# SmartView3

PEWA
Messtechnik GmbH

Weidenweg 21
58239 Schwerte

Tel.: 02304-96109-0
Fax: 02304-96109-88
E-Mail: info@pewa.de
Homepage : www.pewa .de

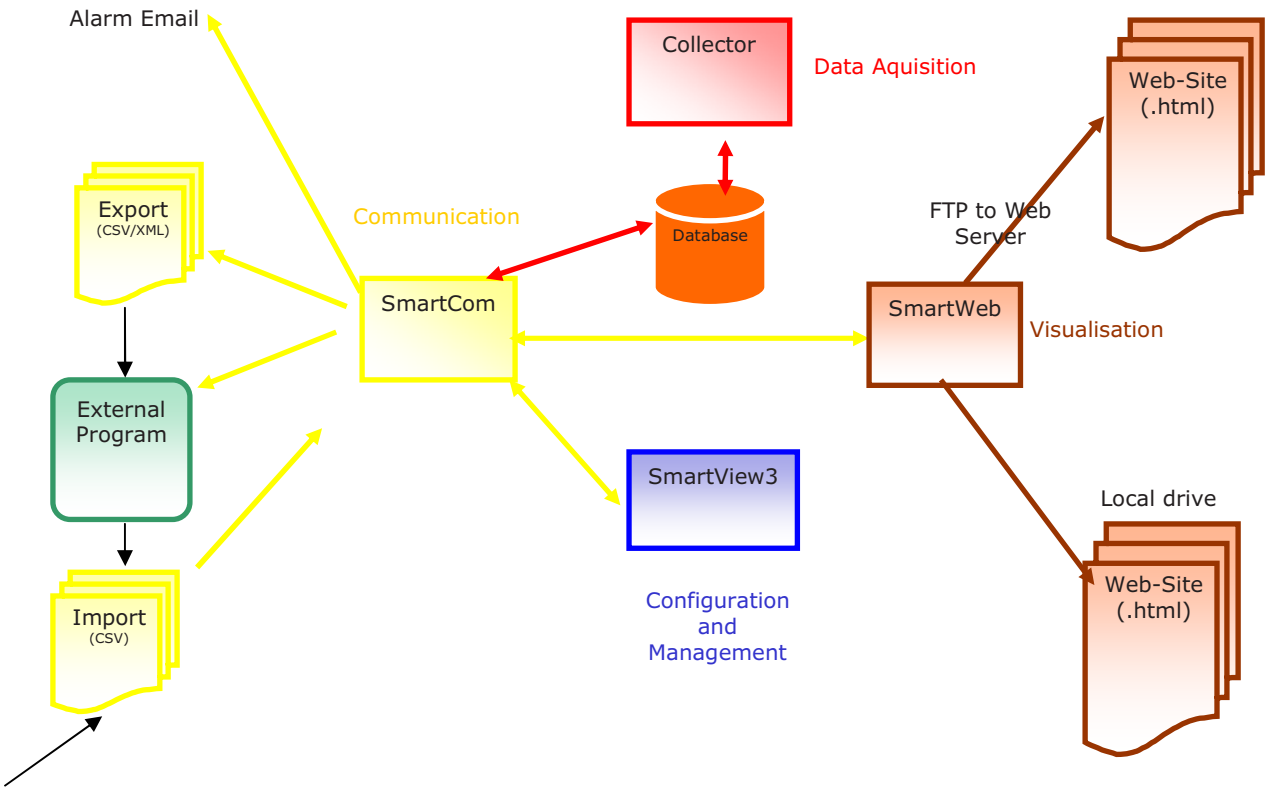
Version 1.4.8
Status: released
Date: August 2008

# Contents:

# 1 Overview

SmartView3 consists of the following independent programs/modules:

1.) SmartView **Collector**: the Collector is used to poll measurement data from the stations and store the data in a MySQL Database.
2.) **SmartCom** is the communication module. This program is used by SmartView and SmartWeb to access the data in the database. It also contains a flexible export/import module, and an alarm module.
3.) **SmartView3** is a PC Client program used to configure all SmartView3 modules.
4.) The **SmartWeb** module generates a web site to visualize the measurement data. Data is displayed in tables and diagrams.



# 2 Installation

**Requirements :**

- PC with Windows 2000/XP or equivalent
- Min. 512 MB Ram (1 GB RAM recommended)
- The required processing speed depends on number of stations/sensors to be polled, the number of (data) pages to be generated, and the poll/update interval.
- Minimum recommended (Collector only) is an Intel Pentium 4 (2 GHz) CPU or equivalent. For larger installations with many stations, short poll/update intervals and web visualization a fast Pentium 4 / HT / Dual Core / Extreme Edition processor may be needed.
- MySQL Version 4.1 or 5.0
- A fast LAN connection to the web server (diagrams, tables, and CSV files can take up quite some space and need to be transferred to the web server)
- Disk Space for database:
  - configuration data: approx 120 Kbytes plus 1 Kbytes per Station
  - per sensor value: approx 120 Byte
  **NOTE:** for database maintenance (optimization, upgrades etc.) additional space is needed. Please ensure there is at least double the disk space available!
- Disk Space for Web-Sites. If you are using SmartWeb to produce web sites with you measurement data, disk space to hold the web pages is required

**Note on Language Selection / Translation:**
All text elements used by SmartView3 are stored in an Unicode text file. With the program files we provide one file with all text elements in English (SmartView3_en.uni), and one text file with “basic” text elements (mainly the text elements used for the Web-Site, and some text elements used for sensor type names etc.) in German (SmartView3_de.uni). If you plan to use a different language, it is recommended to translate the text elements **before** installing SmartView3.
Please **do not modify** the text files provided with the program files, as they are updated with every new version of SmartView3. To create a new translation, please copy the SmartView3_en.uni file (for all text elements) or the SmartView3_de.uni file (for the basic text elements, i.e. the first part of SmartView3_en.uni) to SmartView3_xx.uni (where xx is replaced by an abbreviation for your language, e.g. “hu” for Hungarian, or “es” for Spanish). Then translate the text elements in the new copy. SmartView3 will prompt you to select the language to be used during setup (if there is a file other than SmartView3_en.uni available).
You can remove any entries that you don’t translate from the new text file (see comments in the provided text files). All entries not found in the new text file will always be taken from the “master” file SmartView3_en.uni.

- **Step 1:** Install MySQL (see www.mysql.com for product license information and download).
  **NOTE: MySQL setup must be run with administrator privileges!**
  Please reboot the system after installing MySQL to complete the installation. The MySQL system service needs to be running in order to set up the collector database.
  When installing MySQL, the installation program asks for the kind of database to be installed. Collector does not use transactions, so the “**Non-Transactional Database only**” (MyISAM) option is sufficient for Collector to operate.
  The MySQL setup will ask for the MySQL Admin Password - this needs to be entered in the “Setup Database” dialog (see below).
  **Note:** SmartView uses MyISAM tables without transaction support (for performance reasons). Please adjust the settings in the my.ini MySQL configuration file to your needs- especially the “datadir” parameter - to your needs (see the MySQL manual for more information about parameters for the MySQL database).
  **Note:** if you plan to store many measurement values, make sure that the MySQL data folder is placed on a NTFS partition, as FAT32 partitions only support files up to 4 GBytes.
  **NOTE on MySQL network access:** if you want to run MySQL on a different computer than Collector and SmartCom (which both access the MySQL database), the database (collector) (not the tables in the database) needs to be created manually on the MySQL server, and network access from all computers or from the computer(s) on which Collector and SmartCom are running needs to be allowed for “root” AND for the MySQL user that you configure on the

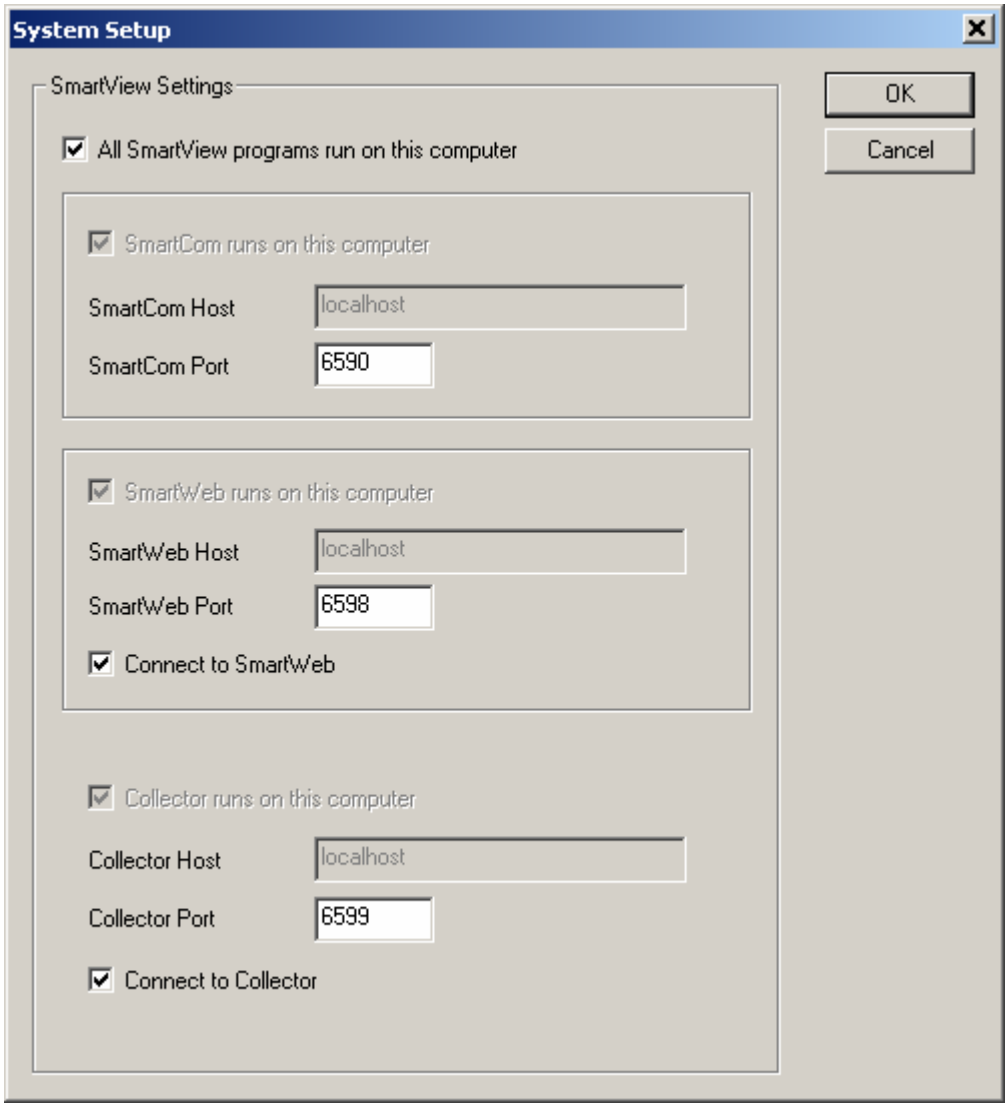
"database setup" dialog (see below). The default password for the MySQL user "collector" is "collector". MySQL does not allow setting the access privileges for the database via network, so the access privileges for the MySQL user (collector) to the database (collector) need to be set on the server manually and the "MySQL user" must be created on the server manually as well. **THIS MUST BE DONE BEFORE YOU PROCEED WITH THE INSTALLATION**!
The database can be set up on the "MySQL Server" by installing SmartView3 and running this setup on the server as well – if the server is Windows based. Then you only need to ensure that the "root" user has network access privileges. The database will then be created and the MySQL user (collector) will be granted network access by the setup procedure described below, and the software can be installed on the "client" computer.
If you run firewall software on the MySQL server computer, please make sure the MySQL port (default 3306) is allowed in the firewall.

- **Step 2:** Create a folder where you like to install the Collector program, e.g. "C:\Program Files\SmartView3". Copy all program files to this folder.
  **Note:** if you install SmartView on a Microsoft® Vista© system, it is NOT recommended to use a folder in "\Program Files" to install SmartView, because Vista will place all files written by program files to a folder below "Program Files" in the user's "VirtualStore" folder, which is a hidden folder. In this case the ini files for the SmartView program modules, and (if you do not specify a different path below) the log files will be placed in that hidden "VirtualStore" folder, and the ini and log files are hard to find. It is recommended to use a folder outside "Program Files" to install SmartView on a Microsoft Vista system, e.g. create a folder "c:\SmartView3" and copy the program files to this folder. You need administrator access rights, or the administrator password to create that folder.

**Step 3:** Start SmartView3.exe. SmartView3 will start SmartCom, and then prompt for installation/configuration information for the different modules of SmartView.
For most of the configuration options (except the MySQL Administrator password), the default values can be used (just click "OK" on the various dialog boxes that are displayed).

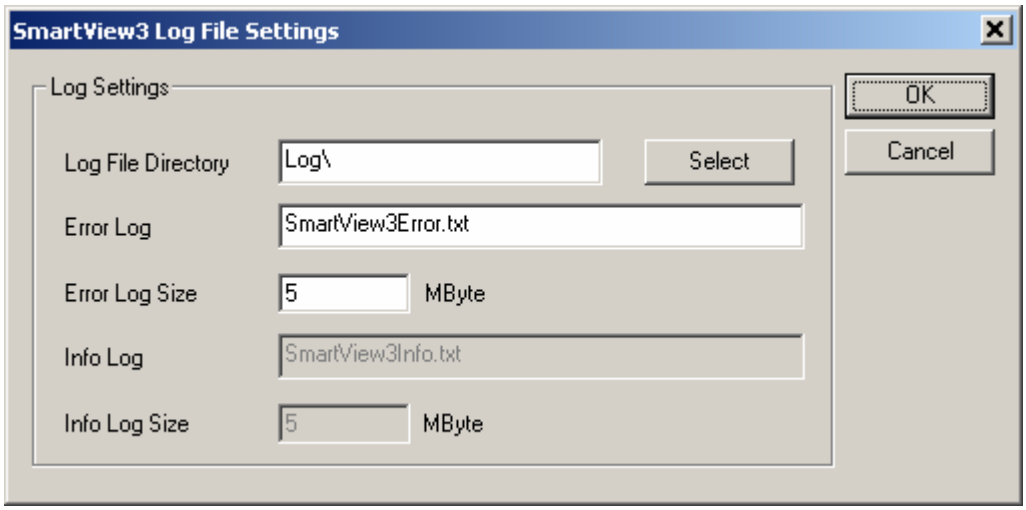
The first dialog box configures, which modules run on the same computer, and which TCP/IP ports are used by the modules to communicate :

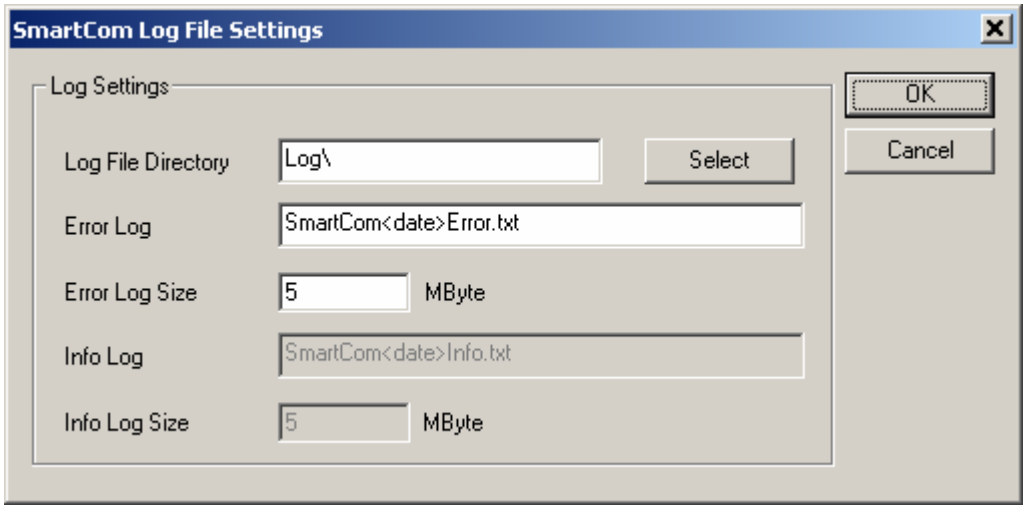
## 2.1 System Setup Dialog




**Note:** if you plan to run all programs only on the local machine, the default settings (host set to "localhost") are sufficient. If you plan to run SmartView3.exe from a different machine, and this way access SmartCom/Collector via Network (or run SmartCom/SmartWeb or Collector on a different computer), you need to uncheck the "All SmartView programs run on this computer" check-box. The host name is then changed to the local computer name instead of "localhost", and this way network access is enabled. If "All SmartView programs run on this computer" is checked, and host is set to "localhost", network access to the programs is not enabled !

Next, the log file settings for SmartView3 and SmartCom are configured:

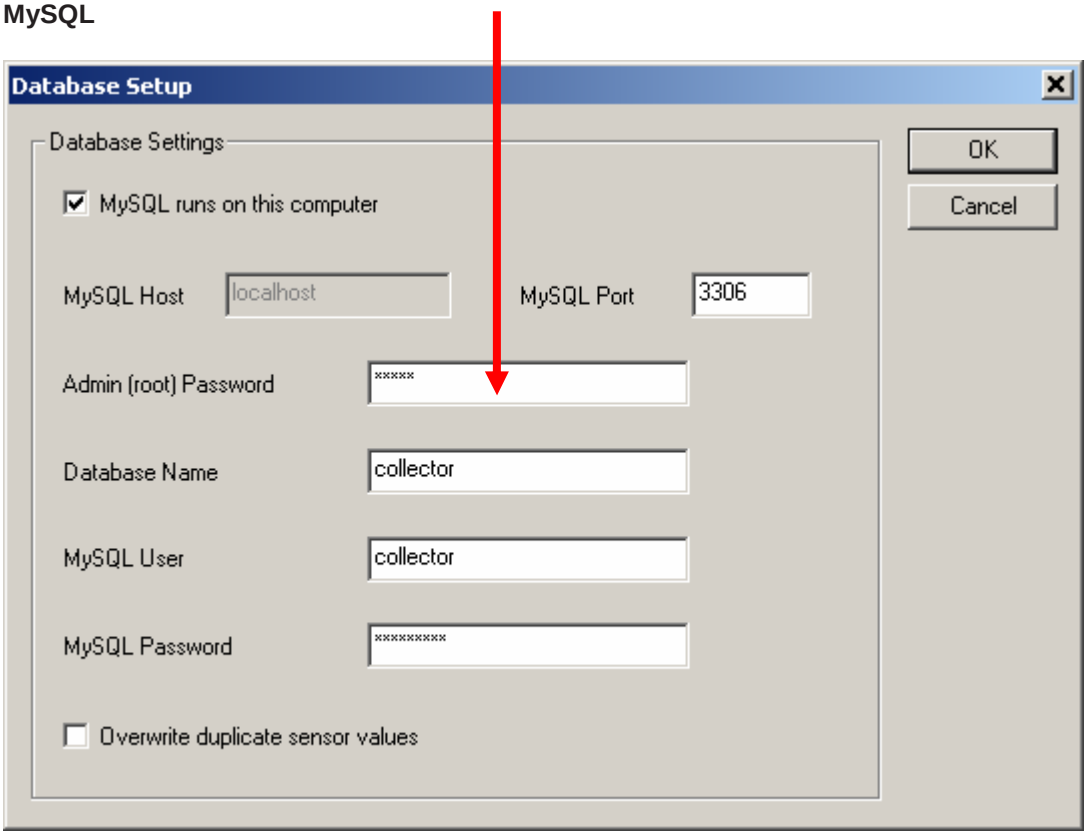



If you do not want to keep log files for SmartCom on a daily basis, please remove the <date> tag from the filenames.

## 2.2 Database Setup

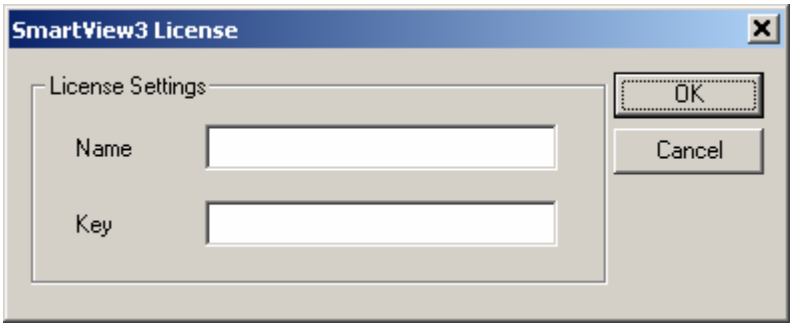
- Then, the parameters for the MySQL database are set.
  **You need to enter the MySQL Administrator password you entered when installing MySQL**



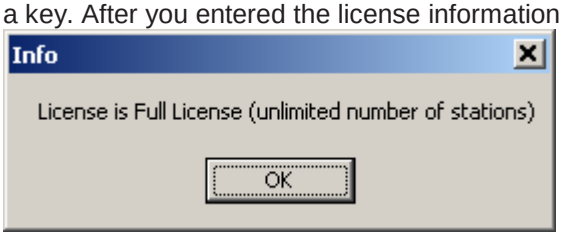

SmartView3 will now start SmartCom to create or update the database
**Note: updating the database may take a lot of time, depending on the amount of measurement data you have already stored in your database. Updating the database from an earlier version requires that there is at least the amount of free disc space on the volume where your MySQL database is located as the sensor_values table requires, because the table may be duplicated during the update process!**

## 2.3 License

- After creating or updating the database, the license information is set up.
  The basic license allows you to run Collector for up to 5 stations – it does not support other features like export/import, alarms or SmartWeb to create web sites.

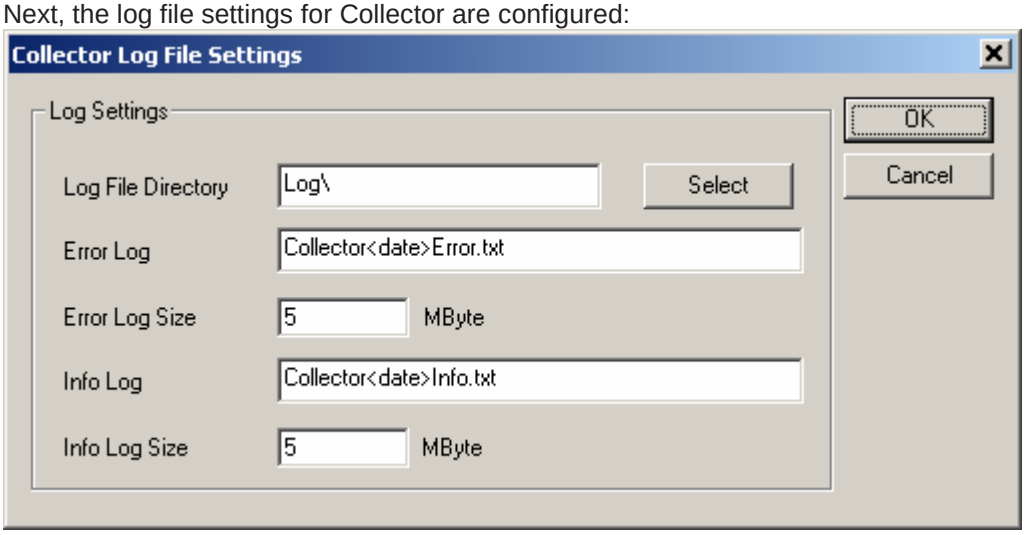



Even if you didn't purchase an "extended" License, you still need to enter a license name and a key. After you entered the license information, a dialog displays your license settings, e.g.:



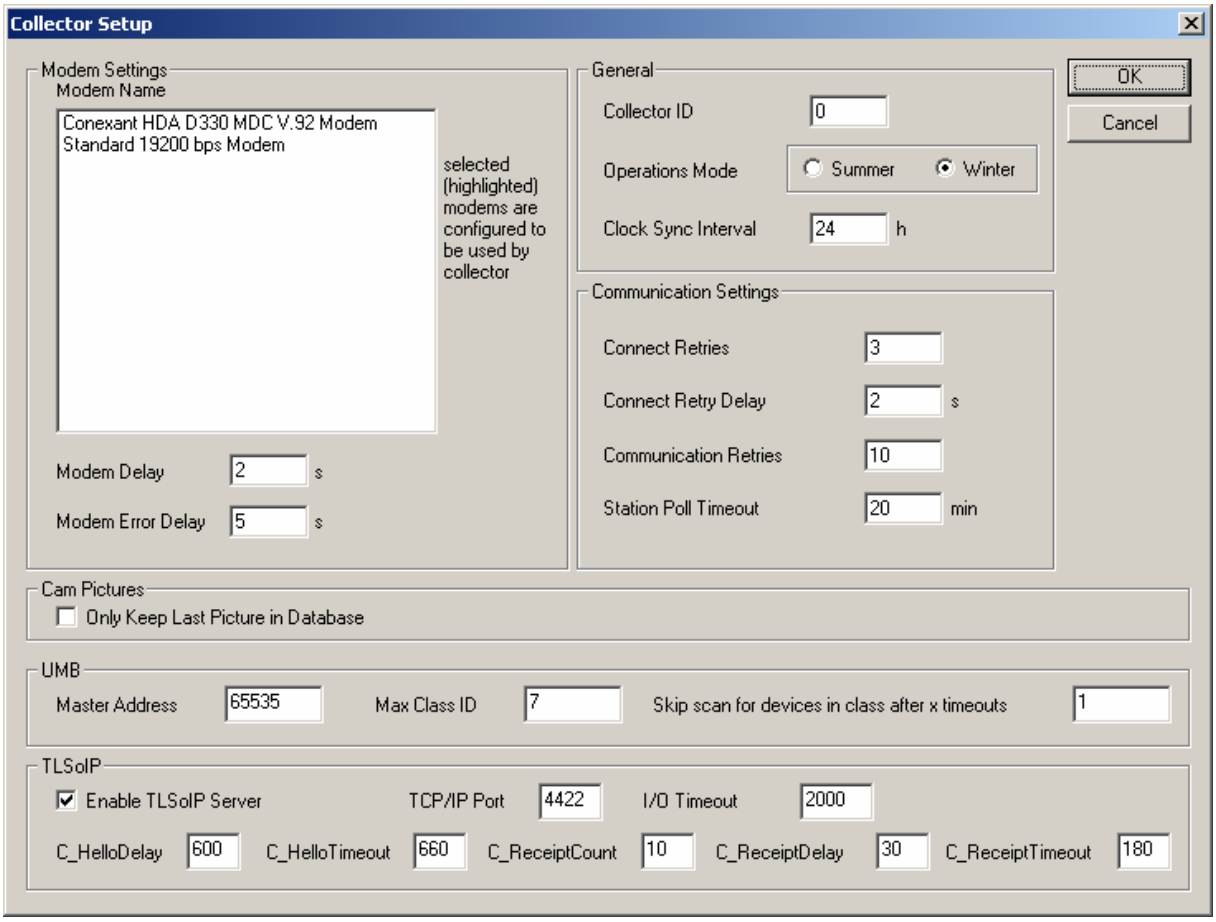
## 2.4 Collector Setup

- Next, the log file settings for Collector are configured:




If you do not want to keep log files for Collector on a daily basis, please remove the <date> tag from the filenames.

Then a dialog to configure other parameters for Collector is displayed:




Here you need to select the modem(s) you would like to use with Collector to connect to the stations (if you use dial up connections to your stations). You can assign the modems to different “modem pools”.

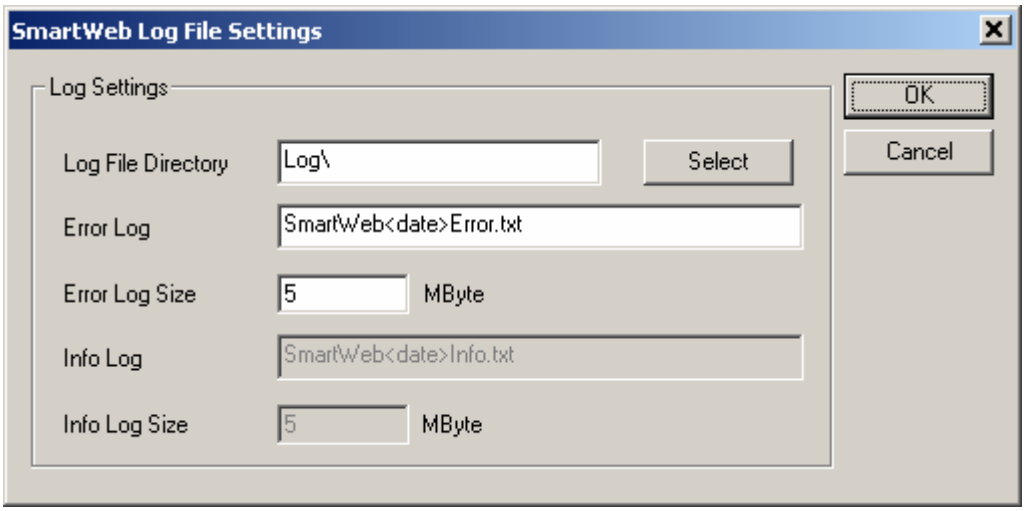
If you plan to use TLSoIP based stations, you need to enable “TLSoIP”, and set the TLSoIP

parameters to your needs.

SmartView3 then prompts to start the Collector. If you did not upgrade from a previous version of Collector, there are no stations configured yet, so you may want to start the Collector later after you configured your stations

## 2.5 SmartWeb Setup

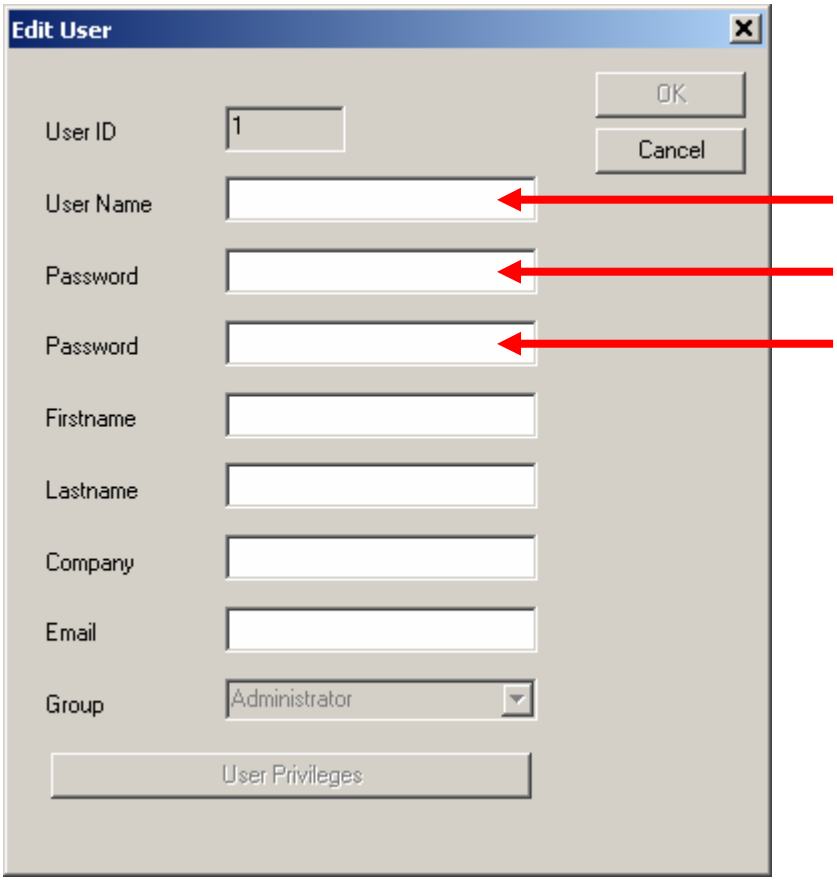
- If your license includes SmartWeb, the SmartWeb log file settings are configured next:




If you do not want to keep log files for SmartWeb on a daily basis, please remove the <date> tag from the filenames.

## 2.6 Administrator User

- Now, the administrator user for SmartView3 is set up. You need to set up at least one administrator user by entering a user name and the password (all other information may be left blank).



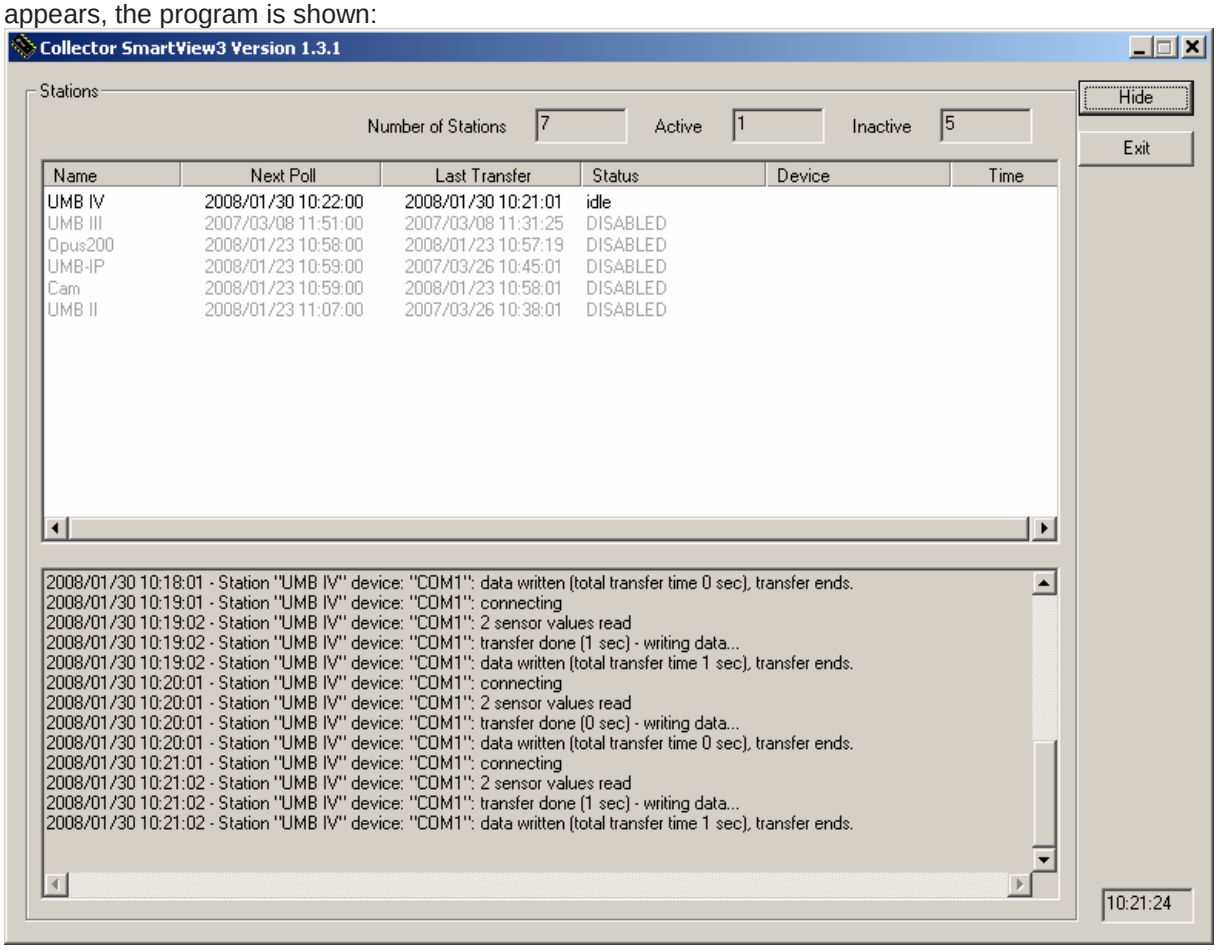

After you entered the user information, SmartView3 is started, and you can configure your stations and web site(s).

# 3 The Modules

## 3.1 Collector SmartView3

Collector runs as a background program. When started, it is minimized, and shows as an icon in the system tray bar:

If you double click this icon, or if you right click the icon and select “show” from the pop-up menu that appears, the program is shown:




The configured stations are shown in the upper part of the window, showing the next station due to be polled first.
In the lower part of the window status/information messages are displayed

Configuration information for Collector is (if not stored in the database) held in the Collector.ini file.

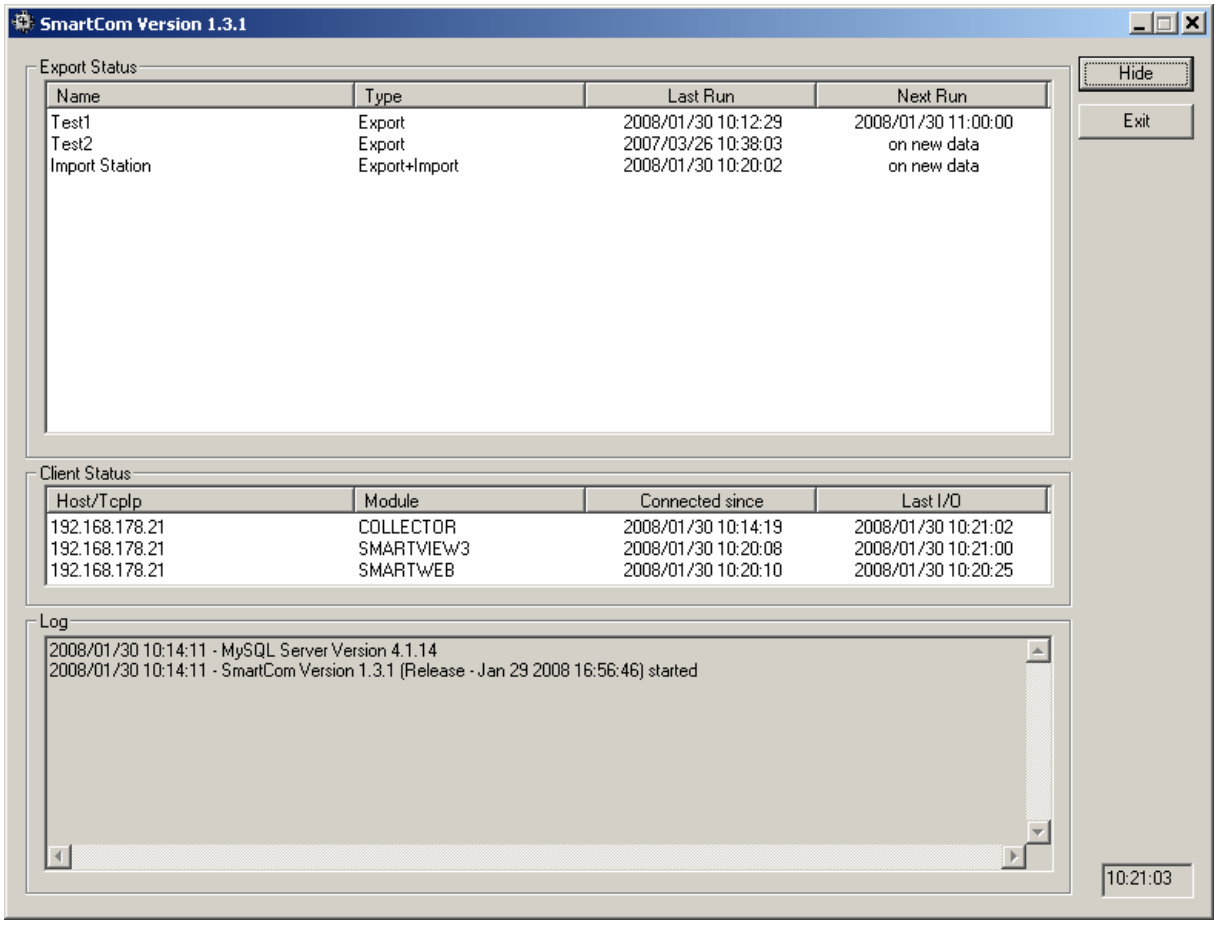
## 3.2 SmartCom

The SmartCom module provides access to the database for SmartView3 and SmartWeb, and contains the “export/import” and “alarm” functions.

Like Collector, SmartCom accesses the MySQL database via the `libmysql.dll` library provided by MySQL. This library must be accessible by the program, i.e. it must be installed on the computer so that it can be found by the operating system (either in an appropriate system directory, or in the directory where the SmartCom program is located).

It is recommended to run SmartCom and Collector on the same computer where the database is located, to minimize network traffic when accessing the database.
Like Collector, SmartCom is run as a background program, and is minimized to the system tray when started.

If you double-click the icon in the system tray, or right click the icon and select “show” from the pop-up menu, the SmartCom window is displayed:




In the upper part of the window, any configured automatic export jobs are displayed. Below that, the SmartView3 modules that are connected to SmartCom are shown. In the lower part, error and information messages are displayed.

Configuration information for SmartCom is held in the SmartCom.ini file.

## *3.3 SmartWeb*

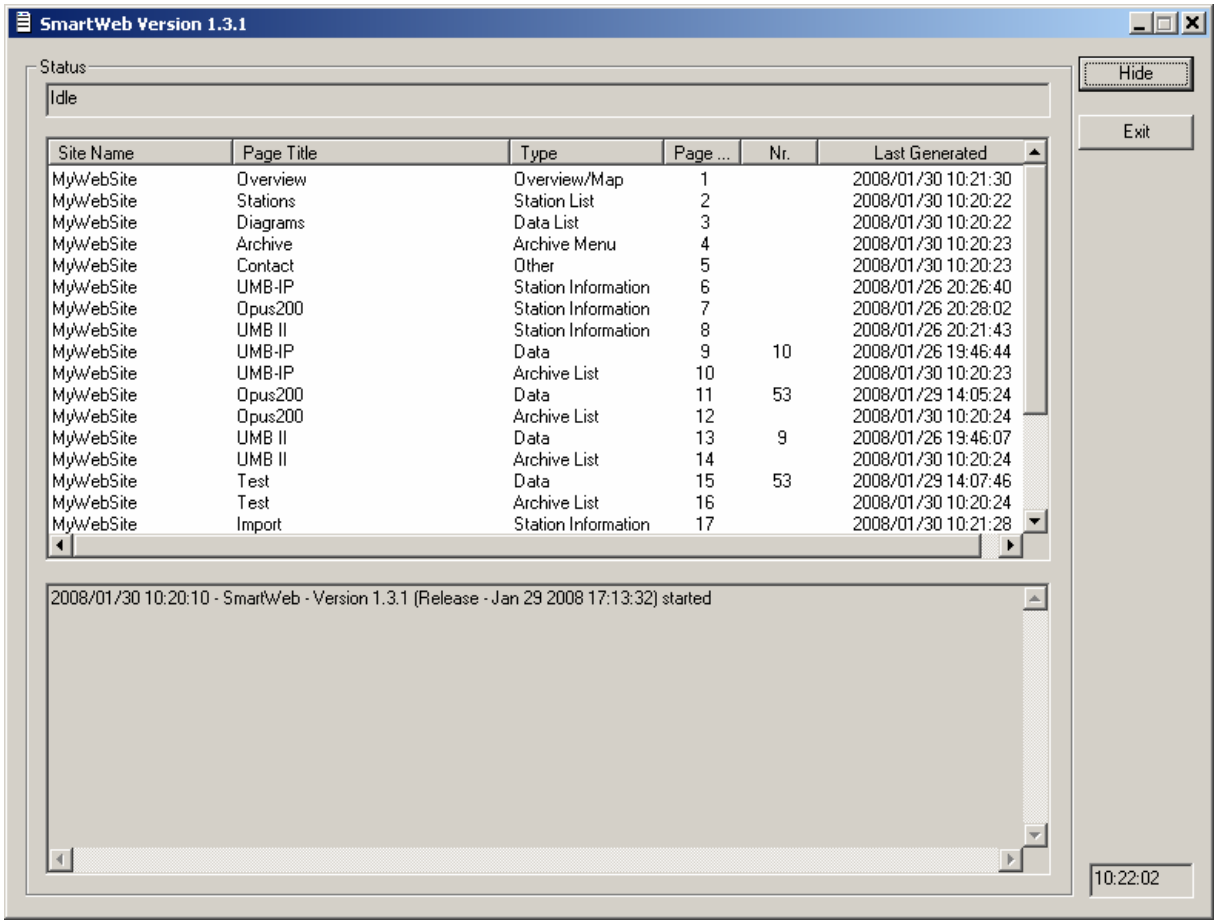
SmartWeb is the module that creates the web pages and uploads these pages to a web server (if ftp upload is configured for a site)

SmartWeb communicates with SmartCom to access the data in the database indirectly.

The program is designed to be run constantly (after the site has been configured initially using SmartView3). You may want to place a link to the SmartWeb program in the “Autostart” folder of your system. Note that SmartCom must be running in order for SmartWeb to start - if you run SmartWeb and SmartCom on the same PC make sure to start SmartCom before starting SmartWeb.

Like Collector or SmartCom, SmartWeb runs as a background program and is minimized to the system tray after startup:

If you double click the system tray icon, or right click the icon and select “show” from the pop-up menu, the program window is displayed:




In the upper part of the window, all pages of all configured web sites are displayed, with information about when the pages were updated last.

In the lower part of the window, error and status messages are displayed.

Configuration information for SmartWeb is held in the SmartWeb.ini file.

## *3.4 SmartView3*

SmartView3 is the configuration program for all functions/modules of the software package.

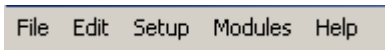
In future releases, SmartView3 may also provide functions to view and analyze the measurement data in diagrams and tables.

SmartView3 communicates with SmartCom to access data - it does not access the database directly.

