Apertum

Airviro Specification

$$\frac{\partial c}{\partial t}+u\frac{\partial c}{\partial x}+v\frac{\partial c}{\partial y}+w\frac{\partial c}{\partial z}=\frac{1}{\rho}\frac{\partial}{\partial z}\left[\rho\left(K_z\frac{\partial c}{\partial z}+w_s c\right)\right]+\frac{Q}{\rho}$$

Airviro Specification v5.00

Part II: Appendices

Airviro Specification v5.00

Part II: Appendices

Amendments

Version	Date changed	Cause of change	Signature
2.30	130809	Changed paragraph E3.30	KH
3.00	141128	Changes to update to 3.00 functionallity	PI
3.11	150420	Changes to update to 3.11 functionallity	GS
3.20	161220	Changes to update to 3.20 functionallity	GS
4.00	180701	Changes to update to 4.00 functionallity	GS
4.00	181003	Review	GS
5.00	220830	Update	GS

Contents

Appendix C: System Databases and their Interfaces

C1 Introduction

The whole Airviro system is based on a large number of databases. From the customers point of view the most useful ones are either to do with time series data or emission data. A concise list of these databases is found below:

C1.1 Databases Related to Time Series Data

time series database	measured data values in different time resolutions
station database	information about data collection sites
parameter database	information about the substances for which time series data exists
key list database	a list of data that require further calculation
bitmap database	information about the presence or absence of data in the time series database

C1.2 Databases related to Emission Data

Point&area_sources database	information about point and area sources
Road database	information about road sources
Grid_layers database	information about grid emissions
Formula database	information about time variation formulae linked with the data database
Fuel&substance_group database	information about fuel types and substance groups linked with the data database
Road_type database	information about road types linked with the road database
Vehicle database	information about different vehicles linked to the road type database
Substances database	the substance list
Searchkeys database	the different searchkey lists

C2 The Time Series Database and Associated Databases

C2.1 The Station Database

The Airviro station database is used to describe a station that will be used to represent a data collection site. If the data is to be loaded manually into the time series database then the information required for the associated station is minimal, but if automatic data collection is going to take place some more detailed information is required. However, the station database does not contain any detailed information about how to retrieve data from the logger. This is handled completely by the external protocols started by the data collection daemon **`cold`** (see *Appendix B3.2*), and in this case the main purpose of the station database is to control when the external protocols are to be run and what resources they need. It also keeps statistics about failed collection attempts.

The ASCII interface to the station database is the command **`stndb`**, but an interface exists in the Indico Administration module. The following fields should always be defined for each station in the station database:

KEY The station key, which is a unique three-letter identifier for the station (this can be any combination of the characters A-Z, a-z and 0-9).

NAME The station name that is used in the Indico administration user interface to the databases and in the Indico presentation module.

OPERATE This should be set to 1 if automatic data collection is active for this station, otherwise it is 0.

BIRTHTIME When you create a station then this field is automatically filled in with the date and time of creation.

DEATHTIME The date and time when a station dies. This should be the time when the station stops collecting data so that the period birthtime-deathtime is the time when the station was used for measuring.

XCOORD The x coordinate of the stations location on the map.

YCOORD The y coordinate of the stations location on the map.

HEIGHT The height of the station above the ground.

STNGROUP A bit mask indicating which station groups the station belongs to. Very useful if there are a lot of stations.

The following fields must also be specified if the station is going to be used in conjunction with automatic data collection:

LOGGING This should be set to 1 if a log file of all the transactions with the station is to be kept, otherwise 0

DUMMY This should be set to 1 if it is just a dummy (i.e. not real) station, otherwise 0.

RAWDATA This should be set to 1 if you want to save the raw data as well, otherwise 0.

XPRNAME The name of the external protocol to use, with or without the “xpr” in front.

XPRARG The argument to send to the external protocol to tell it from when to collect data. Some loggers use a normal date format but other loggers require a pointer. This is sent to the logger as a text string.

RESOURCE This describes what resources can be used for data collection. Device names are separated by a space. This can either be a specific device name such as /dev/cul0p1, or a resource type (such as modem) and an acceptable baud rate

UPDATETIME This is the date and time associated with the last piece of data that was successfully fetched - format: YYMMDD hh:mm represented in database time (usually without Daylight Savings Time)

NEXTCALLTIME This is the date and time when the station is next due to be called - format: YYMMDD hh:mm represented in local time (Daylight Savings Time)

CALLINTERVAL This is the length of time to wait (in minutes) between each call.

