

## Using the Indico Administration Module

How to fetch data automatically from remote stations



# Contents

<b>7.1</b>	<b>Introduction.....</b>	<b>5</b>
<b>7.2</b>	<b>Getting Started.....</b>	<b>6</b>
<b>7.3</b>	<b>Viewing and Editing the Parameter Database.....</b>	<b>8</b>
	7.3.1 Creating and Editing Parameters.....	9
<b>7.4</b>	<b>Understanding the Station Database and Setting Up a Station.....</b>	<b>11</b>
	7.4.1 Viewing the Station Database.....	11
	7.4.2 Creating or Editing a Station.....	11
	7.4.3 Setting Up the External Protocol Information.....	16
<b>7.5</b>	<b>Examining the Data Collection Processes.....</b>	<b>21</b>
<b>7.6</b>	<b>Examining the Database Management Processes.....</b>	<b>23</b>
<b>7.7</b>	<b>Editing the Calibration of the analysers.....</b>	<b>24</b>



## 7.1 Introduction

Automatic data collection is impressive once it is working, but setting up a system to handle it is non-trivial to achieve. The Indico Administration Client has been built to help with this so that the whole process is user friendly, but nevertheless a good understanding of how each part of the data collection works is important if you want to be able to sort out problems.

Data is usually fetched using modems. The Airviro computer first runs an **external protocol**, which contacts a specified remote **station**. The station answers and waits for instructions. The external protocol then sends instructions to the remote station including a request for the latest data. The station responds to the instructions, the computer receives the new data and the conversation is complete. The new data is then loaded into the **time series database** so that other programs (such as the Indico Presentation Client) can access it.

Automatic data collection is carried out via external protocol programs. Every external protocol is different, because every type of logger is different. This means that the documentation cannot really become specific enough to be able to lead you by the hand through the external protocol editor where all the external protocol information is given.

Indico Administration Client is one of the three main tools in Airviro ( Indico Presentation, and Indico Report and Indico Validation), intended for acquiring, storing, editing, presenting, analyzing, reporting and exporting time series data.

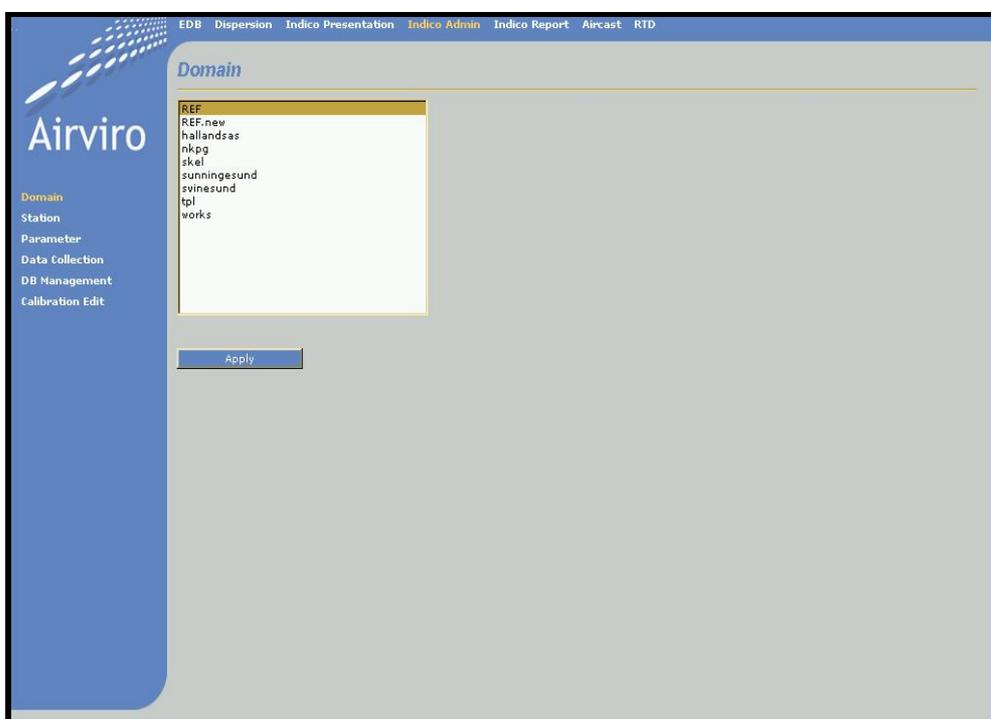
## 7.2 Getting Started

The Indico Administration Client has been designed to make it easier to control and monitor data collection activities.

When Airviro has been installed at the Airviro Server, it is possible to jump to your Airviro URL with a browser over Internet/Intranet.

After logging in with user-ID and password, the user is presented with a list of available **Web Modules** and **Domains**. See Figure.