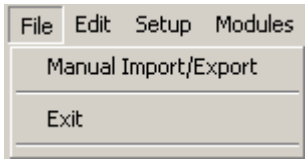
# 4 SmartView3 Menus

## 4.1 Main Menu

The main menu contains the following menu items:



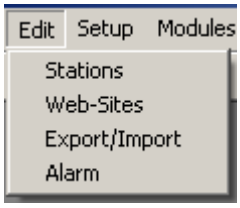

### 4.1.1 File




- Manual Import/Export : configure/run a manual import/export job
- Exit : exit SmartView3

### 4.1.2 Edit

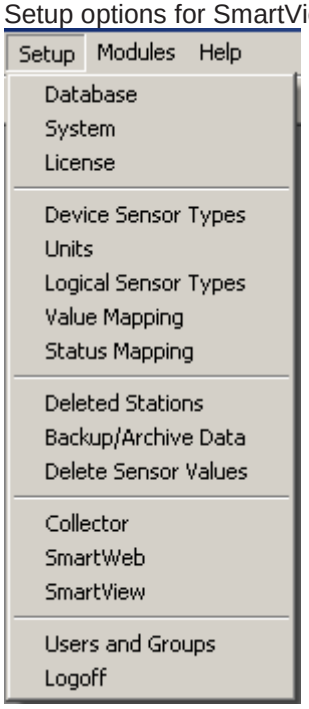
Configure Stations, Web-Sites, Export/Import and Alarm settings




- Stations: view Collector status and configure stations
- Web-Sites: configure web sites
- Export/Import: configure automatic export/import jobs
- Alarm: configure alarm settings for stations and sensors

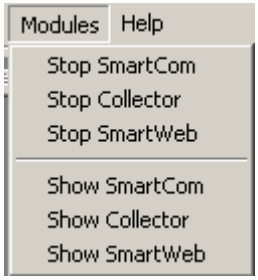
### 4.1.3 Setup

Setup options for SmartView3




- Database : the database settings
- System: SmartView3 system setup (hosts and IP ports for the SmartView3 modules)
- License: setup SmartView3 License
- Device Sensor Types: edit device sensor types
- Units: edit units
- Logical Sensor Types: edit logical sensor types (Sensor Type Templates)
- Value Mapping: edit value mapping templates
- Status Mapping: edit status mapping templates
- Deleted Stations: manage deleted stations, i.e. either “undelete” a deleted station, or remove the station and all related data completely.
- Backup/Archive data: setup automatic backup and/or compression of data
- Delete Sensor Values: delete values for selected sensors from database
- Collector: edit Collector options
- SmartWeb: edit SmartWeb options
- SmartView: edit SmarView3 options
- Users and Groups: edit users and groups
- Logoff: logoff current user

### 4.1.4 Modules




Start/Stop Smartcom: start or stop the SmartCom program
Start/Stop Collector: start or stop the Collector program
Start/Stop SmartWeb: start or stop the SmartWeb program
Show SmartCom: show the SmartCom program window (if running)

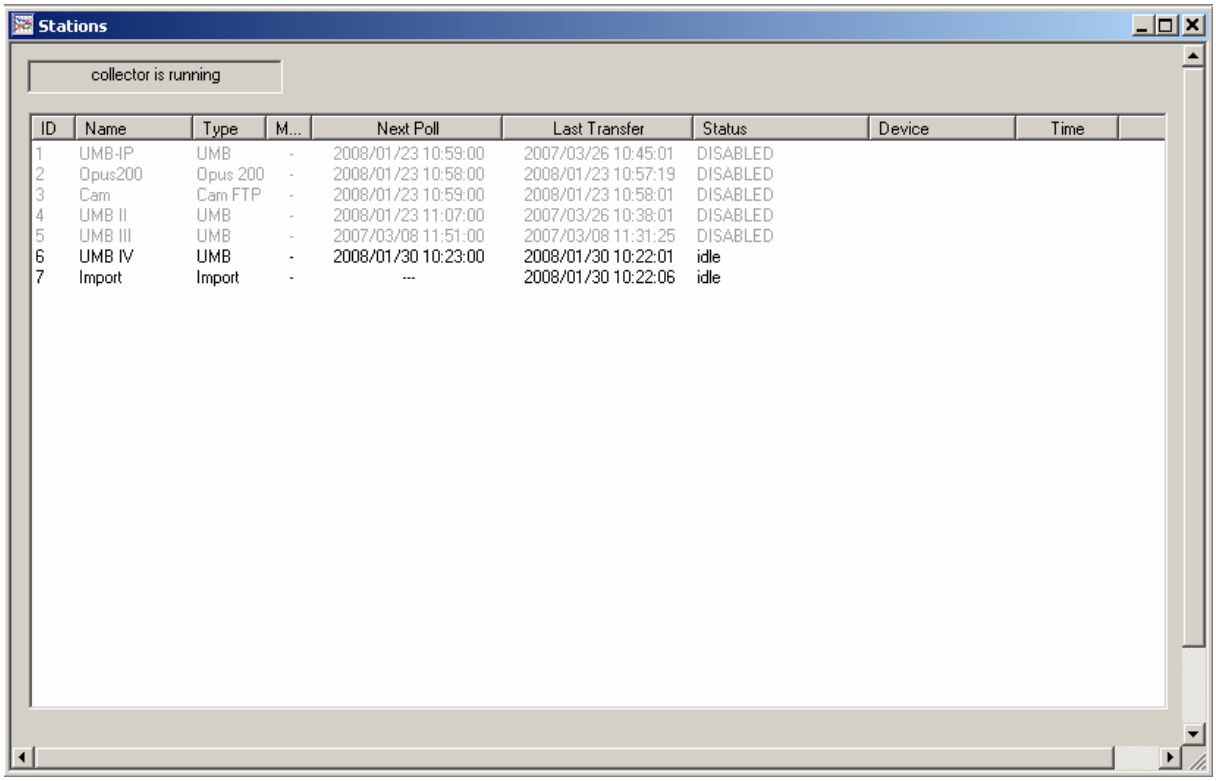
Show Collector: show the Collector program window (if running)
Show SmartWeb: show the SmartWeb program window (if running)

# 5 Configure Stations

To configure stations, select “Edit->Stations” from the main menu.

This opens the station/collector status view window:

## *5.1 Station status*

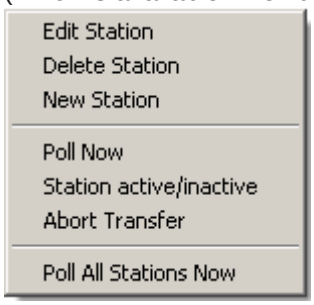



The window shows all configured stations, and the status of the station in Collector, e.g. the status of a station if it is currently polled by collector (and the collector is running).

To edit a station, double click the station in the list, or select the station and select “Station->Edit” from the menu.

The list of stations can be sorted by “ID”, “Name”, “Type”, “Modem Pool”, “Next Poll”, “Last Transfer” or “Status” by clicking on the appropriate column header.

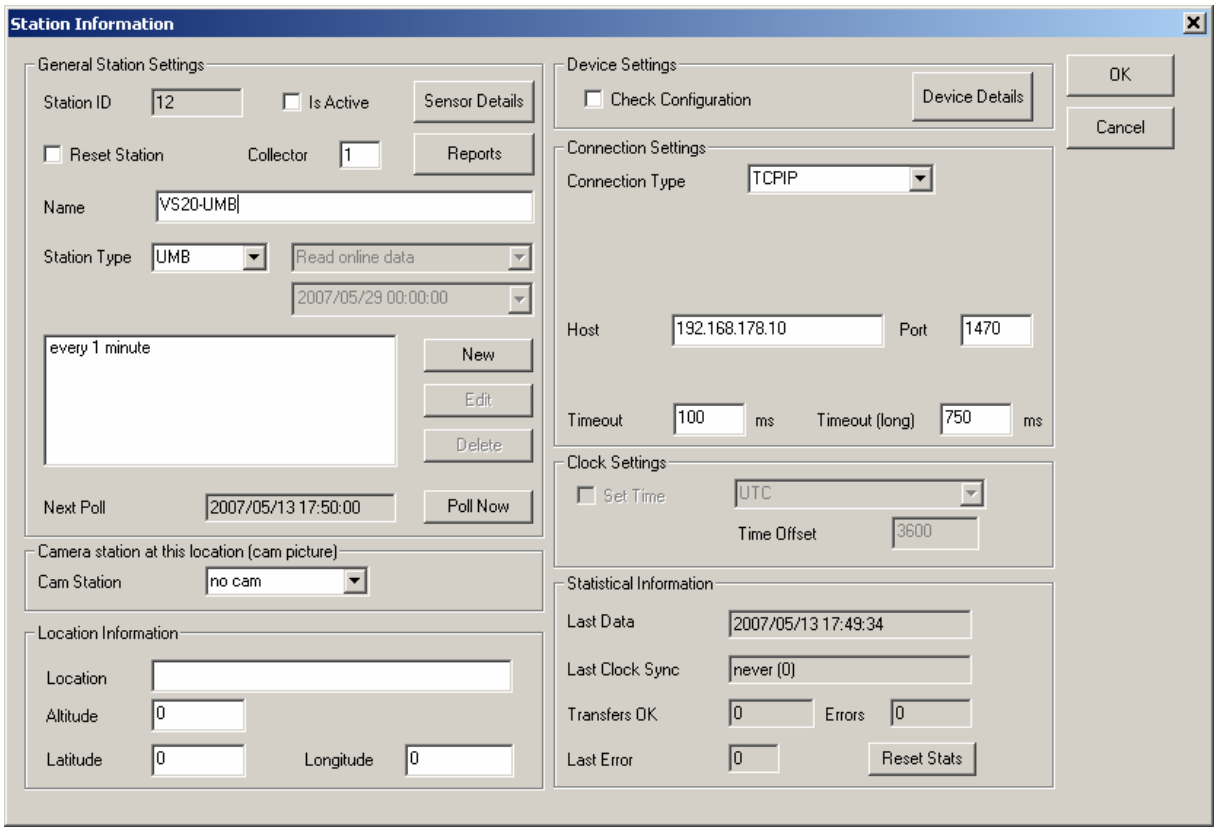
To create a new station, select “Station->New” from the menu, or left click in the station window to show the pop-up menu. The pop-up menu contains the same menu items as the main menu “Stations” (which is available when the station window is opened).




“Edit Station” or “New Station” will display the station configuration dialog (see below).
“Poll Now” will re-set the “next poll” time for the selected station to “now”.
“Station active/inactive” will toggle the active status of the station.
“Abort transfer” will abort the transfer in progress for the selected station

"Poll All Stations Now" will re-set the "next poll" time for all stations to "now".

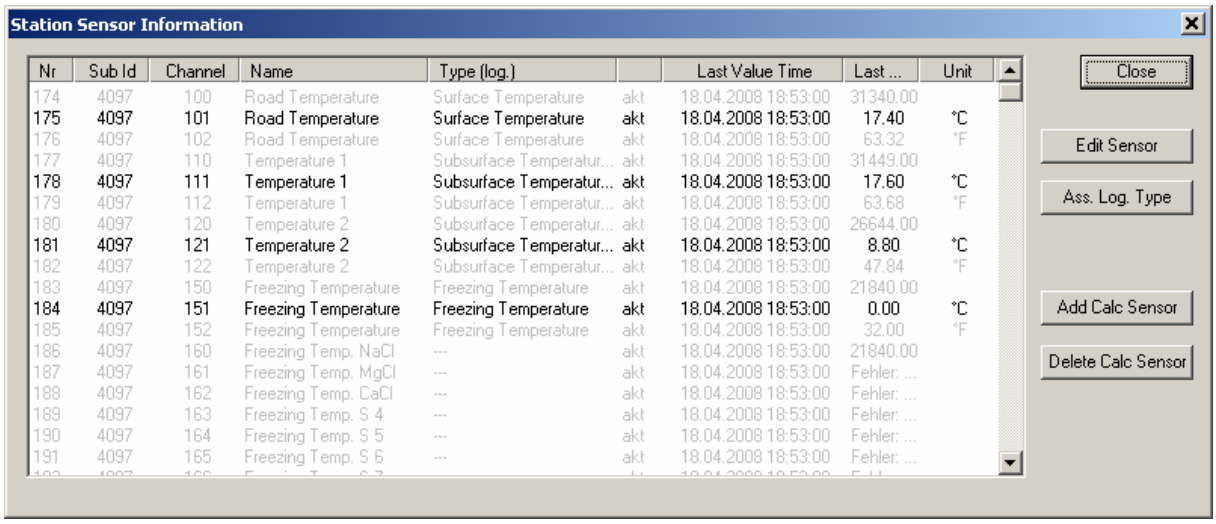
## 5.2 Station Settings




- Station ID : the (program internal) station id
- Is Active: station is active (or inactive)
- Sensor Details: shows information about all sensors of a station (see below)
- Name: the name of the station
- Station Type: the station type
- Read online data/Read stored data/Read all stored data: read online data or data stored in logger. If "read all stored data" is selected, the start time for reading the logger data can be set in the edit field below the selection box.
- Station Schedule: this list contains one or more station schedule entries (see below)
- Next Poll: time station will be polled next
- Cam Station ID: station id of a Cam FTP/HTTP station associated with the station (0 = none)
- Location/Altitude/Latitude/Longitude: location information about station
- Check Configuration: check the station configuration every time a station is polled
- Device Details: shows information about all devices at the station
- Connection Type: the connection type for the station. Depending on the connection type, different connection parameters can be configured
- Timeout/Timeout (long): Timeout settings for device I/O
- Clock Settings: time/clock settings for station.
  **NOTE: when reading data from the device memory (stored data), the clock settings MUST BE SET to the device clock settings, even if the clock is not automatically synchronized- otherwise the time information for the retrieved data may not be correct!**
- Set Time: synchronize the stations clock automatically to the system time, depending on the time zone settings
- Statistical Information: some statistical information about the station

### 5.2.1 Station Sensor Details

Clicking on the "Sensor Details" button opens a window that displays all sensors for a station:
Note: sensor information is only available if the station has been polled successfully at least once.




Disabled sensors are displayed in gray. For all active sensors, the last value time, last value and unit are displayed.

To view/edit the station sensor settings, double click the sensor in the list, or select the sensor and click the “Edit Sensor” button.

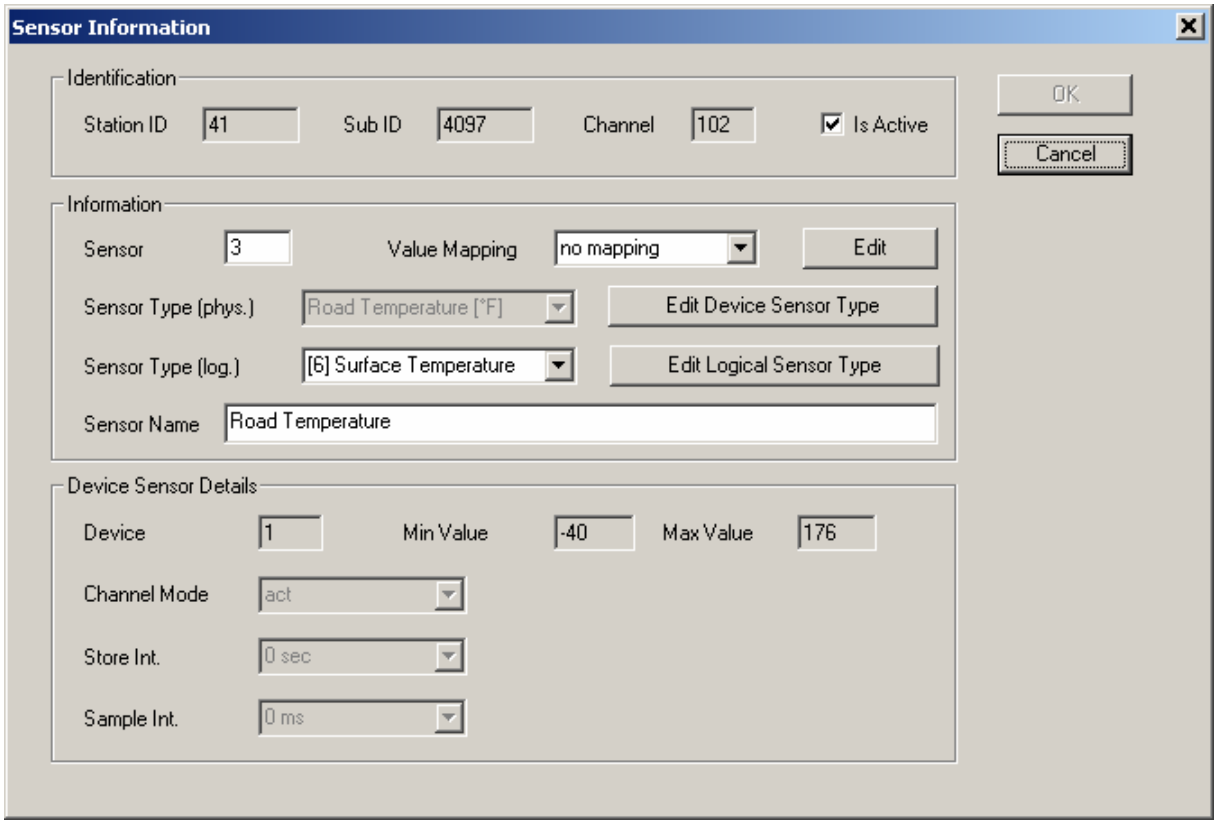
Buttons:

- Ass. Log Type: open dialog to show/change logical sensor type assignments for every active sensor of the station
- Add Calc Sensor: add a calc sensor (see below)
- Delete Calc Sensor: delete a calc sensor (see below)

To assign a logical sensor type to every active sensor.

#### *5.2.1.1 Sensor Information*

For all sensors reported by the device, the “sensor information” dialog is displayed. For “calculated sensors” please see chapter 5.2.1.2 below.

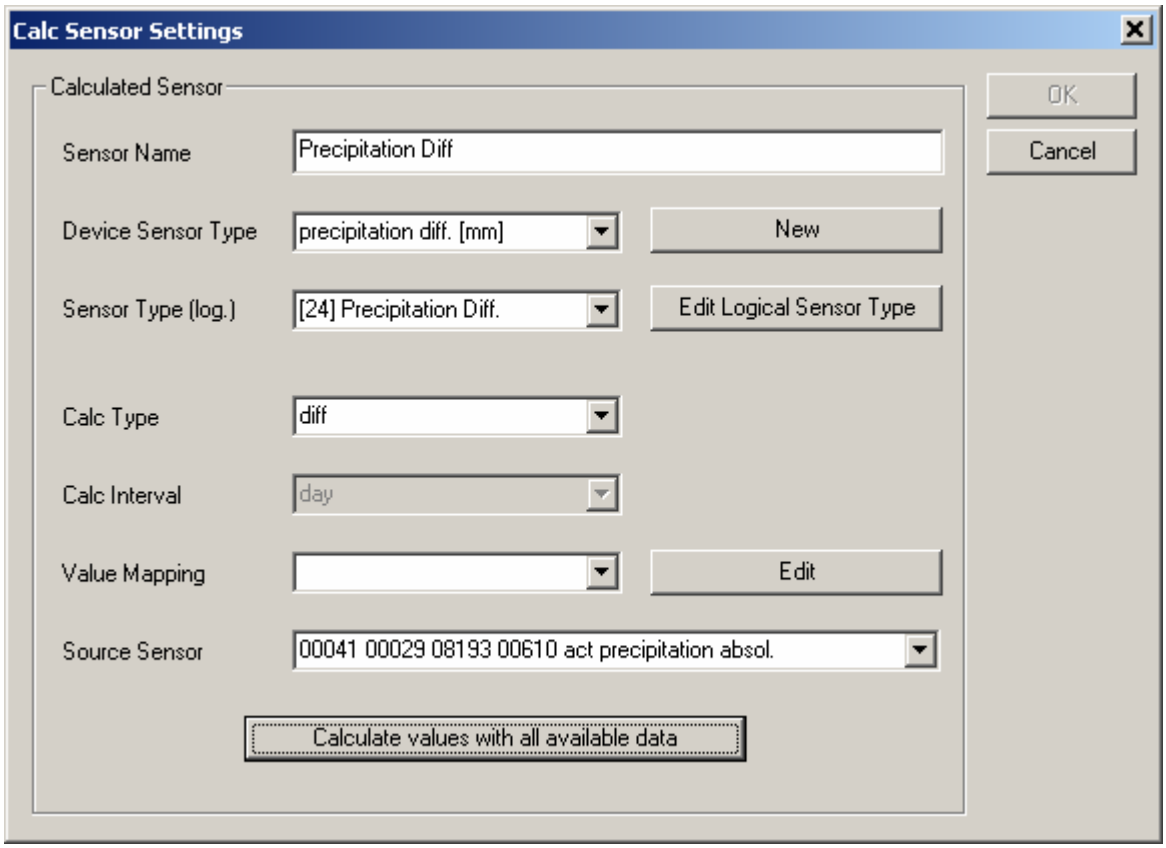



The station sensor information is displayed.
Depending on the station type, following configuration details may be set:

- Is Active: for UMB devices, a selected sensor/channel can be set active (will be read) or inactive (will not be read). For all other device types, this cannot be changed (depends on the device configuration)
- Sensor: The (within a station unique) sensor nr assigned to this channel. The sensor nr is used in web site pages and in export jobs to refer to a specific sensor.
- Value Mapping: mapping of sensor values – applied before values are stored to the database (see Value Mapping)
- Sensor Type (phys): the device sensor type
- Sensor Type (log): the logical sensor type.
  **Note:** SmartView automatically assigns a logical sensor type to each sensor (as far as possible – for some Opus200 sensors it cannot be assigned automatically). This may however not be the correct sensor type. You should check and assign an **unique** logical sensor type to each (active) sensor of the station. If, for example, you have a station with multiple road sensors, please create new logical sensor types (Surface Temperature 2, Road Condition 2 etc.) for each additional road sensor, so each sensor has a unique logical sensor type assigned.
- Sensor Name: the name used for this sensor/channel. For Opus200 devices, this name is stored in the device, and changing the name will result in a configuration change of the device; The change will not be visible until the station is polled next time and the change is applied to the device
- Channel Mode: this can only be changed for Opus200 devices
- Store Interval: this can only be changed for Opus200 devices
- Sample Interval: this can only be changed for Opus200 devices

### *5.2.1.2 Calculated Sensors*

If a “calculated sensor” is added or edited, the following dialog is displayed:
At the moment only “simple” calculations (min/max/avg/sum/diff and value mapping) based on one “souce sensor” are supported.




- **Sensor Name**: name for the calculated sensor
- **Device Sensor Type**: (device) sensor type for the calculated sensor – select from list or create a new type
- Sensor Type (log): logical sensor type for the calculated sensor – select from list or create a new type.
- **Calc Type**: calculation type for sensor:
  Currently following types are supported:
  - Value mapping – use Value Mapping (see below) to calculate new value for each source sensor value
  - Min – calculate minimum value for each given Calc Interval
  - Max – calculate maximum value for each given Calc Interval
  - Avg – calculate average value for each given Calc Interval
  - Sum – calculate sum value for each given Calc Interval
  - Diff – calculate difference for counter values, e.g. precipitation amount from absolute precipitation for UMB R2S sensor
- **Calc Interval**: time interval for min/max/avg/sum calculation
- **Value Mapping**: Value Map to be used for “value mapping” calculation
- **Source Sensor**: the source sensor for the calculation

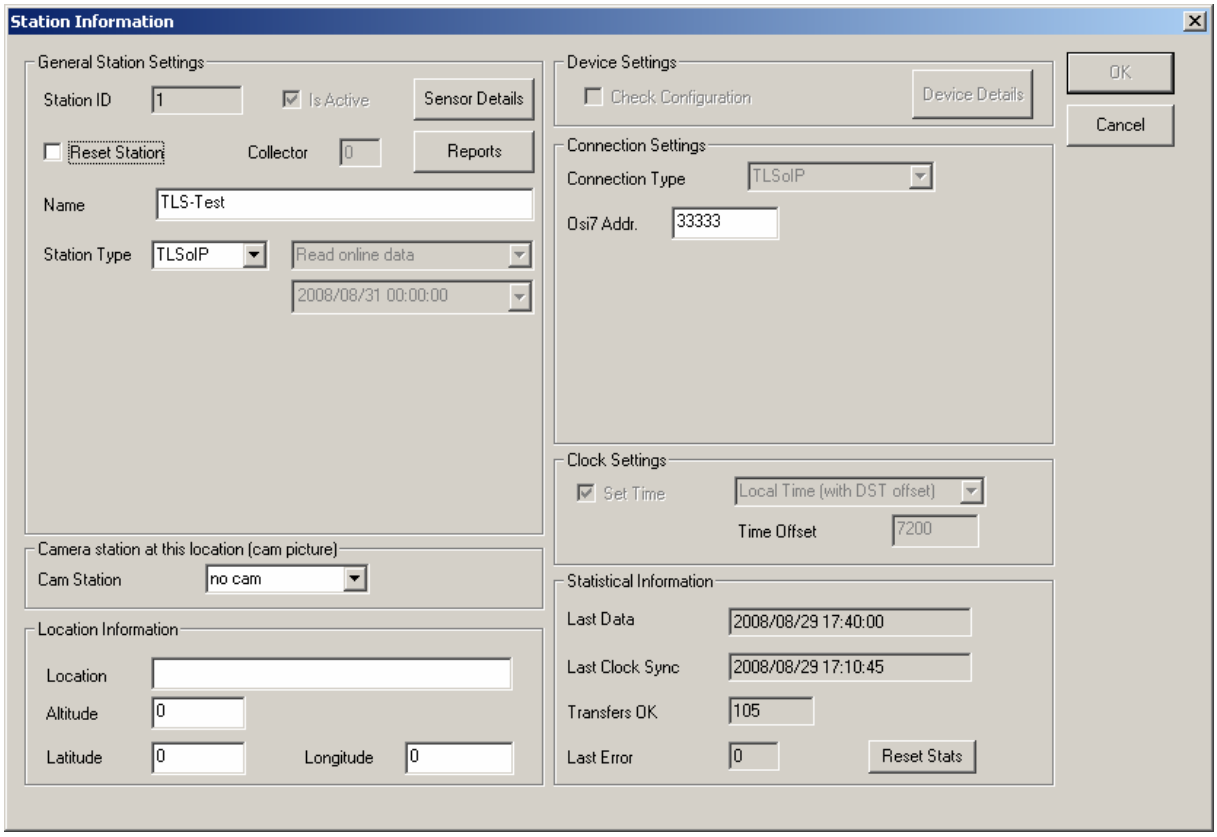
Calculation for a calc sensor is done whenever new data for the source sensor is read from the device. The calculation is done in the “Collector” program after saving the new sensor values.
Using “Calculate values with all available data” you can (re-) calculate the values for this sensor for all available data for the source sensor.

Note: the timestamp for calculated values that are calculated over a given interval (i.e. min/max/avg/sum calculation) is set to the end of the given interval, e.g. if the interval is “day”, the timestamp will be 23:59:59 hours of the given day etc.

### 5.2.2 TLSoIP Stations

Configuration for a TLSoIP station differs from other station types, as in this case the communication between the station and Collector is not initiated by Collector, but by the station. In order to support TLSoIP stations, TLSoIP must be activated in the Collector Setup Dialog (see Collector Setup).

TLSoIP stations are recognized by their unique OSI7 address. If a station connects to Collector, the database is searched for a configured station with this OSI7 address. If not station with this address is found, Collector automatically configures a new TLSoIP station for the OSI7 address.



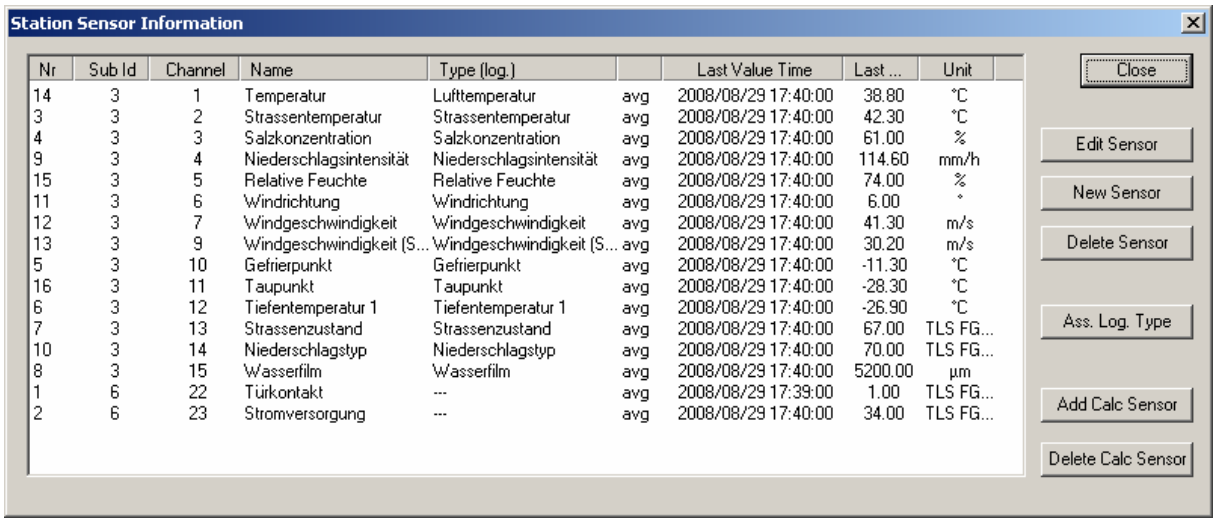

Configuration parameters for a TLSoIP station are (besides general items like an assigned CAM station or the location information) limited to the station name and OSI7 Address.

#### *5.2.2.1 Sensors for TLSoIP Stations*

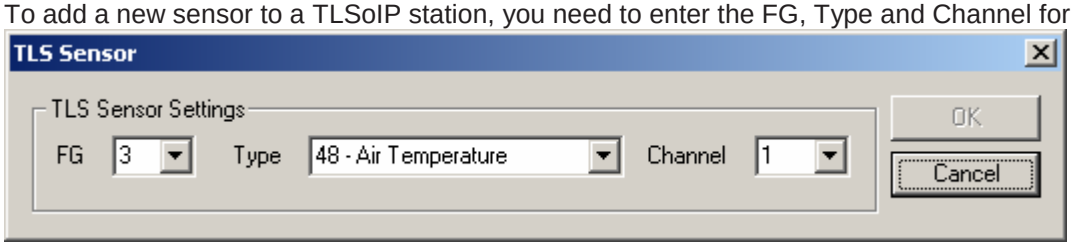
If an TLSoIP station connects, it will typically send “DE-Status” messages for all sensors configured in the station to the server (i.e. Collector). TLSoIP sensors are recognized by their “FG”, “Channel” and “Type” configuration. DE Status Messages only contain FG and Channel information. If no sensor for this FG/Channel can be found in the database, an error message is written to the log file.

If the station reports sensor values, FG, Channel and Type are transmitted with the sensor value. If no sensor for this FG/Channel/Type can be found in the database, Collector will automatically configure a new sensor.

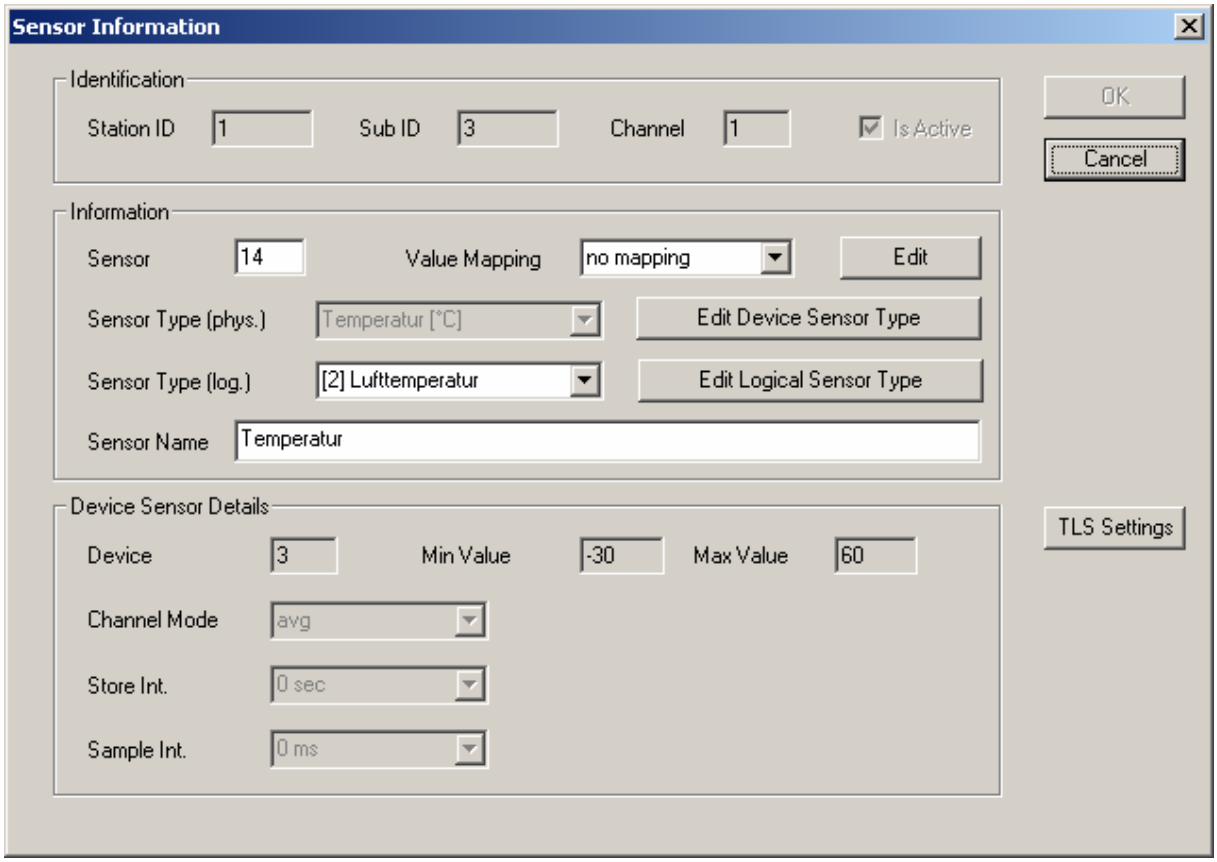
This way **Collector will automatically configure all necessary data for a TLSoIP station if the station does report sensor values for all configured sensors**. If there are sensors configured in the station that are de-activated, or do not report values (e.g. because the sensor is defect), Collector will not be able to auto-configure that sensor – and thus will not be able to display the error status for such a sensor. To cover this, you can manually add/edit/delete sensors for TLSoIP stations:




To add a new sensor to a TLSoIP station, you need to enter the FG, Type and Channel for the sensor:



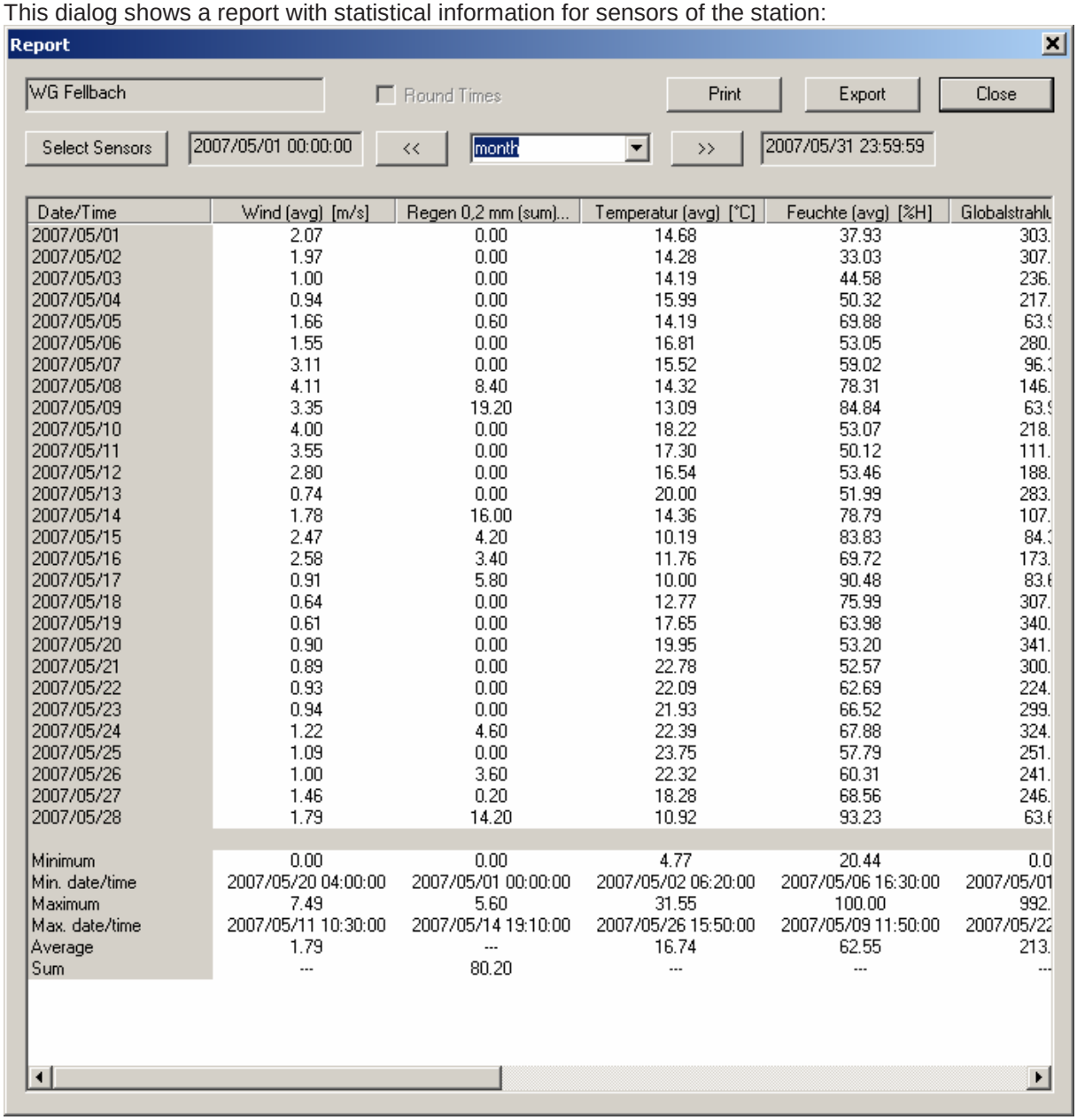

If you edit an already configured sensor for a TLSoIP station, an additional button “TLS Settings” is available to change the FG/Type/Channel assignments for a sensor:




The TLS FG is stored in the “Sub ID” attribute for the sensor (and the “Device Sensor Type” attribute of the Device Sensor Type information), the TLS channel is stored in the “Channel” attribute. The TLS Type is stored in the “Sub-Type” attribute of the Device Sensor Type information.

### 5.2.3 Reports

This dialog shows a report with statistical information for sensors of the station:




| Date/Time | Wind (avg) [m/s] | Regen 0,2 mm (sum)... | Temperatur (avg) [°C] | Feuchte (avg) [%H] | Globalstrahlu |
|---|---|---|---|---|---|
| 2007/05/01 | 2.07 | 0.00 | 14.68 | 37.93 | 303. |
| 2007/05/02 | 1.97 | 0.00 | 14.28 | 33.03 | 307. |
| 2007/05/03 | 1.00 | 0.00 | 14.19 | 44.58 | 236. |
| 2007/05/04 | 0.94 | 0.00 | 15.99 | 50.32 | 217. |
| 2007/05/05 | 1.66 | 0.60 | 14.19 | 69.88 | 63. |
| 2007/05/06 | 1.55 | 0.00 | 16.81 | 53.05 | 280. |
| 2007/05/07 | 3.11 | 0.00 | 15.52 | 59.02 | 96. |
| 2007/05/08 | 4.11 | 8.40 | 14.32 | 78.31 | 146. |
| 2007/05/09 | 3.35 | 19.20 | 13.09 | 84.84 | 63. |
| 2007/05/10 | 4.00 | 0.00 | 18.22 | 53.07 | 218. |
| 2007/05/11 | 3.55 | 0.00 | 17.30 | 50.12 | 111. |
| 2007/05/12 | 2.80 | 0.00 | 16.54 | 53.46 | 188. |
| 2007/05/13 | 0.74 | 0.00 | 20.00 | 51.99 | 283. |
| 2007/05/14 | 1.78 | 16.00 | 14.36 | 78.79 | 107. |
| 2007/05/15 | 2.47 | 4.20 | 10.19 | 83.83 | 84. |
| 2007/05/16 | 2.58 | 3.40 | 11.76 | 69.72 | 173. |
| 2007/05/17 | 0.91 | 5.80 | 10.00 | 90.48 | 83. |
| 2007/05/18 | 0.64 | 0.00 | 12.77 | 75.99 | 307. |
| 2007/05/19 | 0.61 | 0.00 | 17.65 | 63.98 | 340. |
| 2007/05/20 | 0.90 | 0.00 | 19.95 | 53.20 | 341. |
| 2007/05/21 | 0.89 | 0.00 | 22.78 | 52.57 | 300. |
| 2007/05/22 | 0.93 | 0.00 | 22.09 | 62.69 | 224. |
| 2007/05/23 | 0.94 | 0.00 | 21.93 | 66.52 | 299. |
| 2007/05/24 | 1.22 | 4.60 | 22.39 | 67.88 | 324. |
| 2007/05/25 | 1.09 | 0.00 | 23.75 | 57.79 | 251. |
| 2007/05/26 | 1.00 | 3.60 | 22.32 | 60.31 | 241. |
| 2007/05/27 | 1.46 | 0.20 | 18.28 | 68.56 | 246. |
| 2007/05/28 | 1.79 | 14.20 | 10.92 | 93.23 | 63. |
| Minimum | 0.00 | 0.00 | 4.77 | 20.44 | 0.0 |
| Min. date/time | 2007/05/20 04:00:00 | 2007/05/01 00:00:00 | 2007/05/02 06:20:00 | 2007/05/06 16:30:00 | 2007/05/01 |
| Maximum | 7.49 | 5.60 | 31.55 | 100.00 | 992. |
| Max. date/time | 2007/05/11 10:30:00 | 2007/05/14 19:10:00 | 2007/05/26 15:50:00 | 2007/05/09 11:50:00 | 2007/05/22 |
| Average | 1.79 | --- | 16.74 | 62.55 | 213. |
| Sum | --- | 80.20 | --- | --- | --- |

Depending on the selected time interval (can be day, month, or year) the appropriate values are shown in a table.
If "**day**" is selected, average (or sum) values for every hour of the day are shown, and the minimum, maximum, average or sum values for the day are displayed.

If "**month**" is selected, average (or sum) values for every day are shown, and the minimum, maximum, average or sum values for the month are displayed.

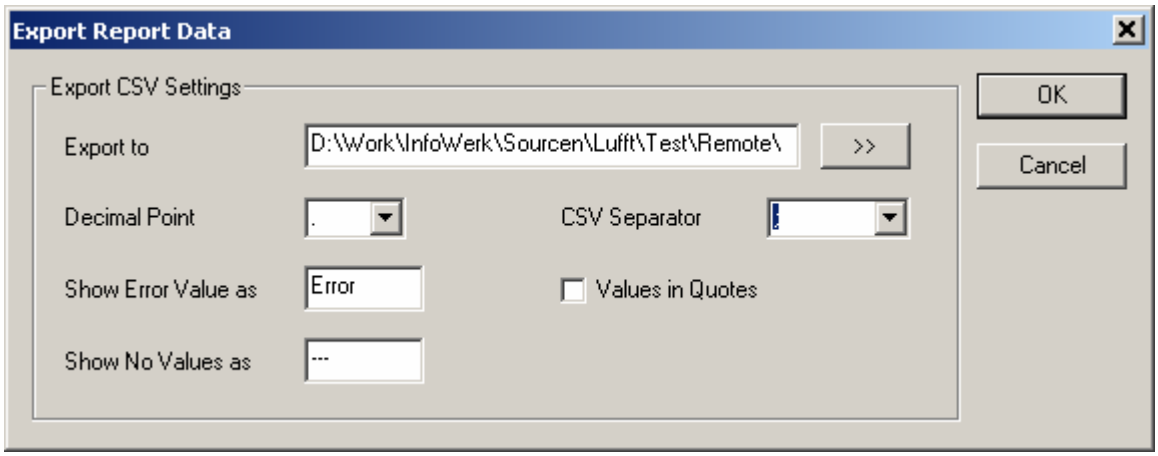
If "**year**" is selected, average (or sum) values for every month of the year are shown, and the minimum, maximum, average or sum values for the year are displayed.

The "**<<**" and "**>>**" buttons can be used to display the previous/next time interval.

Per default, all (active) sensors of a station are displayed. Using the "**Select Sensors**" button, sensors can be selected, and the sequence of the sensors can be altered.

"**Print**" will print the displayed report data.

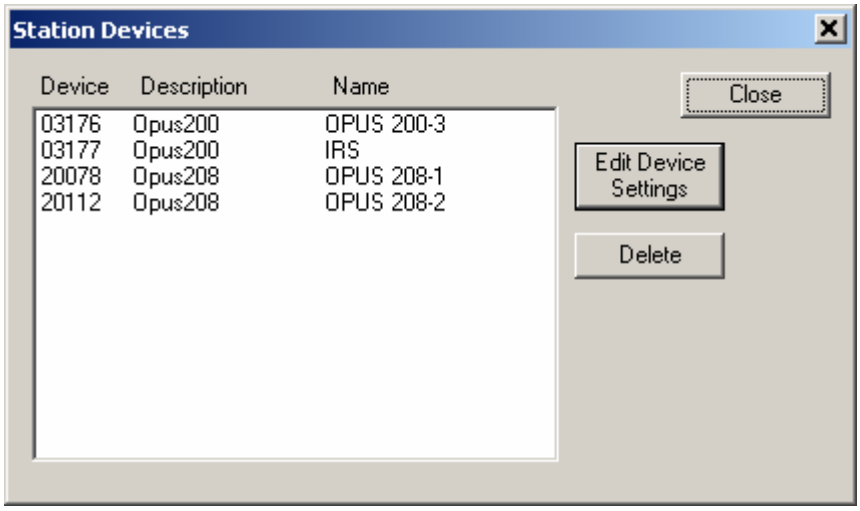
"**Export**" will export the displayed report data to a .csv (text) file – e.g. for import and processing in Microsoft Excel etc. Some parameters can be set to specify the export format:



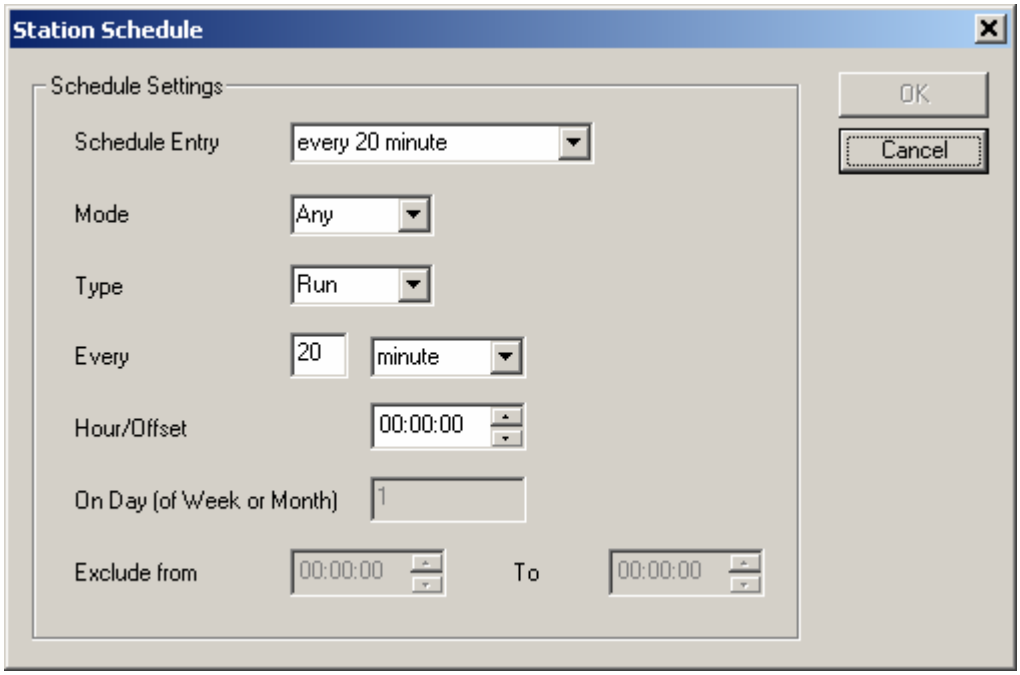

### 5.2.4 Station Device Details

Clicking on the "Device Details" button opens a window that displays all devices for a station.
Note: device information is only available if the station has been polled successfully at least once.