RETRYINTERVAL This is the length of time in minutes to wait to retry if a call has failed.

BADCALLS This is a count of the number of consecutive bad calls to the station.

BADCALLSALRM When BADCALLS reaches this alarm limit the n an alarm can be sent. The alarm can be configured to be sent to the printer or as mail to the systems administrator, although the default is no action at all.

BADCALLSSTOP When BADCALLS reaches this limit then data collection is turned off for this station, i.e. OPERATIONAL is set to 0. An alarm can also be sent which can be configured to be sent to the printer or as mail to the systems administrator, although the default is no action at all.

It is possible to specify a time window to control the times when the station is called by placing restrictions on the days, hours or minutes to call. Intervals are specified with a - e.g. “0-3”, and are separated with a comma. An asterisk (*) implies all options.

CALLWEEKDAYS	Here the days of the week to call in are specified, The days are represented by the number 0-6, where 0 is Sunday.
CALLHOURS	Here the hours to call in are specified using the 24 hour clock.
CALLMINUTES	This specifies which minutes in any hour to call in. Only one interval can be specified, but it may be reversed. For example, 5-20 means that calls will only be made between 5 past and 20 past, and 55-5 means that calls will only be made between 5 to and 5 past.
MAXFACTOR	This is the maximum number of minutes that the data collection may take per hour of fetched data. A constant offset of 3 minutes is added to this to allow for modem initialisation. If this limit is exceeded then the data collection process will be killed. Enter a 0 for no limit. If this time is needed to be given in seconds then an s can be written after it.
HISTGOOD	This is the total number of successful calls that have been made to the station.
HISTXPRBAD	This is the total number of calls that failed due to errors detected by the external protocol.
HISTCOLDBAD	This is the total number of calls that failed due to limitations imposed by **`cold`**, the data collection daemon (e.g. if a process exceeds its maximum running time).

C2.2 The Parameter Database

The Airviro parameter database is used to store all attributes for specific parameters. Data that is to be stored in the time series database must always be linked to a parameter. The same parameter definitions are used by all stations. The ASCII interface to the parameter database is the command **`pardb`** but there is also an interface in the Indico Administration module. The following fields can be specified for each record in the parameter database:

KEY	The parameter key made up of 4 characters which must be unique to each parameter
DESCR	The parameter name, used for identification of the parameter in the user interfaces
UNIT	Units of measurement (e.g. mg/m3).
ALARMLIM	Upper limit for alarm (not used)
GRAPHMIN	Max default value to use for the y-axis on graphs in the Indico presentation module.
GRAPHMAX	Min default value to use for the y-axis on graphs in the Indico presentation module.
MAXEQ	Number of hours for which parameter is allowed to vary within the EPSILON value.
EPSILON	Allowed variation for consecutive values.

IMIN	Absolute minimum for parameter.
IMAX	Absolute maximum for parameter.
GRADIENT	Maximum variation per time unit.

The limits MAXEQ, EPSILON, IMIN, IMAX or GRADIENT are used by the Airviro database manager **avdbm** (see *Appendix B3.1*) to check all incoming data. If any of the limits are exceeded then a status value indicating this is associated with the data values that exceeded the limit.

C2.3 The Time Series Database

All data obtained from measurement stations are spooled to the time series database. The time series database has the capacity to store up to three associated values plus a status value for each time step. These values are called **v1**, **v2**, **v3** and **stat**.

- **v1** is the data value.
- **v2** is the standard deviation or peak value for the data (depending upon whether the logger provides such data or not).
- **v3** is the light intensity for Opsis values.
- **stat** is the status of the data value, provided either by the external protocol or by the limits defined in the parameter database. Up to 16 status codes are available and it is possible to change the texts, assign single digit codes and also to make use of status 9-13. The default status codes that are used are:

0	No value
1	Unchecked value
2	Manually marked invalid
3	Error in value detected by logger
4	Value below absolute min.
5	Value above absolute max
6	Too big gradient
7	Too small variations
8	Too large standard deviation
9	Opsis value
14	Data checked –OK
15	Manually changed

For each time resolution there are two databases:

1. In the **v1&stat** database each 32-bit word stored in the period array contains both **v1** and **stat**. The **v1** value is represented as a 4 byte float according to the IEEE floating-point standard (ANSI/IEEE 754-1985) with one modification: The four least significant bits, i.e. the four least significant bits of the mantissa, are used to store stat. This means that stat can assume values from 0 to 15, and that **v1** is stored with approximately 6 significant digits.