Clicking on the Indico Administration and the frame menu will appear. When they have been selected, the user selects a **Domain**.



The main window is split up into different sections:

- DOMAIN** This allows to select the Project with which is wanted to work. i.e. REF.
- STATION** This provides an interface to the station database, where you first set up new stations from which to collect data. Create new station groups, which can then be assigned to stations. Start the External Protocol Editor, used to configure automatic data collection.
- PARAMETER** This provides an interface to the parameter database. Existing parameters can be altered and new parameters can be created.

**DATA COLLECTION** This enables you to look at a summary of all data collection transactions and alter the state of cold.

**DB MANAGEMENT** This enables you to alter the Airviro database manager.

**CALIBRATION EDIT** Enter zero and span voltage values obtained from the actual analysers.

## 7.3 Viewing and Editing the Parameter Database

The parameter database contains information about the different substances that are measured. This includes gaseous pollutants such as sulphur dioxide, carbon monoxide, etc. and also meteorological parameters such as temperature and wind speed. A parameter can be anything that is measured as a variation against time.

In the **Parameters** on the frames, select **List** and the following sub-windows will appear.

#	Key	Descr	Unit	Alim	Gmin	Gmax	Imin	Imax	Maxeq	Eps	Grad
0001	SO2		ôg/m3	200	0	50	-6e+00	1e+03	0	0.0e+00	-1e+00
0002	NO		ôg/m3	1001	99	100	-5e+00	2e+03	12	1.1e+01	-9e+00
0003	NO2		ôg/m3	120	0	400	-5e+00	2e+03	0	0.0e+00	-1e+00
0004	CO		ppm	15	0	10	-1e+00	1e+02	0	0.0e+00	-1e+00
0008	O3		ppb	200	0	150	-3e+00	5e+02	0	0.0e+00	-1e+00
0021	Benzene		ôg/m3	100	0	150	0e+00	1e+04	6	0.0e+00	-1e+00
0022	Toluene		ôg/m3	5000	3200	-200	0e+00	2e+00	20	3.0e-01	-1e+11
0023	P-Xylene		ôg/m3	100	0	100	0e+00	1e+04	6	0.0e+00	-1e+00
0027	Formaldehy		ôg/mds	100	110	500	9e+28	9e+31	6	-3.4e+38	-1e+07
002y	Toluene b		ôg/m3	5000	3200	-200	0e+00	2e+00	20	3.0e-01	-1e+11
00HC	HC		ôg/m3	1000	0	1000	0e+00	1e+04	6	0.0e+00	-1e+00
0DIS	DIS		ôg/m3	100	0	180	0e+00	1e+07	0	0.0e+00	-1e+00
0LEV	Level 2		mm	5000	1500	2000	0e+00	1e+04	0	0.0e+00	-1e+00
0LMO	LMO		m	0	10	10	-1e+05	1e+05	60	0.0e+00	0e+00
0NOX	NOX		ppb	1500	0	1000	-5e+00	2e+03	0	0.0e+00	-1e+00
PRES	Pressure		mbar	1200	900	1100	8e+02	1e+03	0	0.0e+00	-1e+00
STWS	Std wspee		m/s	100	0	10	0e+00	2e+01	24	0.0e+00	-1e+00
WDIR	Wind dir		Deg.M	0	0	360	0e+00	4e+02	24	0.0e+00	-1e+00
WSPD	Wind spee		m/s	60	0	15	0e+00	4e+01	24	0.0e+00	-1e+00
WSTR	Vert vel			0	-10	10	-5e+01	1e+08	0	0.0e+00	-1e+00

In the first column you see the parameter key - a unique 4-letter identifier refers to each parameter. In the second column is a description of the parameter; so that you know which parameter the key refers to. The fourth column could in the future provide the possibility to have alarm limits, but this hasn't been implemented (concentration level alarms can be implemented using the external protocol).

The third, fifth and sixth column are for use with the Indico presentation module, which displays the units of the graph parameters underneath the graphs, and also needs a default minimum and maximum for each parameter that it uses so that correct scales are created on the graphs.

The last five columns are used to check against any incoming data associated with a particular parameter, unless the data has already failed a check by an external protocol. **Imin** and **Imax** are the minimum and maximum acceptable values, and **MaxEq** is the maximum number of

equal consecutive values that will be accepted, plus or minus the value **Eps. Grad** is the maximum allowable difference between two values of data differing by an hour. If data fails one of these checks then it is assigned a status code indicating where it failed.

### 7.3.1 Creating and Editing Parameters

Click on **Edit** under **Parameters** on the frame to be able to view and edit individual parameters, or to be able to create new ones.