For TLSoIP stations, no "Device Details" are available.



### 5.2.5 Station Schedule




You may configure multiple schedule entries for a station.

- Mode: operations mode for schedule entry (Any, Summer or Winter)
- Type: type of entry (Run or Exclude)
- Every xx interval: run every xx interval, e.g. every 20 minutes, or every 1 day…
- Hour/Offset: hours/min/seconds to be added to begin of interval, i.e. the time of day for interval "day, week, or month"
- On Day(of week or month) – if the interval type is week or month, you can set the day of week or month the station should be polled
- Exclude from/to: for schedule entries of type exclude, this specifies the time (of day) the station will not be polled

Possible time intervals are:

- every xx minute
- every xx 5 minutes (to the full hour, i.e. 00:00, 00:05, 00:10 etc)
- every xx 10 minutes (to the full hour, i.e. 00:00, 00:10. 00:20 etc) [ same as every 2 5 minutes..]
- every xx 12 minutes (to the full hour, i.e. 00:00, 00:06, 00:12, 00:18…)
- every xx 20 minutes (to the full hour, i.e. 00:00, 00:20, 00:40…)
- every xx 30 minutes (to the full hour, i.e. 00:00, 00:30,…)
- every xx hour
- every xx day
- every xx week

# 6 Configure SmartWeb pages

## *6.1 Overview*

The SmartWeb configuration consists of the following objects:

### 6.1.1 Site

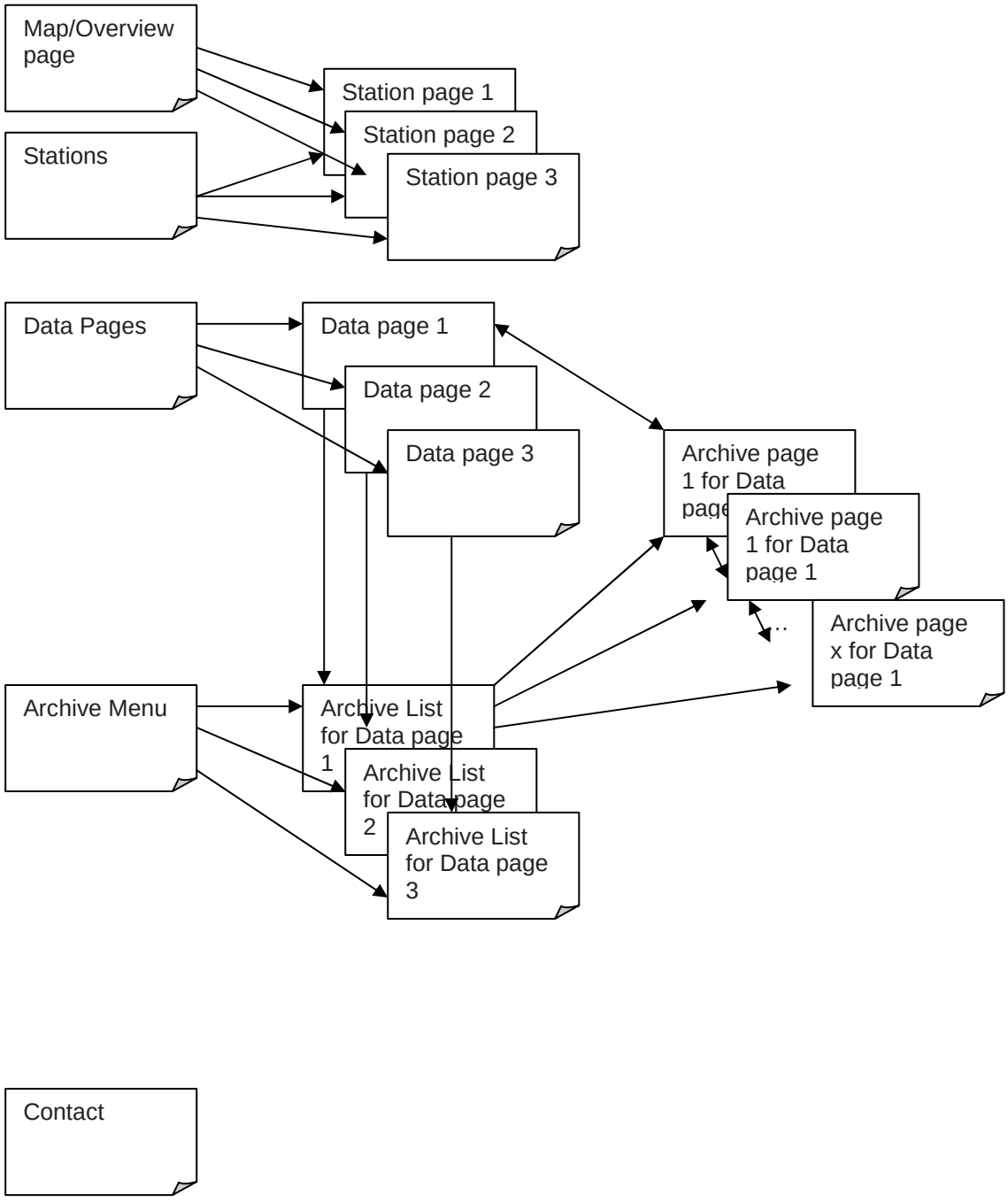
A site refers to a “web site”, which may contain data from multiple stations. A “Site” consists of “Pages” of different types (see below).
The pages that are generated by SmartWeb for a site are always stored on the PC where SmartWeb runs (in the directory indicated by the “local path” field of the Site configuration), and can be viewed using a web browser (i.e. Internet Explorer, Mozilla Firefox, or Opera…).
The “start” page of a site is the file “index.html” in the “local path” directory – open that file in your browser to view the data.
The pages of a “site” can of course also be accessed via a web server. You could either install a web server (like Apache) on the same PC where you are running SmartWeb, and configure the web server to use the “local path” of the site as “document root”, or configure SmartWeb to use the “document root” (or a directory below that) as local path. Or you can configure SmartWeb to transfer all files to an external web server.

**Note:**

The generated website contains active elements and Javascript code used for navigation and popup texts. Make sure your web browser is set up properly to allow the use of active elements.

When viewing the generated web site from the local disc with Microsoft Internet Explorer, Internet Explorer always pops up a warning about “active elements”, and blocks these elements.

The structure of a SmartWeb generated web-site is as follows:

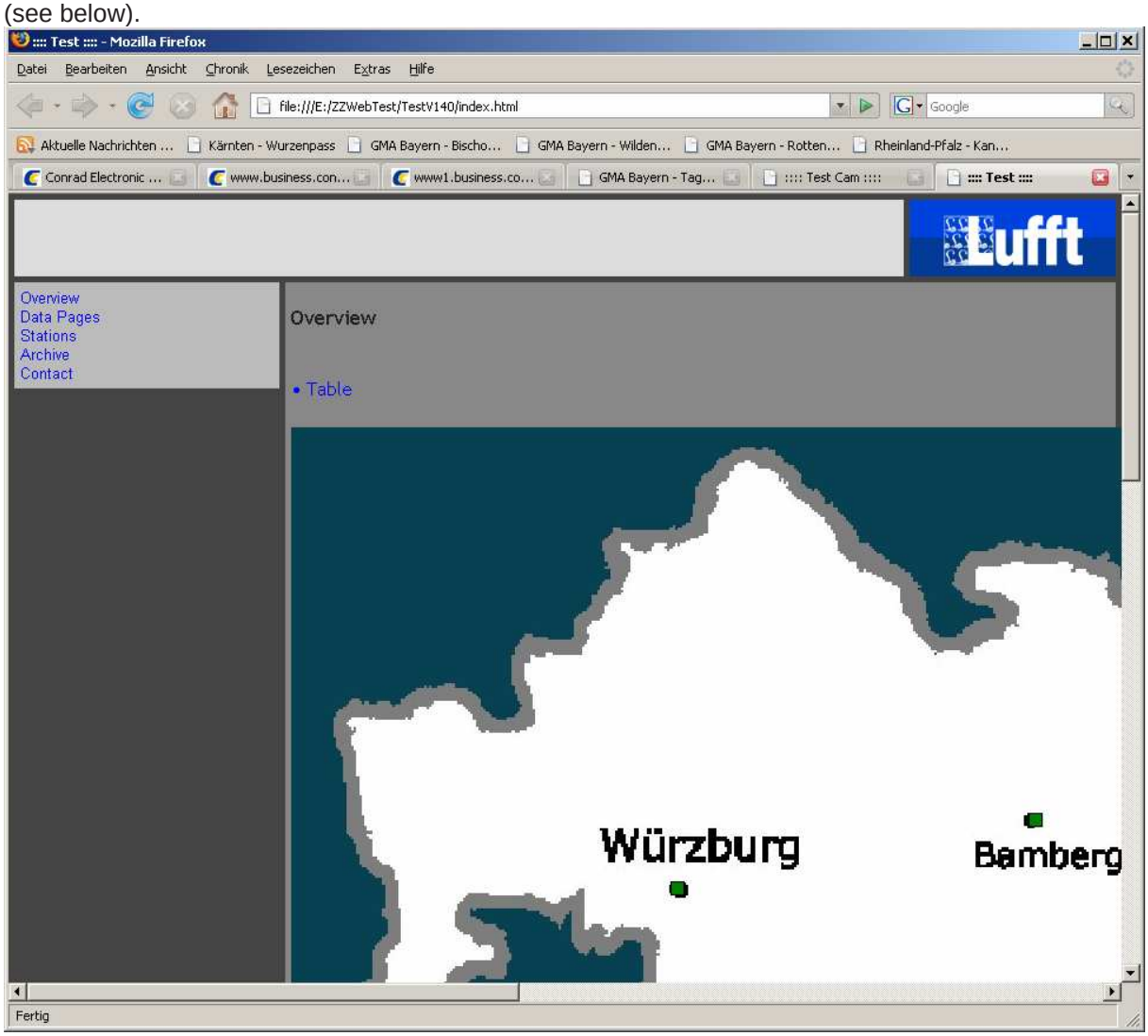


### 6.1.2 SmartWeb Pages

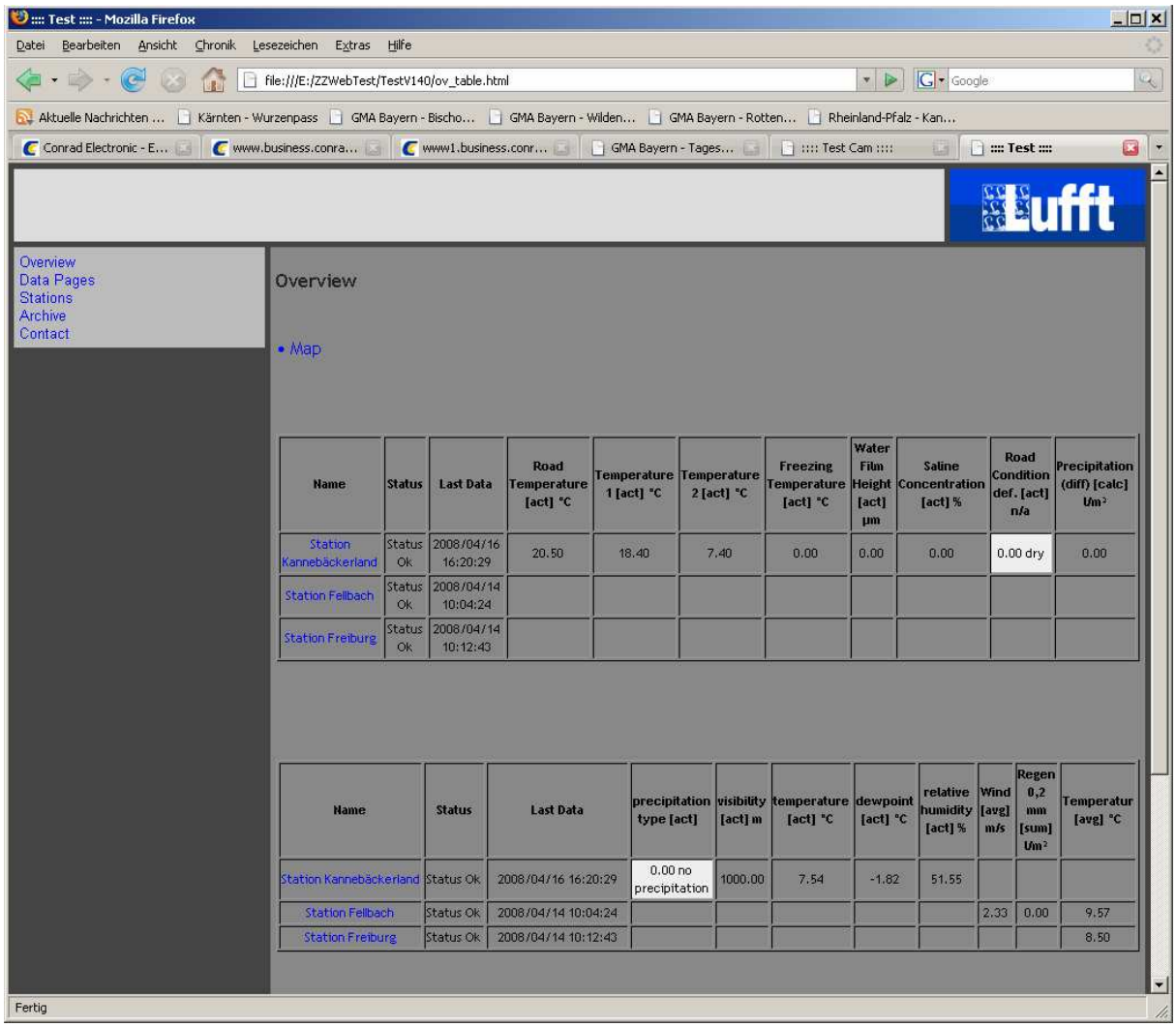
SmartWeb will generate following types of pages:

#### *6.1.2.1 Map page- "Overview"*

This is the "start" or "home" page of the site. It shows the status of all stations configured for this site either in a table (with a link to the "station page" for this station), or (graphically) on a map.
The map page is the first menu item on the menu generated by SmartWeb on the left side of each page.
The "title" of this page is used in the menu, and can be customized in the "edit page" dialog for this page.
If you loaded a map for this page, and placed your stations on the map using the appropriate dialog (see below), the stations will be displayed on a map as colored squares, with the color indicating the status of the station, i.e. green when there was no error, or red if the station could not be polled or there were error or alarm values for sensor values. If you scroll over a station icon on the map, the station name and status is displayed, as well as the last sensor values (if you configured sensors to be shown on the "station icons"). Clicking on the station icon will take you to the appropriate station page (see below).



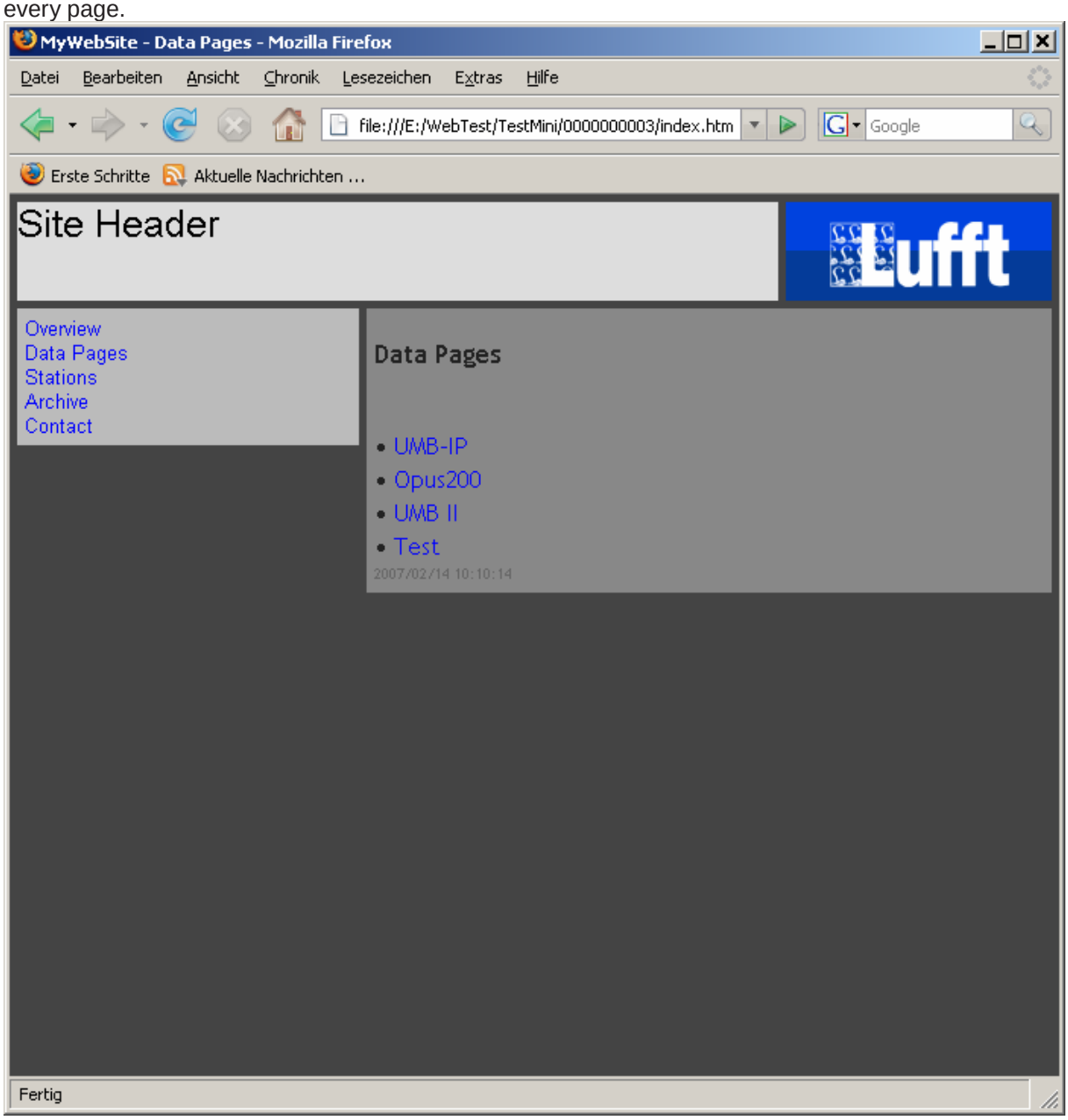

On the "map page", a link to the "Table" overview is provided.
If no map is configured, only the "Table" overview is created.




The “table overview” shows status of all stations, and the sensors configured for the “station icon” of each station in a table. Sensors with identical logical sensor type, value type and unit are shown in the same column. There is a param for the “Overview page” (see below) to configure the maximum number of sensors displayed in one table.
If there is a map configured for the Overview page, a link back to the map page is available on top of the table overview page.

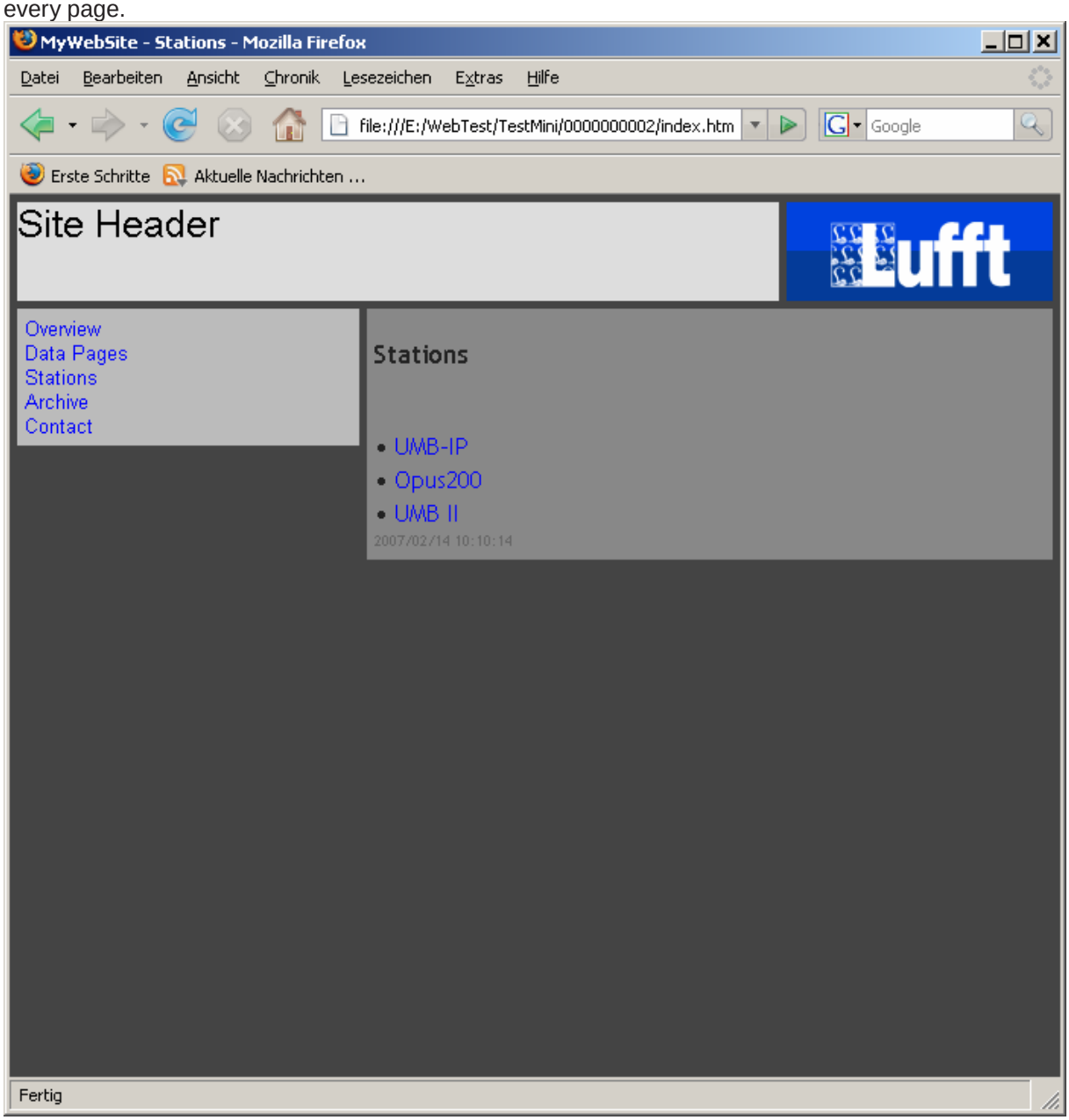
### *6.1.2.2 data list page*

This page contains a list (with links) to all data pages generated by SmartWeb. A “data page” is a page showing sensor values in diagrams and tables.
The “data list page” is the second menu item on the menu created by SmartWeb on the left side of every page.



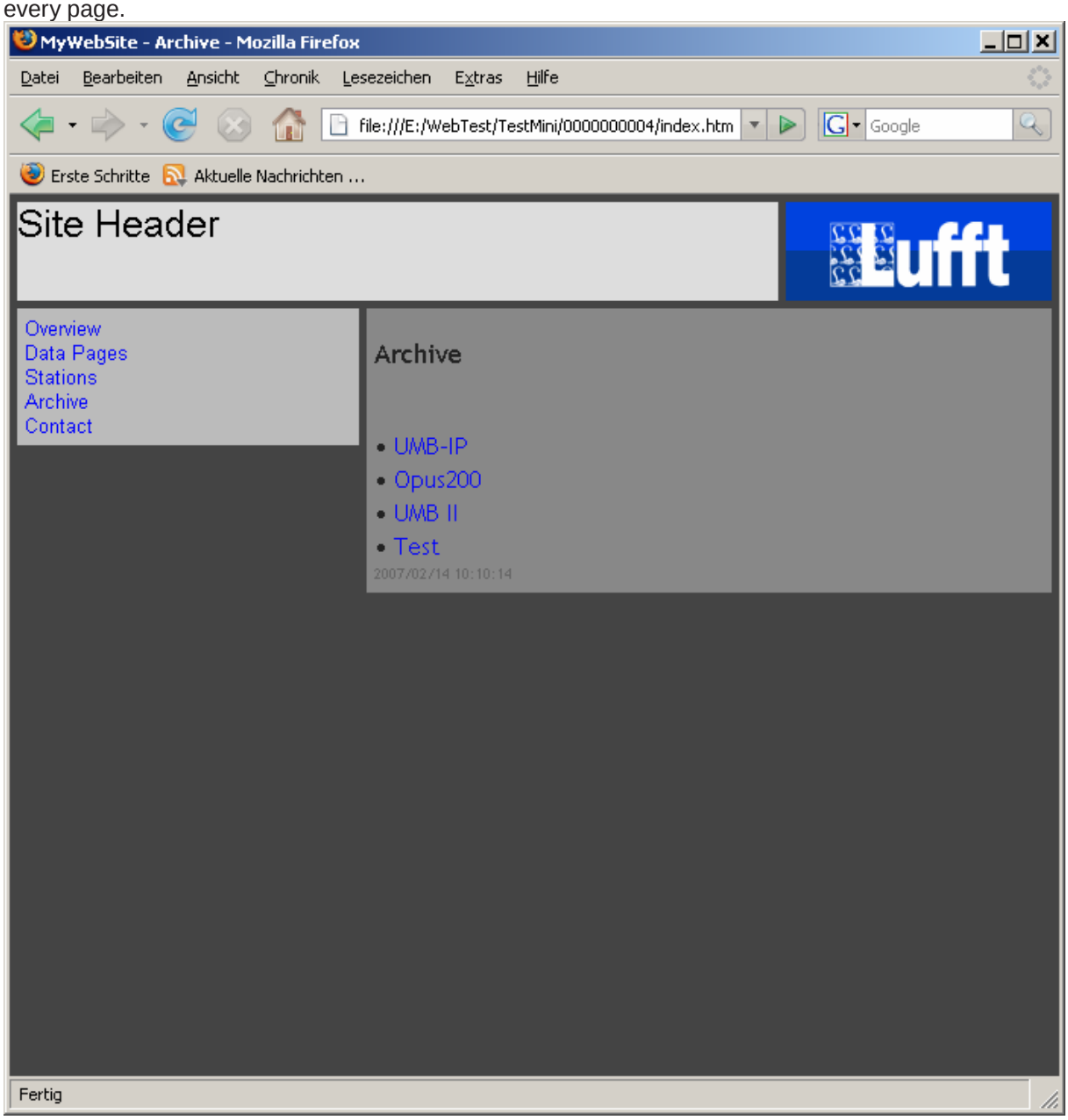
### *6.1.2.3 station list page*

The “station list” page lists all “station pages” with a link to the appropriate station page – similar to the “map page” if the map page is not configured to show the stations on a map. The station list page does not show status information for a page.
The “station list” page is the third menu item on the menu created by SmartWeb on the left side of every page.



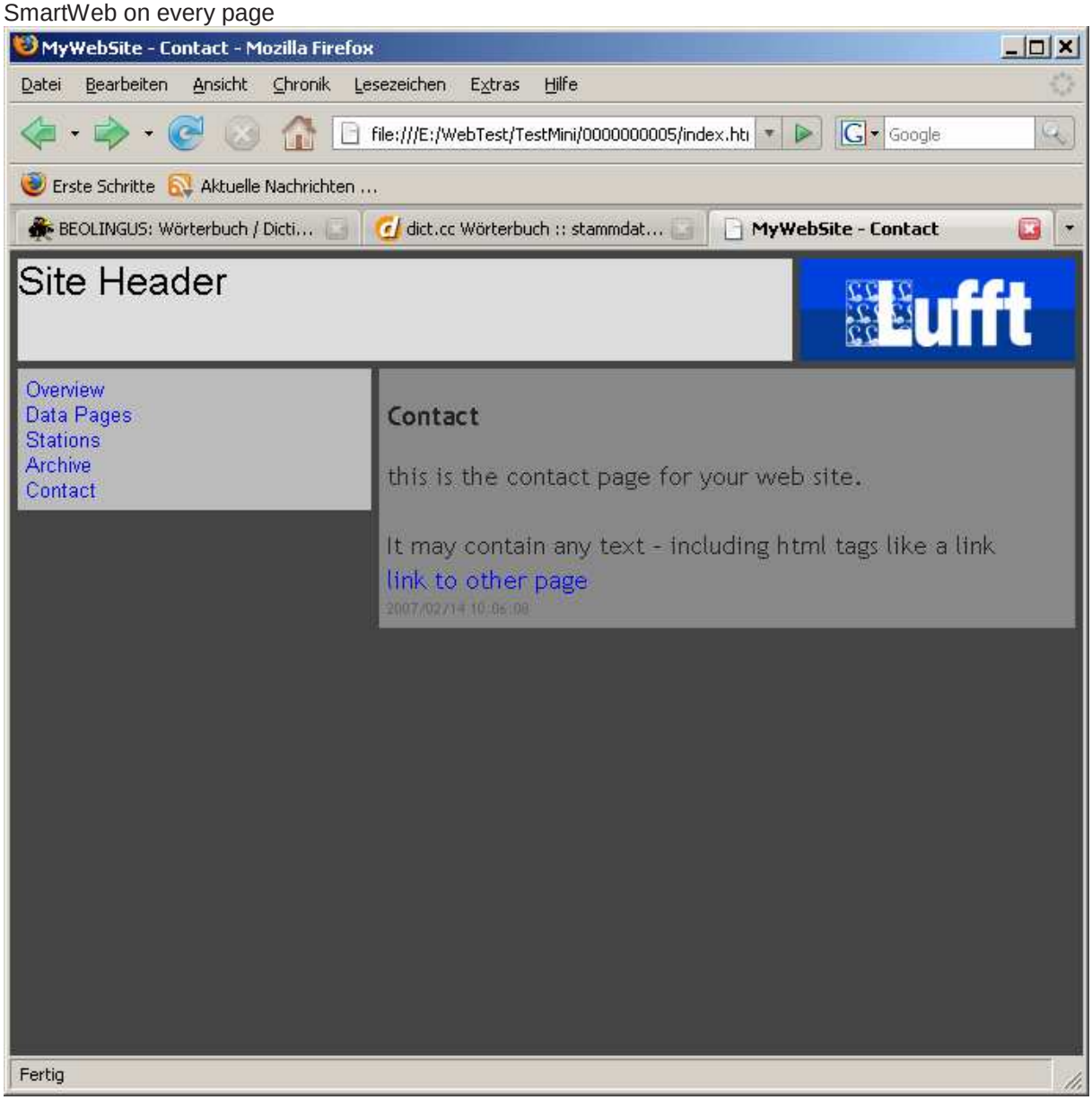
#### *6.1.2.4 archive menu page*

The “archive menu” page contains a list with all “archive list” pages. There is an “archive list” page for every data page configured.
The “archive menu page” is the fourth menu item on the menu created by SmartWeb on the left side of every page.



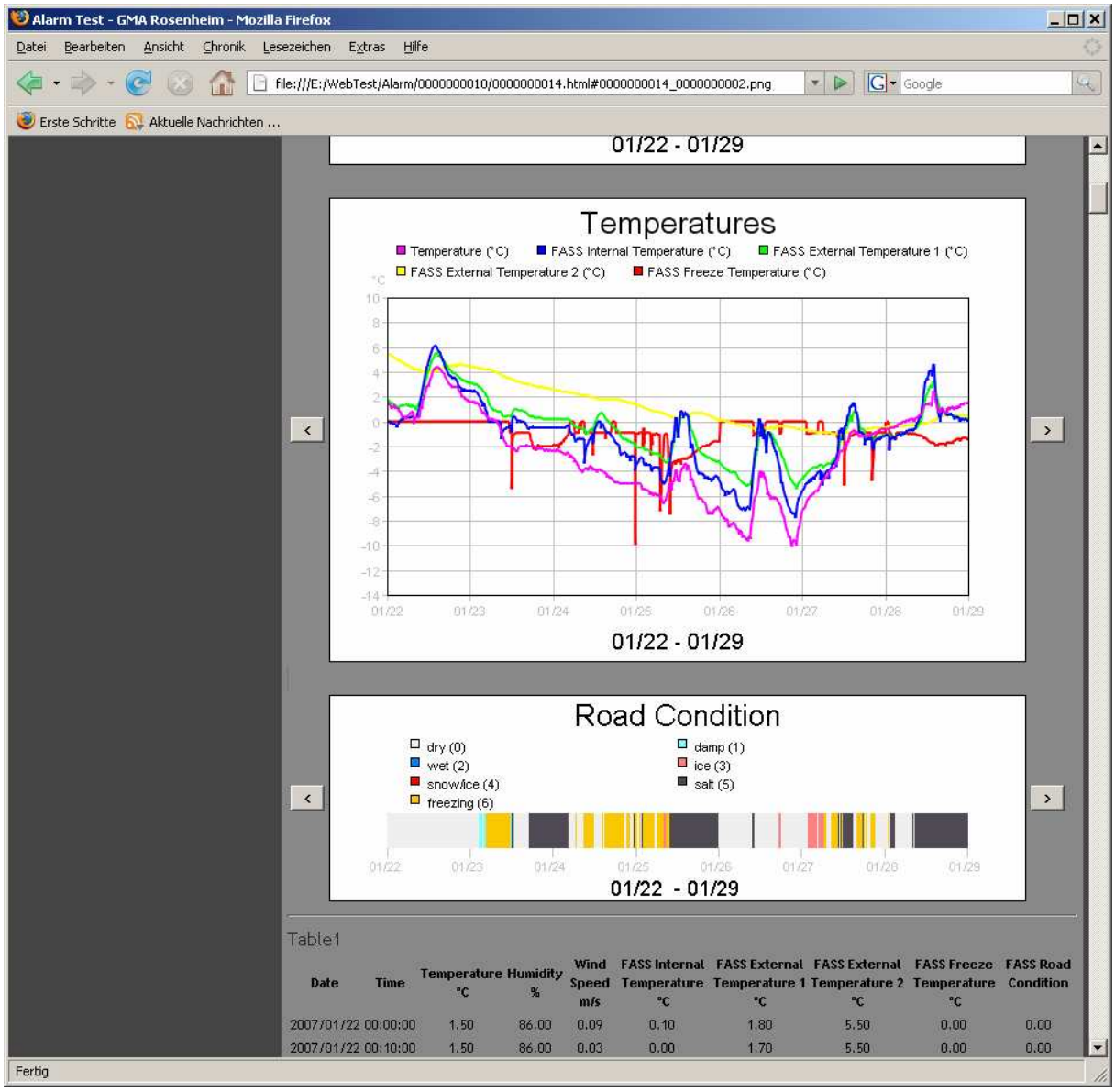
### *6.1.2.5 other page / Contact*

This is the “contact” page (a page of type “other page”). It is the fifth item on the menu created by SmartWeb on every page



### *6.1.2.6 data page*

Data pages are used to display data in diagrams, tables and reports. The sensors that are displayed on a data page can be selected from all stations that are configured for a site.
A Data page always displays data for a specified (fixed) time interval. If this time interval is in the past, the page is referred to as “archive page” (and is no longer updated by SmartWeb).



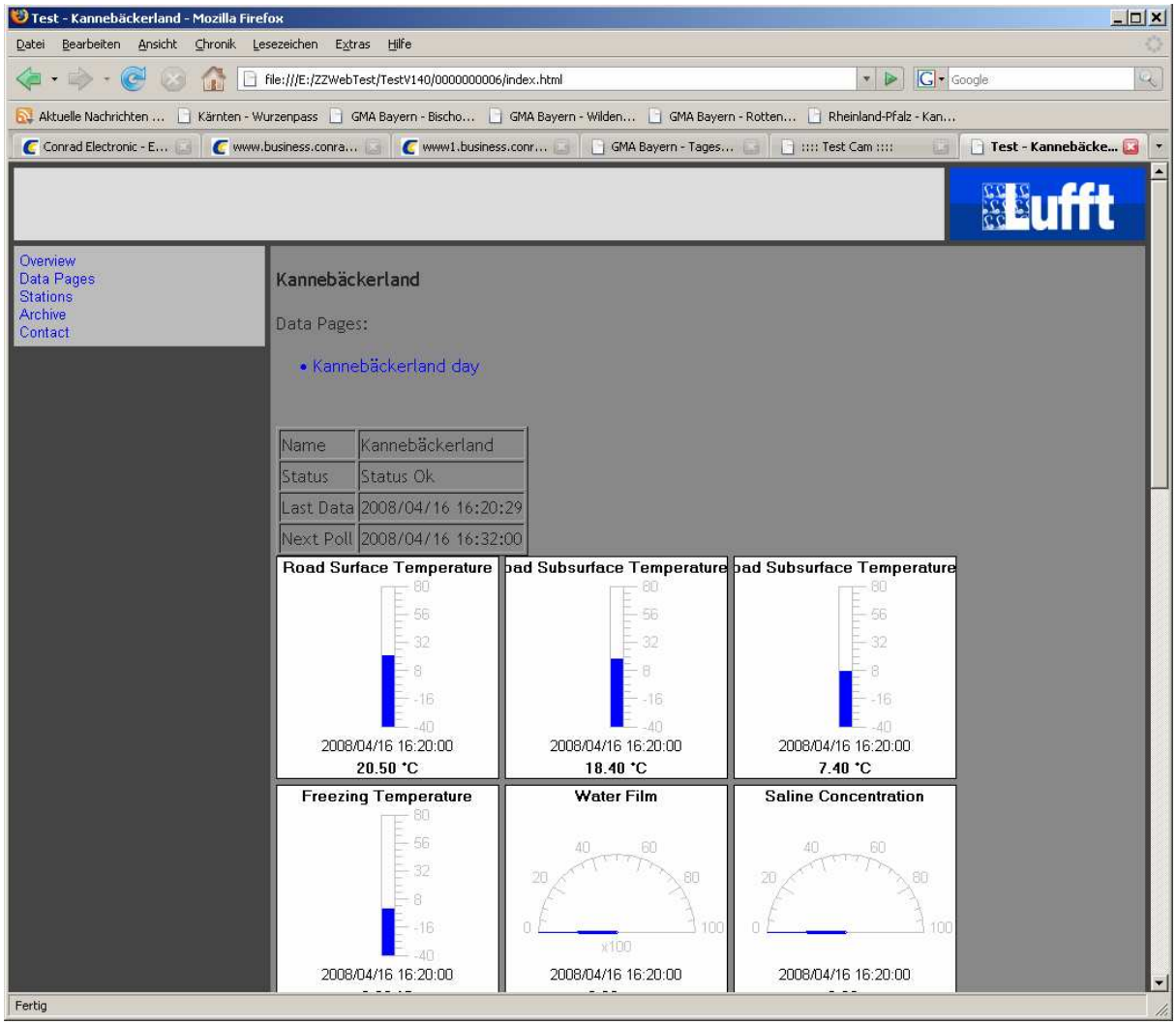
### *6.1.2.7 station page*

There is one “station page” for every station configured to a web site.
On the station page, the last sensor values for selected sensors of a station are displayed. If there is a cam station associated with the station, the cam picture is shown on the station page as well.
In addition, some statistical information about the station is displayed.

Last sensor values can be displayed in tables and/or as “analog gauge” displays for every sensor.

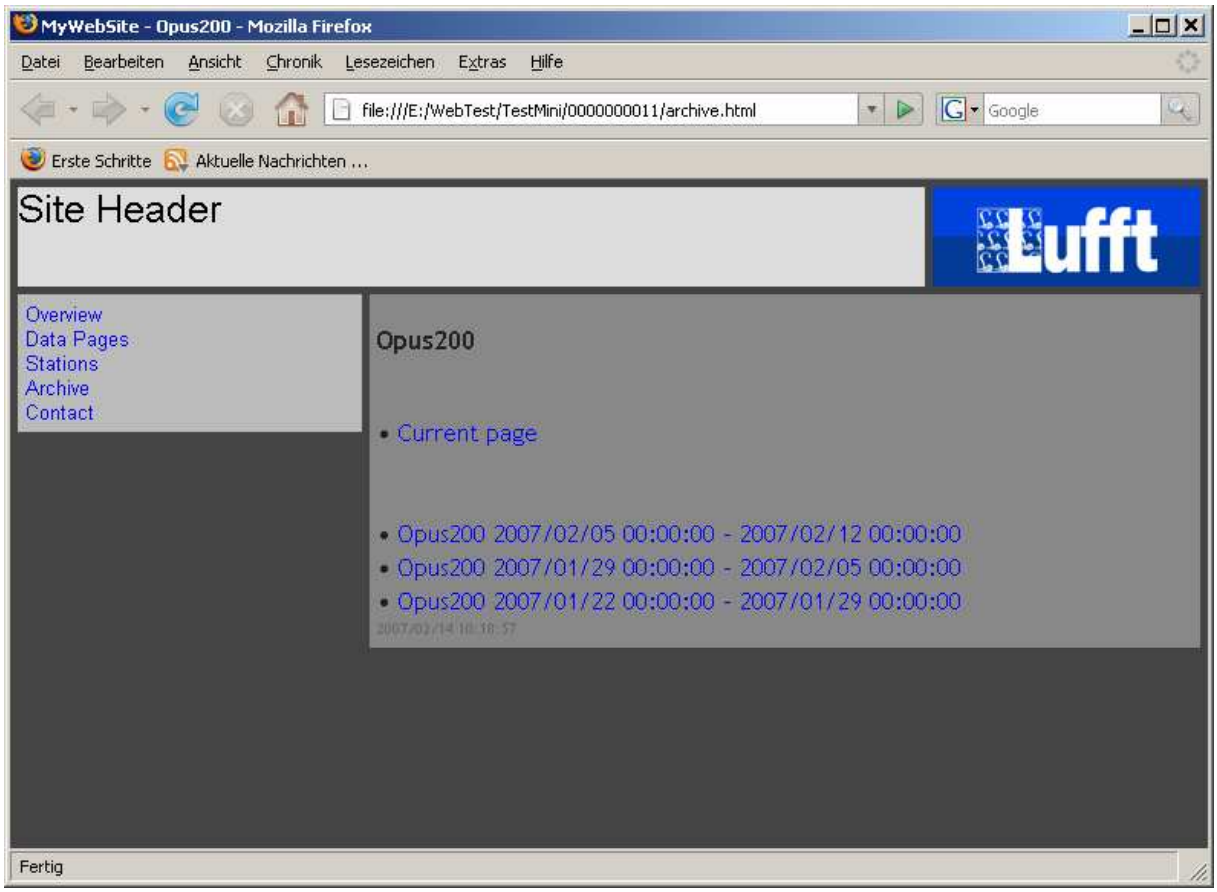
“Reports” that show min/max/average/sum values for selected sensors for specific time intervals can be added to the station page.

Links to “data pages” for the station are provided on top of the page




### *6.1.2.8 archive list page*

For every data page there is one “archive list” page.
An “archive list” page lists all “archive” pages for a data page. As mentioned above, an “archive” page is a data page that is no longer updated, i.e. contains data from the “past”.




#### *6.1.2.9 archive page*

An archive page is an “old” data page, i.e. a data page that is not updated anymore, because its time range is “in the past” (e.g. last week, if the data interval for the data page is “one week”).
Archive pages can be accessed either via the “Archive Menu” / “Archive List” page, or by clicking the appropriate navigation button on the “data page” (which is the “current page” on the archive list page).

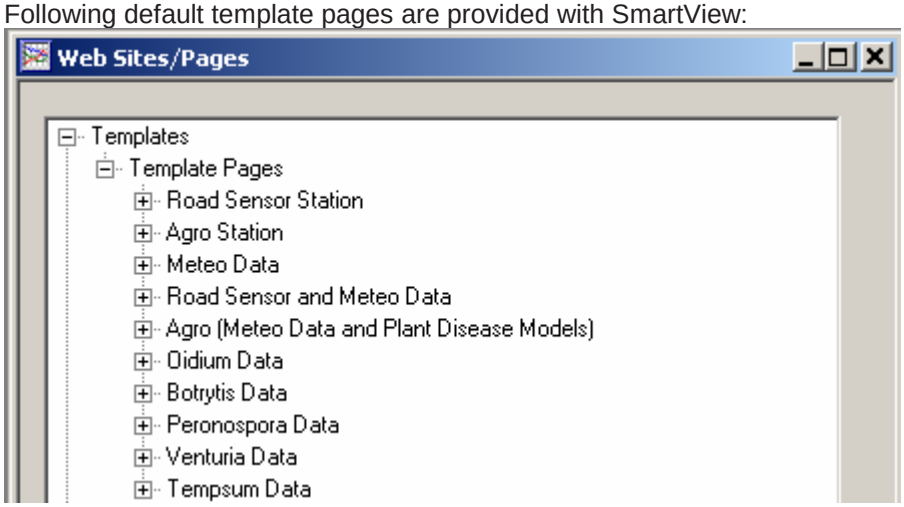
## 6.2 Getting Started

To create a SmartWeb site, select menu item “Sites” from “Settings”. This will open a window titled “Web Sites/Pages”.

If no web-sites are configured, only the “Templates” are displayed,

Template pages are used to provide an easy way to create identical station and data pages for multiple stations.
In Template pages, logical Sensor Types are used instead of “real” sensors. When a template is used to create a station or data page, the “real” station sensors are mapped to the logical sensors used in the template (the same applies to template export/import jobs, see below).

Following default template pages are provided with SmartView:




- **Road Sensor Station**: a “station” page with analog gauges etc. for a typical road sensor station with one road sensor
- **Agro Station:** a “station” page with analog gauges etc for a typical meteo/agro station
- **Meteo Data:** a “data” page with diagrams and tables for a typical “Meteo” data station (air temperature, rel. humidity, dewpoint, air pressure, precipitation amount and type, wind speed and direction, solar radiation, leaf wetness)
- **Road Sensor and Meteo Data:** a “data” page with diagrams and tables for “Meteo” data and data for one road sensor
- **Agro (Meteo Data and Plant Disease Models):** a “data” page with diagrams and tables for “Meteo” data and data for all 5 plant disease models (Oidium, Botryis, Peronospora, Venturia and TempSum) available from Lufft.
- **Oidum/Botrytis/Peronospora/Venturia/Tempsum:** “data” pages with diagrams for the respective plant disease models available from Lufft.

You can alter the existing templates, or create new templates, to your needs.
**Note:** if you have multiple sensors of the same type (e.g. air temperature, visibility, or road sensors) at your stations, you need to set up own logical sensor types (e.g. “air temperature 2”, “visibility 2”, “road condition 2” etc) for the multiple sensors, and use those new logical sensor types in the template diagrams (and assign them to the appropriate sensors in the station configuration) in order for SmartView to pre-select the appropriate sensors when assigning station sensors to templates (see below). If you don’t create new logical sensor types, SmartView cannot pre-select the proper station sensors, and you will have to manually select the sensors when assigning the page template.