2. In the **v2&v3** database each 32-bit word stored in the period array contains both **v2** and **v3**. The **v2** value is represented as a 4 byte float according to the IEEE floating-point standard with one modification: The seven least significant bits, i.e.

the seven least significant bits of the mantissa, is used to store **v3**. This means that **v3** can assume values from 0 to 127, and that **v2** is stored with approximately 5 significant digits. Note that the Opsis light intensity is measured in percent and therefore varies between 0 and 100.

The advantage of storing **v1&stat** and **v2&v3** in separate databases is that many stations only deliver **v1** and there is no need to store **v2** or **v3**. This significantly reduces the size of the database on disk.

Before data is loaded to the time series database it is checked by **avdbm**, the database manager. First the following conditions are checked in the time series database key (explained in *C2.3.1*):

- the **station key** must exist in the station database
- the **time resolution** must exist
- the **parameter type** must be valid
- the **parameter key** must exist in the parameter database
- the **instance** must be alphanumeric

If the data still has status 1 (i.e. if it was not checked by the logger) then for each parameter each data value is checked against the following criteria which are specified in the parameter database:

- the value is larger than the minimum level
- the value is lower than the maximum level
- the gradient is less than the maximum gradient
- the number of consecutive equal values is less than the maximum allowed number
- the standard deviation is less than half the data value (this check is only for OPSIS data)

C2.3.1 Time Series Database Key

Data in the time series database is referenced by a key, assembled from several logical parts and is a total of 16 bytes long. A schematic description of the key is usually written as **SSSRTPPPPNNNtttt**. Each part of the key can be described as follows:

SSS This is the key to the station database. It associates this stored data with a particular measuring station. This key can be used to retrieve information about the station from the station database.

R This is the time resolution i character that indicates what time resolution this block of data has. The time resolution of the data decided the size of one data block and can be one of the following:

Char	Name	Resolution	One block
*	day	86400 s	240 days
+	hour	3600 s	10 days

s	half hour	1800 s	5 days
i	twenty minute	1200 s	80 hours
q	quarter	900 s	60 hours
t	ten minute	600 s	40 hours
f	five minute	300 s	20 hours
,	minute	60 s	4 hours

T This is the parameter type character that indicates the type of the stored parameter, i.e. which of the values **v1**, **v2**, **v3** that is relevant for this parameter and how they should be interpreted. For compatibility with previous releases, other letters are accepted here.

Char	**v1**	**v2**	**v3**
M	value	-	-
K	value	peak	-
v	Value	-	-
O	value	std.dev.	light

PPPP This is the key to the parameter database. It associates this stored data with a certain parameter. This key can be used to retrieve information about the parameter from the parameter database.

NNN This is a character string representing the **instance**. It separates the identical parameters measured at the same station. It has different interpretations for different stations. For OPSIS stations it means the path number and for other stations it usually is interpreted as the height of the measuring instrument relative to the ground.

tttt This four byte group is binary data. It is the time for the first value in the block in the database record. The time is given as a 32-bit word containing the number of seconds since 1 January 1970.

Different subsets from the value key have been given their own names:

SSSRTPPPPNNNtttt is called the **value key** since it points out a certain block of values in the database.

SSSRTPPPPNNN is called the **series key** since it refers to a whole series of values.

TPPPPNNN is called the **channel key** since it corresponds to a channel on a specific station.

SSSRT is called the **station channel group key** since it refers to a set of channels on a station with the same time resolution and parameter type.

C2.4 The Keylist Database

The keylist database is a list of search keys for records in the time series database which require further processing by Airviro. For each block there is a 32-bit word with bits set for special tasks to be performed on the corresponding block in the time series database. When the task has been successfully performed the bit is cleared and when all bits are cleared, the whole record is deleted. This mechanism is used for batch processing, e.g. mean value calculation, that only takes place at certain times. Further processing can be one or more of the following, depending upon which combination of bits are set:

Value	Bit	Description
1	0	Calculate half hourly mean values from minute, five-minute, ten-minute or fifteen-minute mean values
2	1	Calculate hourly mean values from minute, five-minute, ten-minute, fifteen-minute, twenty-minute or half-hour mean values.
4	2	Calculate daily mean values from lower resolution values.
8	3	Calculate measurements from voltage values.
16	4	Calculate hourly sums from 5 minute values.
32	5	Calculate daily sums from hourly values.
64	6	Calculate NO2 values from No and NOX.
128	7	Adjust TEOM values.