The screenshot shows the 'Parameters' administration interface. On the left, a list of parameters is displayed, with 'Std\_wspee' highlighted. The main area is divided into three sections:

- Main Information:** Contains fields for 'Name' (Std\_wspee), 'Key' (STWS), and 'Info'.
- Presentation Information:** Contains fields for 'Unit' (m/s), 'Graph Min.' (0), 'Graph Max.' (10), and 'Alarm limit' (100).
- Data checks:** Contains fields for 'Max number of equal values' (24), 'Allowed Variation' (0), 'Absolut Min' (0), 'Absolut Max' (20), and 'Max gradient (per time unit)' (-1).

At the bottom left, there are 'Add' and 'Delete' buttons. At the bottom center, there is an 'Apply' button.

In the left side you will see a list of all the parameters currently defined. To view all the details stored about a given parameter, just click on the parameter name. You will see all the information that was displayed in the parameter database summary, this time much more nicely presented. Alternatively click on the **Add** button in the bottom below left corner to want to create a new parameter.

The most important piece of information for each parameter is its parameter key, and once a parameter has been assigned a key you cannot alter the key. The parameter also needs a name so that you can recognise it in the parameter list. Fill in these two pieces of information next to **Name** and **Key**.

Next fill in the **Presentation information**. The **Unit**, **Graph min.** and **Graph max.** are required by the Indico Presentation Client when it draws a graph involving the parameters. It is

always important for a person looking at a graph to know what units it represents. For most gaseous pollutants this will be  $\mu\text{g}/\text{m}^3$ . Fill in the likely minimum and maximum value that would be required for the graph axis. This can be altered in the Indico Presentation Client but a default value is required when the graph is first produced. The **Alarm limit** is currently not implemented so it doesn't matter what number you enter there.

### 7.3.1.1 Validating the Incoming Data

Now move onto the **Data checks** section, which defines various limits to check the incoming data. All incoming data is checked in some way - there are some checks that are carried out by the external protocol but most data is checked against these limits here. The data is then assigned a status code, which either indicates the first check that the data failed or that the data is OK. The data is still stored in the time series database whether or not it is OK (Indico Validation), but users using the Indico Presentation Client can choose to only use data with certain status codes.

A consecutive number of equal values often indicates instrument failure, so set **Max number of equal values** to what you consider would indicate such a failure. The **Allowed variation** is related to this - if the difference between two adjacent values is less than the allowed variation then they are considered to be equal. Instruments also usually have a measurement range outside of which their readings cannot be relied upon. Enter these limits next to **Absolute min** and **Absolute max**. **Max gradient** is the maximum allowable difference between two hourly values of data.

When you have finished defining the parameter click on **Apply** to save your data.

## 7.4 Understanding the Station Database and Setting Up a Station

### 7.4.1 Viewing the Station Database

Each site where data is measured is called a station. If you click on **List** under **Station** on the frame and a summary of the station database will be presented on the screen:

Station	Op	Bad	Update	Next
GK3 Kungsgatan	NO	50s	031008 15:00	031104 14:46
GM1 Shell	NO	0	900122 12:00	700101 01:00
GM2 Lejonet2	NO	0	910823 12:00	910823 14:35
GM3 Jarnbrott	NO	99s	910116 15:00	910701 14:10
GM4 Risholmen	NO	0	910116 15:00	910701 14:10
GM5 Femman	NO	0	900101 01:00	700101 01:00
GO1 EMEST1	NO	14s	900122 12:01	010803 12:24
GO2 Molndal	NO	0	900122 12:00	700101 01:00
GO3 Rya	NO	0	900122 12:00	700101 01:00
GO5 Jarntorget	NO	0	900122 12:00	930407 01:00
GOX EMEST1	NO	14s	900122 12:01	010803 12:24
GOY EMEST2	NO	0	900122 12:00	700101 01:00

In the first column you see the station key. Each station is referred to by a unique 3-letter identifier, which is the internal name of the station. In the second column is a description of the station. This is a name associated with the station key so that you know which station the key refers to.

The third column shows whether the station is in operation or not, i.e. if it is due to be called automatically at some specific time. The next column shows any bad (unsuccessful) calls and can also show whether any alarm limits have been reached. The last two columns show the date and time of the last successfully fetched data, and the next call time (this is only relevant for operational stations).

### 7.4.2 Creating or Editing a Station

To see more information about a particular station click on **Edit** under **Station** on the frame. A hair cross appears that waits for you to choose the area on the map that you are interested in. Select an area with the mouse by pressing and holding the left mouse button in one corner, dragging it over the area and then releasing it. Alternatively, if you are not interested in a particular area and would like to see all stations, click on **Show all**.