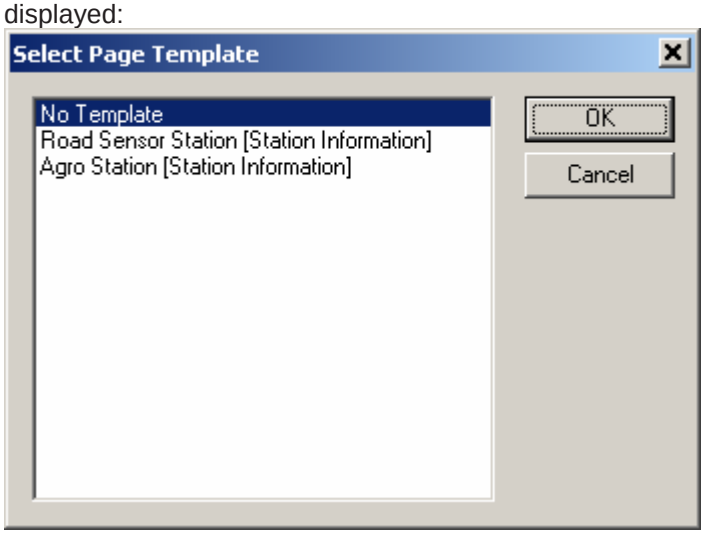
To create a **new web site**, you can either select the option “New” from the “Web-Site” menu in the menu bar, or from the pop-up menu that appears when you right-click at the empty window. This opens the “Site Edit Dialog” (see Edit Site Dialog ).

**Note:** Before you add a station to the web-site, you need to poll the station at least once to read the station's sensor configuration. For stations of type UMB, you also need to activate the specific channels (sensors) you would like to use before you add the station to the web site.

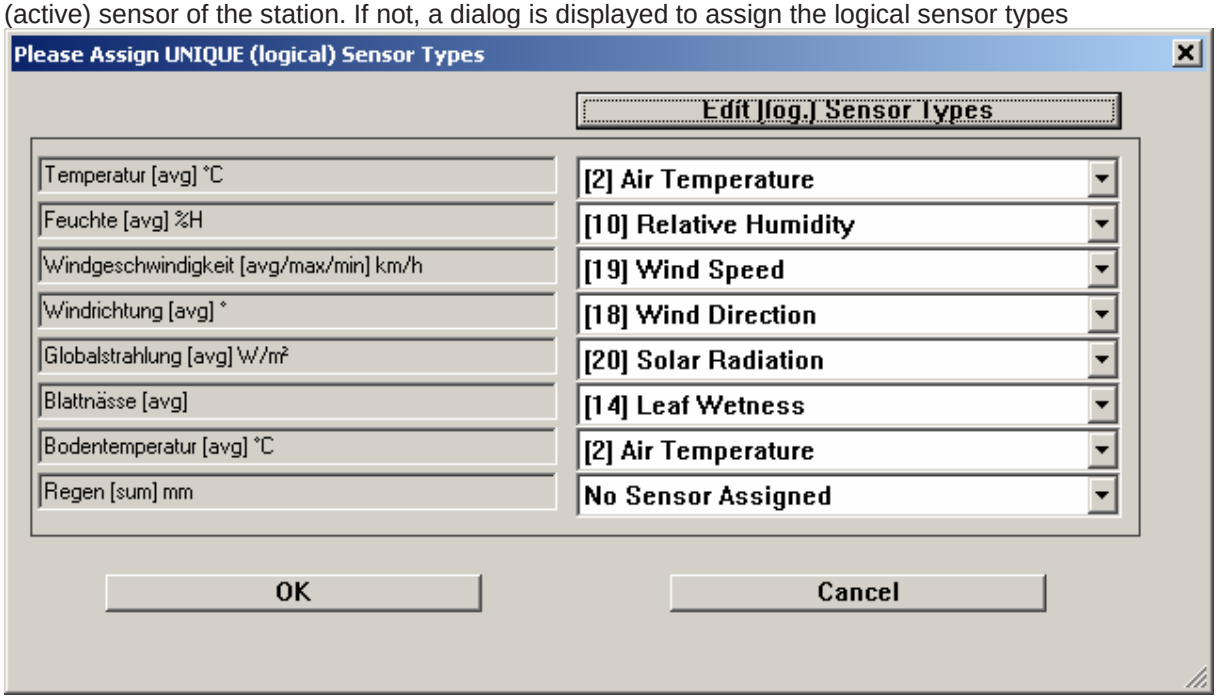
If you plan to use plant disease model calculations for your (agro) station, it is recommended to configure the Export/Import jobs for the plant disease models first (see below), so that the "import" sensors for the plant disease models are available when setting up the station for the web site

Select the stations to be used in this site by clicking the "Edit Stations" button (see Select Site Stations).
Click OK to save the new site.

For each station you add to the web site, a dialog to select a (station) template page for the station is displayed:

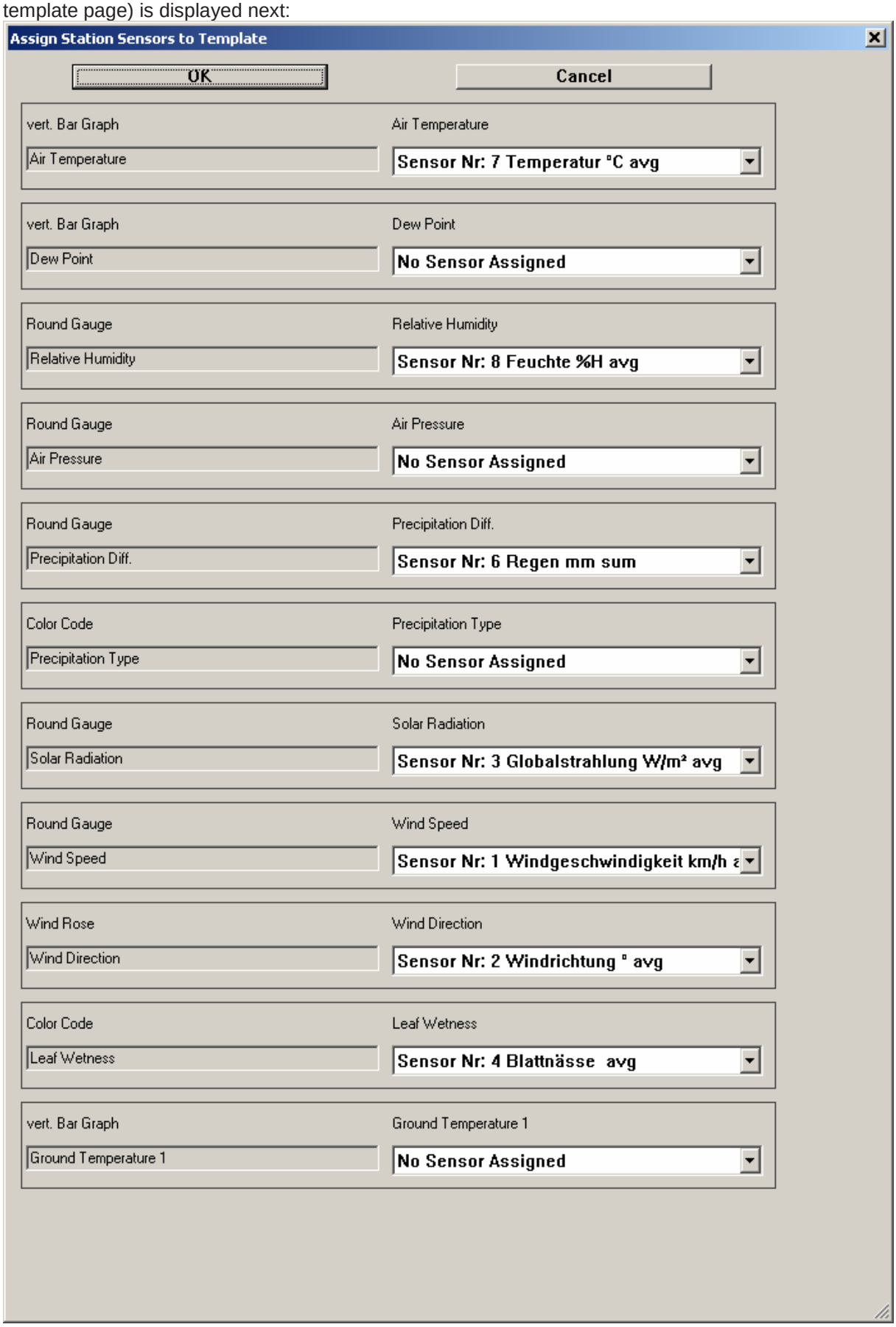



You can set up a station without using a template page.
If you selected a template, SmartView checks if there is a logical sensor type assigned to every (active) sensor of the station. If not, a dialog is displayed to assign the logical sensor types




You can select a logical sensor type from the drop down box. If there is no pre-defined sensor type that fits you sensor, you can create a new sensor type using the button "Edit (log.) Sensor Types.

A Dialog to assign the station sensors to the page template (i.e. to all the page elements of the template page) is displayed next:

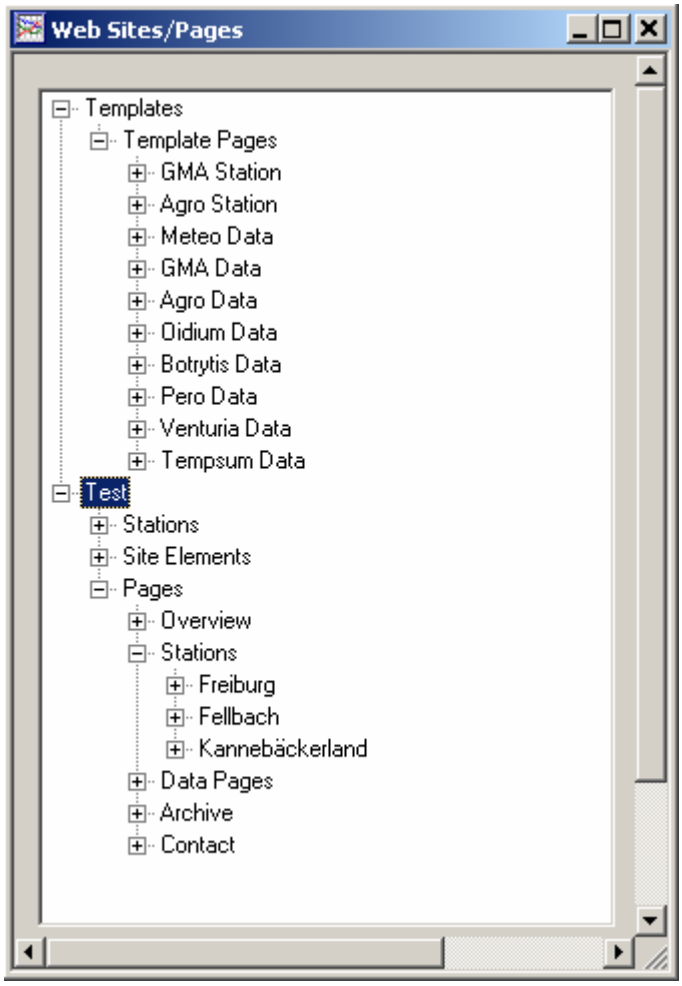



The page elements that don't have a sensor assigned will not be added to the resulting page.

For station pages, a table containing all active sensors of the station is created automatically.

Click “OK” to add the station (and station page) to the web site.

The pages for the new site will be created, and displayed in the Web Site/Pages window.

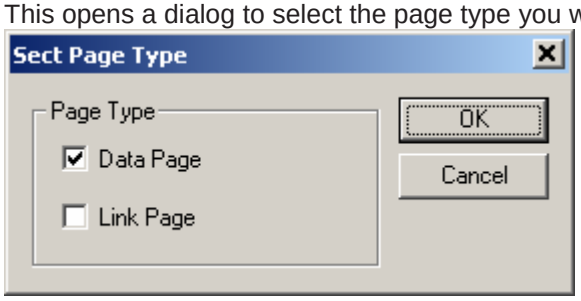



The window displays all elements a SmartWeb web site consists of. See Web-Site Elements below.

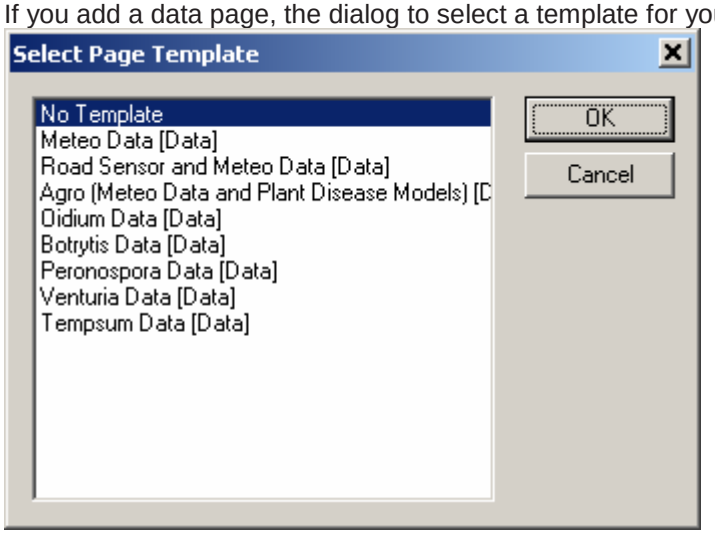
You can expand the display to show subordinate elements by clicking on the “+” signs.

Now you can add “data” pages to display data in diagrams and tables (the station page only shows the last sensor values).

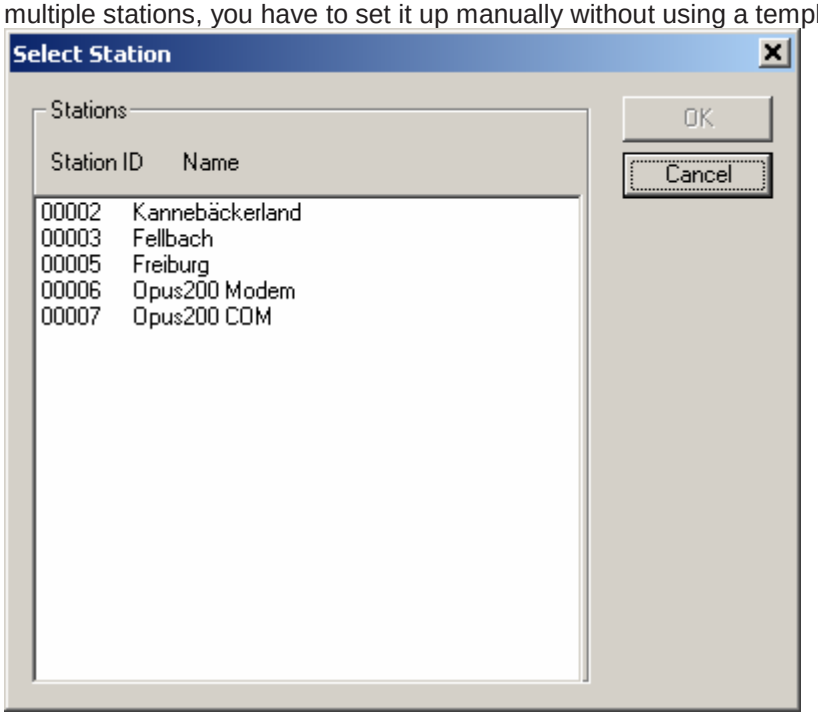
You can either double click on “Pages” (which opens a dialog showing all pages of the site, that has buttons to add and delete pages), or you can select “Pages” or an entry below “Pages” with the left mouse button, and then open the context menu with the right mouse button and select “new”.
This opens a dialog to select the page type you would like to add



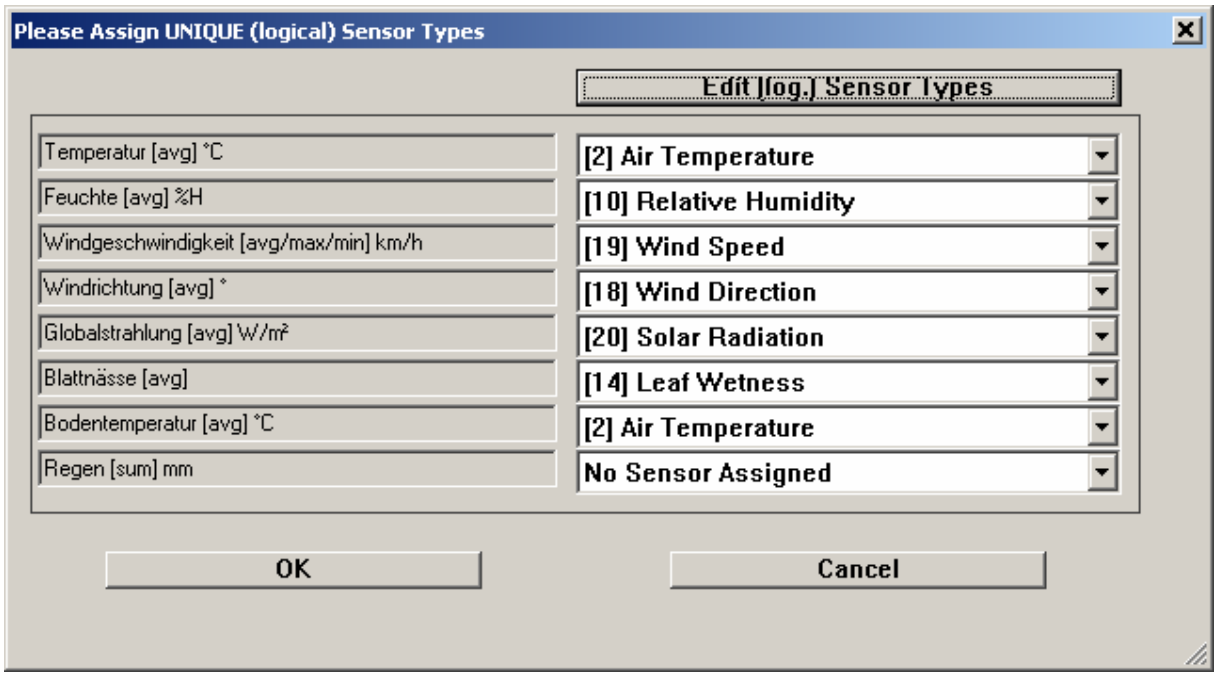
You can either add a data page, or a “link page” (see below).
If you add a data page, the dialog to select a template for your data page is displayed :



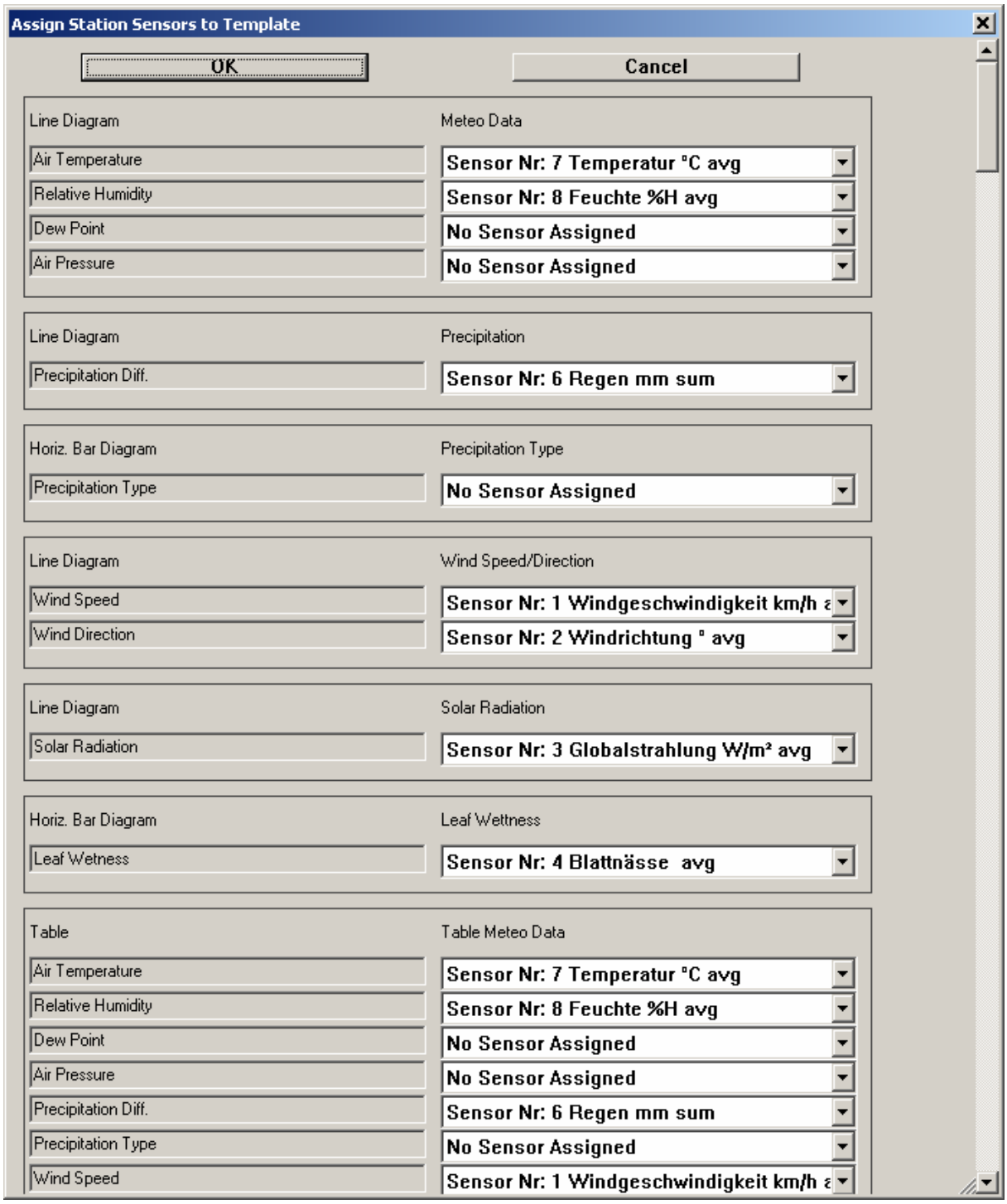

If you selected a template page, a dialog to select the station is displayed. Template pages can only be used to show data from a specific station. If you like to add a data page that contains data from multiple stations, you have to set it up manually without using a template.




Select the station you want to use for the data page.
SmartView checks if there is a logical sensor type assigned to every (active) sensor of the station. If not, a dialog is displayed to assign the logical sensor types.

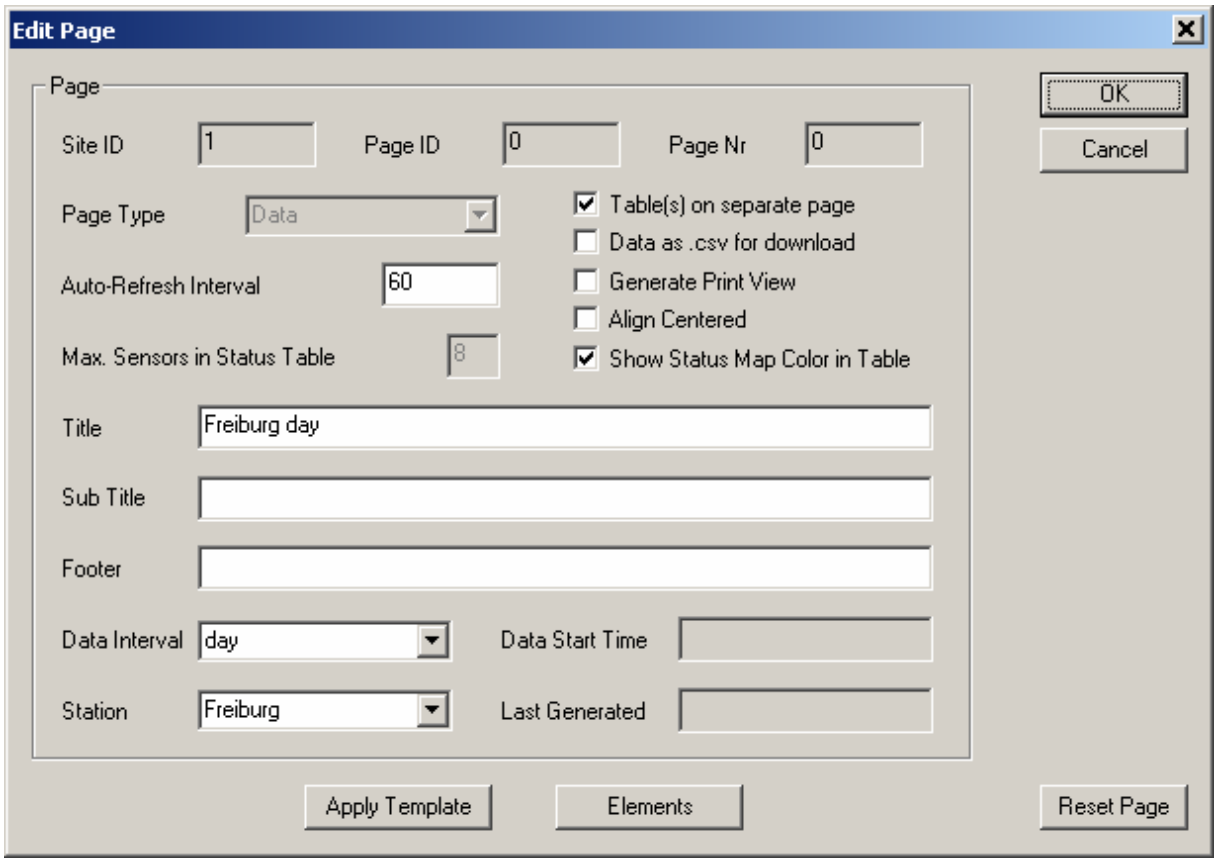



Next, the dialog to assign the station sensors to the page elements (diagrams, tables etc), is displayed. If there are many page elements on the page, you may have to scroll down to see all assignments.




Only page elements that have sensors assigned will be added to the resulting page.

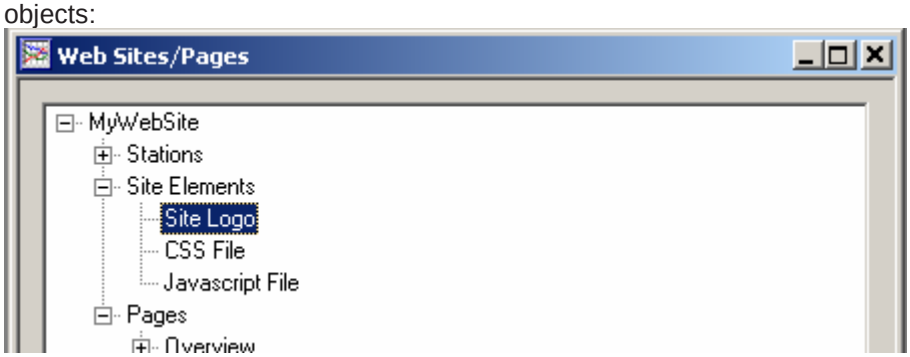
A dialog showing the new page is shown:




You can change the pre-set page title and data interval etc. for the page, and check/modify the page elements etc. before actually saving the page by clicking “OK”. Of course you can change the page at any time, or re-apply a template page later.

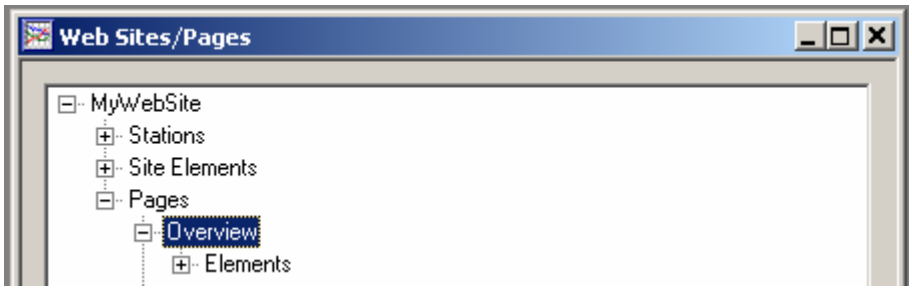
**Customizing the general layout of your web site:**

If you want to use a different site logo, click on the “Site Elements” entry to show the subordinate objects:




Now double click on the “Site Logo” entry to open the “Edit Site Element” dialog for the logo. Now you can change the logo by clicking the “Read from File” button (see Site Elements, Edit Site Element). Please make sure your logo has the same size (175x65 pixel) as the one provided.

If you would like to show your stations on a map, double click the first entry under “Pages” named “Overview”.




Click the “Edit Map” button on the right side of the Edit Page dialog that is displayed (this button is only displayed for the Overview page).
Now the “Edit Page Element” dialog is displayed, where you can load the map to be used by clicking the “Read form File” button. Close the Edit Page Element dialog with “OK”
Now the “Map Editor” - see Edit Map will be displayed, and you can position your stations on the map. All stations are placed in the upper left corner of the map on top of each other if the dialog is opened the first time. Move the stations to the required position on the map by clicking and holding the left mouse button on the station (the rectangle), and release the mouse button.

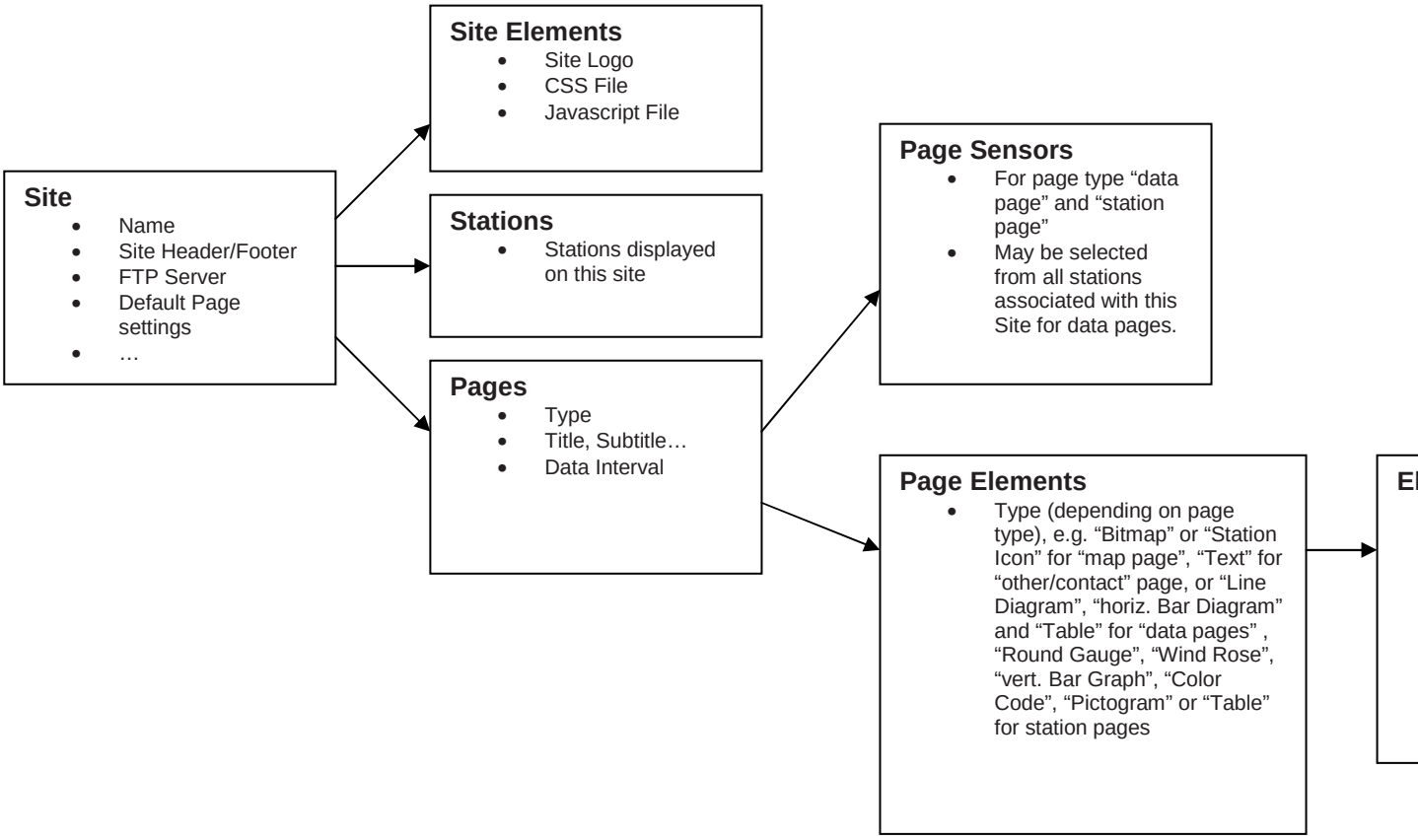
Click “OK” - this will save the page.

If you didn’t start SmartWeb already, start it now (e.g. by selecting “Start SmartWeb” from the “Modules” menu).

**Note:** SmartWeb will now create all pages. It will create data pages for all available data in the database - i.e. if you already have data in the database, it may take some time for SmartWeb to generate all pages and upload them to the configured web server.

## 6.3 Web-Site Elements

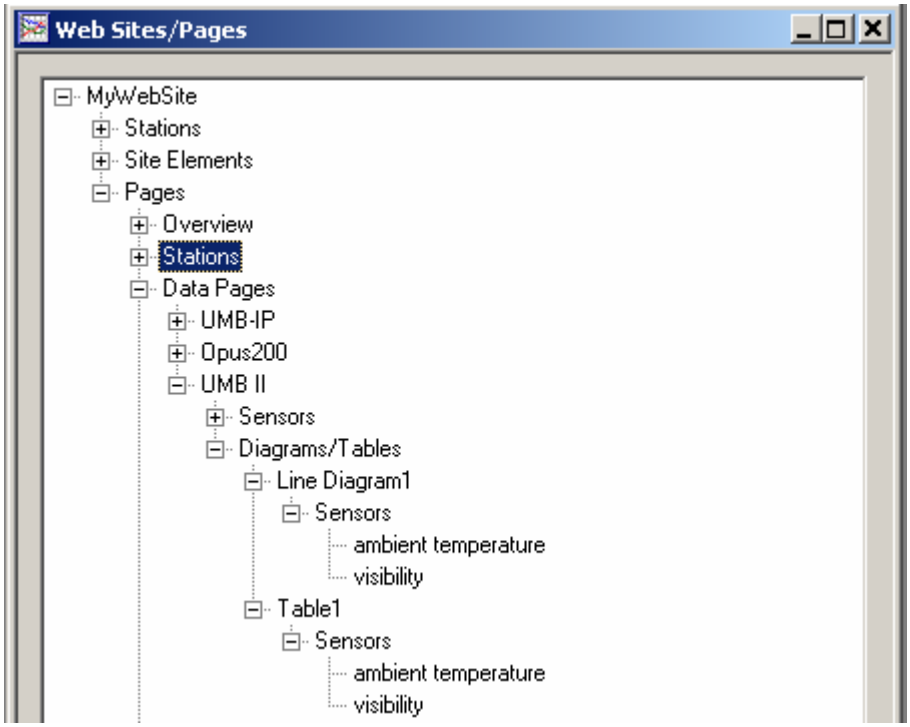
The following diagram shows the elements a SmartWeb web site consists of.



## 6.4 Customizing Pages and Diagrams

You can customize the layout of the diagrams for each data page to meet you needs.
Diagrams and Tables are shown as subordinate elements of data pages in the “Web Site/Pages” display.

A double click on an entry/element in the “Web Site/Pages” display opens a dialog depending on the type of entry you select. Clicking the right mouse button when an entry/element is selected opens a context sensitive menu that allows you to edit, delete, or add a new element of the selected type.




The sensors associated with each diagram or table are shown as subordinate elements of the respective diagram or table:

A double click on a page (all entries subordinate to “Pages/Data Pages” ) opens the “Edit Page” dialog. Here you can configure attributes for the page like the page title and sub tile, or the data interval to be used in diagrams on data pages.

A double click on the “Sensors” entry subordinate to a page opens the “Select Sensor” dialog. If the data page is associated with a specific station, only sensors for this station can be selected. If it is not associated with a specific station, sensors from all stations configured for this web site can be selected.

A double click on the “Diagrams/Tables” entry subordinate to a page opens the “Page Elements” dialog. Here you can edit, delete and add new diagrams or tables, or re-arrange the sequence of page elements (diagrams or tables).

A double click on a diagram or table opens the “Edit Page Element” dialog. Depending on the page element type (e.g. “Line Diagram”, “horiz. Bar Diagram” or “Table” for data pages), the attributes for the page element like name, line diagram axis scaling etc. can be configured.

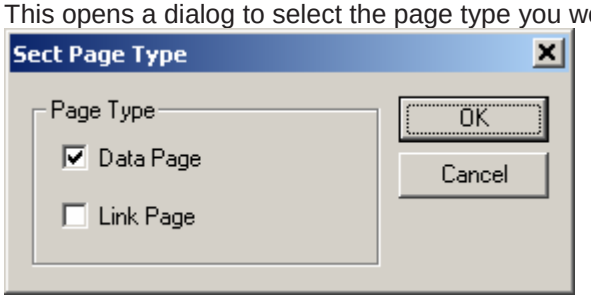
A double click on the “sensors” entry of a page element opens the “Select Sensor” dialog for the page element. The sensor(s) assigned to the page element can be edited. If a page element can contain more than one sensor, like tables or line diagrams, the sequence of the sensors can be changed here.

A double click on a specific sensor of a page element opens the "Edit Page Element Sensor" dialog for this sensor. Depending on the page element type, the color of a sensor (for line diagrams) or the status mapping (for horiz. Bar Diagrams, vert. Bar Graph, Color Code, Table or Pictogram) can be modified.

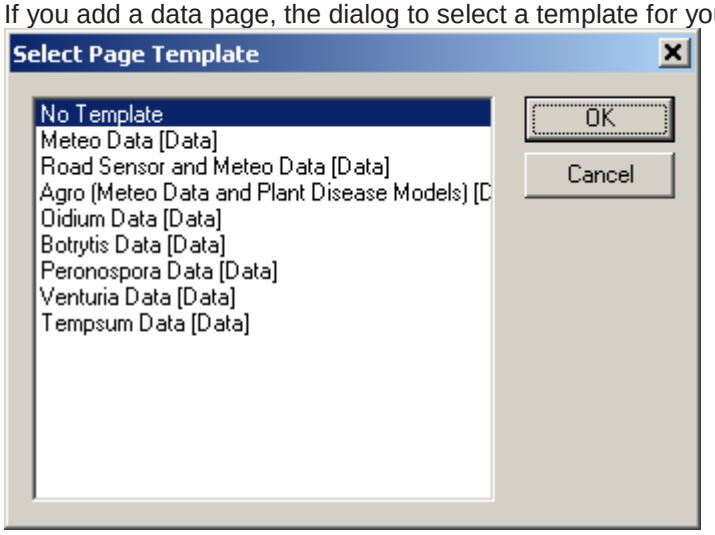
## 6.5 Adding new Data Page

To add a new "data page" to a site, you can either double click on the "Pages" element of the web site, which will open the "Pages" dialog where you can add a page by clicking the "New" button, or by selecting the "Pages" entry (or a page entry subordinate to the "Pages" entry) in the "Web Sites/Pages" display and clicking the right mouse button – which will pop up a menu where you can select "New".

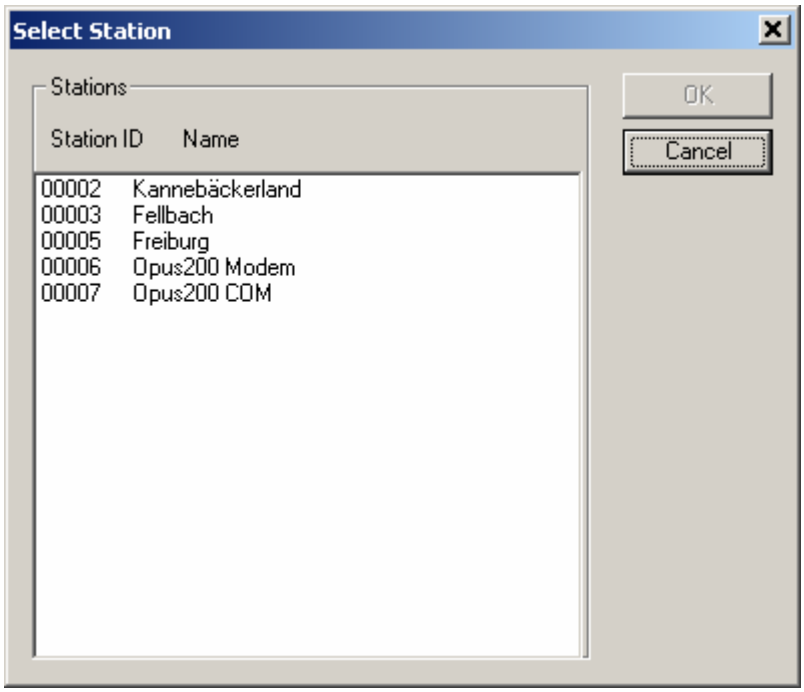
This opens a dialog to select the page type you would like to add



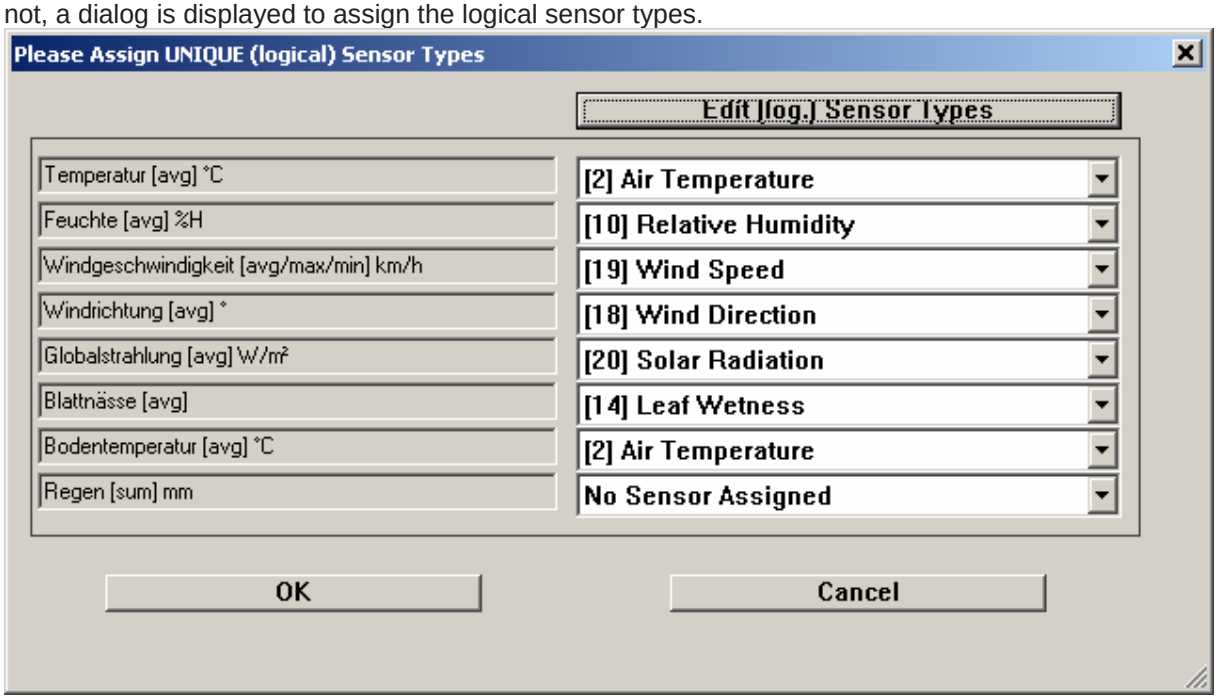

You can either add a data page, or a "link page" (see below).
If you add a data page, the dialog to select a template for your data page is displayed :



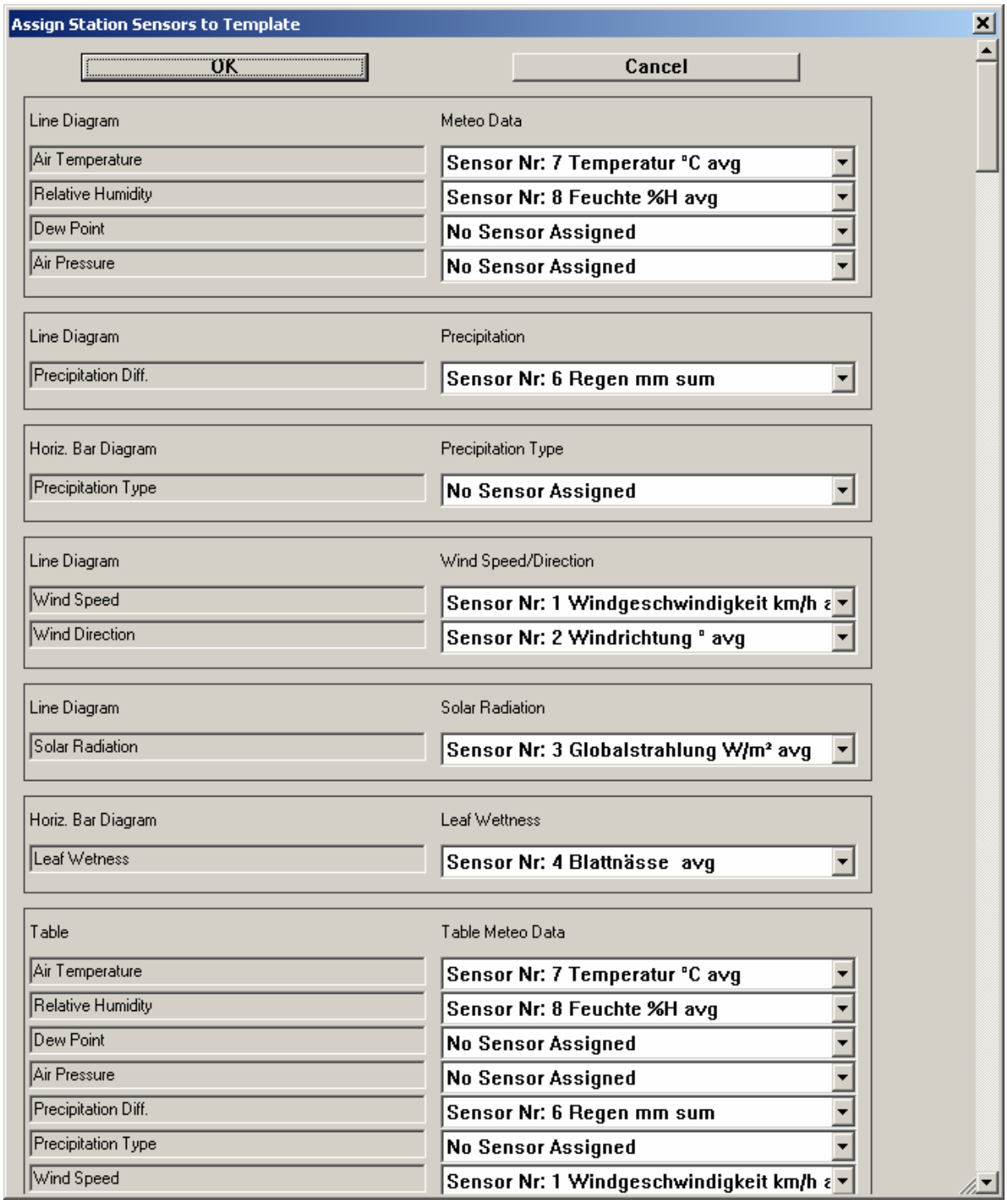

If you selected a template page, a dialog to select the station is displayed. Template pages can only be used to show data from a specific station. If you like to add a data page that contains data from multiple stations, you have to set it up manually without using a template.