The contents of the keylist database can be displayed/printed using the showkeyl command. The contents of the keylist database can be modified using the initkeyl command although this is not usually necessary as the relevant bits are set automatically after data collection.

The programs mean, scale, sumts, calcNO2 and adjustTeom scan the keylist database every time they are run. Keys for records in the time series database that require processing are read from the keylist database. The corresponding records are then processed and the keylist record replaced with the appropriate flag reset. Data to be stored in the database is spooled to the database manager for storage.

C2.5 The Bitmap Database

The bitmap database stores the presence or absence of measured values in the database as a compact (one bit per value) bit stream. This database is updated by the database manager during storage of new values. Note that it is just the existence of data that is stored here, and not whether the data is marked as good or not.

All resolutions are stored in the same physical database. Each value in the time series database corresponds to a bit in the bitmap database. If the status field for a value in the time series database is zero the corresponding bit is clear, otherwise it is set. The first bit in the block is bit 0 of word 0, the second is bit 1 of word 0.

Since there are 32 bits in each word used, each block therefore contains 32 times the number of time steps so the bitmap database is approximately 32 times smaller than the time series databases together.

C3 The Emission Database and Associated Databases

C3.1 Introduction

The emission database consists of the following subdatabases:

- substances
- searchkeys
- point&area_sources
- fuel&substance_group
- formula
- vehicle
- road_type
- road
- grid_layers

The substances database has only one record. This block is shared between 1023 substance descriptions of 31 bytes each. A substance is referenced by the description number and not by its name.

The searchkeys database contains 5 searchkey lists linked to searchkey1, searchkey2, searchkey3, searchkey4 and searchkey5 in the EDB. The first searchkey list contains 128 places and the other 3 contain 32 places each. Each searchkey may be up to 31 bytes long.

It is possible to add grid layers of emissions to an EDB, however this can only be done via an ASCII interface to the grid_layers database and cannot be done using the EDB user interface. Another way of adding grids is using Wedbed.

C3.2 Point and Area Sources

Each record of the point&area_sources database contains the following information:

X1	x coordinate.
Y1	y coordinate.
X2	Upper right x coordinate (for area sources).
Y2	Upper right y coordinate (for area sources).
NAME	The name of the source (up to 47 characters).
INFO	A string for information (up to 47 characters). Can be used for searching.
INFO2	A second string for information (up to 63 characters). Can be used for searching.
ADDRESS	The street address of the source (up to 47 characters).
POSTADDRESS	The post address of the source (up to 47 characters).
INFOGIVER	The name of the supplier of information (up to 47 characters).
DATE	Manually entered date showing when the information was obtained
CHANGED	A date that is automatically updated when the source is altered
MISC	Miscellaneous information (up to 47 characters).

MAXEFFECT	Maximum effect for energy producers. If this field is non-zero then FUEL must reference a fuel with a non zero energy value.
CHIMNEY HEIGHT	The height of the chimney in metres.
GASTEMPERATURE	Temperature of the exhaust gas in degrees Celsius.
GAS FLOW	Exhaust gas velocity in metres per second.
FUEL	A reference to the fuel database.
FORMULA	A reference to the formula database. Has to be non zero.
SEARCHKEY1	A reference to the first searchkey database.
SEARCHKEY2	A reference to the second searchkey database.
SEARCHKEY3	A reference to the third searchkey database.
SEARCHKEY4	A reference to the fourth searchkey database.
SEARCHKEY5	A reference to the fifth searchkey database.
CHIMNEY OUT	Outer diameter of the chimney in metres.
CHIMNEY IN	Inner diameter of the chimney in metres.
HOUSE WIDTH	The width of surrounding buildings in metres.
HOUSE HEIGHT	The height of surrounding buildings in metres
NOSEGMENTS	Number of directions to specify information about surrounding houses.
BUILD_WIDTHS	Building width in NOSEGMENT directions.
BUILD_HEIGHTS	Building height in NOSEGMENT directions.
BUILD_LENGTHS	Length of building in NOSEGMENT directions.
BUILD_DISTFARWALL	Distance to the most distant wall of the building in NOSEGMENT directions.
BUILD_CENTER	Distance to the centre of the building in NOSEGMENT directions.
ALOB	A text string used to store application specific data.
POINT0 **POINT1** : **POINTn**	This is used if there isn't a reference to the fuel database. Each POINTi describes a different substance and value.
FUEL EMISSION	This is used instead of the POINTs if there is a reference to the fuel database and gives the emission of the fuel in tons per year.