Then double click on the station, i.e. Molndal, and **Station** sub-window appears. The user can also use the toggle button **Edit mode** to open these sub-windows.

In the left side of the sub-window is a list of existing stations. If you want to edit an existing station, just click on the station name in the list and the station details will appear. If you want to create a new station, then click on the **Add** button in the bottom below left hand corner. The station sub-window disappears and a hair cross appears, waiting for you to define the location of the new station. Click on the map at the site of the station. The stations sub-window reappears.

- **Name Information**

Start by filling in the name of the station. Click in the box beside **Name** and enter a name that will be recognised by you and other users as identifying the station. Next fill in the **Key**. This is the system name for the station and must be a unique 3-letter identifier, and conventionally follows a naming pattern. The first letter represents the location of the station, for example all stations in Göteborg have G as the first letter of the key. The second letter usually describes the type of data measured at the station. The most important types are O for Opsis analysers and M for Campbell loggers. The last letter distinguishes between different stations of the same type. You do not have to follow this convention - as long as the key is unique, the station database will be happy.

- **Main information**

Next comes **Main information**, but this is actually the last thing to fill in. Once the station is fully configured for automatic data collection, fill in the date and time for the next data

collection beside **Time to call** and activate the toggle button beside **Operate**. Once data collection is running the **Latest update** field automatically updates itself.

This **argument** is automatically update by the data collection daemon after a successful data fetch. Each protocol requires a specific argument that specifies the start date and time of the data to be collected. This has to be in the correct format for the station that is contacted to understand what is required.

If you are using the “ops” protocol then the argument is the date and time e.g. 9406281500. If the OPSIS station has several paths then there must be an argument for each path. The arguments should be separated by a space.

If you are using the “cam” protocol then you must give the pointer position e.g. 1047. If you give a pointer position 0 then all data in the logger will be fetched.

Each protocol requires an argument describing the start time of the data required. The argument is required by the external protocol program. This takes different formats for different protocols - for example, Opsis loggers take a normal date and time format but Campbell loggers require a pointer which does not seem to have anything to do with time at all. Enter a suitable argument beside **Argument**. This argument is automatically updated after data has successfully been collected from the station. A resource is also normally required as a means of contacting the station, and this is usually a modem.

- **Limits**

Move on to **Limits**. You can define your own alarm limit and stop limit, which typically take values of 5 and 10 respectively. When the number of consecutive bad calls reaches the alarm limit, a warning alarm message can be sent, and when the stop limit is reached another alarm message can be sent and the station ceases to be operational. When this happens you must first find out and solve the problem, then reset the number beside **Bad calls** to 0 and make the station operational again.

---

**Note:** It is usually much easier to copy an existing station when creating a new station. First select the station that you want to copy and then click on **Add**. All the information will be copied to the new station.

---

- **Station group**

Finally you can choose a **Station group** to associate your station with. These groups can be created in **Station groups** in the **Station** menu on the frame. A station can then be assigned to one or several station groups. When you look at time series data in Indico Presentation Client you can choose to only look at stations for a particular station group. This is useful if you have a system with a large number of stations.

Station groups can also be used for dealing with mobile stations. Every time a mobile station moves it becomes a new station with a new station key. However the mobile station can have its own station group so that it is easy to use Indico Presentation Client to select all stations that are actually the same, mobile station.

- **Configuration**

### 7.4.2.1 Restricting the Call Times

Click on **Time restrictions** and the Time restrictions sub-window will appear. Here you can restrict the times and the days when the station is to be called. Under Hours click on the hours that the station may be contacted (note that you can click on All for a short-cut). The hours that will be used are those that are shown with highlight buttons. Do the same for Weekdays.

Next fill in **Miscellaneous information**. First comes the **Call interval** (in minutes). This describes how long to wait in between each call to the station. Of course this depends on the restrictions that you have made under **Hours** and **Week days** - if you have restricted calls to be only between the hours of 6 and 7 then the station will only be contacted between the hours of 6 and 7, even if the call interval is 180 minutes. Try to call fairly frequently to avoid loss of data - it is best to call at least once a day to collect hourly data, and if you want to have fairly recent data during the day you will need to call 3 or 4 times a day, or even every hour.

The **Retry interval** is the time to wait (in minutes) after a bad call before retrying. The optimum time to wait varies between systems, but 3 minutes is quite popular.

**Start minute** and **Stop minute** restrict the parts of the hour to use when calling a station. For example, if **Start minute** is 10 and **Stop minute** is 30, then calls to the station may only be made between 10 past and half past the hour. This can be very useful in helping to avoid more than one process trying to use the modem at the same time.

