/off for relevant species.
`Agent Decay?`	Switch agent decay on/off.
`Dry Deposition?`	Switch dry deposition on/off.
`Wet Deposition?`	Switch wet deposition on/off.
`Max deposition height`	Maximum deposition height (in m). Default = boundary layer depth. Particles below min(Max deposition height, boundary layer depth) will contribute to dry deposition. Useful for situations where material is not well mixed within the boundary layer (e.g. near source problems). For use with particles only (i.e. not to be used with puffs). The number of particles released may need to be increased to prevent noisy deposition fields.
`Mesoscale Motions?`	Switch modelling of unresolved mesoscale motions on/off.
`Chemistry?`	Switch chemistry on/off.
`Turbulence?`	Switch Turbulence on/off.
`A1`	Parameter of the puff scheme. Default value = 50.
`A5`	Parameter of the puff scheme. Default value = 3.
`A7`	Parameter of the puff scheme. Default value = 2.
`Vertical velocity`	Default = 'Yes'.

`Skew TimeSkew`, `Velocity Memory Time` and `Inhomogeneous Time` have to satisfy

$$\texttt{Skew TimeSkew} \leq \texttt{Velocity Memory Time} \leq \texttt{Inhomogeneous Time}.$$

Appendix — The format of headed input files

A headed input file consists of one or more blocks of data. Other material can be added before, between or after the blocks without having any effect on the model (provided the material can't be interpreted as a block). This can be useful for adding comments etc. Different types of block are possible. Each type of block has a unique block keyword associated with it and the block type also determines whether blocks of that type must be unnamed or named blocks (a given type cannot allow both named and unnamed blocks). The structure of an unnamed block is as follows

> Block keyword
> Comma separated column keywords
> Rows of comma separated data items
> Blank line (or end of file)

while a named block has the following form

> Block-keyword Block-name
> Comma separated column keywords
> Rows of comma separated data items
> Blank line (or end of file).

Spaces can be included or omitted before, between or after any of the elements (block keyword, block name, comma, column keywords, data item) and optional commas are allowed at the end of the comma separated lines. An exception where trailing commas are not optional concerns a block with a single column — if one wants to input a blank piece of information then this must be followed by a comma to distinguish it from a blank line signifying the end of the block — however hopefully one would not want to do this often in practice. Each line of data must be interpretable as containing the same number of items as there are column keywords (the reason why there is room for interpretation here is that a line ending in a comma or a comma and some spaces can be interpreted as ending with the comma or ending with a blank string following the comma). Block keywords and column keywords are not case sensitive, but block names and data items may be, depending on how they are used in the model. Other than within a block, any line begining with a recognised block keyword (possibly preceeded by spaces) is interpreted as the start of a block — if the rest of the block does not conform to the required format an error message will be given.

The column keywords which are recognised by the model will vary from block type to block type. Unrecognised names are permitted but will result in warning messages and the values of the corresponding data items will have no effect on the model. From the perspective of the file format, there is no need to include all or indeed any of the recognised column keywords; however the model will of course complain if there is inadequate information given. Duplicate recognised column keywords will result in an error message. Valid data items (e.g. numbers, strings, blanks etc) corresponding to each column keyword will vary from block type to block type. However we note that, because spaces before and after data items are not significant, blank entries and zero length strings are interpreted the same way. The column keywords can be in any order, but the order of the data items must correspond to the order of the headers. Block keywords, block names and column keywords can never be blanks.

If there is more than one file in a set of headed input files, the names of the second and subsequent files must be given in an unnamed block in the first file of the set.

A number of unnamed blocks of the same type have the same effect as if the blocks were combined into one (with the set of column keywords being given by the union of the sets of column keywords in the individual blocks). This can be useful for example if one wants to add some extra species but don't want to edit the reference file of species, or if one has radioactive and non-radioactive species — by putting these in separate blocks there is no need to have all the column keywords which relate to radioactivity for the non-radioactive species. The order of unnamed blocks and lines within such blocks can affect issues of appearence (e.g. the order in which output fields appear in an output file), and, as noted above, any block giving the names of the second and subsequent files in a set must lie in the first file in the set. Apart from these trivial situations, the order of the blocks, the way they are distributed between the files, and the order of lines in the blocks, is irrelevant.

For named blocks one cannot have more than one block of the same type with the same block name. The order of named blocks and the way they are distributed within the files is irrelevent. However the order of rows within a named block is usually relevent as such data usually takes the form of arrays (e.g. the Eta Definition example discussed in §2 above).