The purpose of the data database is to store information about point and area sources. The **X1** and **Y1** coordinates are the primary keys for the database.

C3.3 Fuels and Substance Groups

Records in the **fuel&substance_group** database have one of two different formats depending on whether they are describe a fuel type or a substance group. For a fuel type the following information is stored:

NAME	The name of the fuel (up to 19 characters)
INDEX	Just an index. The primary key of the fuel.
ENERGYVALUE	The energy content of the fuel in MJ/kg.
SUBSTANCE	The substances contained in the fuel specified as an index to the substance database followed by a value and then either a % or g/MJ, depending upon how the contents of the substance are specified. Up to 254 different substances can be specified.

If the emission of a substance is expressed as a percentage then the total emission is calculated as:

$$\frac{MaxEffect}{EnergyValue} * emission$$

Otherwise if the emission is expressed in g/MJ the total emission is calculated as

Max Effect*emission

For a substance group the following information is stored:

NAME	The name of the substance group (up to 19 characters)
INDEX	Just an index. The primary key of the substance group.
ENERGYVALUE	Must be zero.
SUBSTANCES	The substances contained in the substance group specified as an index to the substance database followed by a value which is the percentage (weight) content of the substance in the substance group. Up to 254 different substances can be specified.

C3.4 Time and Temperature Variation Formulae

Each record of the formula database contains the following information:

NAME	The name of the formula (up to 19 characters)
INDEX	The primary key of the formula.
TYPEDAY	Description of how the emission of a source varies with time. This is specified as 4 columns, one for each day type (working days, Friday day types, Saturday day types and Sunday day types). Each column has 24 rows, one for each hour of the day. These figures are relative.
MONTH	Describes how emission changes with the 12 months of the year. These figures are relative.
SCENARIO	Describes how the emission, in percent, changes with scenario. These figures are absolute.
TEMP	This field describes how the emission changes with temperature. There are 30 values in an interval from -30 to -30 degrees Celsius. Each value represents a range of 2 degrees.
GASFLOW	This field describes, in percent, how the gas flow changes with emission. The first value specifies the gas velocity when the emission is 0-5% of max emission. There are 20 values that must be specified, so the last value specifies the gas velocity when the emission is 95-100%.

C3.5 Vehicles

Each record of the vehicle database contains the following information:

NAME	The name of the vehicle (up to 19 characters)
INDEX	The primary key of the vehicle.
SUBSTANCE	The substances emitted by the vehicle.
SPEED 1	The emission for each substance for 11 definable speeds.
SPEED 2	
:	
SPEED 11	
MAXEMISSION	Not used for the moment.

C3.6 Road Types

Each record of the **road_type** database contains the following information:

NAME The name of the road type

INDEX The primary key of the road type

VEHICLE A reference to a vehicle in the vehicle database. A road type can reference up to 10 vehicles. After each vehicle reference a TYPEDAY, MONTH, SCENARIO and MINSPEED must follow. For each vehicle associated to the road type the following information is stored:

TYPEDAY This describes how the emission from the road type changes with time. There are 5 columns for the different day types (working days, Friday day types, Saturday day types, Sunday day types). These figures are absolute when used with sources connected to a fuel definition, otherwise they are relative.

MONTH Describes how the emission varies with the months of the year. These figures are absolute when used with sources connected to a fuel definition, otherwise they are relative.

SCENARIO Describes how the emission varies with scenario. These figures are absolute.

MINSPEED The lowest speed of vehicles using this road type.

MAXSPEED The highest speed of vehicles using this road type.

C3.7 Road Sources

Each record of the road database contains the following information:

X1	Lower left corner x-coordinate for a rectangle surrounding the road.
Y1	Lower left corner y-coordinate for a rectangle surrounding the road.
X2	Upper right corner x-coordinate for a rectangle surrounding the road.