**Max factor** is the maximum number of seconds that is allowed to fetch **one** hour's worth of data. For each call the maximum allowable call time is calculated, which depends on the number of hours of data to be collected, plus a default connection time that is added because it sometimes takes a while to establish contact with a station. If a call exceeds its maximum call time then the process is terminated by the collection daemon. A value of 0 here means no restrictions.

Click on **OK** to save you choices and to return to the **Stations** sub-window.

Hour	Week day	Misc. information
00 - 01	Sunday	Call Int. 240 (min)
01 - 02	Monday	Retry Int. 0 (min)
02 - 03	Tuesday	Start min. 0
03 - 04	Wednesday	Stop min. 59
04 - 05	Thursday	Max Factor. 0 min/h
05 - 06	Friday	
06 - 07	Saturday	
07 - 08		
08 - 09		
09 - 10		
10 - 11		
11 - 12		
12 - 13		
13 - 14		
14 - 15		
15 - 16		

Buttons: All None All None Ok Cancel

### 7.4.2.2 Defining the Protocol to Use

Here you define which data collection protocol to use, along with a few extras required by the protocol. However most of the information for the external protocol is entered with the **Argument and Resources** (page13 and page 15).

The station is however not yet ready to be automatically contacted to fetch data. The external protocol needs a lot of information about the setup of the logger that it will be contacting, and you now have to start the external protocol editor, which lies under **Protocol Settings** in the **Stations** windows (page16).

### 7.4.2.3 Miscellaneous information

Click on **Miscellaneous** and sub-window will appear.

In **Geographic info** you can move the station and define its height above the ground.

The **Time Information** is birth and death time. The birth time of a station is automatically stored here when a new station is created. If a station ceases permanently to be operational then you may want to call it “dead” and you enter the death time. In the future this might be more relevant and most be used with mobile stations.

Under **Protocol, Resources** can be the name of the resource (device file) such as `/dev/cuL00` or to be able to specific from a set of modems it can be specified by:

<b>type (T)</b>	Indicates the type of resource (ex TModem, TDirect).
<b>speed (S)</b>	Specifies which speed the resource supports (ex S2400 or S1200).
<b>property (P)</b>	Specifies a certain type of modem (ex PHST, PDiscovery)

Most external protocols require a resource to use (such as a modem). For systems that have several modems with different capabilities, it is possible to just specify a resource type to use, so that any available resource matching that type can be used.

Alternatively, you may instead just specify a specific resource, but it is better to give the resource type, as this is unaffected if the actual resource changes name or if more resources become available when the system is expanded.

If you have got 3 modems, two of which work at 2400 bauds and one at 1200, then you would prefer to use a 2400 modem, and beside “resources” you type

Tmodem, B2400 Tmodem, B1200

This means “use type modem at 2400 bauds, but if none are available use type modem with 1200 bauds”.

Under **Statistics** a summary of good and bad calls to the station is presented.

Now fill in the **Attributes**. Here you can specify three additional features connected with the data that is collected. Choose **Logging** if you want to keep a log of all transactions with the

station (you usually want this with live stations). Choose **Dummy station** if the station is not actually operational but data appears at regular intervals anyway (for example if it is distributed from another computer system). Setting the **Dummy station** flag means that you can set the **Operate** flag under **Main information** but it will be ignored by the data collection daemon.

Choose **Save raw data** if you want to save the data as a file in the format that it arrives in as well in standard format in the time series database. This is sometimes needed to sort out problems with data collection but in normal cases is not necessary.

Click on **OK** to save your choices.

**Stations >> Miscellaneous**

**Geographic info.**

X Coord.  Y Coord

Height(m)

**Time information**

Birth time

Death time

**Protocol**

Resources

**Statistics**

Successful collections

Extern protocol failures

Collection daemon failures

**Attributes**

Logging  Dummy Station  Save Raw Data

### 7.4.3 Setting Up the External Protocol Information

This is the most complicated part of setting up an operational station and is intended for use only by users who have had system administration training. It requires detailed information about how the logger has been configured and which parameter is measured on each channel.

Choose the correct protocol from the protocol list that appears.

Now choose a station, a list of all stations that have already been configured for the protocol will appear, and you can choose one to **Edit Mode**. This gives you a list of all the stations that have been set up in the station database, which will use the protocol that you have currently chosen.