Select the station you want to use for the data page.
SmartView checks if there is a logical sensor type assigned to every (active) sensor of the station. If not, a dialog is displayed to assign the logical sensor types.

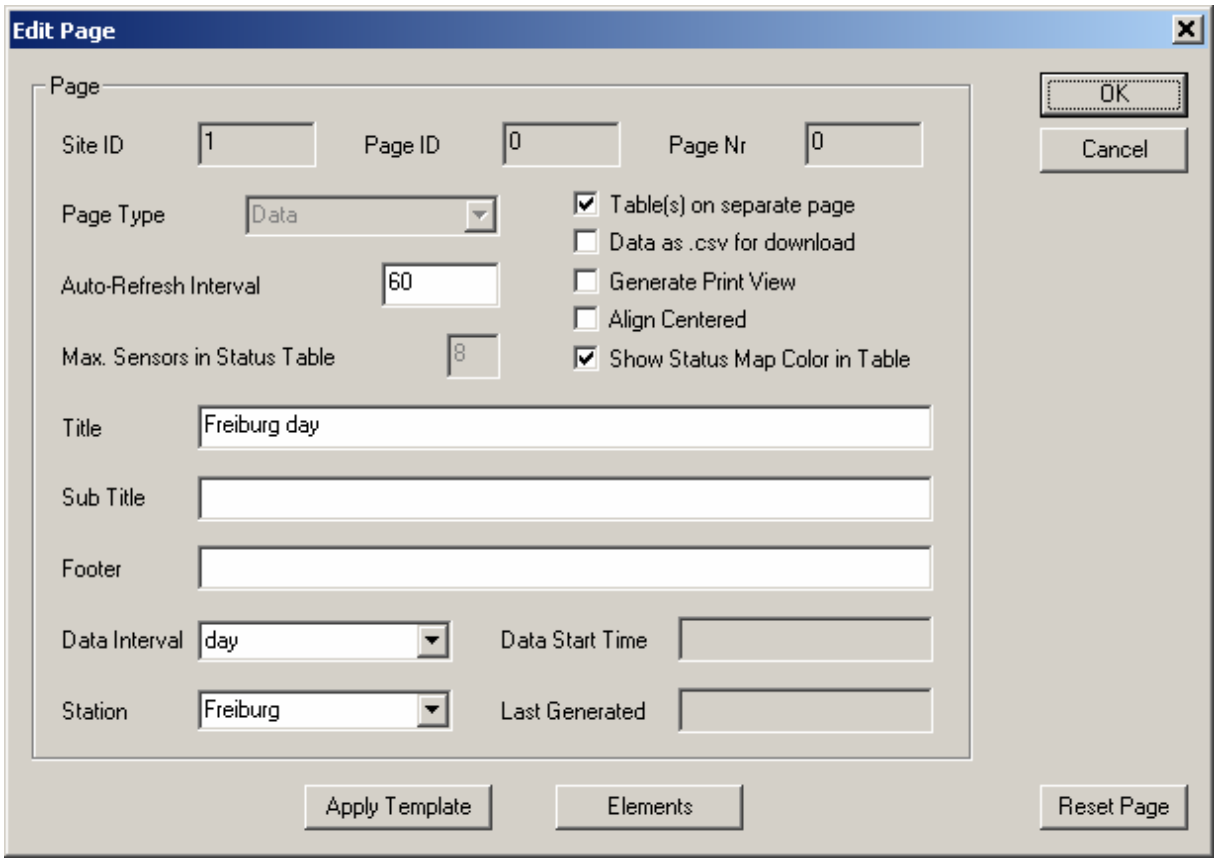



Next, the dialog to assign the station sensors to the page elements (diagrams, tables etc), is displayed. If there are many page elements on the page, you may have to scroll down to see all assignments.




Only page elements that have sensors assigned will be added to the resulting page.

A dialog showing the new page is shown:



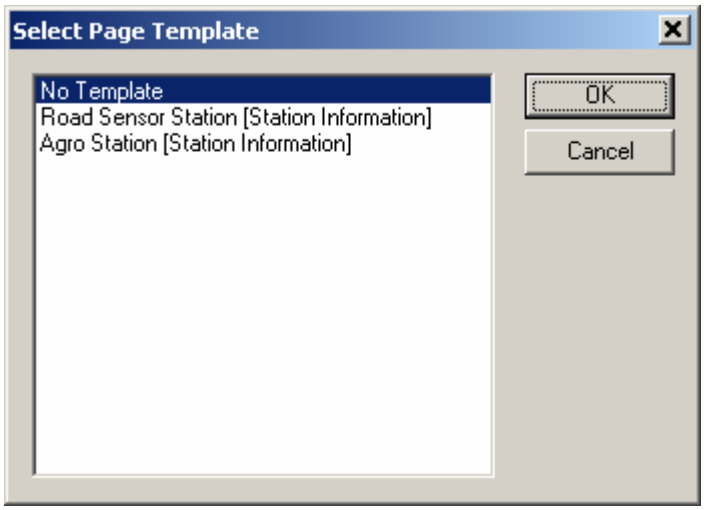

You can change the pre-set page title and data interval etc. for the page, and check/modify the page elements etc. before actually saving the page by clicking “OK”. Of course you can change the page at any time, or re-apply a template page later.

## 6.6 Adding Stations

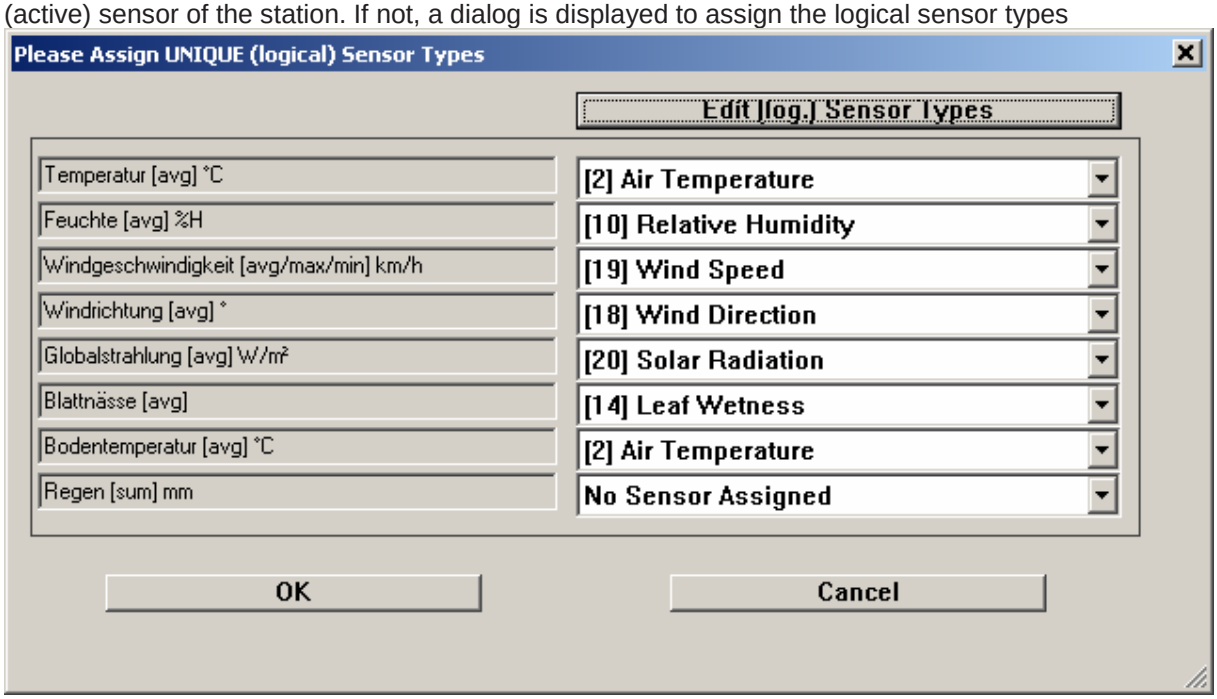
To add a new station to a SmartWeb site

- ➔ Add the new station to the Collector configuration –> SmartView3 menu Edit -> Stations
- ➔ Have the station polled at least once by Collector, so the sensor configuration of this station is read from the station
- ➔ Select “Web-Sites” from the “Edit” menu in SmartView3
- ➔ Double click the web site entry
- ➔ Click “Edit Stations” and add the station to the site

For each station you add to the web site, a dialog to select a (station) template page for the station is displayed:

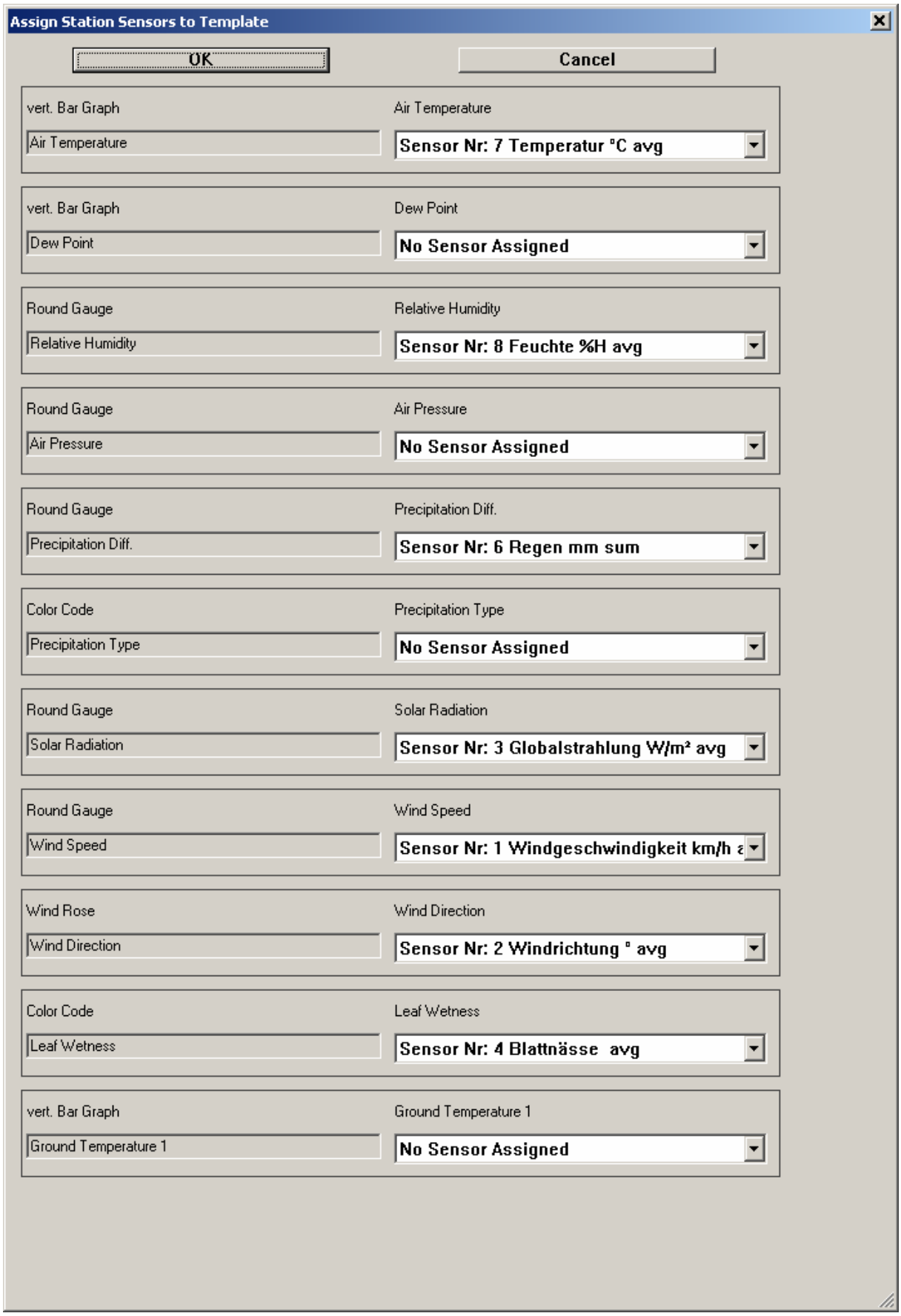



You can set up a station without using a template page.
If you selected a template, SmartView checks if there is a logical sensor type assigned to every (active) sensor of the station. If not, a dialog is displayed to assign the logical sensor types




You can select a logical sensor type from the drop down box. If there is no pre-defined sensor type that fits you sensor, you can create a new sensor type using the button “Edit (log.) Sensor Types.

A Dialog to assign the station sensors to the page template (i.e. to all the page elements of the template page) is displayed next:



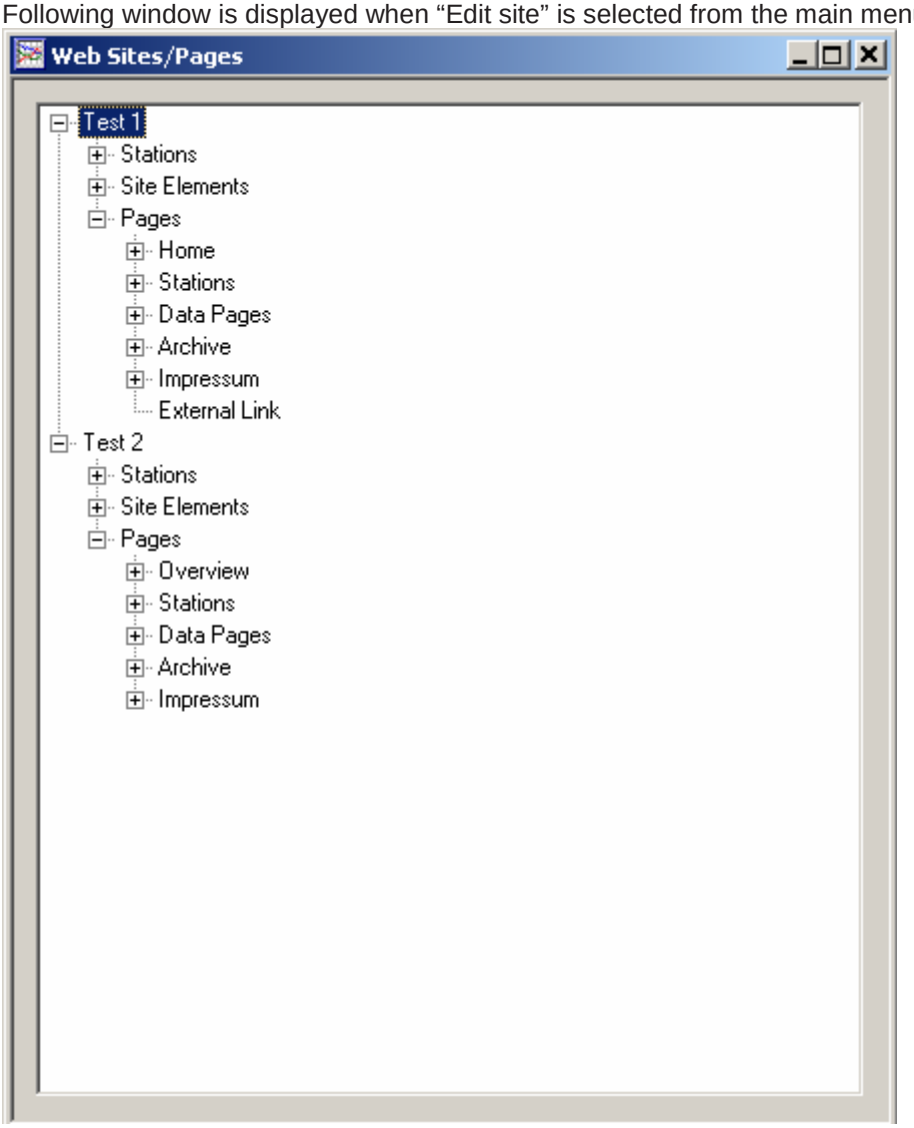

The page elements that don't have a sensor assigned will not be added to the resulting page. For station pages, a table containing all active sensors of the station is created automatically.

Click “OK” to add the station (and station page) to the web site.

➔ Customize the new data page to your needs.

## 6.7 Site configuration dialogs

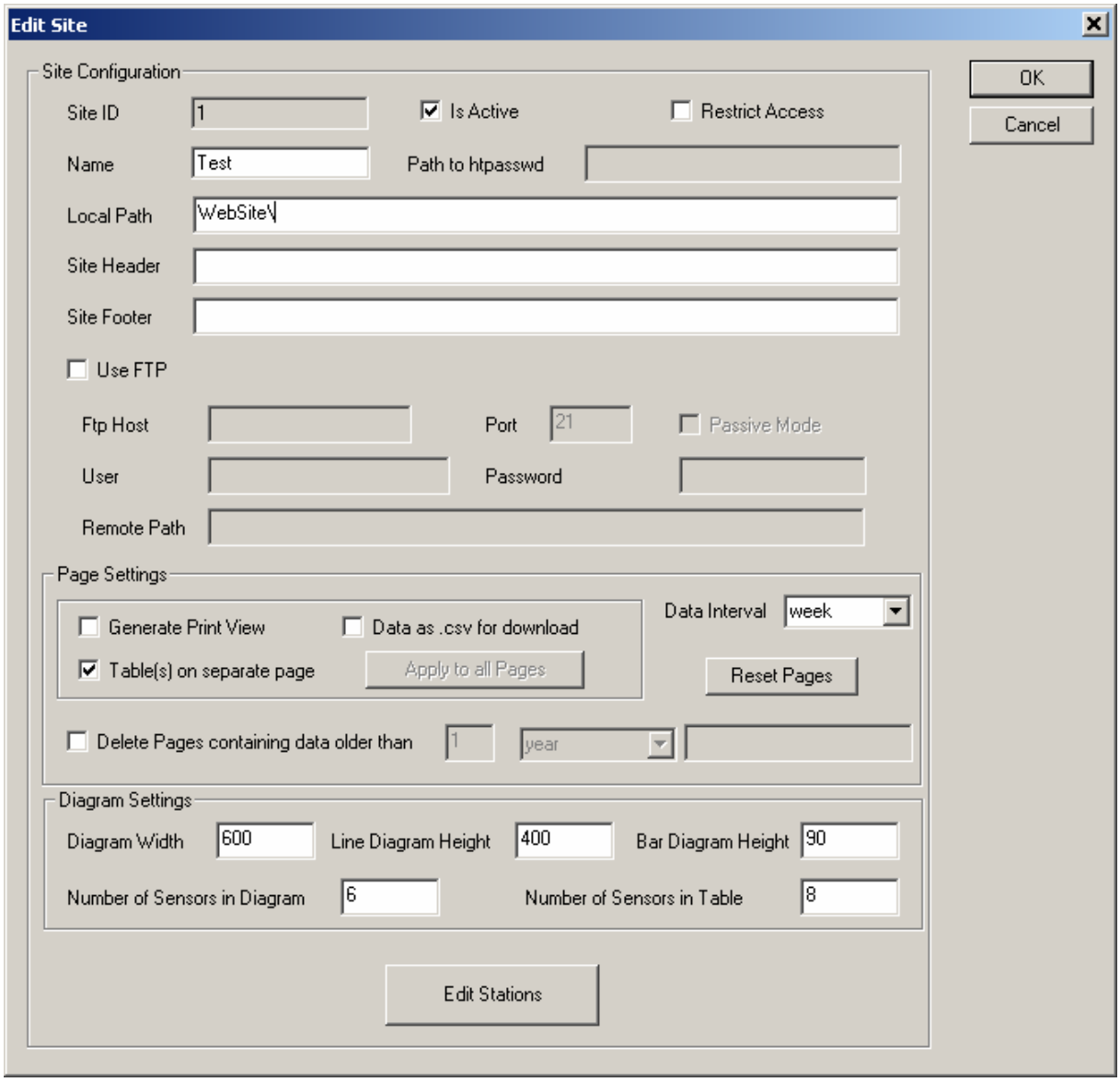
Following window is displayed when “Edit site” is selected from the main menu :




The main menu will now contain a menu “Web-Sites”, from which actions like “edit” “delete” or “new” can be selected. Depending on the element that is selected in the window, an appropriate dialog will be displayed. The same functions (edit/delete/new) can be selected from a pop-up menu that appears if a “left click” with the mouse is done on the selected element.

The function “Check Site Sensor Configuration” will go through all configured stations and web sites, and remove inactive (or not longer configured) sensors from the web site configuration (i.e. the tables and diagrams on station and data pages, and the “station icon” on the overview page). Note that this could lead to tables and/or diagrams without any configured sensors.
It will also add active (new) sensors to the respective pages – but will not add the new sensors to tables or diagrams. However, as the new sensors are added to the respective pages, they are available to configure new diagrams.

### 6.7.1 Edit Site Dialog




**Input Fields:**

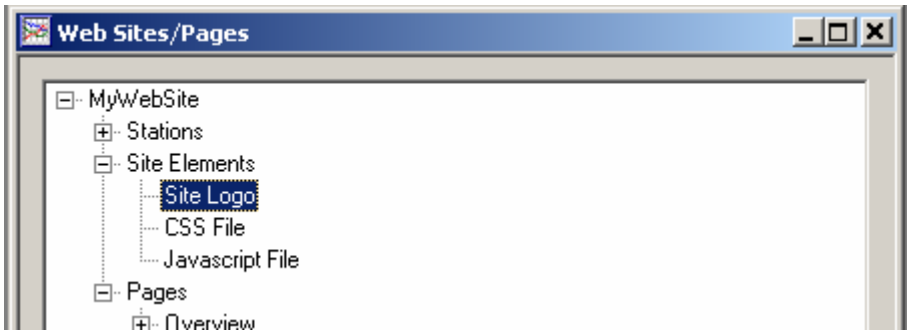
- **Site ID**: the site id for this site (read only)
- **Is Active**: the site is active (will be generated/updated) or is not active
- **Restrict Access**: if this box is checked, .htaccess and .htpassword files will be created to restrict access to the pages of the site.
  **NOTE**: This only works for Apache web server!!
- **Path to htpasswd**: (only if “Restrict Access” is checked) the **complete/absolute** path to the htpasswd file (without the filename!). If ftp transfer to the web server is used, the complete/absolute path to the directory where this ftp user accesses the server needs to be specified here. If the web server is on the local PC, this should be set to the local path.
- **Start Page ID**: The page id of the page that is to be displayed as start page of the site (default 1 = map page)
- **Local Path**: the local path where all generated pages are written to. All files are written to the local hard drive using this path. If Use FTP is specified, the files are transferred using ftp as well
- **Remote Path**: the path (below the directory where the ftp user is logged on to the server) on the server where the site files are to be placed
- **Site Header**: The site header. This header will appear on top of all pages of the site
- **Site Footer**: The site footer. This footer will appear at the bottom of all pages of the site

- **Use FTP**: if checked, the generated files are uploaded to the specified server.
  NOTE: as the image files (for diagrams) and html files with the measurement data in tables can get quite big, a fast connection to the web server is recommended
- **FTP Host**: the host name or ip address for the ftp transfer
- **FTP Port**: the port for the ftp connection to the server
- **Passive Mode**: use ftp passive mode to connect to server
- **FTP User**: the ftp user for the ftp transfer
- **FTP Password** : the ftp password for the ftp transfer
- **Generate Print View** : default for new pages - include a “print view” of the page
- **Data as .csv for download**: default for new data pages - include data as .csv file for download
- **Table(s) on separate page**: default for new data pages - place tables on a separate page
- **Data interval**: default data interval for new data pages (not used with template pages).
- **Delete Pages containing data older than**: automatically delete data pages that show data which is older than the configured time period.
- **Diagram Width**: the width for all diagrams on all data pages for this site (not used with template pages)
- **Line Diagram Height**: the height for all line diagrams on this site (not used with template pages)
- **Bar Diagram Height**: the height for all bar diagrams on this site (not used with template pages)
- **Number of Sensors in Diagram**: maximum number of sensors in a line diagram
- **Number of Sensors in Table**: maximum number of sensors in a table

**Buttons:**

- **Edit Stations**: this opens the “Select Site Stations” dialog, where you can specify which stations are displayed on this site. See Select Site Stations
- **Apply to all Pages:** applies “Generate Print View”, “Data as .csv for download” and “Table(s) on separate page” to all pages of this site
- **Reset Pages:** re-sets all pages of this site - this will cause SmartWeb to re-generate all pages for the site

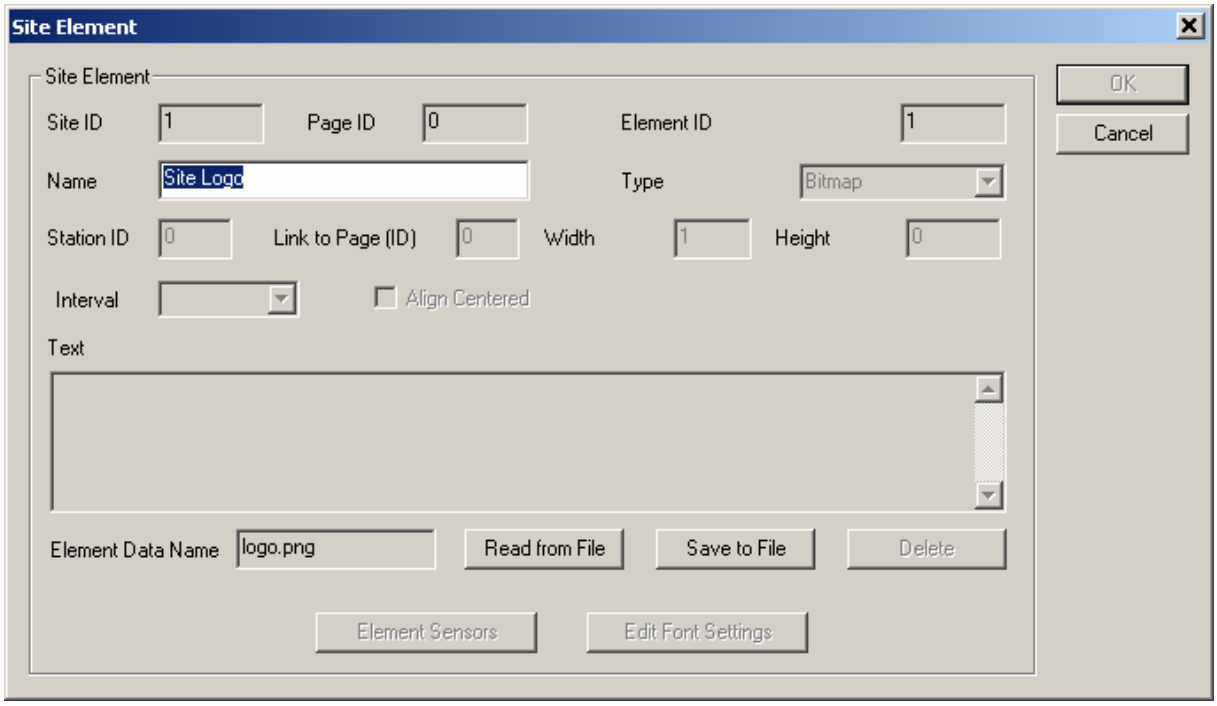
### 6.7.2 Site Elements




A site always has 3 “Site Elements” - a Site Logo , a CSS File and a Javascript File. These can be edited by double-clicking on the entry in the list or by selecting the entry and clicking the “edit” button.

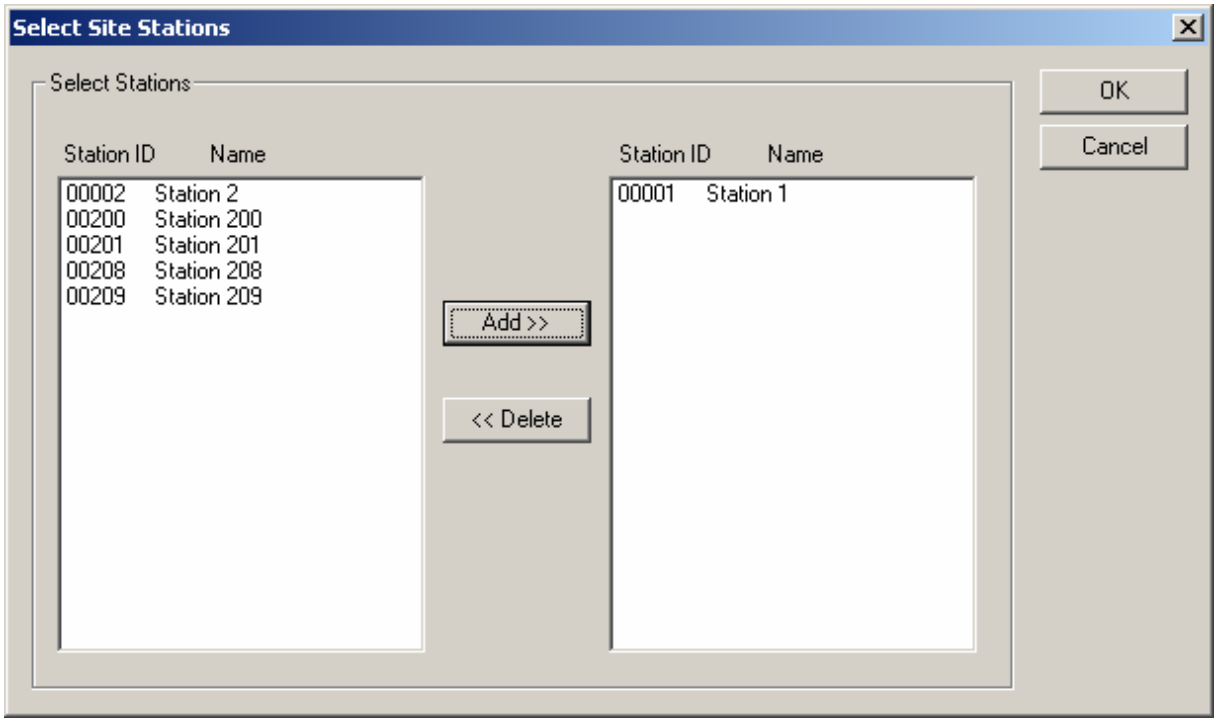
See chapter Styles.css for a description of the .css file

### 6.7.3 Edit Site Element



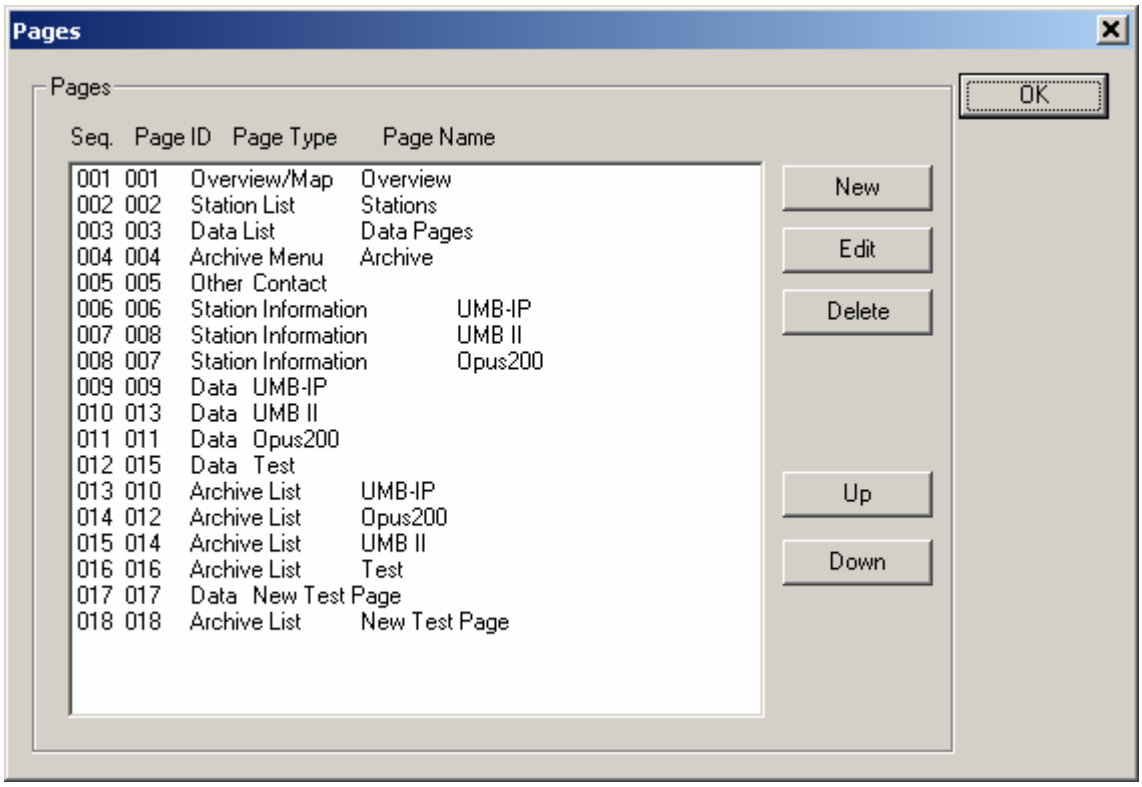

The Site Elements “Logo”, “CSS File” or “Javascript File” can be edited here - i.e. they can be read from a file on the hard drive, or written to a file on the hard drive to be edited and then again read from the file and stored in the database for use by SmartWeb.

### 6.7.4 Select Site Stations




This dialog is used to select the stations to be used on a site. To select a station, mark it in the left list box and click “add”. To remove a station mark it on the right list box and click “delete”.

### 6.7.5 Pages




This dialog shows all pages for the site.
**Note** : only “data” pages and “external link” pages can be deleted or added to a site- all other pages are pages that are always present on a site.
Deleting a data page will delete the associated “archive list” page.

The “Up” and “Down” buttons can be used to change the sequence of the pages, which changes the sequence of entries in the “Station List”, “Data List” and “Archive Menu” page.

Following page types exist:

#### *6.7.5.1 Map Page*

The map page is the default start page for a site. It shows (if a map has been loaded to the page) the stations as icons on a map, and the status of the station (status of last transfer). If no Map has been loaded, the stations and their status is listed.

A Map Page has following “Page Elements”:
One element of type Bitmap which contains the image of the map to be used
One element of type “Station Icon” for each station that is configured to this site

The page elements are maintained automatically, and can not be deleted - but can be edited.
See Edit Page, Edit Map, Page Elements.

#### *6.7.5.2 Station List Page*

This “menu page” contains a list of all stations configured for this site. This page does not have page elements that can be edited.

#### *6.7.5.3 Data List Page*

This “menu page” contains a list of all data pages configured for this site. This page does not have page elements that can be edited.

#### *6.7.5.4 Archive Menu Page*

This “menu page” contains a list of all “Archive List” pages. This page does not have page

#### *6.7.5.5 Other Page / Contact*

This is the “contact” page for the site. It contains one Page Element of type “Text”. This text is placed “as is” on the page and may contain any valid html tags.

#### *6.7.5.6 Station Page*

This page shows information about a station, the sensors of the station, and the last measurement values for each sensor.
It may contain “page elements” for

- station information
- one (or more) tables with sensors to display the last sensor value for a sensor
- “Analog Gauge” displays for sensor values to display the last sensor value graphically, e.g. as “Round Gauge”, “Wind Rose”, “vert. Bar Graph”, “Color Code” or “Pictogram”.

#### *6.7.5.7 Data Page*

A data page shows measurement values of sensors configured for this page in diagrams and tables. Page Elements are Line Diagrams, Bar Diagrams and Tables. Sensors can be configured from all stations that are configured for the site. Diagrams and Tables can be configured from all sensors that are configured to the page.

#### *6.7.5.8 Archive List Page*

This “menu” page lists all “Archive Pages” for the site. There is an Archive List page for every Data Page.

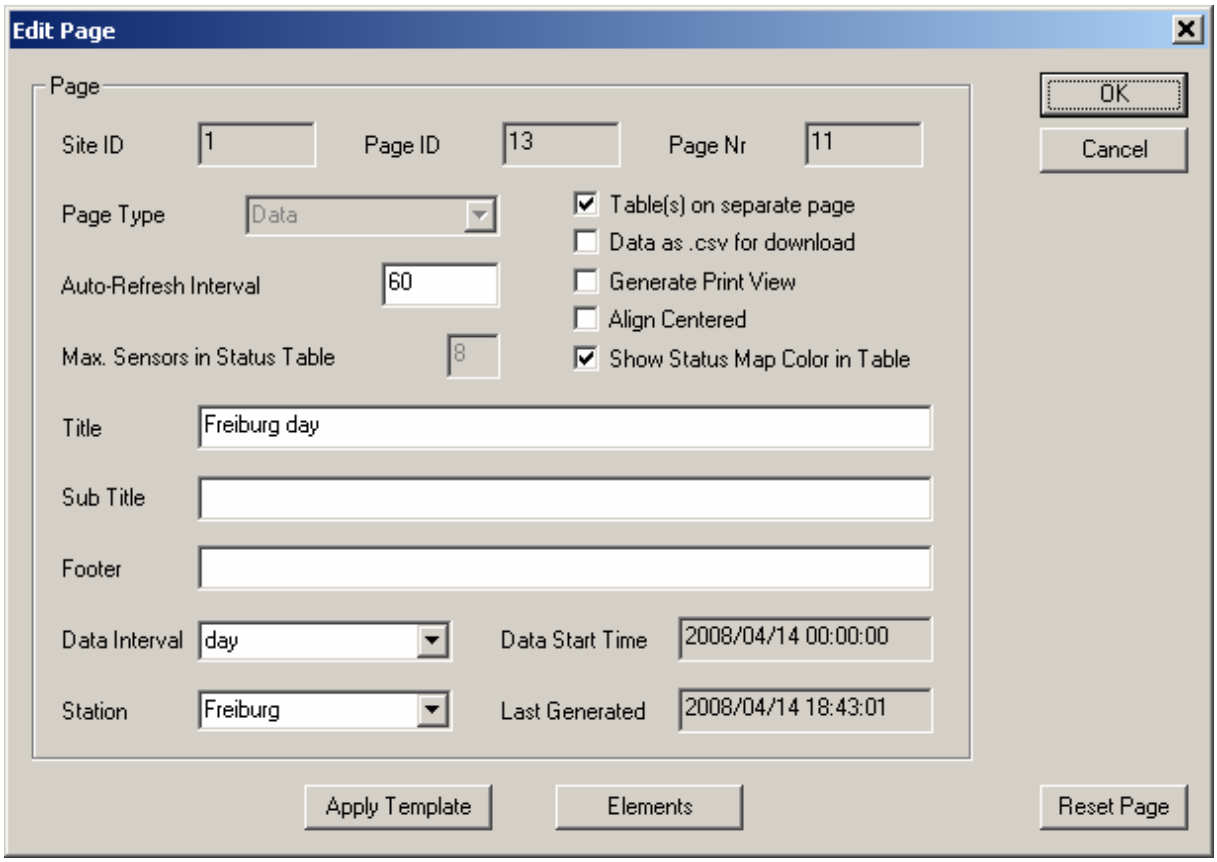
#### *6.7.5.9 Archive Page*

An Archive Page is a “Data page” that has been closed, i.e. the time interval for the page is in the past - and no more data is added to this page.

#### *6.7.5.10 External Link*

A link to a web-site extern to the SmartView3 generated site. This will appear on the main menu.

### 6.7.6 Edit Page

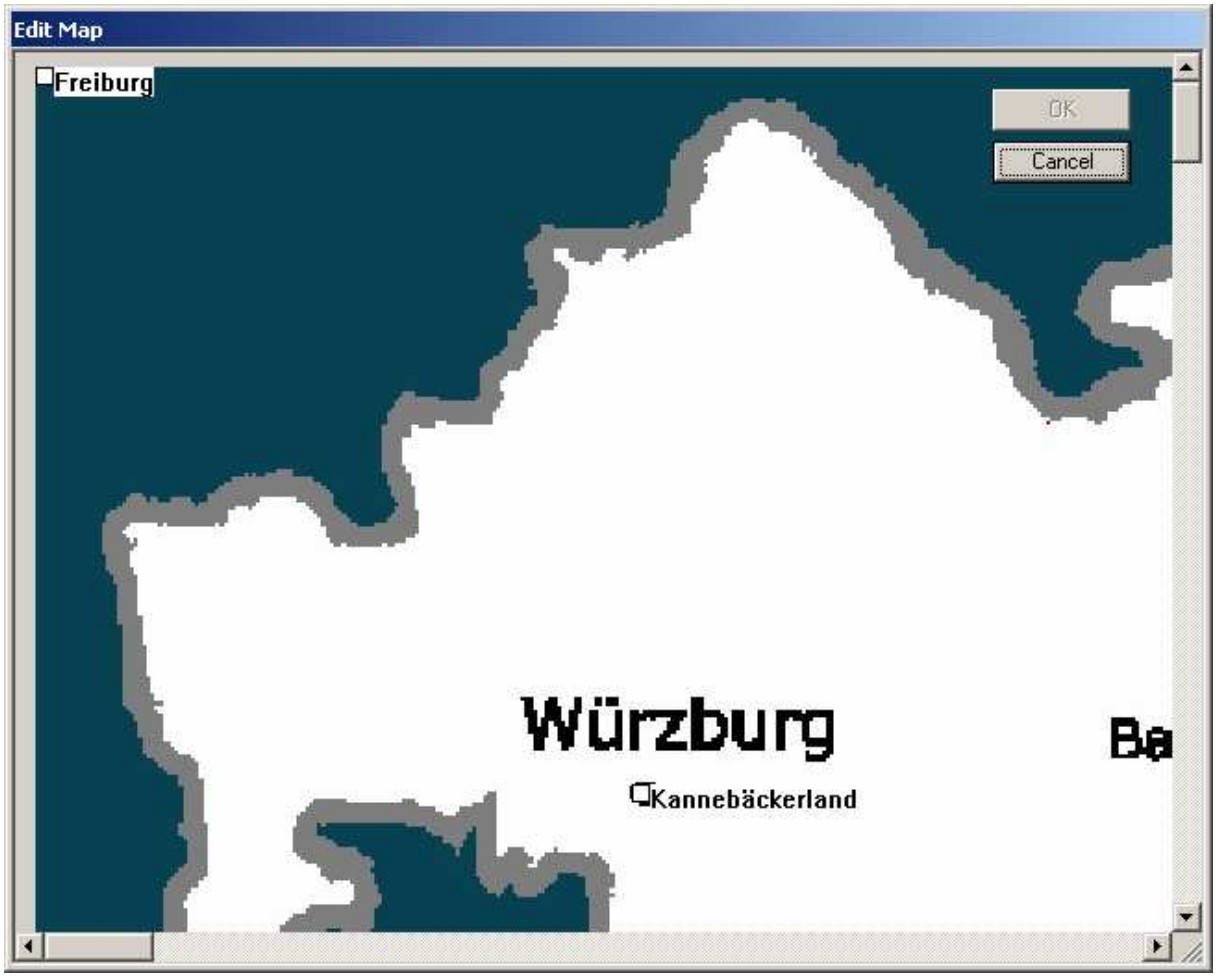



- **Table(s) on separate page:** if checked, tables are placed on a separate html page (data page only)
- **Data as .csv for download:** if checked, data for all sensors of a data table are included as .csv file for download from the page
- **Generate Print View:** if checked, a separate html page for printing without site header/footer and menu, and with white background is generated, and a link to this page is shown on the corresponding page.
- **Align Centered:** align elements (diagrams, tables etc) centered on page
- **Show Status Map Color in Table:** if selected, all values sensors in tables that have a status map assigned will be shown with the appropriate color as background of the table cell.
- **Auto-Refresh Interval:** For pages of type “Map”, “Station” or “Data” the auto-refresh interval (in seconds) can be configured. An interval of 0 seconds disables auto refresh for the page.
- **Max. Sensors in Status Table:** (overview page only) this parameter sets the maximum number of sensors that are displayed on a status table. If the stations contain more than that number of sensors, multiple tables are created.
- **Title:** the title for this page
- **Sub Title:** the sub title for this page
- **Footer:** the footer for this page
- **Data Interval:** the data interval for a data page
- **Station ID:** the associated station id for a page

Buttons:

- **Apply Template:** you can (re-) apply a page template to data or station pages. The appropriate dialogs (see “Getting started” or “adding data pages” above) are displayed.
- **Elements:** show/edit the elements (diagrams, tables etc) for the page, and change the sequence of elements on the page
- **Reset Page:** resetting a page will cause SmartWeb to re-generate the page. For data pages, all “archive” pages with all available data for the page/station are re-generated.

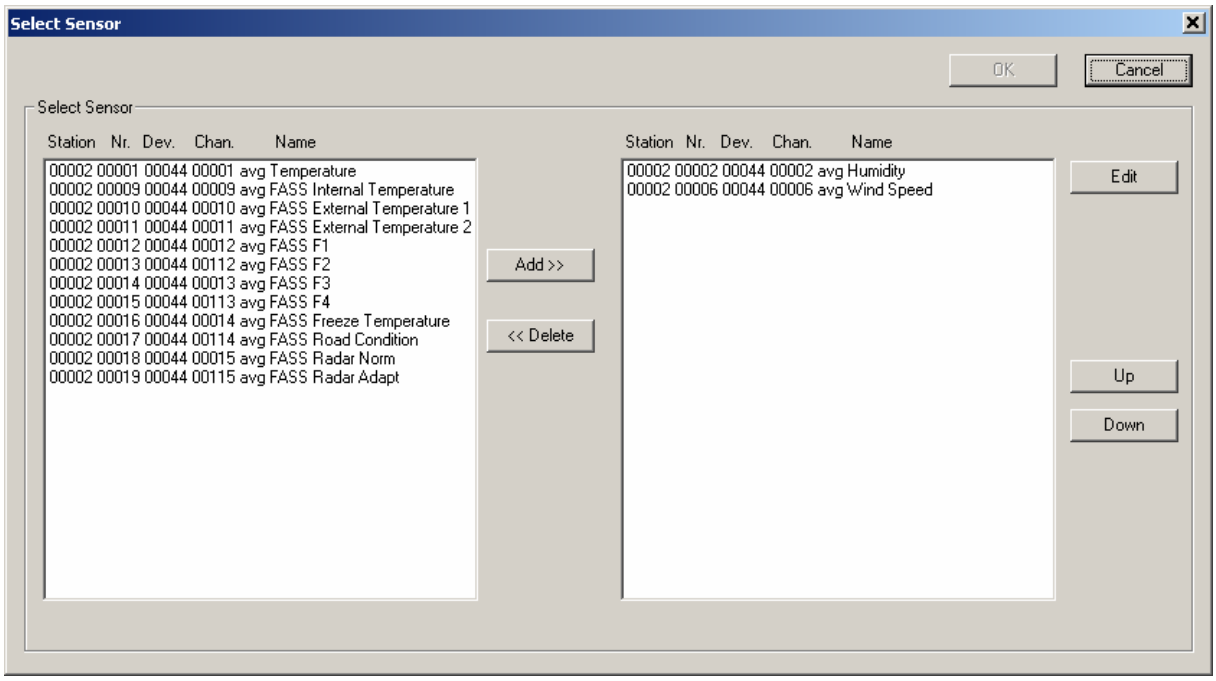
### 6.7.7 Edit Map




This is the map editor window (Map page only). Click on the white square of the station and move it (by holding the left mouse button) to the desired position on the map, and release the mouse button. If the map is bigger than your screen, you may need to move the station icon, and scroll the dialog, and then move the station icon further.