Y2	Upper right corner y-coordinate for a rectangle surrounding the road.
NAME	The name of the road (up to 47 characters).
INFO	An information string (up to 47 characters).
INFO2	A second information string (up to 63 characters).
VEHICLES	Number of vehicles per day.
CORRFACTOR	A correction factor for the emission. The calculated emission will be multiplied by this factor.
NOLANES	Number of lanes of the road.
SPEED	The average speed of the vehicles on the road.
ROADTYPE	A reference to the road type database. Must be non zero.
CONGESTIONLIMIT	Limit of traffic flow when the congestion speed should be used instead of the normal speed.
CONGESTIONSPEED	Congestion speed.
SEARCHKEY1	A reference to the searchkey1 database.
SEARCHKEY2	A reference to the searchkey2 database.
SEARCHKEY3	A reference to the searchkey3 database.
SEARCHKEY4	A reference to the searchkey4 database.
SEARCHKEY5	A reference to the searchkey5 database.
WIDTH	The width of the road.
DISTHOUSES	The distance to the surrounding houses.
SLOPE	The slope of the road in percent.
HEIGHT	Height of surrounding houses in 12 directions.
ALOB	A text string used to store application specific data.
X0 Y0 **X1 Y1** : **Xn Yn**	Maximum of 20 coordinate pairs giving the vector chain defining the road.

The X1, Y1, X2, Y2 coordinates are the primary key for the road database.

C3.8 Grid Layers

The **grid_layers** database is defined as follows:

A grid layer consists of two parts:

- a description part consisting of
  - Information about the position and geometry of the grid layer.
  - Specification of which data are common for the whole grid layer and which are not.
  - The common data.

The description part contains the following field:

X	Lower left x coordinate of the grid.
Y	Lower left y coordinate of the grid.
NX	The number of grid squares in x direction.
NY	The number of grid squares in y direction.

DX	The width of one grid square in metres.
DY	The height of one grid square in metres.

- a part containing grid cell information consisting of
    - location of the grid cell
    - The emission of the grid cell
    - The specific data for each grid cell.

The following information can then either be specified as common for the whole grid layer or as specific for each grid cell.

NAME	The name of the source. Maximum 47 characters
INFO	A string for information (up to 47 characters). Can be used for searching.
INFO2	A second string for information (up to 63 characters). Can be used for searching.
ADDRESS	The street address of the source (up to 47 characters).
POSTADDRESS	The post address of the source (up to 47 characters).
INFOGIVER	The name of the supplier of information (up to 47 characters).
DATE	Manually entered date showing when the information was obtained
CHANGED	A date that is automatically updated when the source is altered
MISC	Miscellaneous information (up to 47 characters).
FUEL	Specifies which substance group to use. This should be set to zero if substances are to be used. Note that references to the fuel database may not be used.
FORMULA	A reference to the formula database. Has to be non zero.
SEARCHKEY1	A reference to the first searchkey database.
SEARCHKEY2	A reference to the second searchkey database.
SEARCHKEY3	A reference to the third searchkey database.
SEARCHKEY4	A reference to the fourth searchkey database.
SEARCHKEY5	A reference to the fifth searchkey database.
SUBSTANCE	One field for each substance emitted from the grid cells. All grid cells must emit the same substances. These fields should be left out if substance groups are used.

The grid cell information has the following syntax:

```
<xpos>,<ypos>[,<data1>,...,<datan>],<sgemi> |
[<subst>,<subst>,...]
```

<xpos>	is the (lower left) x-coordinate for the sub area.
<ypos>	is the (lower left) y-coordinate for the sub area.
<data1>	is the first of the grid specific fields defined in t he grid description file.
<datan>	is the nth of the grid specific fields defined in the grid description file.
<sgemi>	is the emission for a substance group. The value is a float and the unit is tons/year. Either this or the <subst> field must be specified.

<subst> is a floating point value containing the emission value for its corresponding substance in the grid description file, the unit is tons/year. If the grid uses substance groups this field should be left out.

C3.9 The Temperature Database

This database can be used during a search in the EDB by using the option Temperature dependent on time. The EDB must then find out the temperature frequencies that are typical for the chosen time restrictions. This is done by accessing the EDB temperature database. The database is created with several years of temperature data as input, which is taken from the meteorological mast data in the time series database. The temperatures are assumed to be normally distributed. Thirteen main groups of data are stored: the first group is for the whole year and then there is one for each month. For each group 24 mean values, one for each hour, and their low and high standard deviations are stored. These values are calculated using a moving average of size three.