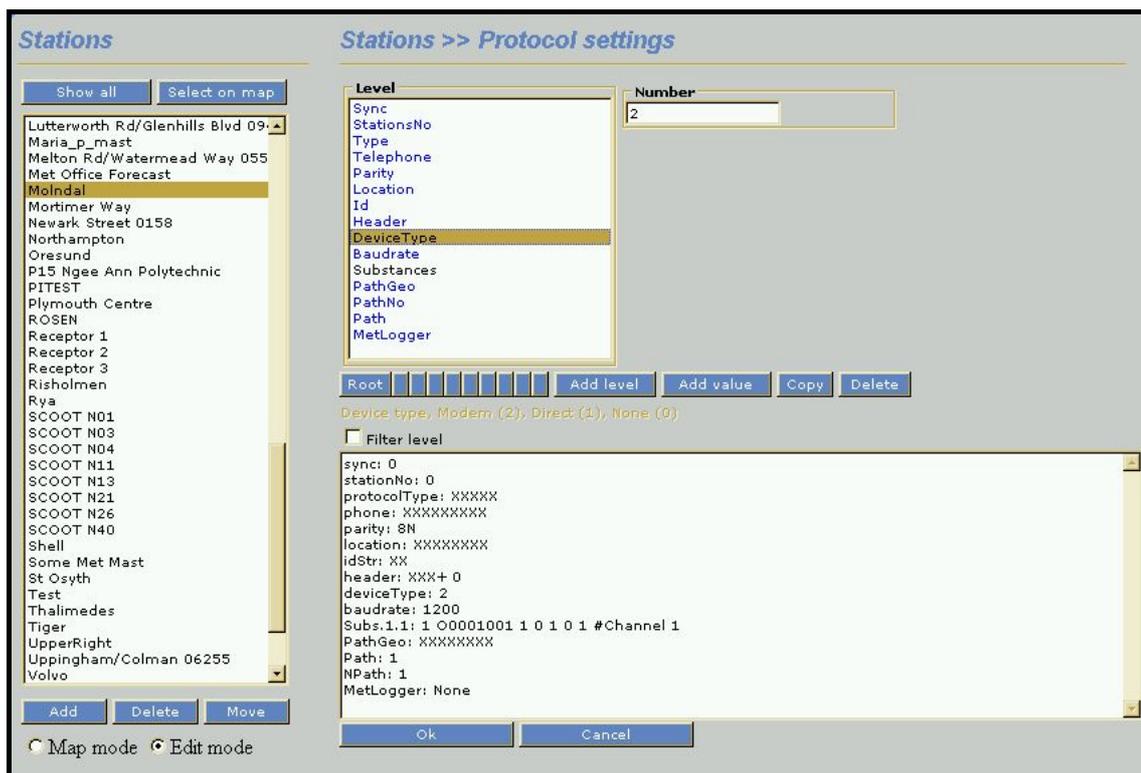
Once you have loaded a station your external protocol editor should look something like the one shown. Level sub-window is a list of properties. Each property specifies some specific information about the logger such as the way the channels are configured or the way in which it can be contacted.

Each property (Level) must therefore be given the correct value or values (Number). A property will enable you to edit it.

With the button **Root** you have the possibility to return to the root level. There you will be able to add properties and to enter values to each one of them.

You can clone (copy) these complex properties until you have the correct number of them.

Activating the filter level the data-header of the list disappears.



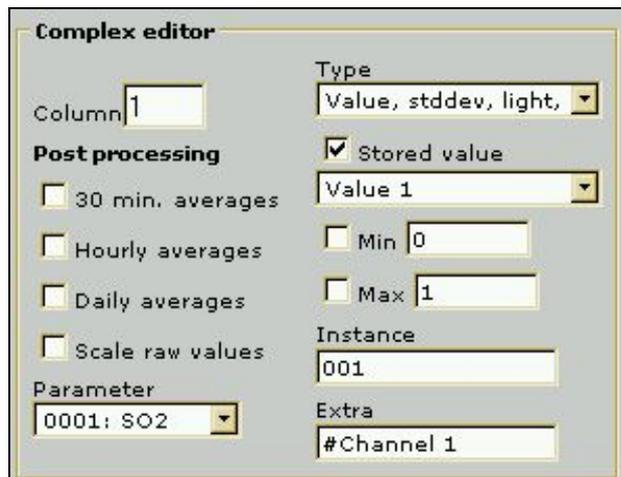
**Note:** you can request to the System Administrator to configure the “Import” protocol that is used it to extract existent data from a station.

### 7.4.3.1 The Complex Editor

This complex editor is the same regardless of the type of protocol you are using. However some parts of it might be irrelevant for certain protocols, so they will appear in grey, which means that you cannot edit them.

Each parameter can have more than one attribute. For example, every single piece of data has a status value associated with it, and this is called the status attribute. The data value must always be stored as Value 1 attribute. The maximum value or standard deviation must always be stored as the Value2 Attribute. The Light level must always be stored as the Value3 Attribute. The

status value must always be stored as the status attribute. If you do not want to spool the data to the value database then choose Ignore for the attribute.