To load (or change) the map : edit the “Bitmap” page element of the “Map” page.

### 6.7.8 Select Sensor




This dialog shows sensors to be selected. This dialog is used wherever Sensors can be selected within SmartView3 (e.g. Sensors for a data page, sensors for diagrams or tables, sensors to be exported etc.).
Multiple sensors can be selected from the left side (list of available sensors).

**Note:** if you edit a template page element, a similar dialog to select logical sensor types (instead of “real” station sensors) is displayed.

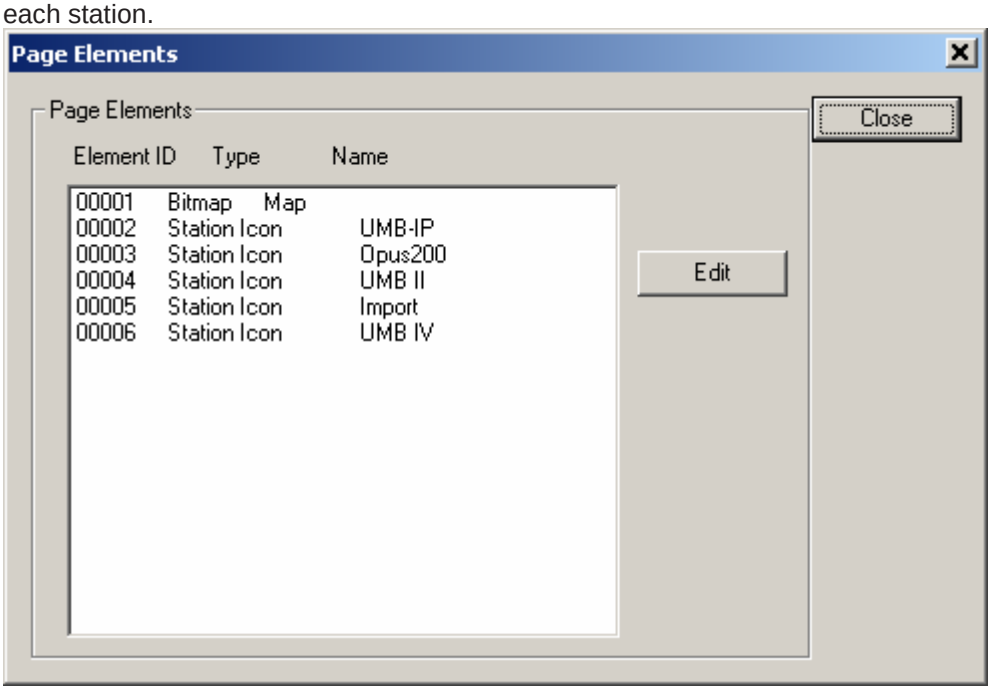
**Page Sensors**: all sensors from all stations configured to the site are shown on the left, and can be added to the page.
**Page Element Sensors**: all sensors configured for the page are shown on the left. Sensors can be added to a line diagram (up to the “number of sensors per line diagram”; or if the maximum of 4 units per line diagram has been reached). One sensor can be added to a bar diagram - the number of value/color intervals is asked for, and the sensor is added to the page element sensor list for every value interval - this way you can configure the color for each value interval for a bar diagram.

The sequence of the selected sensors can be changed by selecting a sensor and using the “up/down” button.

### 6.7.9 Page Elements

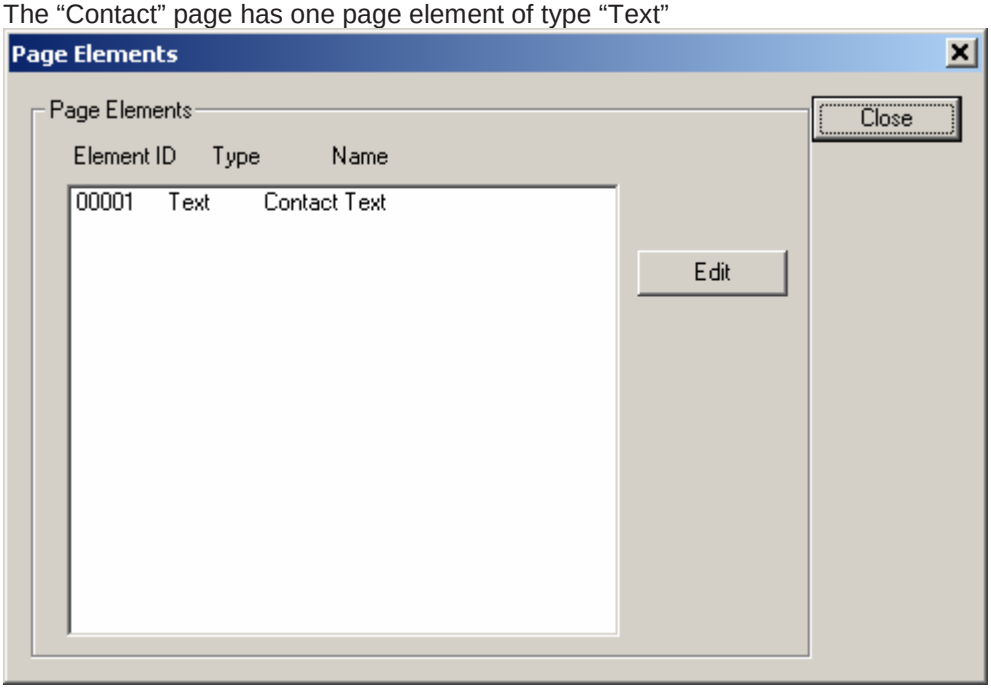
Only “Map”, “Contact”, “Data” and “Station” pages contain page elements.

- The “Map” page has one element of type “Bitmap” and one element of type “Station Icon” for each station.



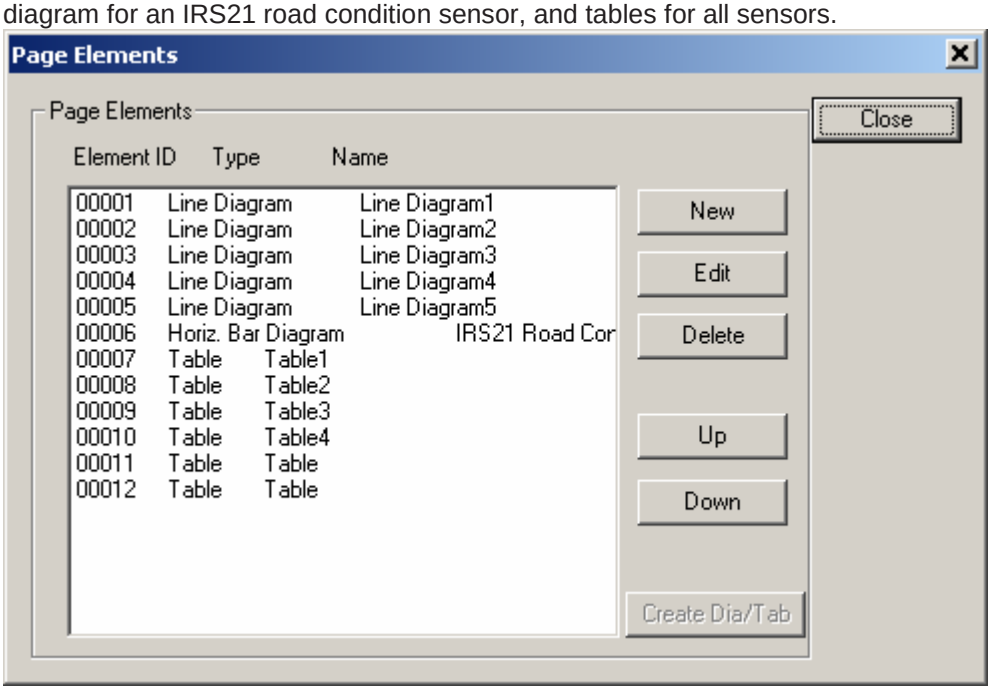

The sensors assigned to the station icon determine, which sensor values are displayed when you “scroll over” the station icon with the mouse (map display). They determine which sensors (and in which order) are shown on the overview table as well.

- The “Contact” page has one page element of type “Text”

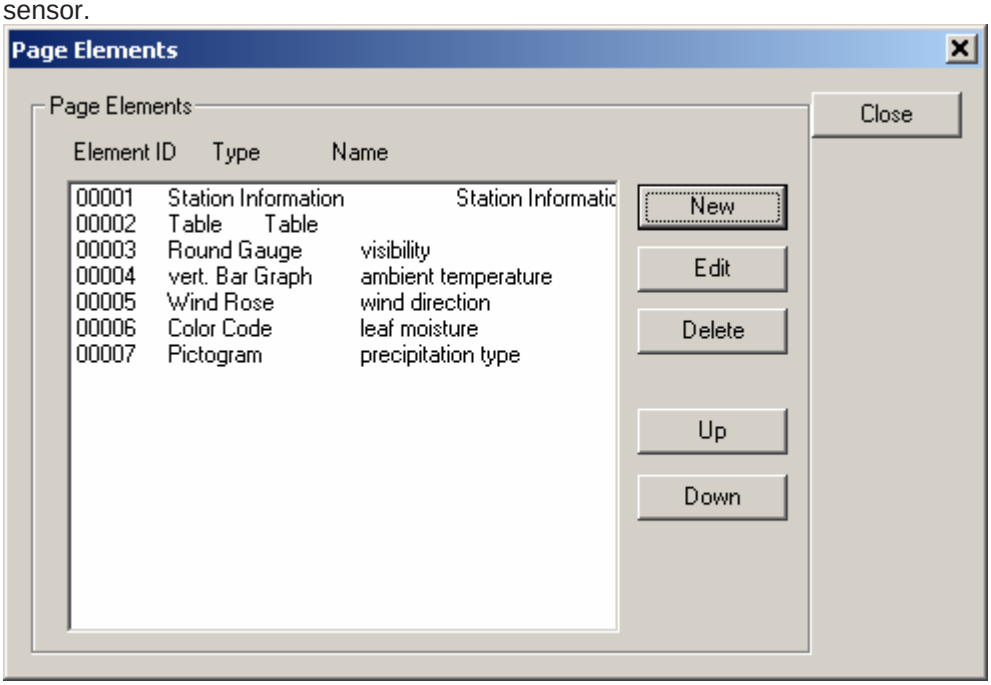



- A “Data” page can have one or more page elements of type “Line Diagram”; “Bar Diagram” “Table”, “Text” and/or “Report”.
  If a data page does not have any page elements, default elements can be created by clicking “Create Dia/Tab”.
  This will create line diagrams for all sensors except an IRS21 road condition sensor, a bar

diagram for an IRS21 road condition sensor, and tables for all sensors.




- A “Station” page always contains a page element of type “Station Information”.
  You can select which station information items to be displayed by editing this page element item.
  It also always contains a “Table” element to display the last (current) sensor value for each configured sensor.
  It may contain one or more “Report” page elements with specific time intervals (day/month/week).
  It may also contain one or more “Text” page elements.
  A “Station” page may contain “analog gauge” displays like “Round Gauge”, “vert. Bar Graph”, “Wind Rose”, “Color Code” or “Pictogram” to display the last (current) sensor value for a sensor.




The sequence of the page elements can be changed by selecting an element and using the “up/down” button.

#### *6.7.9.1 Page Element Types*

Following page element types exist:

**Station Icon :**
An Icon for a station on the map page (map page only).

**Bitmap**
A bitmap file - used with Site Elements for the site logo, and for the map page for the map.

**Line Diagram**
A line diagram with up to 4 different Units/Y-Axis (data page only)

**Horiz. Bar Diagram**
A horizontal bar diagram with different colors depending on the sensor value (data page only)

**Table**
A table with sensor values (data page only)

**Text**
A text element (station page, data page, and other/contact page)

**File**
A file - used for Site Elements CSS file and Javascript File.

**Round Gauge**
A round gauge display similar to a speedometer.

**Wind Rose**
A wind rose display to display the wind direction.

**Vert. Bar Graph**
A vertical bar graph similar to a thermometer.

**Color Code**
A color code display where a value is mapped to a color (similar to the horizontal bar diagram).

**Pictogram**
A pictogram display where a value is mapped to a pictogram.

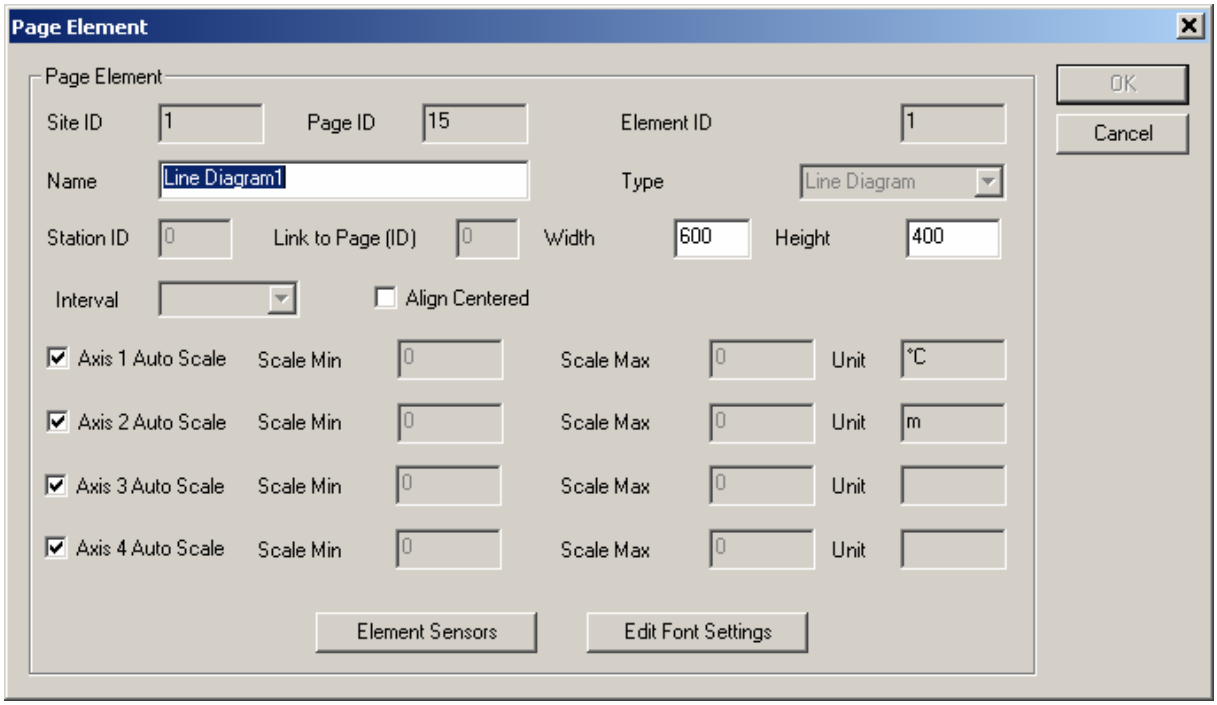
**New Line**
A “New Line” page element to separate analog gauges – so the next gauge will be displayed on a new line.

**Report**
A “Report” page element shows data for the selected sensors for a specified time interval. Report page elements are similar to the “Report” function available per station in SmartView3, see Chapter 5.2.3 Reports.

### 6.7.10 Edit Page Element

Depending on the Page Element Type (see above) the Edit Page Element Dialog shows different input fields :

*6.7.10.1 Edit Line Diagram:*

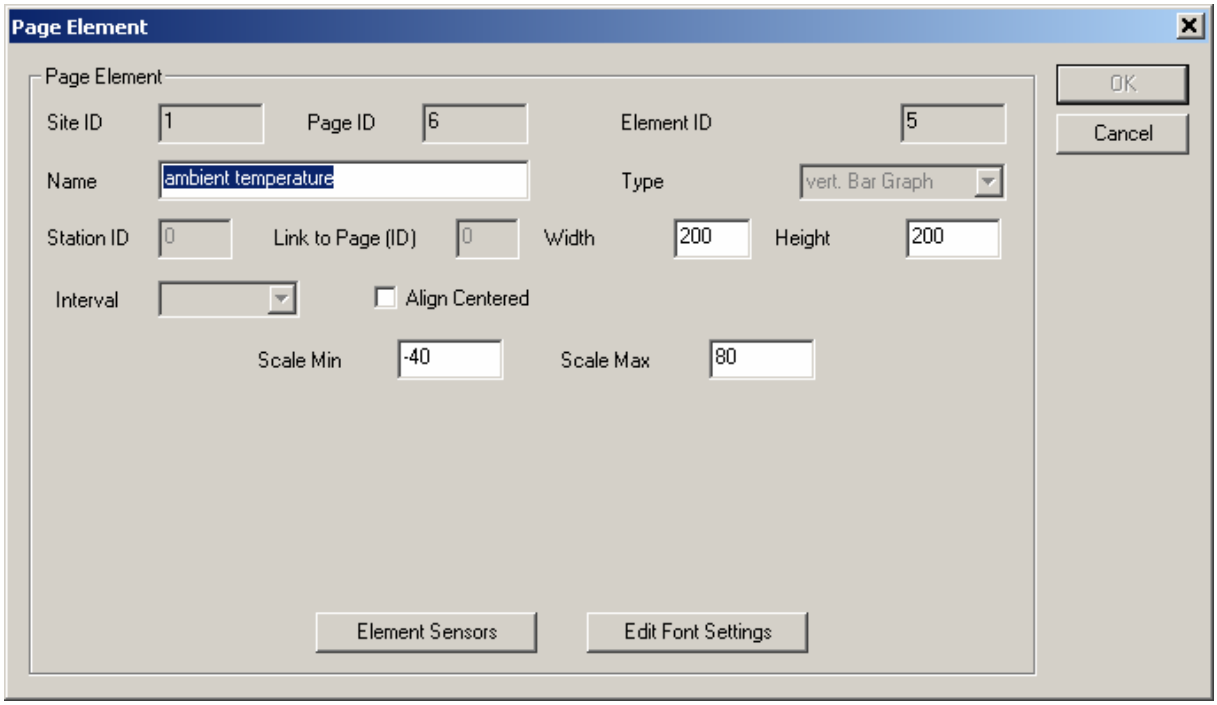



- **Name:** the name of the page element
- **Width:** the width of the element
- **Height:** the height of the element
- **Interval:** data interval for the element (only for “Report” elements on station page).
- **Align Centered:** align element centered on page.
- **Axis x Auto Scale:** Line Diagram only: auto scale diagram to min/max values in interval
- **Scale Min:** if Auto Scale is off, the minimum value for y-axis of the diagram
- **Scale Max:** if Auto Scale is off, the maximum value for y-axis of the diagram
- **Unit:** the unit of the axis of the diagram

**Buttons:**

- **Element Sensors:** show/edit the element sensors. See Select Sensor
- **Edit Font Settings:** for diagram/analog gauge only; See “Edit Font Settings”

*6.7.10.2 Edit "Round Gauge" or "vert. Bar Graph",*

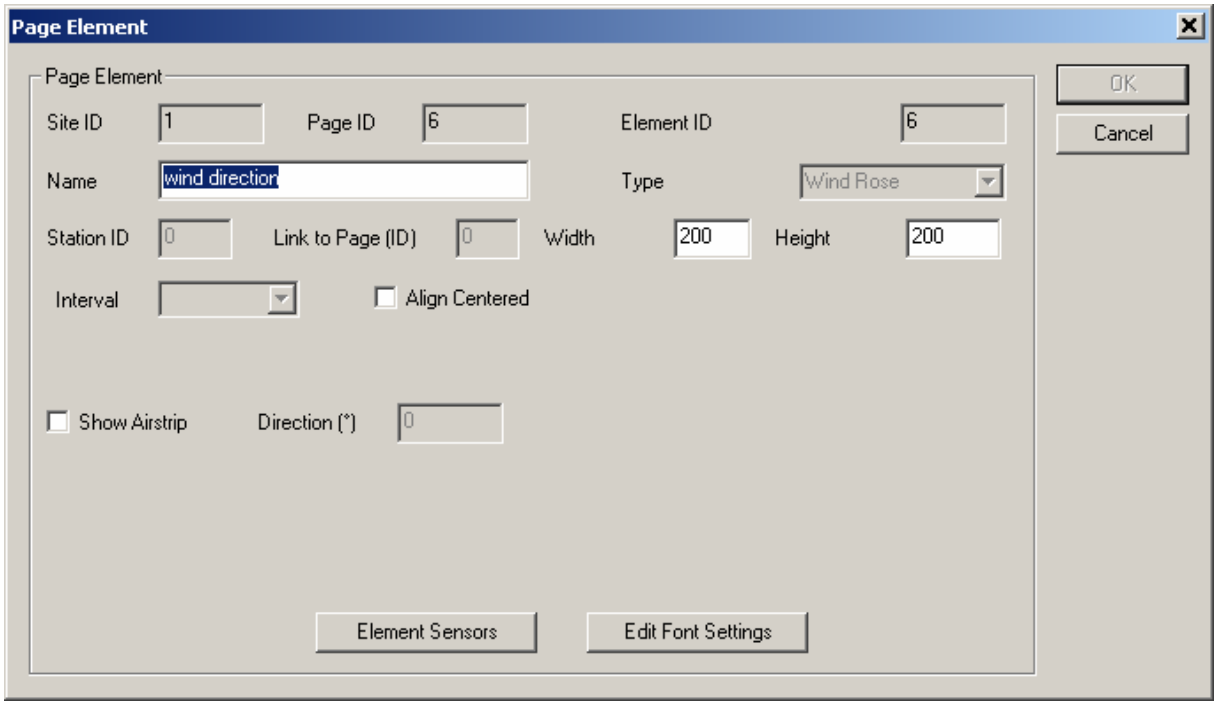



- **Name:** the name of the page element
- **Width:** the width of the element
- **Height:** the height of the element
- **Align Centered:** align element centered on page
- **Scale Min:** the minimum value for the scale of the display
- **Scale Max:** the maximum value for the scale of the display

**Buttons:**

- **Element Sensors:** show/edit the element sensors. See Select Sensor
- **Edit Font Settings:** for diagram/analog gauge only; See "Edit Font Settings"

### *6.7.10.3 Edit Wind Rose*

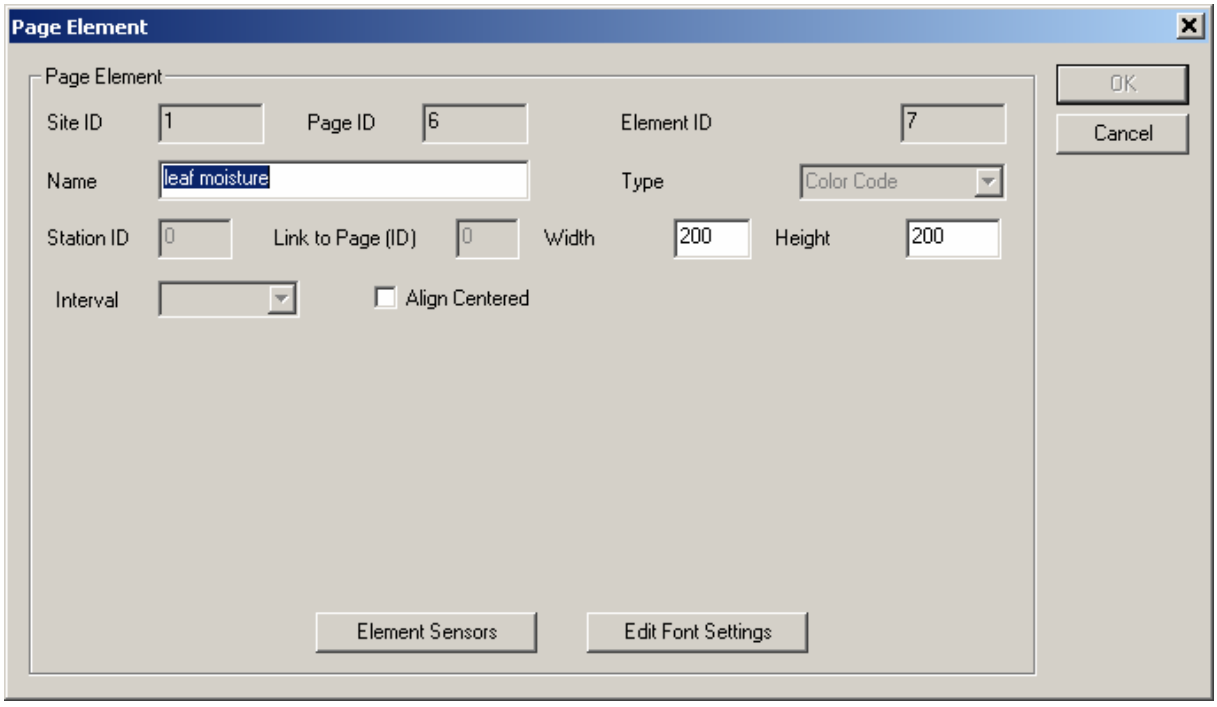



- **Name:** the name of the page element
- **Width:** the width of the element
- **Height:** the height of the element
- **Align Centered:** align element centered on page
- **Show Airstrip:** if this box is checked, an “airstrip” is shown in the center of the wind rose, aligned in the angle specified as “**Direction**”.

**Buttons:**

- **Element Sensors:** show/edit the element sensors. See Select Sensor
- **Edit Font Settings:** for diagram/analog gauge only; See “Edit Font Settings”

### *6.7.10.4 Edit vert. Bar Diagram, Color Code or Pictogram*



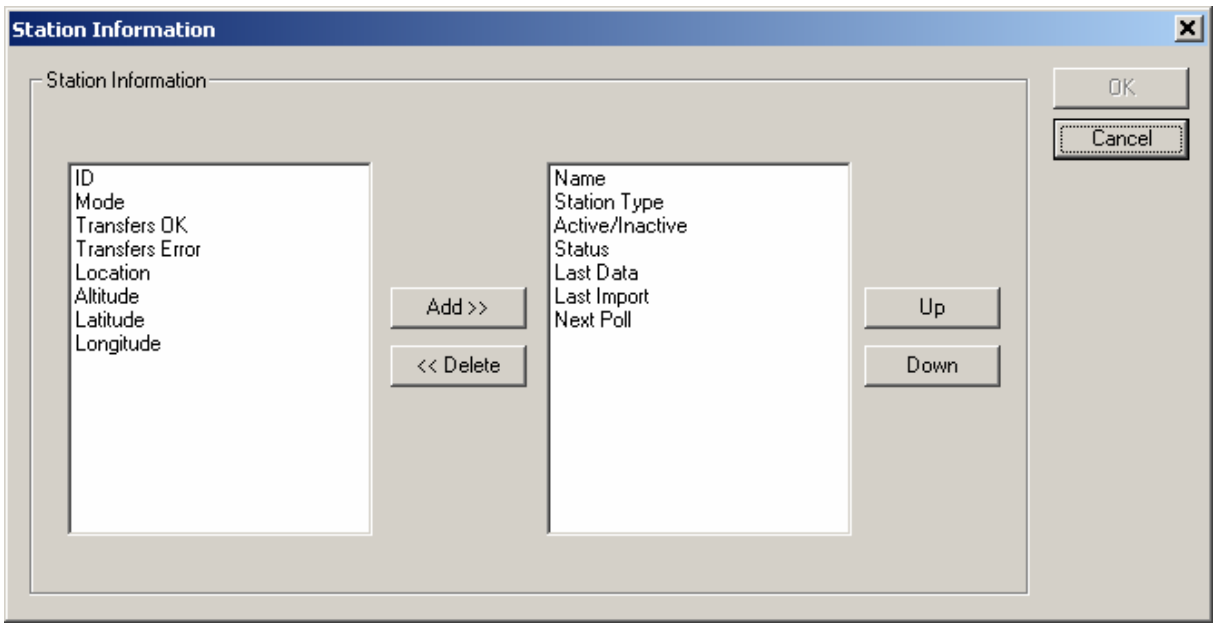
- **Name:** the name of the page element
- **Width:** the width of the element
- **Height:** the height of the element
- **Align Centered:** align element centered on page

**Buttons:**

- **Element Sensors:** show/edit the element sensors. See Select Sensor
- **Edit Font Settings:** for diagram/analog gauge only; See "Edit Font Settings"

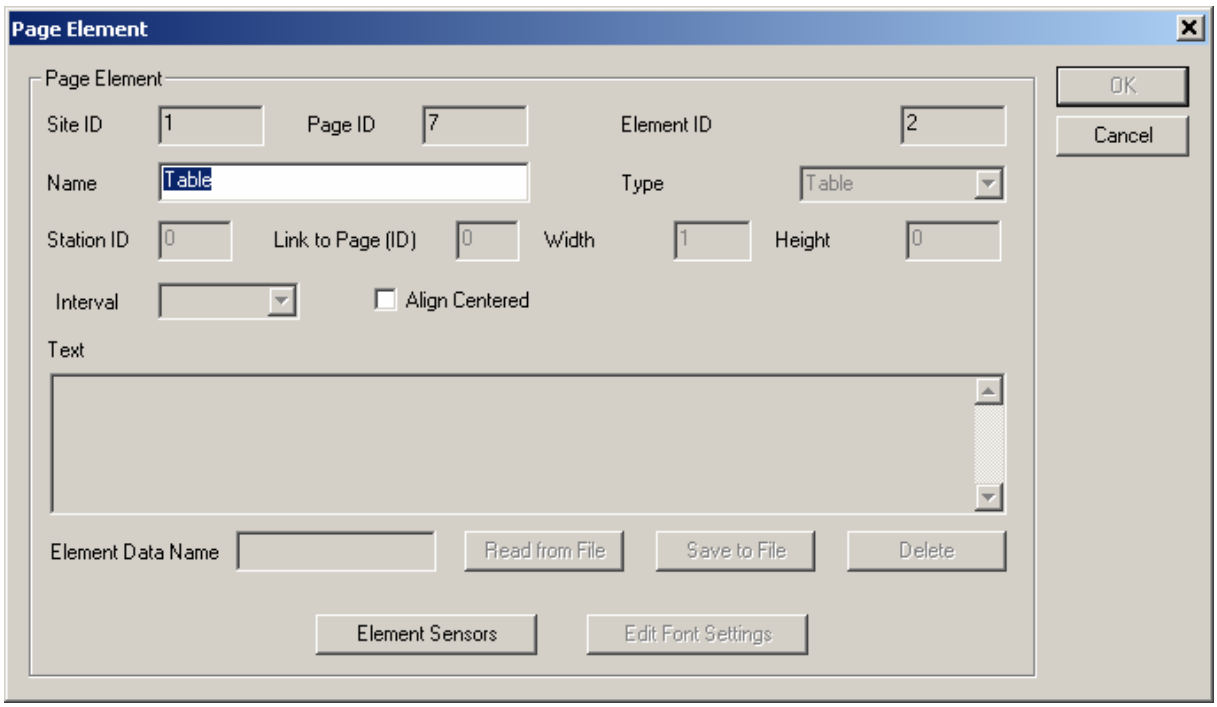
**Note:** to assign specific colors and/or pictograms to values for these diagram types, Status Mapping must be configured for the "Element Sensor" of this diagram.

*6.7.10.5 Edit "Station Information"*




Select which station information is included in the station information table on the station page.

*6.7.10.6 Edit "Table":*



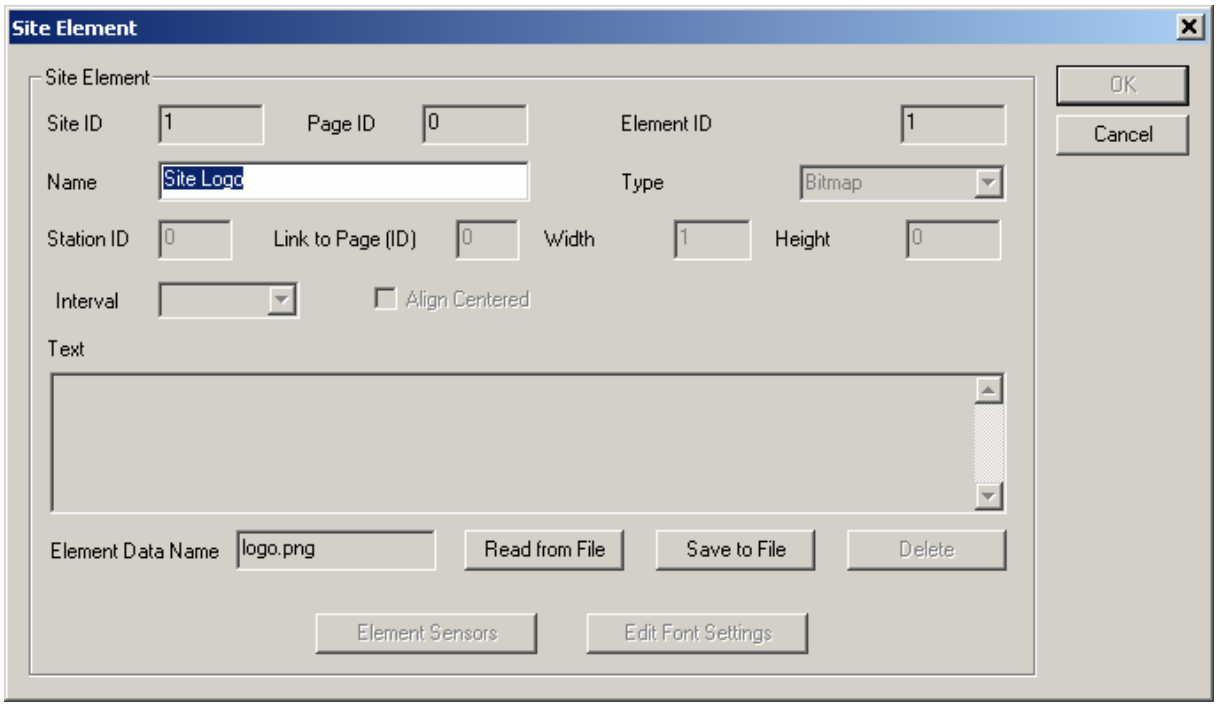

- **Name**: the name for the table
- **Align Centered**: align element centered on page

**Buttons:**

- **Element Sensors:** show/edit the element sensors. See Select Sensor

**Note:** to display a descriptive text for a value, Status Mapping may be configured for the "Element Sensor" in a table.

*6.7.10.7 Edit Bitmap:*



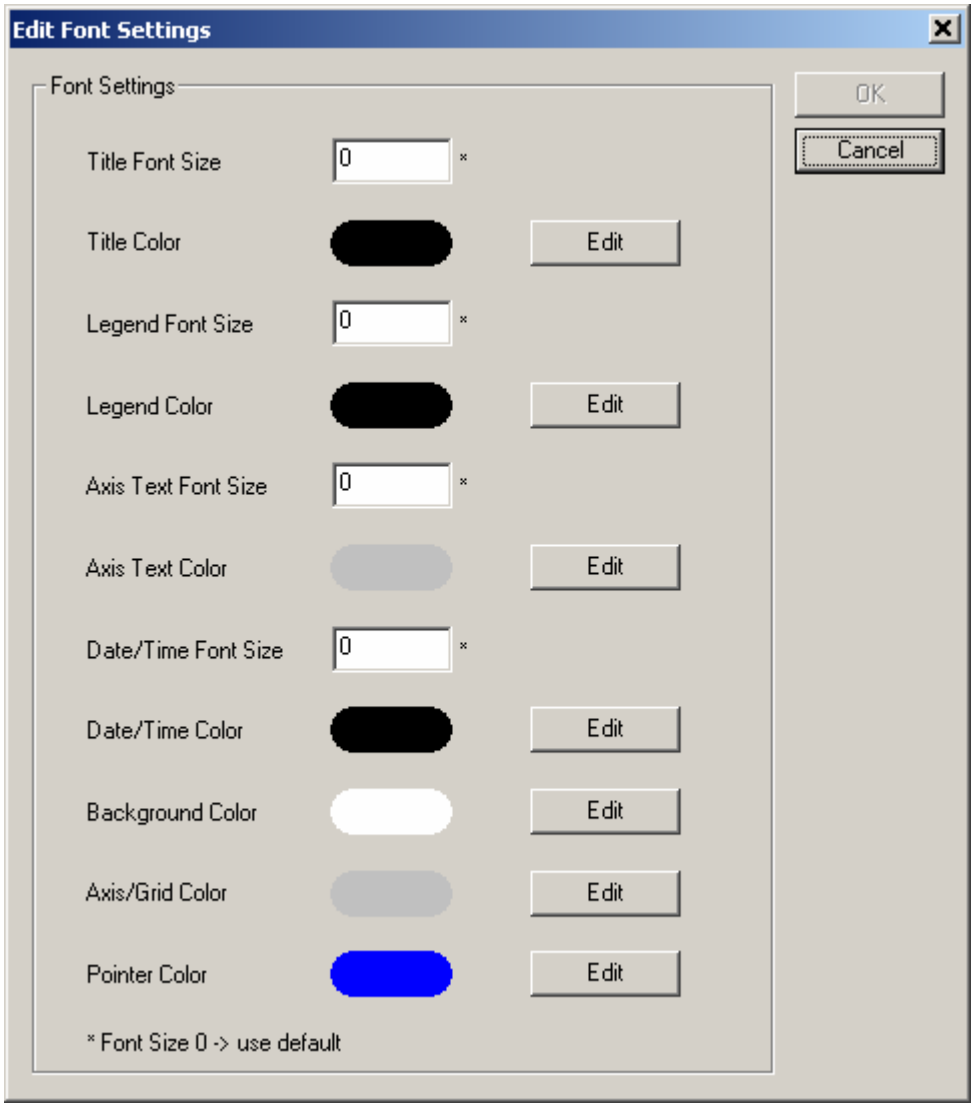

- **Text:** (Text element only) Text for the element - may contain any valid html tags
- **Element Data Name:** Bitmap and File only - the file name

**Buttons:**

- **Read from File:** read Bitmap or File from hard disc
- **Save to file:** save Bitmap or File to hard disc
- **Delete:** delete the data (file)

### 6.7.11 Edit Font Settings

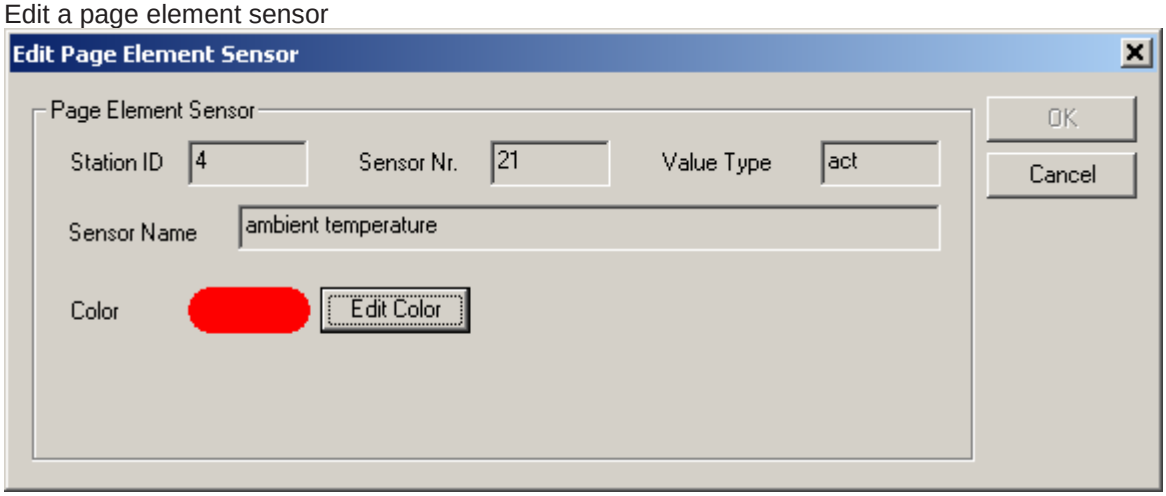
This dialog allows configuration of font and color settings for diagrams.



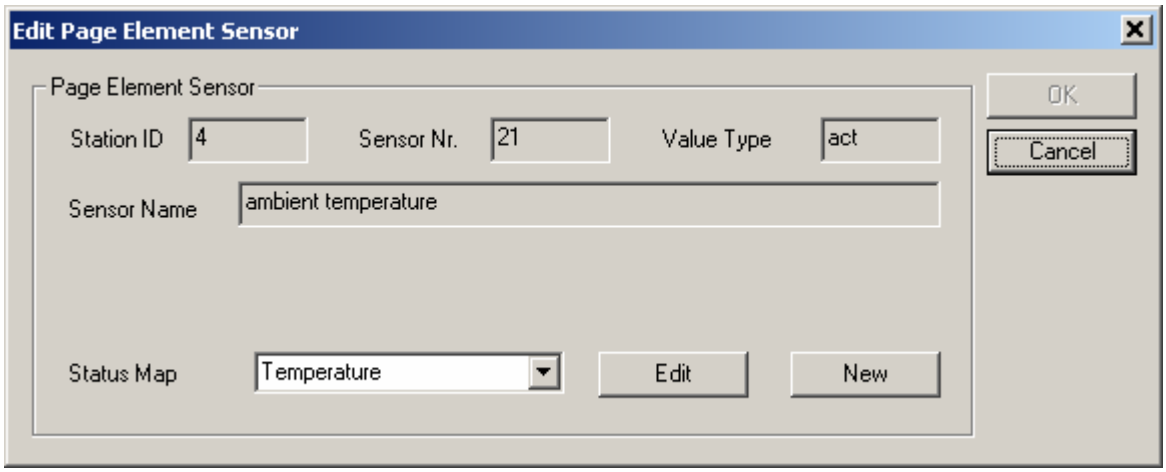

- **Title Font Size:** the size for the title text. 0 = use default size. For “analog gauge” diagrams the default size is calculated depending on the size (height) of the diagram.
- **Title Color:** the color for the title text.
- **Legend/Value Font Size:** for “Line Diagrams” and “horiz. Bar Diagrams”: the size of the font used for the diagram legend text. For “analog gauge” diagrams the size of the font used for the value text.
- **Legend/Value Color:** the color for the legend or value text.
- **Axis Text Font Size:** the size for the axis (line diagram) or scale (analog gauge) text.
- **Axis Text Color:** the color for the axis (line diagram) or scale (analog gauge) text.
- **Date/Time Font Size:** the size for the date/time range (line and vert. bar diagram) or date/time (analog gauge) text.
- **Date/Time Color:** the color for the date/time range (line and vert. bar diagram) or date/time (analog gauge) text.
- **Background Color:** the background color for the diagram.
- **Axis/Grid Color:** the color used to draw the grid or scale for the diagram or gauge.
- **Pointer Color:** the color used to draw the pointer of an analog gauge.

### 6.7.12 Edit Page Element Sensor

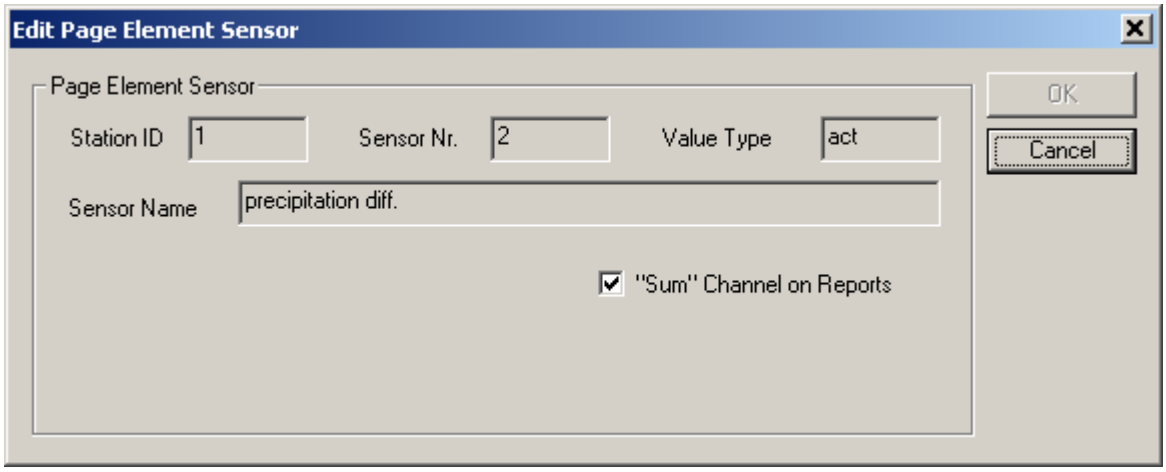
Edit a page element sensor




For a line diagram sensor the color of the curve in the line diagram can be specified.




For a “horizontal bar diagram“, “color code”, “table” or “pictogram”, “status mapping” is used to specify which color/text/pictogram is assigned to a specific sensor value.

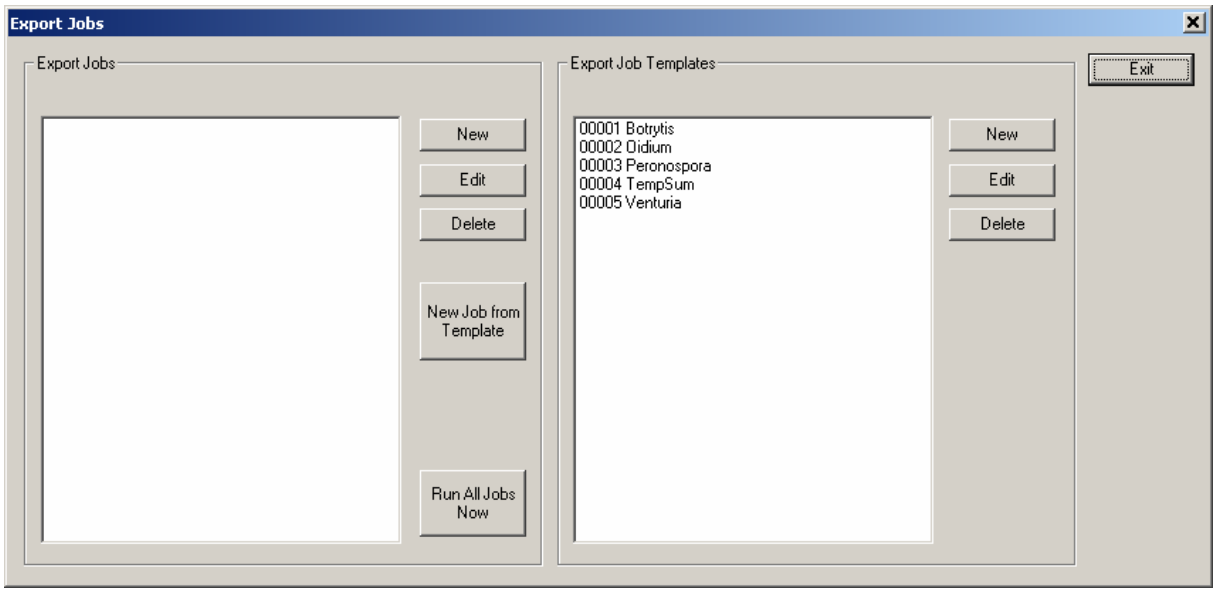



For “reports” a sensor/channel can be configured to show sum values instead of average values.