C4 The Topography and Physiography Database

The topographic and physiographic information in Airviro are stored in an ASCII-format file. The file contains a header followed by several sections of data, each of which has a heading starting with a #. Lines starting with a ! are comment lines and are ignored.

Each data section represents a layer of information in a delimited area. This area is divided into a grid with a defined resolution. The first value in each data section refers to the lower left corner of the grid.

C4.1 The Topography and Physiography File

C4.1.1 The Header

The header gives the size of the grid. The resolution of the grid is stored in the definition file for the maps.

nx ny Size of grid (number of grid boxes nx*ny)

C4.1.2 The Topography Section

The topography section starts on a new line with the header #topo. The data begins on the next line and describes the average height in metres above sea level for each grid box.

C4.1.3 The z0 Section

The z0 section starts on a new line with the header #z0. The data starts on the next line and describes the surface roughness for each grid box. These values give the height above the ground where the wind ceases. See *Appendix D3.3* for a definition of z0.

C4.1.4 The Physiography Sections

The physiography is split up into different classes and there is one section for each class. Each class type is assigned a number n and each section begins on a new line with the heading #physio n. The data begins on the next line and describes the fraction of land use in percent for each grid box. Currently the following class types are used:

0	Water
1	Urban area
2	Open area
3	Forest

C4.1.5 The Building Height Section

The building height section starts on a new line with the header #househgt. The data starts on the next line and gives the average height of the buildings in the grid box.

C4.1.6 The Influence Section

The influence type section starts on a new line with the header #influ. The data begins on the next line and gives the influence class for each grid box. The influence classes are stored in a separate resource file and within each class the meteorological conditions are assumed to be similar.See appendix D3.3 for an explanation of how the influence affects the simulations.

C4.1.7 The Heat Island Section

The heat island section starts on a new line with the header #heat. The data begins on the next line and gives the value 1 if the heat island effect is present in the grid box and 0 otherwise. See *Appendix D3.3* for an explanation of how the heat island effect is used.

C4.2 Example

```
!Grid breakdown of Mumindale
8  4
#topo
!height of each gridbox above sea level
10 11 12 11 13 11 13 15
11 12 14 15 20 23 25 29
11 13 18 17 16 18 17 21
12 12 13 19 18 17 20 23
#z0
!height above ground where wind stops
0.35 0.40 0.41 0.36 0.48 0.40 0.20 0.30
0.36 0.37 0.39 0.40 0.43 0.29 0.30 0.32
0.30 0.31 0.32 0.33 0.43 0.40 0.39 0.41
0.29 0.31 0.34 0.35 0.34 0.34 0.23 0.34
#physio 0
!percentage water
10 00 10 10 20 20 30 30
10 00 00 10 10 10 00 30
20 20 50 40 10 20 30 30
30 30 10 10 25 25 15 30
#physio 1
!percentage urban area
10 20 10 10 20 20 30 30
10 30 30 20 20 30 20 30
```

```
10 20 00 10 20 10 30 30
20 40 00 05 25 25 15 20
#physio 2
!percentage open area
20 20 30 30 20 20 30 30
20 30 30 30 30 20 30 30
30 20 20 00 70 60 30 30
10 30 10 05 25 30 30 10
#physio 3
!percentage forest
60 60 50 50 40 40 10 10
60 40 40 40 40 40 50 10
40 40 30 50 00 10 10 10
40 00 80 80 25 20 40 40
#househgt
!average height of buildings in each gridbox
15 15 15 15 15 15 20 20
15 20 22 15 15 20 20 20
15 15 00 15 15 15 28 28
28 28 00 15 20 20 15 15
#influ
!influence area
1 1 1 1 1 1 2 2
1 1 1 1 1 2 2 2
1 1 1 1 2 2 2 2
1 1 1 2 2 2 2 2
#heat
!heat island effect
0 0 0 0 0 0 1 1
0 1 1 0 0 1 0 1
0 0 0 0 0 0 1 1
0 1 0 0 1 1 0 0
```

Note that it is not necessary to structure the contents of each subsection so that the grid size corresponds to the number of rows and columns. All that is important is that each section starts on a new line with the section header on a separate line.

C5 System Maps

Maps used in Airviro are either based on shape files or on GIF pictures or a combination of the two. This gives a possibility to use high resolution vector maps within Airviro.