The image shows a 'Complex editor' dialog box with the following fields and options:

- Column: 1
- Type: Value, stddev, light
- Post processing:
  - 30 min. averages
  - Hourly averages
  - Daily averages
  - Scale raw values
- Parameter: 0001: SO2
- Stored value:
- Value 1: dropdown menu
- Min: 0
- Max: 1
- Instance: 001
- Extra: #Channel 1

You do not always need all these attributes – it depends on the logger that you are fetching the data from. You define which attributes you will use next to “type”. If “type” is “opsis” then there must be a value1, value2, value3 attribute. Each attribute corresponds to separate column of data so must be defined as a separate substance.

You can store the value of each parameter (stored value) and also the options of the postprocessor (30min average, hourly average, daily average, scale raw value). These toggle button to be able to set up automatic, i.e., calculation of hourly and daily average. These choices are then converted a hexadecimal code.

The instance is a three letters code used to differentiate between cases when the same parameter is measured more than once at the same station. In normal cases the instance is usually the height of the analyser above the ground.

When you have finished supplying the data to the external protocol, save the data by choosing **Ok** under the **Protocol settings** sub-windows.

#### 7.4.3.2 An Example of an Entry for the Campbell External Protocol Configuration File

```

Substance.1.01      1 MTEMP002 1 0 1 4 100 #Temperature, 2m
Substance.1.02      1 MDTMP008 1 0 1 4 100 #DiffTemp, 2 and 8m
Substance.1.03      1 MWSPD010 1 0 1 4 100 #WindSpeed, 10m
Substance.1.04      1 MWDIR010 1 0 1 4 100 #WindDirection, 10m
Substance.1.05      1 MSTWD010 1 0 1 4 100 #Std.dev, wind dir, 10m
Substance.1.06      1 MSTWS010 1 0 1 4 100 #Std.dev, wind speed, 10m
Substance.1.07      1 MSVWS010 1 0 1 4 100 #Std.dev, vertical, 10m
Substance.1.08      1 MGLOB002 1 0 1 4 100 #Global radiation, 2m
Substance.1.09      1 MTIDF002 1 0 1 4 100 #Humidity, 2m
Substance.1.10      1 MRAIN002 1 0 1 4 100 #Rain, 2m
Substance.1.11      1 MERRBATT 1 0 2 0 0.0 #Battery check
Substance.1.12      1 MERRSIGN 1 0 2 0 0.0 #Signature check
Substance.1.13      1 MERRFANS 1 0 2 0 0.0 #Fan check
Substance.1.14      1 MERRGIVA 1 0 2 0 0.0 #Transducer check
Baudrate            1200

```

```

Device type      Modem
Distribution key  0
Id (text)        Kungsängen
Parity           8N
Telephone number 125367
Protocol type    Binary
NM0.rowType.1   014
Station number   0
Sync            0
Time resolution.1 3600
Modem timeout    30

```

### 7.4.3.3 An Example of an Entry for the Opsis External Protocol Configuration File

The following is an example of the most advanced type of configuration. The Opsis station has two beams and a meteorological logger linked to it.

```

Logger type      DL10
Number of paths  3
Path number.1    1
Path number.2    2
Path number.3    0
Path geometry.1  814 188 5
Path geometry.2  534 290 5
Substance.1.1    1 00001001 1 4 7 1 2 #SO2
Substance.1.2    1 00003001 1 4 1 0 0.0 #NO2
Substance.1.3    1 00008001 1 4 1 0 0.0 #O3
Substance.2.1    1 00001002 1 4 1 0 0.0 #SO2
Substance.2.2    1 00003002 1 4 1 0 0.0 #NO2
Substance.2.3    1 00008002 1 4 1 0 0.0 #O3
Substance.3.1    1 MWSPD006 1 4 1 1 0.0 #Wind speed
Substance.3.2    1 MDUMB001 1 0 0 4 100 #Dummy
Substance.3.3    1 MWDIR006 1 4 1 4 0.0 #Wind direction
Substance.3.4    1 MDUMB002 1 0 0 4 100 #Dummy
Substance.3.5    1 M0004003 1 4 1 4 0.0 #CO
Substance.3.6    1 MPM10004 1 4 1 4 0.0 #PM-10
Substance.3.7    1 MDUMB004 1 4 2 4 90.0 #Dummy
Baudrate         1200
Device type      Modem
Location         G. Rådstugugatan
Parity           N
Telephone no.    0102672369
Type            Ascii
Station number   0
Time sync        0
Time resolutio  3600
Modem timeout    30

```

### 7.4.3.4 Starting the Data Collection

It is now time to test the automatic data collection. Return to the Stations window, and select your station. Select **Edit Mode** and click on **Operate** so that the toggle button beside it shows as

selected. Fill in Time to call with a time 5 minutes from now. Fill in **Latest update time** as some time a few hours ago. It is not a good idea to try and collect too much data at once, especially with a new station, which might not work at first, as it is easy to forget to define something important. Save the station and wait for the call time. Check that the station appears in the data collection idle queue (click on **Data Collection.**). Watch the modem and make sure it calls at the correct time.