# 7 Configure Export/Import

Export/Import tasks are configured as “Export Jobs” in SmartView3. There are “manual” Export/Import jobs, or Export/Import jobs that are processed automatically.
The difference between “manual” and “automatic” is, that “manual” jobs are processed within SmartView3, and only run when you select the job from the list of manual jobs via “file->export/import” in Smartview3, and click the “Run Job” button. All paths (for export file, import file, or external programs) refer to the PC where you run SmartView3.
“Automatic” Export/Import jobs are configured in SmartView3 via “Edit->Export/Import”, and are processed by the SmartCom program – so all paths (for filenames or programs) refer to the PC where SmartCom runs – which could be different from the PC where you run SmartView3. “Automatic” Export/Import jobs are run “as configured”, e.g. every day or week or month etc. (see below).

If you click “Export/Import” from the “Edit” menu (or “Manual Export/Import” from the “File” menu), the following dialog is shown:




On the right side, template export jobs are shown. There are pre-defined template export/import jobs for the plant disease model calculations available from Lufft. You can create your own Export Job Templates.

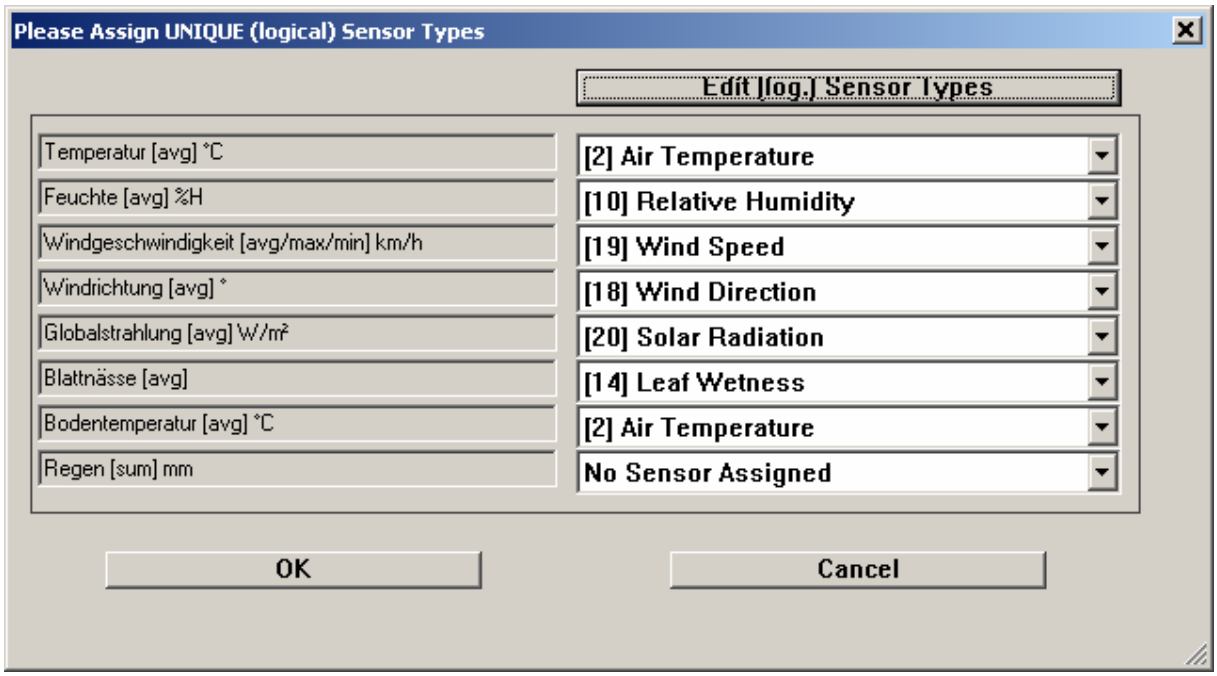
Export Job Templates use “logical” sensor types instead of “real” station sensors. All other attributes for a template job are the same as for a “real” export/import job.

You can eiher create a new export job “from scratch”, or you can create a new job using a template job.

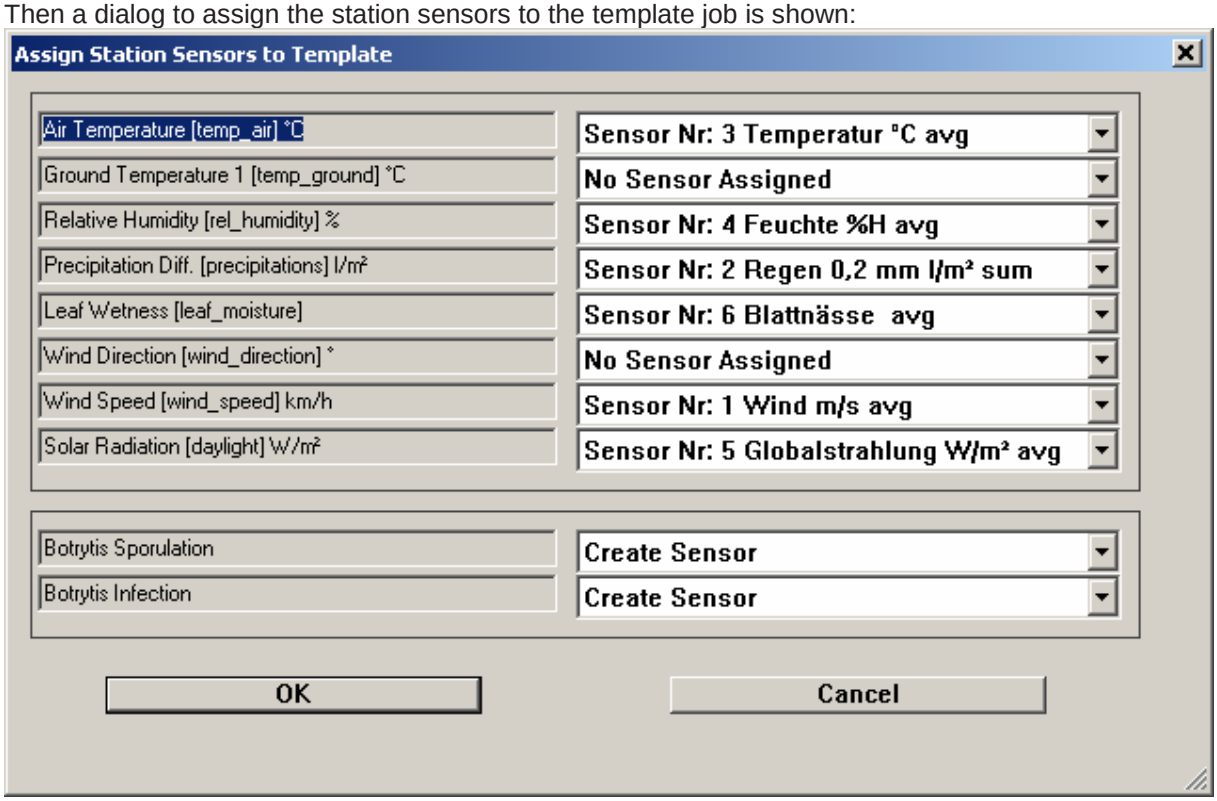
## *7.1 Create an Export/Import job using a Template Job*

If you want to use a template job, select the template job to be used on the right side, and click “New Job From Template”.
A dialog to select the station for the new job is displayed. If you plan to export data from multiple stations in one job, you will need to manually configure the job “from scratch”.

If not all sensors of the station have a logical sensor type assigned, a dialog to assign the logical sensor types is shown:



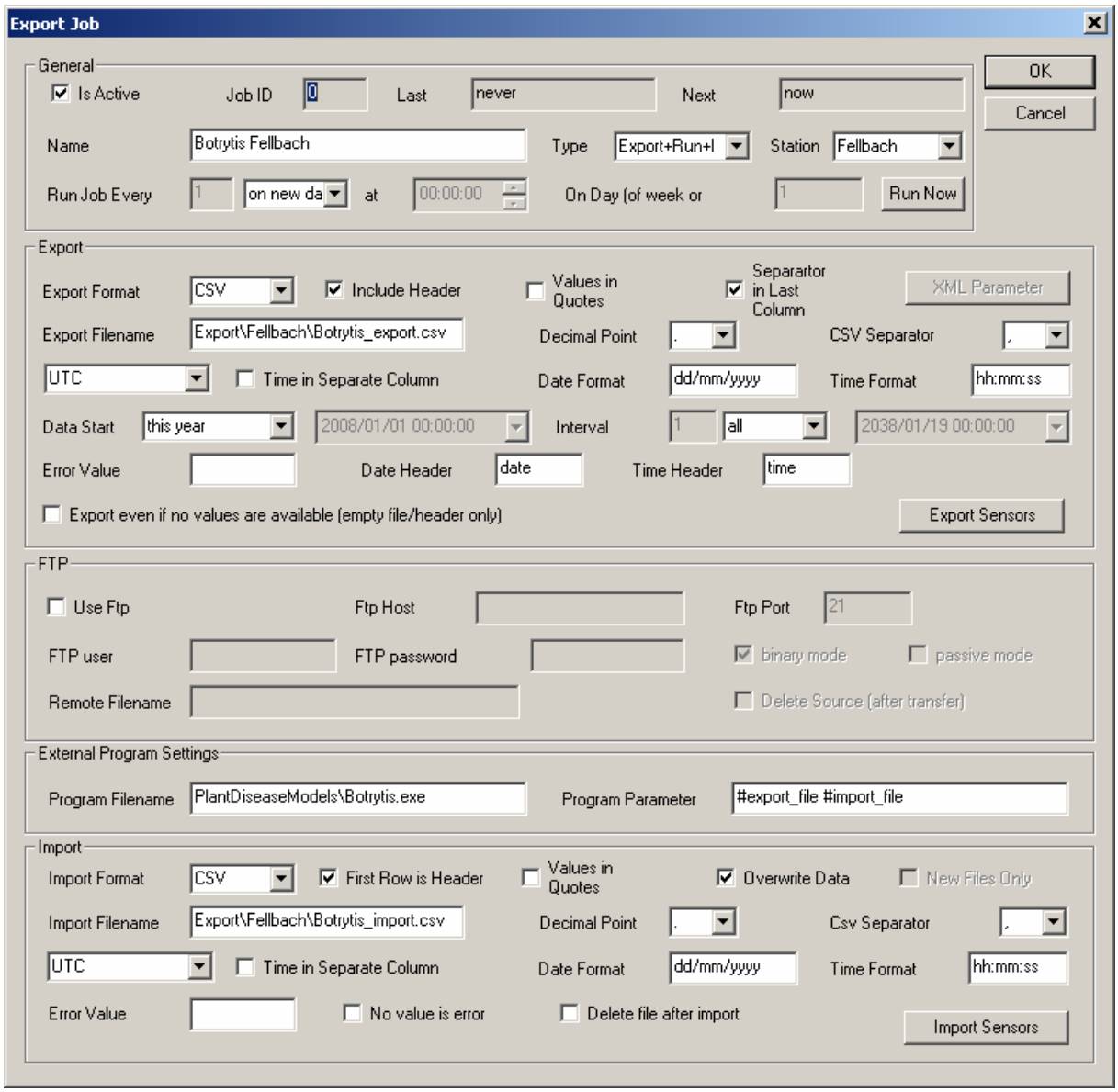

Then a dialog to assign the station sensors to the template job is shown:




If (which usually is the case) there are no appropriate import sensors for the station available, the sensors will be created. For the pre-defined job templates (the plant disease models), this will be done completely automatically.

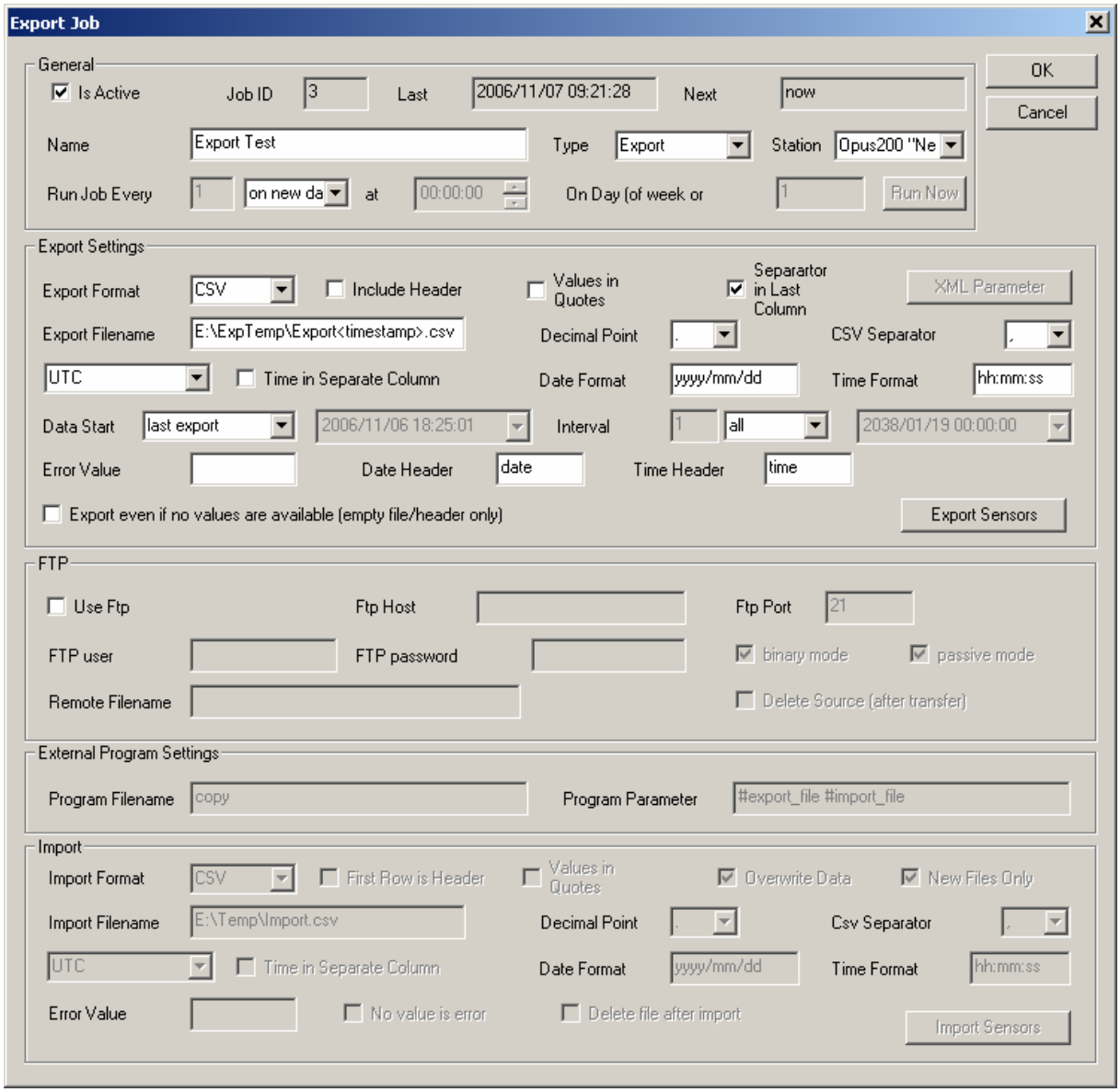
The resulting export job is shown next, and can be customized if needed.

**Note:** if you plan to use the plant disease models provided by Lufft, you need to place them in a folder “PlantDiseaseModels” below the progam folder, or you will have to change the path for the “Program Filename” of the template job (or the resulting job) accordingly.




Details of this dialog are described in the next chapter.

## 7.2 Export Job Dialog




General parameter:

- ➔ Is Active: the Export/Import job is active (or not active).
- ➔ Name: a name for this job
- ➔ Type : type of job; can be “Export”, “Export+Run”, “Export+Run+Import”, “Run+Import”, “Import”, “Run”, “FTP”, “Export+Import” or “Run (external export)”
  **Export**: data for the specified “Export Sensors” (see below) are exported in either .csv or .xml format.
  **Export+Run**: data is exported, and an external program is called.
  **Import:** data for the specified “Import Sensors” (see below) are imported from a file in .csv format
  **Export+Run+Import**: data is exported in .csv format, an external program (e.g. to calculate forecasts or “derived sensor values”) is run (which writes its result into a new file in .csv format), and the resulting data is imported (from the .csv file the external program wrote).
  **Run+Import**: the external program is called, and then the specified file(s) is/are imported
  Import: the specified file is imported. The filename may contain “wildcard” characters “*” or “?”, e.g. “C:\Import*.csv”
  **Import**: the specified file(s) is/are imported. The filename may contain “wildcard” characters

"*" or "?", e.g. "C:\Import*.csv"
**Run**: the specified program is executed (no export or import is done)
**FTP**: the specified file(s) is/are downloaded from the server to the location specified as "import filename". No export or import is done. The filename may contain "wildcard" characters "*" or "?".
**Export+Import:** data is exported to and imported from the specified file name. No program is called in between export and import. May be used to scale/change sensor values and re-import the values as a different sensor (e.g. scale generic mV or mA sensor value to a specific sensor with proper name/unit etc.).
**Run (external export):** the specified external program is run. Export sensor can be configured, but no export file is written. The external program is responsible for reading the appropriate sensor values from the database itself (used for custom specific export functions, e.g. export via SOAP client).

- ➔ **Station:** the station associated with this export job. If a station is selected, only sensors from this station can be selected as "export" or "import" sensors. If no station is selected, sensors from all stations can be used in export and import in this job.
- ➔ **Run Job Every / at / on day** (of week or month) : for "automatic" jobs only, you can specify the interval the job is run (e.g. every hour, every week etc). For "manual" jobs, this input fields are disabled
- ➔ **Run Now:** clicking this button will reset the "next run" time to now.

Export Settings :

- ➔ **Export Format:** CSV or XML
- ➔ **Include header:** write a line with column names as first line of the export file
- ➔ **Values in quotes:** (CSV only) put all values in quotes ("…")
- ➔ **Separator in last column:** (CSV only) write a separator character at the end of the last column
- ➔ **XML Parameter:** (XML only) displays a dialog to configure the XML output format (see below)
- ➔ **Export filename:** name of the export file. The name may include the tags <date> or <timestamp> which will be replaced by the actual date (yyyymmdd) or timestamp (yyyymmddhhmmss) when the job is run
- ➔ **Decimal point:** (CSV only) – the character to be used as decimal point in numbers ("." or ",")
- ➔ **CSV separator:** (CSV only) – the character to be used as separator between columns (",", ";" or "TAB")
- ➔ UTC/ Local Time (no DST offset) / Local Time (with DST offset): Timezone information used for timestamps in export.
- ➔ **Time in separate column:** (CSV only) use separate columns for date and time in export file
- ➔ **Date format:** (CSV only) format string for date. yyyy = 4 digit year, yy = 2 digit year, mm = month, dd=day – or "special format" **UTC-TIMESTAMP** -> date/time is an UTC-Timestamp (integer value).
- ➔ **Time format:** (CSV only) format string for time. hh=hour, mm=minute, ss=second
- ➔ **Data start:** start time of data in export : first available, last_hour, yesterday, last week, last month, last year….or fixed start date/time
- ➔ **Interval:** the time interval of data to be exported : hour/day/week/month/year/all or fixed end date/time
- ➔ **Error value:** string to be used to indicate an error value
- ➔ **Date header:** header for date column
- ➔ **Time header:** header for time column
- ➔ **Export Sensors:** opens a dialog to configure the sensors to be exported

FTP:

- ➔ **Use ftp:** for jobs of type "Export" or "Export+Import", the export file will be transferred to a server using ftp after the export file has been written. For jobs of type "Import" the import file will be transferred from a server using ftp before it is processed.
- ➔ **Ftp host:** DNS name or IP address of ftp server
- ➔ **Ftp port:** port to be used for ftp (default : 21)
- ➔ **Ftp user:** ftp user name
- ➔ **Ftp password:** password for ftp user
- ➔ **Binary mode:** transfer file as binary
- ➔ **Passive mode:** use ftp passive mode
- ➔ **Remote filename:** name of file on ftp server

- ➔ **Delete Source (after transfer):** delete the source file after it has been transferred successfully

External Program Settings

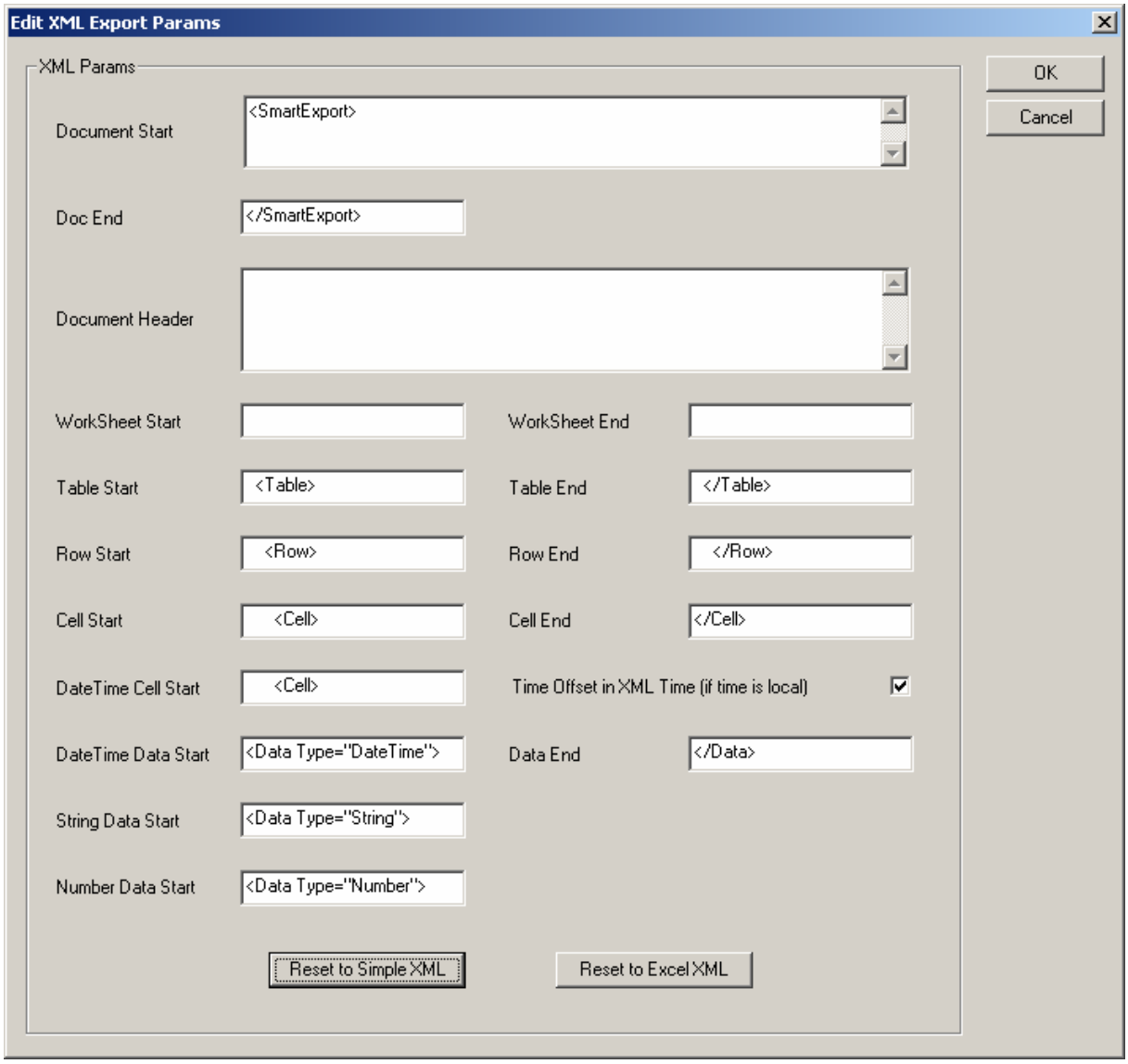
- ➔ **Program filename:** name of the external program (including path)
- ➔ **Program parameter:** parameter to be used for the external program. The special tags "#export_file" and "#import_file" will be replaces by the name of the export or import file respectively

Import

- ➔ **Import format:** only CSV is supported as import format at the moment
- ➔ **First row is header:** the first row in the import file contains a header (e.g. with column names).
- ➔ **Values in quotes:** the values in the import file are in quotes
- ➔ **Overwrite data:** if this checkbox is checked, new imported data will overwrite data for the same time that is already stored in the database.
- ➔ **New Files Only:** only import files that have a "last modified" timestamp newer than the timestamp this job was run last.
- ➔ **Import filename:** name of the import file. The name may include the tags <date> or <timestamp> which will be replaced by the actual date (yyyymmdd) or timestamp (yyyymmddhhmmss) when the job is run.
  Note: for Jobs type "Import", "Run+Import", wildcards are supported in the filename. For jobs type "FTP" the value of "import filename" only indicates the folder where the files are to be transferred to – please use in this case an appropriate folder name (e.g. c:\download*.*)
- ➔ **Decimal point:** the character used in the import file as decimal point "." or ","
- ➔ **Csv separator:** the character used in the import file as csv separator "," or ";" or "TAB").
- ➔ UTC/ Local Time (no DST offset) / Local Time (with DST offset): Timezone information used for timestamps when data is imported (timestamps are always converted to UTC when stored in database).
- ➔ **Time in separate column:** date and time are separated (by a csv separator character) in the import file
- ➔ **Date format:** (CSV only) format string for date. yyyy = 4 digit year, yy = 2 digit year, mm = month, dd=day – or "special format" **UTC-TIMESTAMP** -> date/time is an UTC-Timestamp (integer value).
- ➔ **Time format:** (CSV only) format string for time. hh=hour, mm=minute, ss=second
- ➔ **Error value:** string that is used to indicate an error value
- ➔ **No value is error:** if there is no value for a sensor (column) for a given time in the import file, it is treated as an error value.
- ➔ **Delete file after import:** delete the import file after it has been successfully imported.
- ➔ **Import Sensors:** opens a dialog to configure the sensos to be imported (see below).

**Note**: the first column in an import file must always be the date (in the format specified). If date and time are in a single import column, the date/time string must always start with the date string. If date and time are in separate columns, the first column must be the date column, and the second column must be the time column.

## 7.3 XML Parameter Dialog




This dialog allows to configure the XML output format for XML Export.
The exported XML document always has the following structure :

- XML Header (fixed)
- **Document Start**
  - **Document Header** (if specified) – must be a valid XML construct
  - **WorkSheet Start** (if specified)
    - **Table Start** (if specified)
      - **Row Start**
        - **DateTime Cell Start** – **DateTme Data Start** - **Data End** – **Cell End**

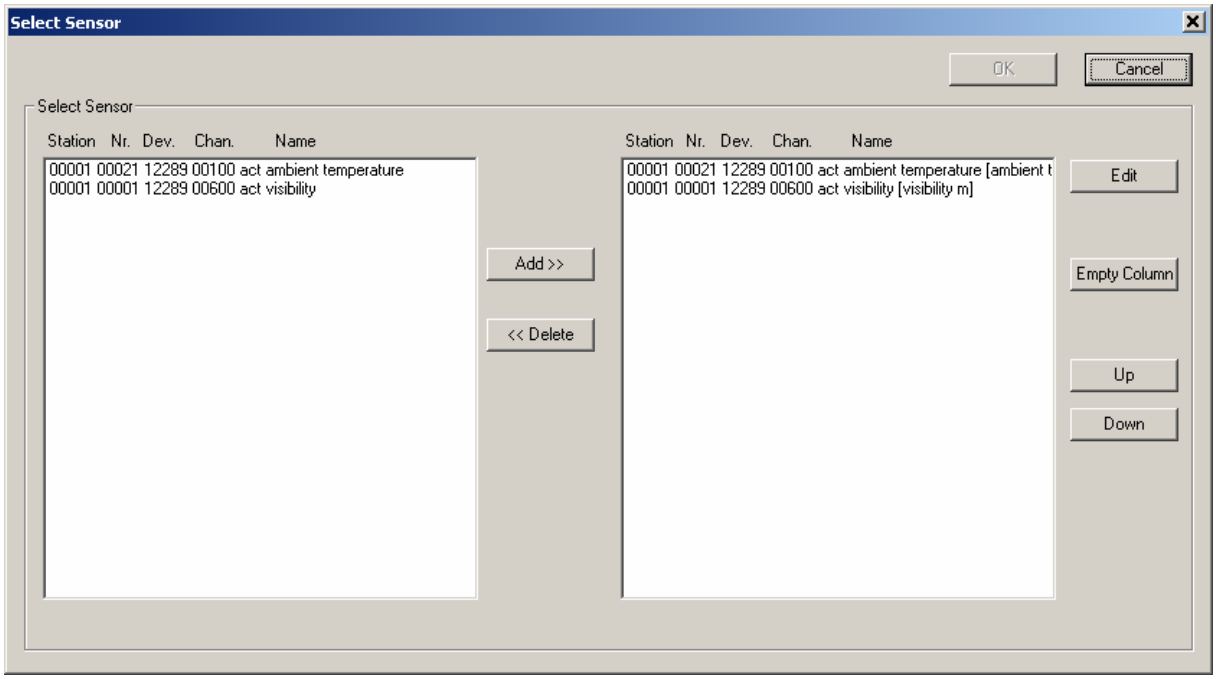
        or
        - **Cell Start** – **String/Number Data Start** – **Data End** – **Cell End**
      - **Row End**
    - **Table End** (if specified)
  - **WorkSheet End** (if specified)
- **Document End**

The "**Time Offset in XML Time**" option controls, if the XML formatted date/time will include the time offset to local time (if Time is local is configured for the Export Job) as specified by the XML standard, or not. Note: Microsoft Excel does not support the time offset in the XML Date Time string.

**Reset to Simple XML** : will set all XML tags to produce a simple XML output
**Reset to Excel XML** : will set all XML tags so the XML file can be imported to Microsoft Excel

## *7.4 Export Sensors*

The “Select Sensor” Dialog is displayed to select the sensors to be exported:



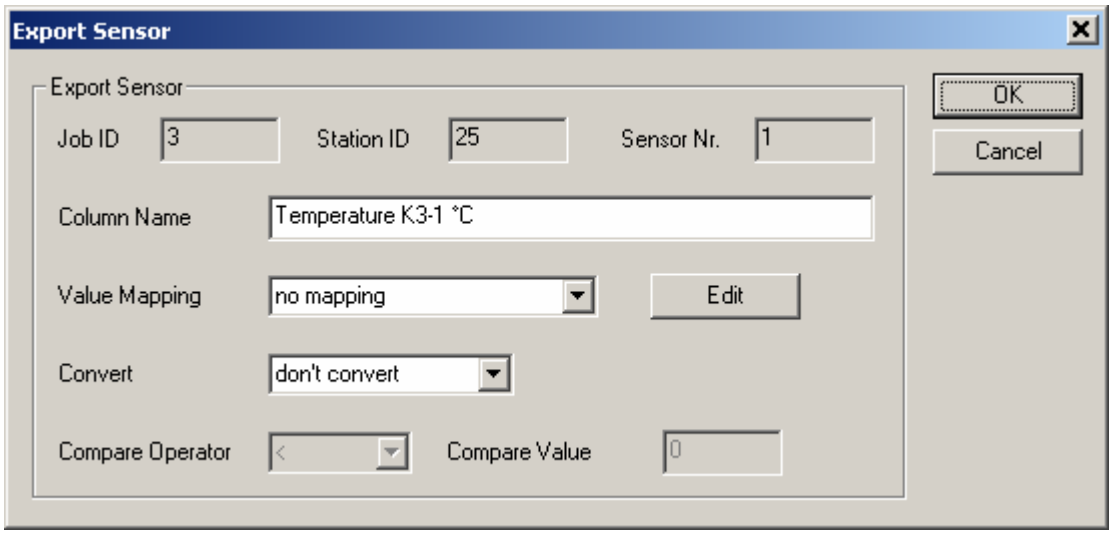

If a station was configured for the export job, only sensors from that station are available on the left side to be selected for the export. A sensor may be selected multiple times.
Empty columns can be inserted into the export by clicking “empty column” on the right side.

**Note:** for template jobs, an appropriate dialog to assign logical sensor types instead of “real” sensors is displayed

The sequence of the sensors can be changed by selecting a sensor and using the “up/down” buttons.

For each sensor you add to the export, the “Export Sensor” Dialog is displayed to set following options for the sensor:

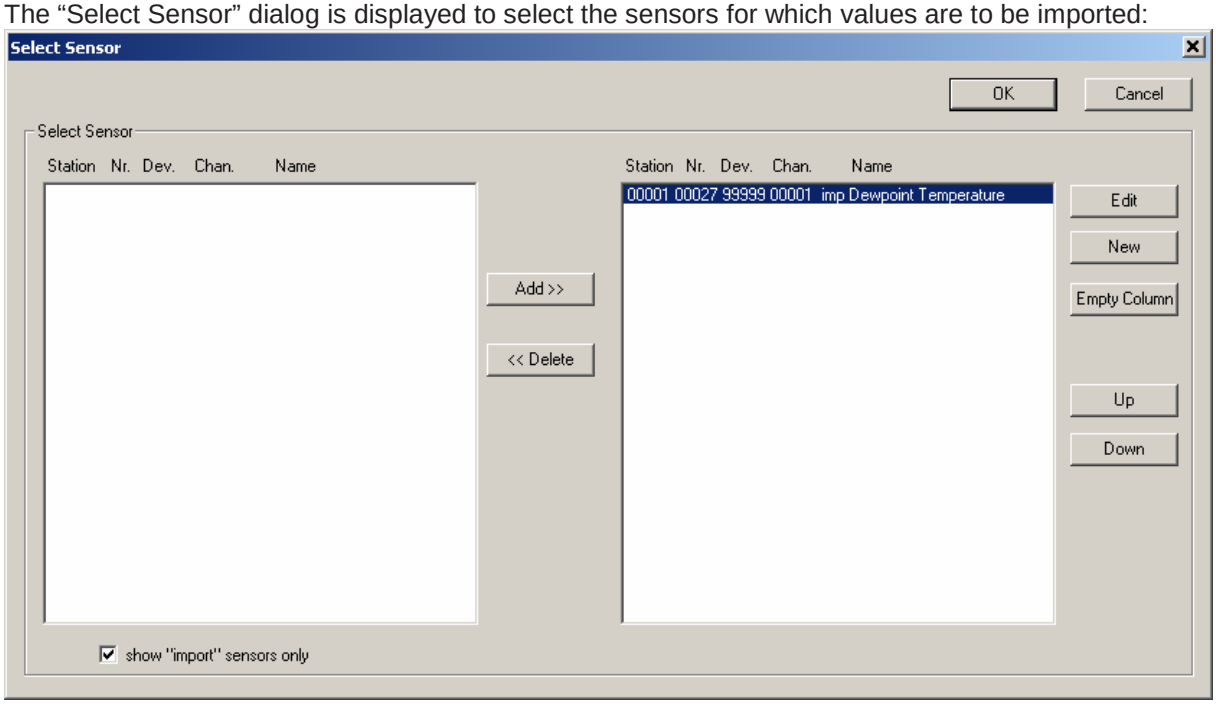
Add sensors to be exported in the order they should appear in the export file (which is the order they are displayed on the left side).




- **Column Name**: the name of the column (default : sensor name and unit). This name will appear for the column in the first line if “include header” is specified in the Export Job settings.
- **Value mapping**: value mapping (offset and scale or table mapping) to be applied to exported values
- **Convert**: the measure values for the exported sensor can be converted as follows :
  - **Don’t convert**: the value will be written as “floating point” value
  - **To bool**: the value will be converted to a bool value (false = 0, true =1) depending on “**Compare operator**” and “**Compare value**”
  - **To int**: the value will be converted to a (signed) integer value

## *7.5 Import Sensors*

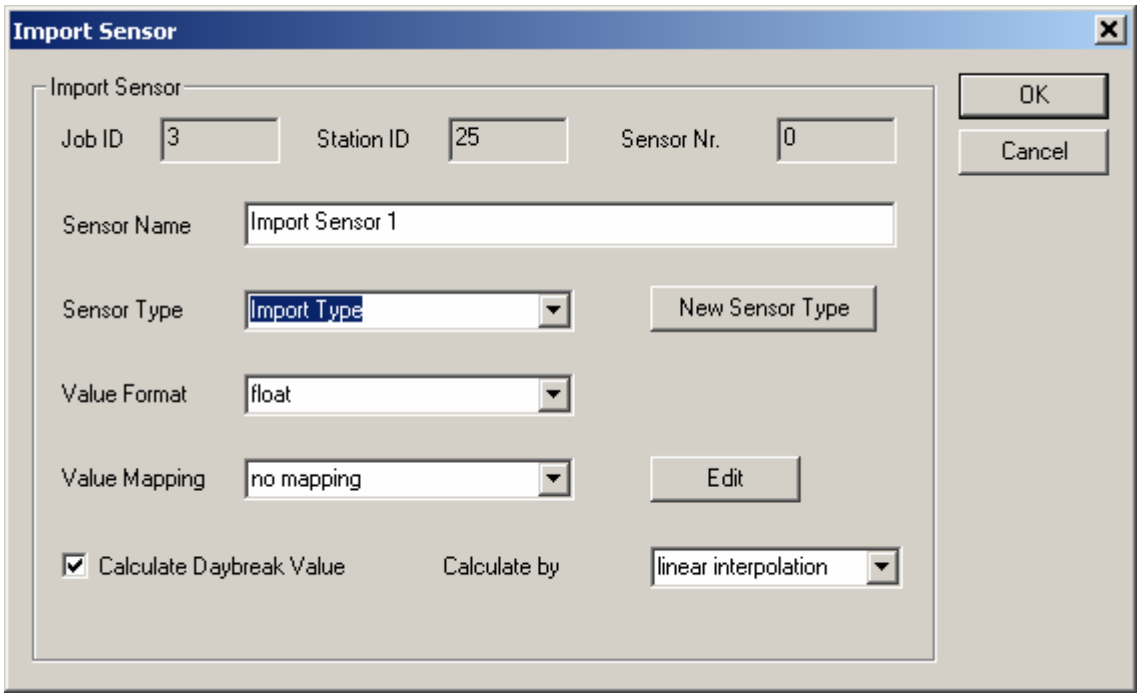
The “Select Sensor” dialog is displayed to select the sensors for which values are to be imported:




If a station is configured for the export job, only sensors of that station are displayed to be selected. If **show “import” sensors only** is checked (which is the default), only “import” sensors (that have been created using “New”) are displayed for selection.

**Note:** for template jobs, an appropriate dialog to assign logical sensor types instead of “real” sensors is displayed

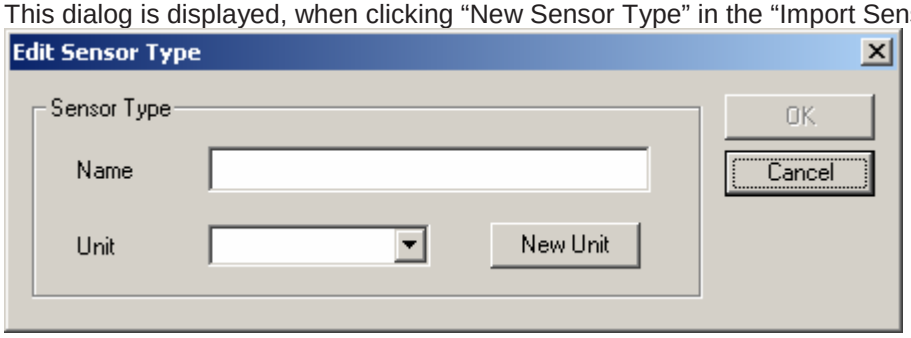
For every selected sensor (or when clicking “New”), the “Import Sensor” dialog is displayed:




- **Sensor Name**: the name of the sensor
- **Sensor Type**: the type of the sensor. Select the type from the listbox, or click “New Sensor Type” to create a new type (see below). Unit information is associated with the Sensor Type
- **Value Format**: the format of the value to be imported: float, bool or int
- **Value Mapping**: the value mapping (offset and scale or table mapping) to be applied to values when imported
- **Calculate Daybreak Value**:: calculate a “day break” value (local time with DST, as interval start value for days (e.g. diagrams in day/week/month interval) if there is a value from the day before available.
- **Calculate by**: calculation type of daybreak value : linear interpolation, last (day before) value or next (current day) value for value at daybreak.

### 7.5.1 New Sensor Type

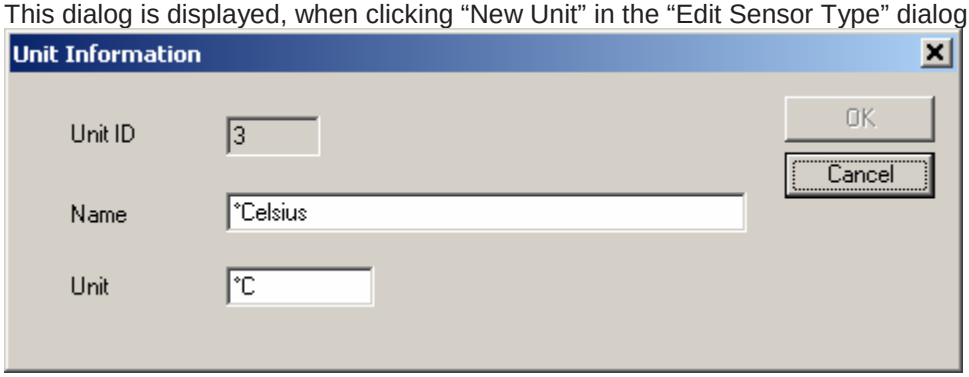
This dialog is displayed, when clicking “New Sensor Type” in the “Import Sensor” dialog:




- **Name**: the name of the sensor type
- **Unit**: the unit for this sensor type. Select the Unit from the listbox, or click “New Unit” to create a new Unit (see below).

### 7.5.2 New Sensor Unit

This dialog is displayed, when clicking “New Unit” in the “Edit Sensor Type” dialog :

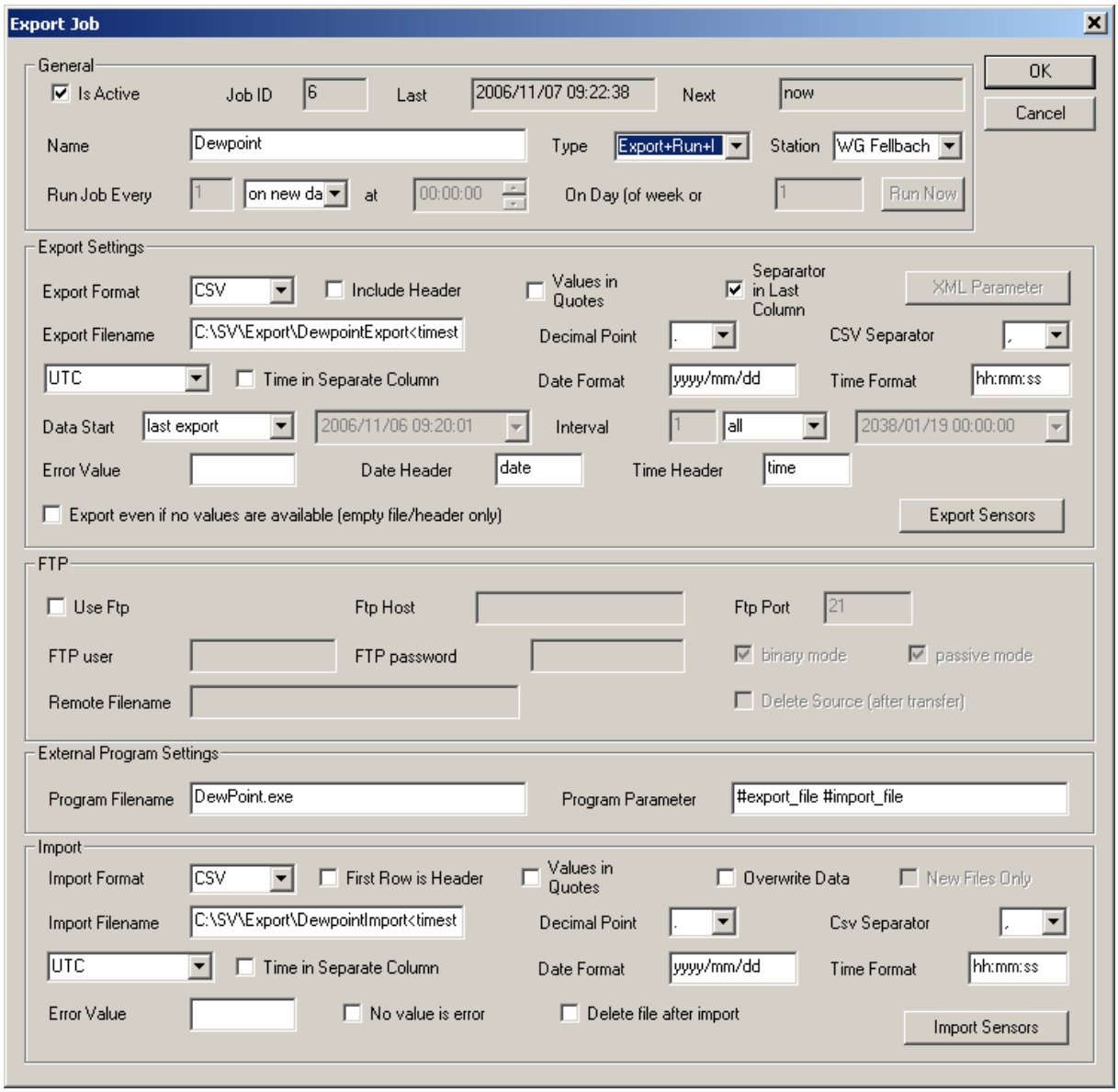



- **Name**: the name for the Unit
- **Unit**: the unit (symbol or abbreviation) for the unit

## *7.6 Dewpoint Calculation*

Dewpoint calculation is done by the program “Dewpoint.exe”. The calculation is done using an “Export/Import” Job type “Export+Run+Import” :

e.g.:




Following parameter must be set for the dewpoint calculation :

Export:

- Export Format: CSV (default)
- Include Header: No!
- Values in Quotes: No (default)
- Separator in Last Column: No (default)
- Export Filename: a valid Filename
- Decimal Point : '.' (Point) (default)
- CSV Separator: ',' (Comma) (default)
- Timezone: must be the same for export and import, but does not matter for dewpoint calculation (best set to UTC) (default)
- Time in Separate Column: No (default)
- Date Format : yyyy/mm/dd (default)
- Time Format: hh:mm:ss (default)
- Data Start: last export
- Interval : all
- Error Value: (none/default)
- Date Header: date (default)
- Time Header: time (default)
- Export even if no values are availale : no (default)

- Export Sensors : dewpoint calculation needs the following 2 sensor types being exported in this sequence :
  1.) Temperature in ℃
  2. ) rel. Humidity in %

External Program Settings:

- Program Filename: Dewpoint.exe (assuming the program file is in the same folder as SmartCom.exe)
- Program Parameter : #export_file #import_file (default)

Import:

- Import Format: CSV (default)
- First Row is Header: No!
- Values in Quotes: No (default)
- Overwrite Data: No (default)
- Import Filename: a valid filename
- Decimal Point : '.' (point, default)
- CSV Separator : ',' (comma, default)
- Timezone: same settings as for Export above – UTC recommended (default)
- Time in Separate Column: no (default)
- Date Format: yyyy/mm/dd (default)
- Time Format: hh:mm:ss (default)
- Error Value: (none/default)
- No value is error: no (default)
- Import Sensors: dewpoint calculation will write one "column" with measure values to the output file, which is the
  1.) dewpoint temperature in ℃

## *7.7 Integrate "plant disease models"*

To integrate plant disease models like "oidium" or "botrytis", you can use the pre-defined "Template" jobs as described above.