When it has finished the call, look in the data collection queue again to see if the call was successful (i.e. 0 bad calls). If this is the case then you have successfully set up automatic data collection, otherwise you have to check everything that you defined very carefully and try to work out what is wrong. Good luck! Examining the Data Collection Processes.

## 7.5 Examining the Data Collection Processes

Click on **Data Collection** on the frame. This shows the processes that are under the control of the data collection daemon (**cold**), together with their current status. First you see the state of **cold**, which in most ordinary cases should be **MULTIPLE**, i.e. it can deal with more than one data collection process at any one time. Next come the database(s) that currently have operational stations. In most cases there will be just one database although in some cases there can be more. The other important things here are three queues, called Idle, Wait and Running.

The screenshot shows the 'Data Collection' interface with the following sections:

**Collection Mode:** Multiple (dropdown), Shutdown, Clear selections, Freeze Window

**Idle Queue:**

Idle	Bad	Next	Resources	Nr	Database path
1:RO3 - ROSEN	0	031222 23:35	1	1:	/usr/airviro/data/nkpg/
1:GK3 - Kungsgatan	0	031223 05:30	1	2:	/usr/airviro/data/skel/
3:HA2 - Bjarred	0	031223 06:02	1	3:	/usr/airviro/data/hallandsas/
3:HA1 - Lya	0	031223 06:04	1	4:	/usr/airviro/data/svinesund/
3:HA4 - Norrlycke	0	031223 06:06	1		
3:HA3 - Severtorp	0	031223 06:08	1		
2:SM0 - Met_Tuvan	0	031223 07:05	1		
1:MM0 - Kungsangen/X	0	031223 07:35	1		

**Buttons:** Force, Force w. Debug, Reset, Stop

**Wait Queue:**

Wait	Timeout	Resources	Nr	Pri	Resource	Good	Bad	State
			1:	0	/dev/cua0	799	61	-

**Buttons:** Kill

**Running Queue:**

Running	Elapsed	Max	Resource	Inactive	Station	Op	Bad	Update	Next
			UV0	Sunningen	M0	0	000430	04:00	000510 12

**Buttons:** Kill, Start

The Idle queue contains all the stations that are operational. These stations are to be contacted at the time shown under Next. You can also see the number of bad calls and the resources that may be used.

The Wait queue is for stations that are due to be contacted but have to wait until a resource is available, for example if the modem that is to be used to contact the station is taken up by another process.

The Running queue shows any current activities - all stations from which data is currently being fetched. The length of time that the current collection has taken so far is also shown.

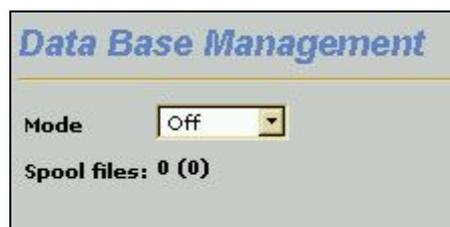
At the left is a list of all the resources that are available to be used by the data collection daemon. These resources keep a count of good and bad calls that they have made since the collection daemon was started.

This summary of the data collection processes only describes what was happening at all the moment.

## 7.6 Examining the Database Management Processes

Now click on **DB Management** on the frame. This brings up a small sub window where you can view or alter the state of the Airviro database manager. The statuses have following effect:

- Off: the database manager is shutdown
- Idle: the database manager is closed. There is no spooling of data
- Single: the database is open and incoming data is spooled
- Multi: the database management for the several stations



In the **Spool files** the user can identify the number of spool files and invalid files (0), in the spool directory of the current Domain.

---

**Note:** For a normal system running data collection the state of both of them should be **Multiple/Multi**. This means that the database manager can deal with more than one database, and the collection daemon can coordinate data collection from several stations at the same time.

---

## 7.7 Editing the Calibration of the analysers

Click on **Calibration Edit** on the frame and enter zero and span voltage values obtained from the actual analysers.

Chanel	Zero	Span
0 dummy	0	0
1 dummy	0	0
2 NOX	0	1000
3 NO	0	1000
4 NO2	0	1000
5 SO2	0	1000
6 Dust	0	1000
7 Filter	0	1000
8 dummy	0	1000
9 dummy	0	1000
10 Rack Temp	0	1000

**Note:** Calibrations sub-windows will appear if, zero and span voltage is settings in the Protocol the property (level).