# 8 Alarms

SmartView3 can notify a group of recipients if an alarm condition occurs.

Alarm conditions are:

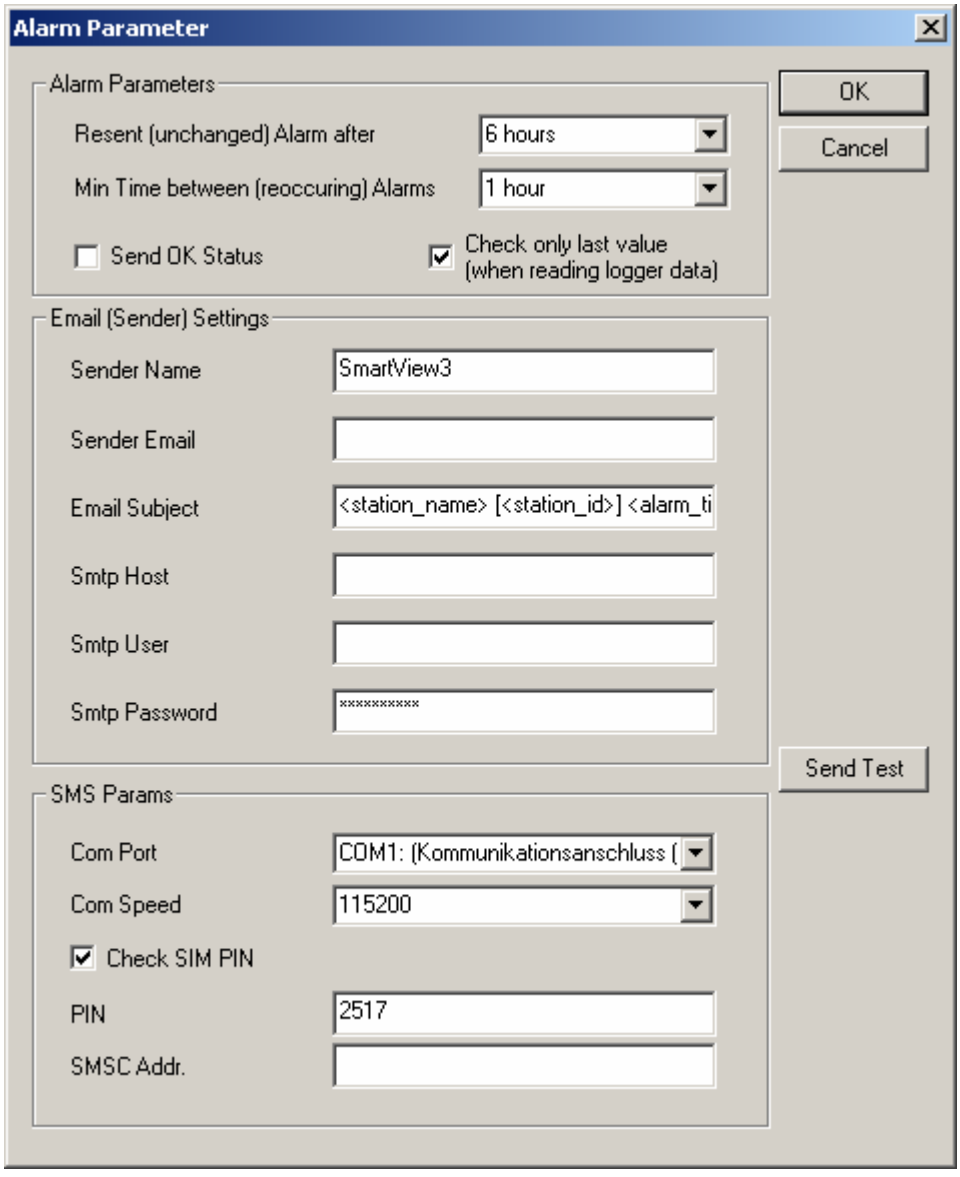
- An error occurred when polling a station (station could not be polled)
- An error occurred when reading values for a specific sensor (the sensor could not be read)
- An error occurred when processing an export/import job
- A sensor value exceeded configured minimum/maximum values
- A sensor value matched a “status” alarm or warning condition

Alarms can be sent as Email, or (through an attached GSM modem or mobile phone) as SMS message.

Alarms are configured via menu “Edit->Alarms”.

If this menu item is selected the first time, the “Alarm Parameter” dialog is displayed to configure the settings needed to send an email.

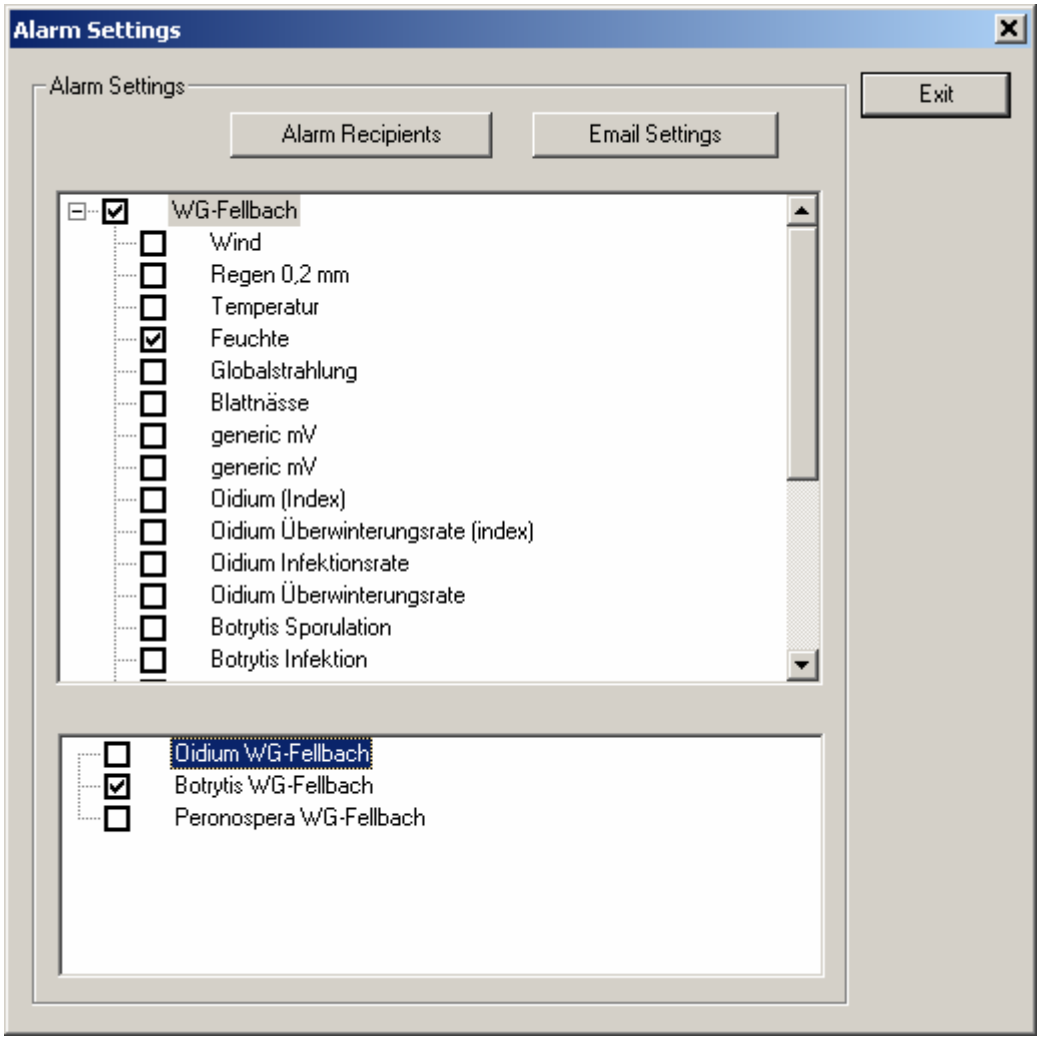
## *8.1 Alarm Parameter*



- **Alarm Parameters**:
  - **Resent (unchanged) Alarm after**: specifies the time period after which an alarm condition **that did not change** (e.g. the sensor value is still above the confiogured maximum) will be sent again.
  - **Min. Time between (reoccurring) Alarms**: specifies the minimum time period that has to expire before a specific alarm (for a station/device/channel/reason) will be sent again **if the alarm does occur again** (e.g. the sensor value was back below the configured maximum, and went up above the configured maximum again).
  - **Send OK Status**: send an "OK" alarm if the alarm condition is cleared (e.g. the value is below the configured maximum)
  - **Check only last value (when reading logger data**): when reading data from a data logger memory, or when importing more than one data row for a sensor, only the last (newest) sensor value is checked for alarm condition. Otherwise all read/imported values are checked for alarm condition.
    Note: if all read/imported values are checked, only the first occurance of an alarm condition is reported depending on the "Min Time between alarms" setting.
- **Email (Sender) Settings**:
  - **Sender Name**: the sender name as it will appear in the email send by SmartView3
  - **Sender Email**: the senders email address – i.e. the email address of the smtp account you use to send the email
  - **Email Subject**: the subject line that will be used for all alarm emails: Following "Tags" are supported (and will be replaced by the appropriate values when the alarm email is sent) :
    <station_name>: the name for the station that caused the alarm
    <station_id>: id for the station that caused the alarm
    <alarm_time>: time when the alarm occurred
    <alarm_reason>: reason for the alarm (error/warning/alarm…)
  - **Smtp Host**: the Smtp (email) host used to send the email
  - **Smtp User**: the user used to send the email (and to log on to the smtp server)
  - **Smtp Password**: the password for the smtp account.
- **SMS Params:**
  - **Com Port** on which the modem or mobile phone is attached. A list of available Com ports is displayed for selection.
    Note: this can be a "real" com port or a virtual com port, e.g. if the mobile phone is connected via Bluetooth (most mobile phones that support Bluetooth will appear as a Bluetooth com port even if no special drivers for the mobile phone are installed)
  - **Com Speed:** connection speed for com port. Note: setting the speed is important if the gsm modem is set to operate on a specific speed (e.g. not set to "auto speed"), or if the gsm modem does not support the pre-configured speed if 115200 baud.
  - **Check SIM PIN:** specifies weather the SIM Pin is checked/set via appropriate AT commands before the modem/mobile phone is used by the application.
    Note: SIM PIN checking is usually only necessary for serial attached gsm modems. Mobile phones usually require to enter the PIN manually before the mobile phone can be used at all.
  - **PIN:** the SIM PIN to be used (if "Check SIM PIN" is enabled)
  - **SMSC Addr:** the Address (phone number) of the SMS Service Center of the mobile service provider. Usually, this can be left empty – the pre-configured SMSC address (usually pre-configured with the SIM card) will be used per default.

If you have entered this basic information, the Alarm Settings dialog is displayed :

## 8.2 Alarm Settings




Here you can activate/deactivate alarms for stations and sensors of a station, and for export jobs, by clicking in the appropriate check box.

Alarms are always send to an “alarm recipient group”, i.e. to one or more (email) recipients.

If you did not configure any alarm recipients yet, the Alarm Recipient dialog is displayed (see below).

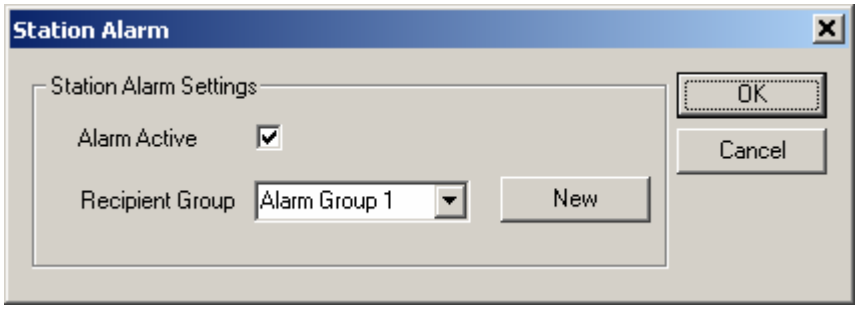
Alarms can be activated for a “station”, or for a specific “sensor” of a station.

**Error Alarms**:
Activating alarm for a station will cause alarms being generated for error conditions, like if a station could not be polled, or if a sensor of a station reported an error – and are send to the “alarm recipient group” configured for the station. “Error” alarms for a specific sensor (e.g. a device reported an error condition for a sensor) are only generated (and send to the alarm group configured for the station) if alarms are activated for the station AND the sensor.

**Value Alarms**:
Activating alarms for a sensor will cause alarms being generated because a sensor value was below a specified minimum value, a sensor value was above a specified maximum value, or a sensor value was “mapped” to a warning or alarm condition via “Status Mapping” (see below). Sensor “value” alarms are sent to the recipient group configured for the sensor. Only if alarms are activated for the station as well, error conditions will cause an alarm to be generated (see above).

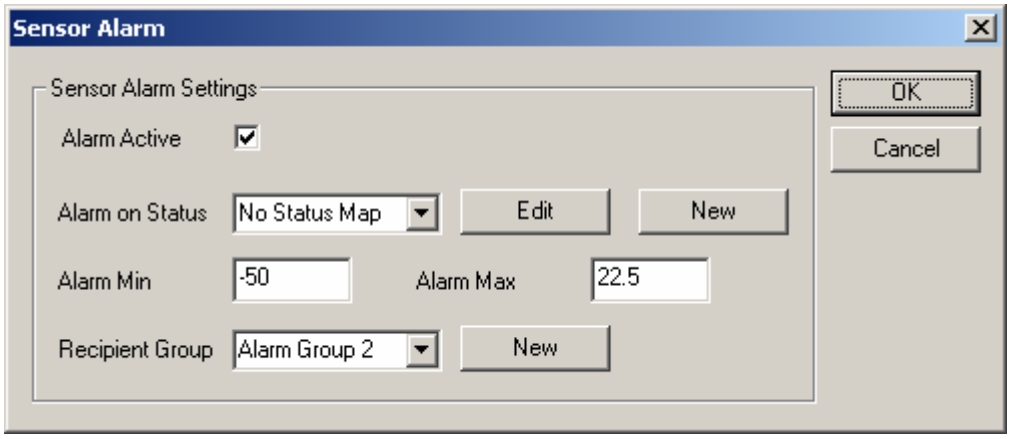
If you activate alarms for a station, the “Station Alarm” dialog is displayed.
If you activate alarms for a sensor, the “Sensor Alarm” dialog is displayed.

### 8.2.1 Station Alarm (for error alarms)




You can select the alarm recipient group for alarms for this station from the box, or create a new group. A “station alarm” is generated if a station could not be polled for new data, or if the station reported an error for a specific sensor (if alarms for that sensor are active as well).

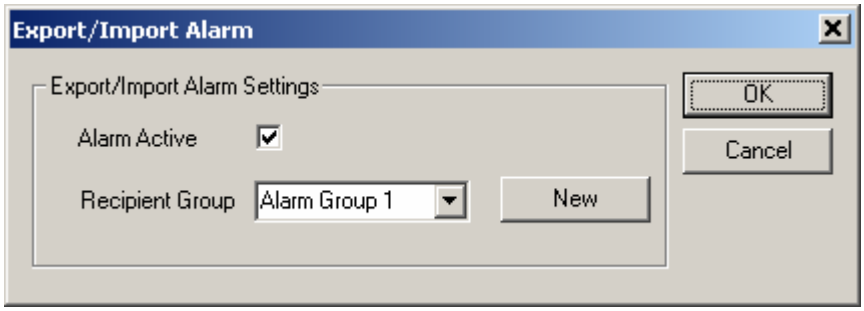
### 8.2.2 Sensor Alarm (for value alarms)




Here you can set/edit the “status mapping” to be used to determine a warning/alarm condition, or you can set the lower and upper alarm limits for the sensor. Default is the minimum/maximum value a sensor reports.

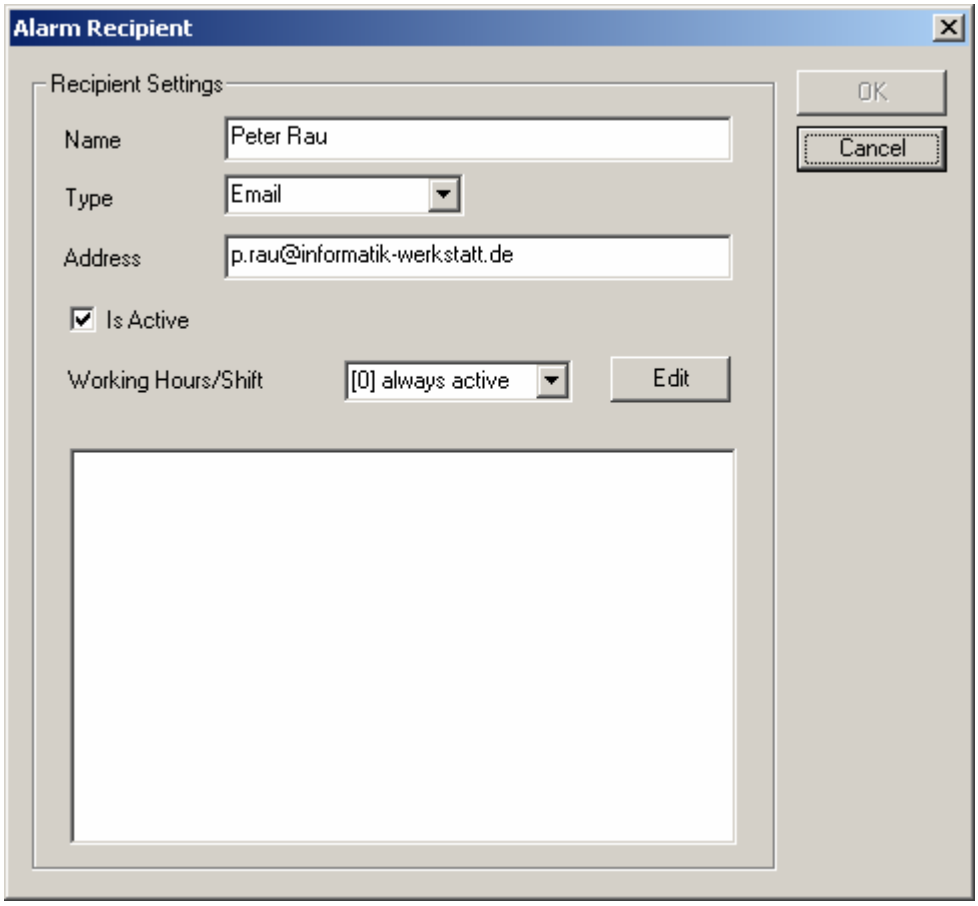
Besides that, you can select the alarm recipient group for alarms for that sensor, or create a new group.

### 8.2.3 Export Alarm




You can select the alarm recipient group for alarms for this Export/Import job from the box, or create a new group.
An “Export/Import” alarm is issued if the export/import did fail, i.e. no data could be read from the import file.
Alarms for imported sensor values can be set using the “Sensor Alarm” for the import sensor (see above).

### 8.2.4 Alarm Recipient



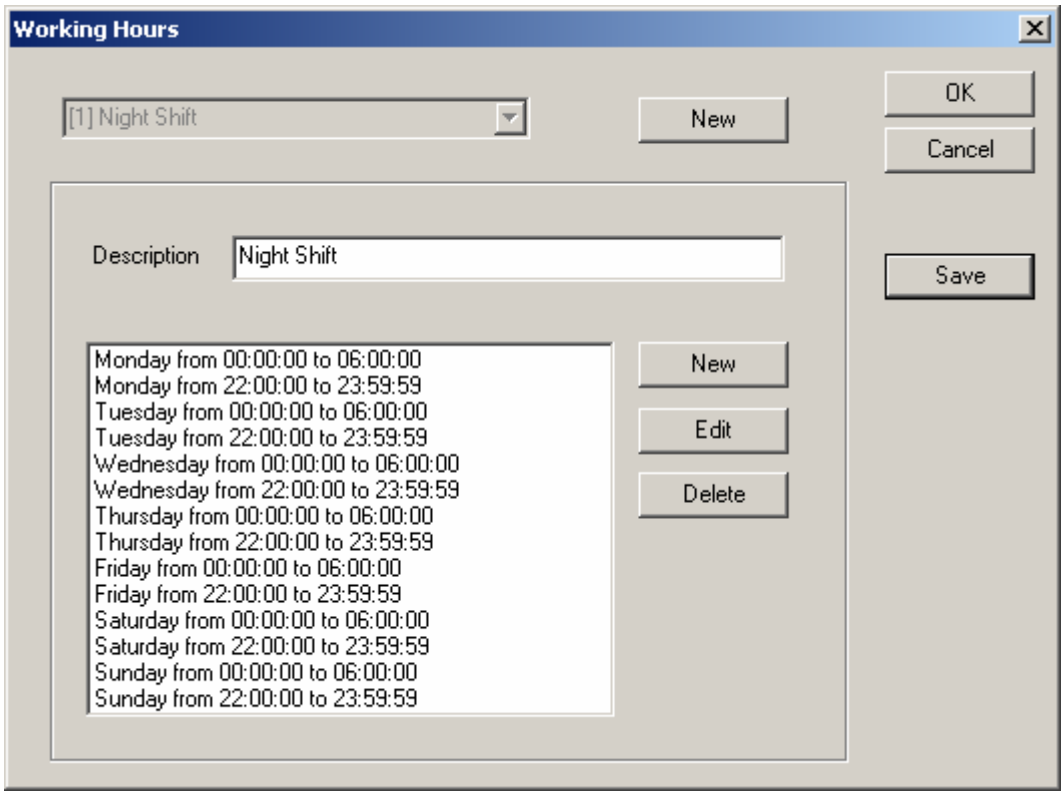

Enter the name and the email address (for type “Email”) or mobile number (for type “SMS”) for the alarm recipient here.
Note: it is recommended to specify mobile phone numbers in international format, e.g. for a recipient with a German mobile phone it would be +491729876543

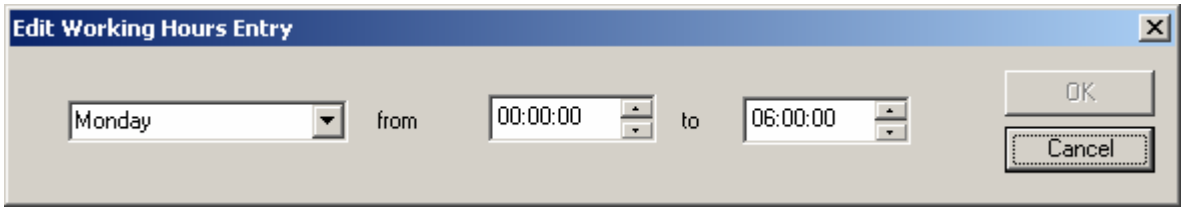
This (first) alarm recipient is then added to the first recipient group :

You can assign a “Working Hours” definition to an alarm recipient. An alarm recipient will only receive alarms when “on duty” if a Working Hours definition is assigned.

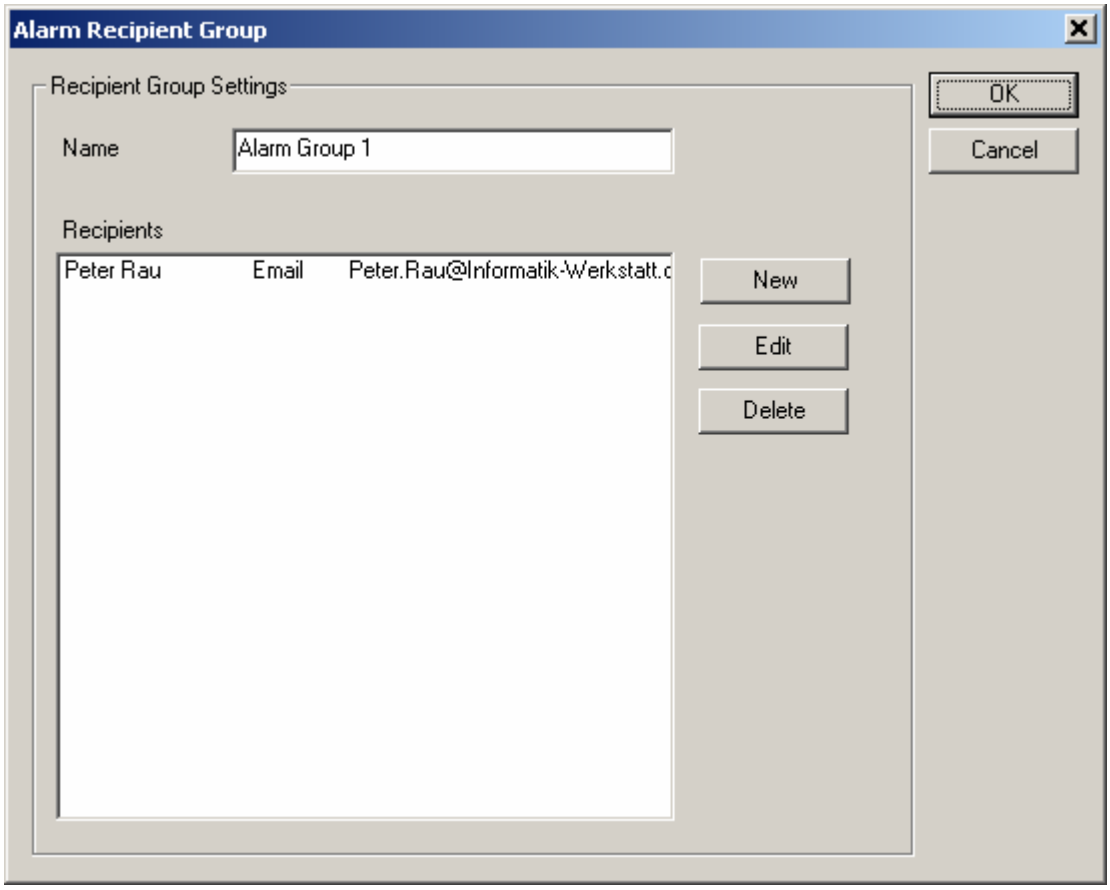
### 8.2.5 Working Hours

Working Hours are configured by specifying start/end times for each day of the week. There can be multiple entries for a day of week, to cover “night shifts”.


Working Hours
[1] Night Shift
New
OK
Cancel
Description
Night Shift
Save
Monday from 00:00:00 to 06:00:00
Monday from 22:00:00 to 23:59:59
Tuesday from 00:00:00 to 06:00:00
Tuesday from 22:00:00 to 23:59:59
Wednesday from 00:00:00 to 06:00:00
Wednesday from 22:00:00 to 23:59:59
Thursday from 00:00:00 to 06:00:00
Thursday from 22:00:00 to 23:59:59
Friday from 00:00:00 to 06:00:00
Friday from 22:00:00 to 23:59:59
Saturday from 00:00:00 to 06:00:00
Saturday from 22:00:00 to 23:59:59
Sunday from 00:00:00 to 06:00:00
Sunday from 22:00:00 to 23:59:59
New
Edit
Delete


Edit Working Hours Entry
Monday
from
00:00:00
to
06:00:00
OK
Cancel

### 8.2.6 Alarm Recipient Group

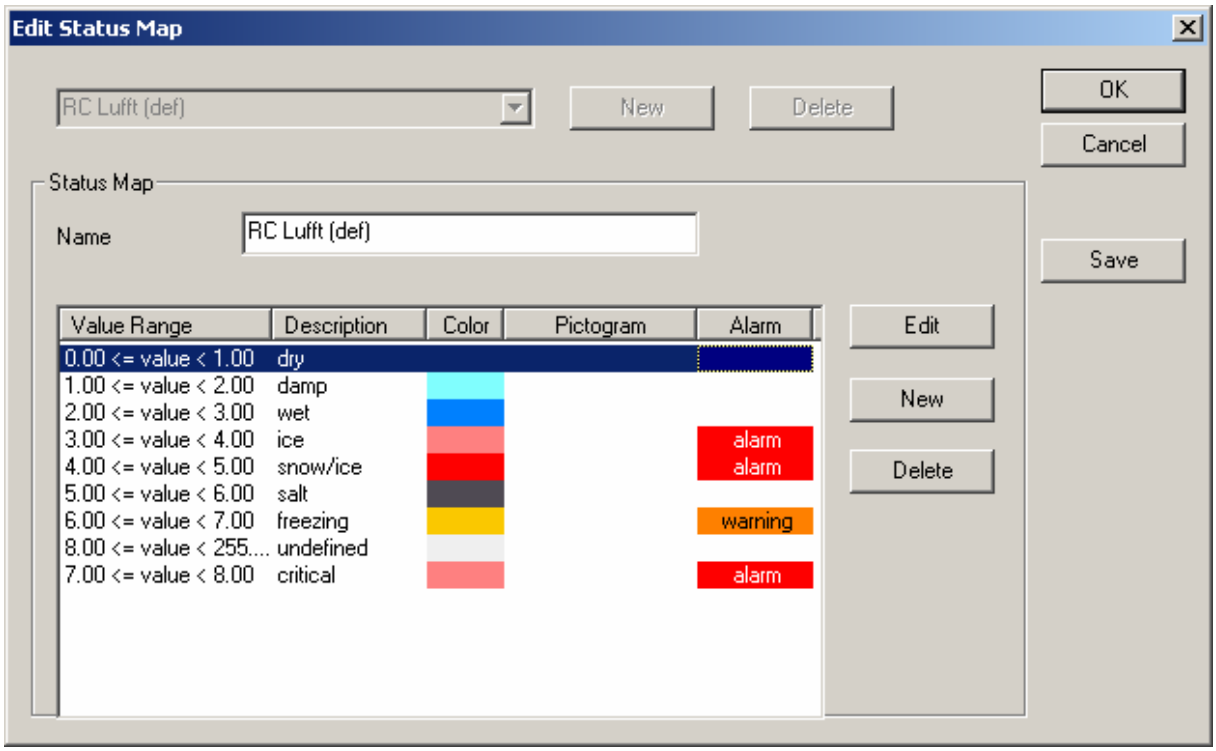



You can add more recipients to this alarm group.

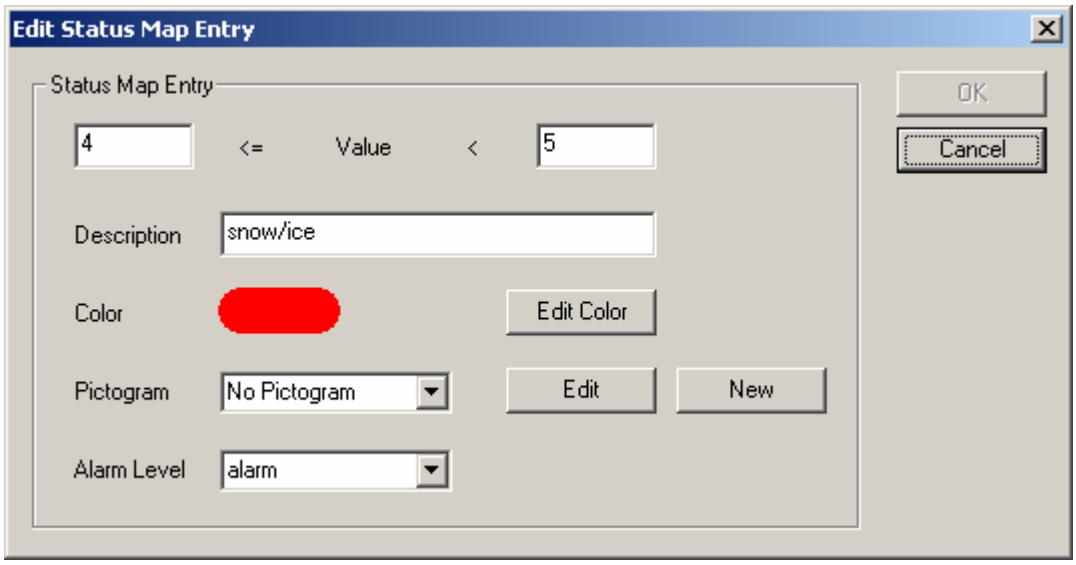
# 9 Status Mapping

A “Status Map” is used to assign a color, a descriptive text, a pictogram and an alarm condition to a specific value. Status Maps are used in “horiz. Bar Diagrams”, “Color Code” displays, “Pictogram” displays and “Tables”, and in the sensor alarm configuration.

A “Status Map” is assigned to a “Page Element Sensor” using the “Edit Page Element Sensor” dialog.

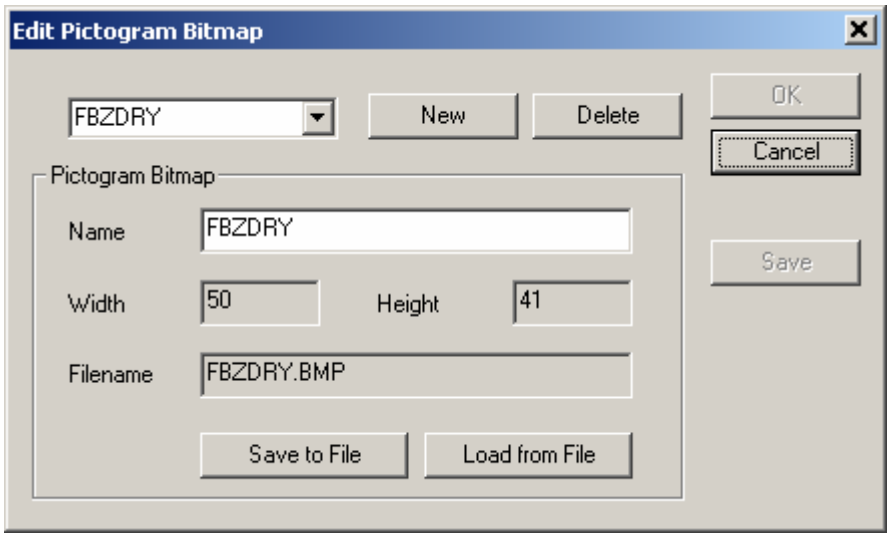



The description, color, and pictogram bitmap are assigned to a value range in “Status Map Entries”.




For “Pictogram” displays a bitmap must be assigned to all entries of a “Status Map”.

To assign a pictogram bitmap to a status map entry, a pictogram bitmap already loaded can be selected from the drop down box, or a new pictogram bitmap can be loaded from the hard drive by clicking “New”.



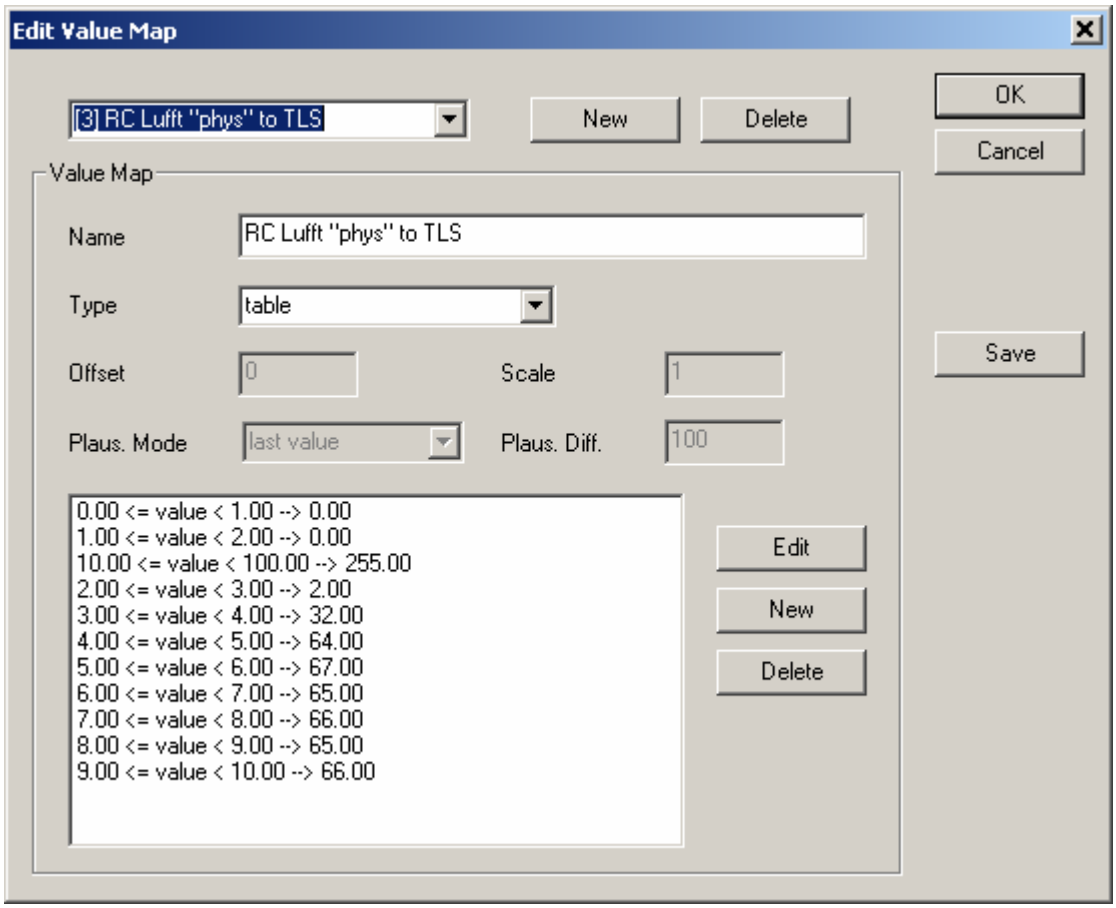

Pictogram bitmaps can be loaded from a file on the hard disc, or written to a file on the hard disc. The name for the pictogram bitmap can be edited.

**Note:** only bitmap files in “BMP” format are supported at the moment.

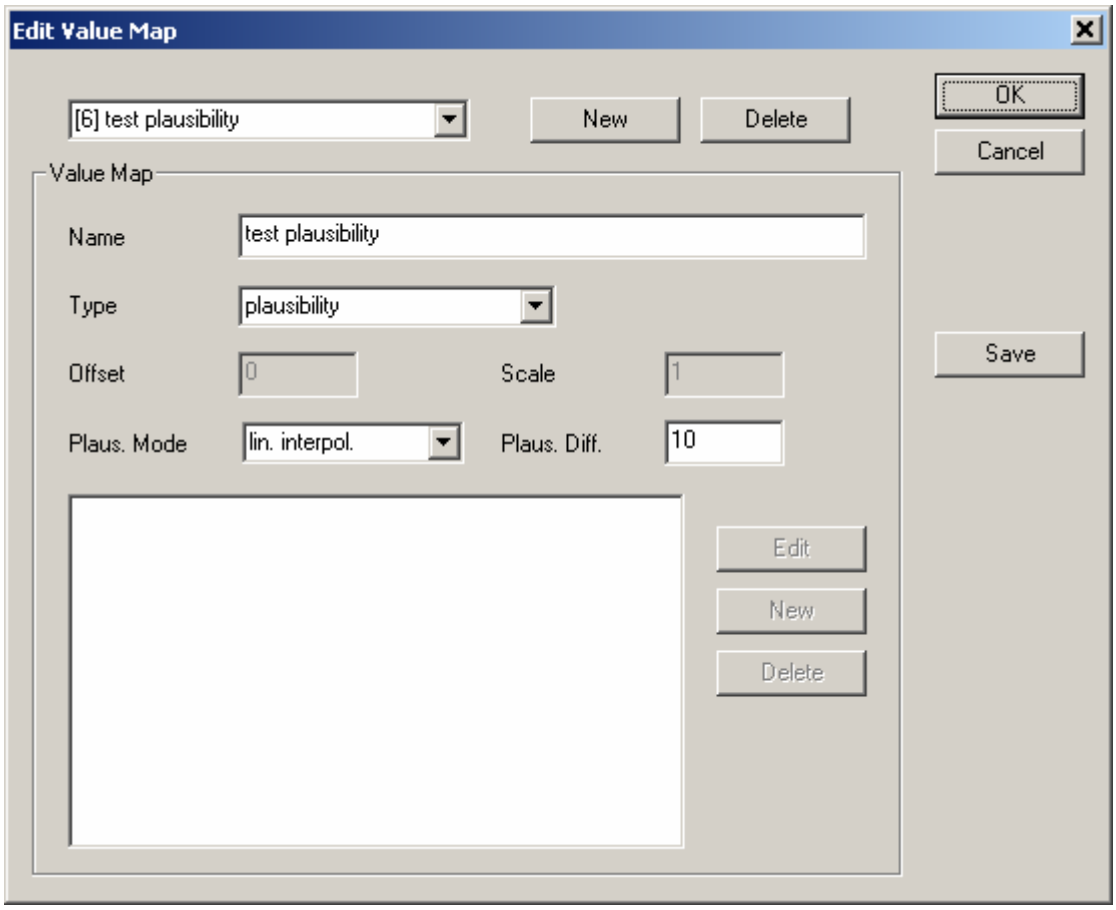
# 10 Value Mapping

Value Mapping can be configured for station sensors (will be applied when sensor values are read), export sensors, and import sensors. The dialog to configure value mapping can be selected from the appropriate dialogs (see above).

Value Mapping can be applied using “offset and scale”, or using a table to map values.




A special value mapping “plausibility checking” can be applied to UMB device sensors :

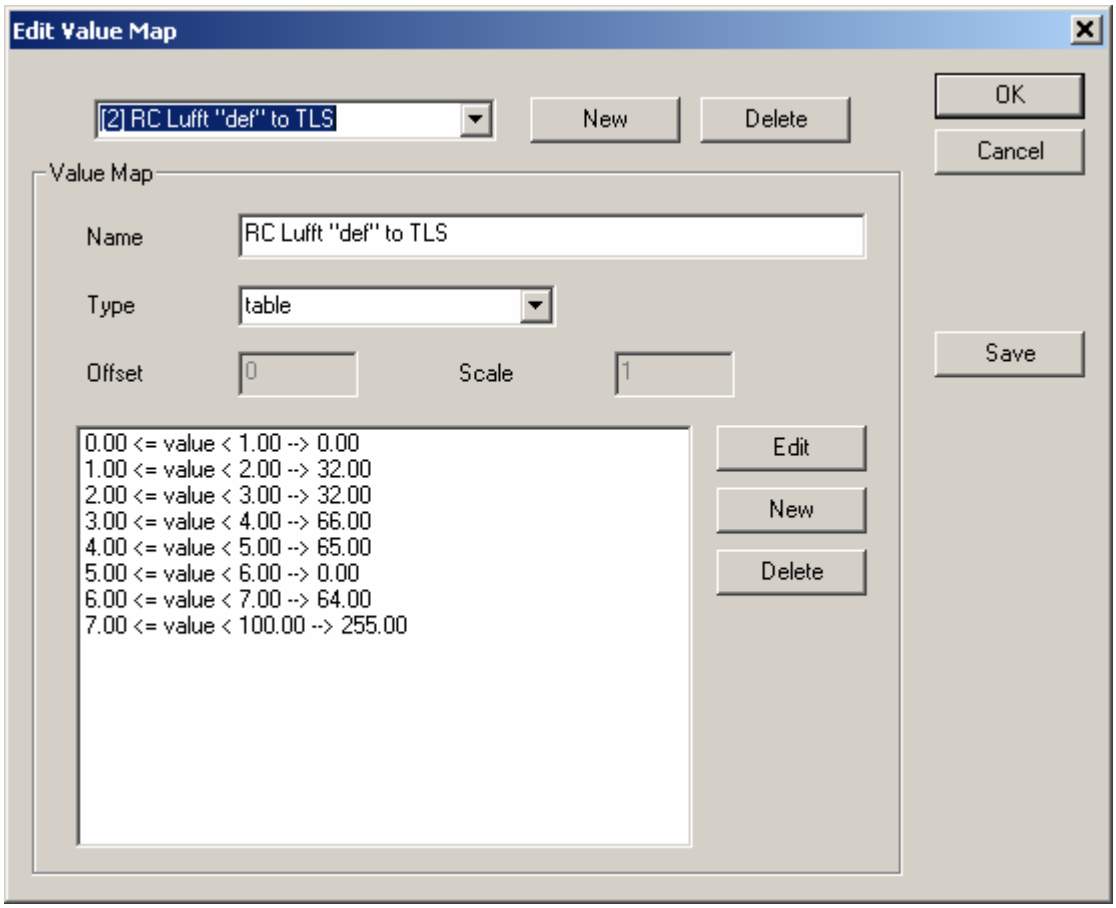



To check if the current value is a valid value, the last values for this sensor (within the last 24 hours) are read from the database. If there is no valid value for the sensor within the last 24 hours, no plausibility checking is applied.

There are 3 different plausibility modes:

- Error: if the difference between the actual value and the last value for this sensor is larger than “Plaus. Diff”, the value is set to “Error”
- Last value: if the difference between the actual value and the last value for this sensor is larger than “Plaus. Diff”, the value is set to the last value
- Lin. Interpol.: if the difference between the actual value and the last value for this sensor is larger than “Plaus. Diff”, the value is set to “Error”. If the next value is a valid value, the last (error) value is updated to a value between the 2 valid values (linear interpolation).

For “table” mapping, values within a specific range can be mapped to a new value. E.g. the Road Condition “Lufft coded” can be mapped to road condition in “TLS” format :




# 11 User and Privileges

With SmartView3, you can set up users, groups and privileges for groups and users. Only an administrator user (with the appropriate privileges) can add new users and/or groups and edit the privileges of groups and users.

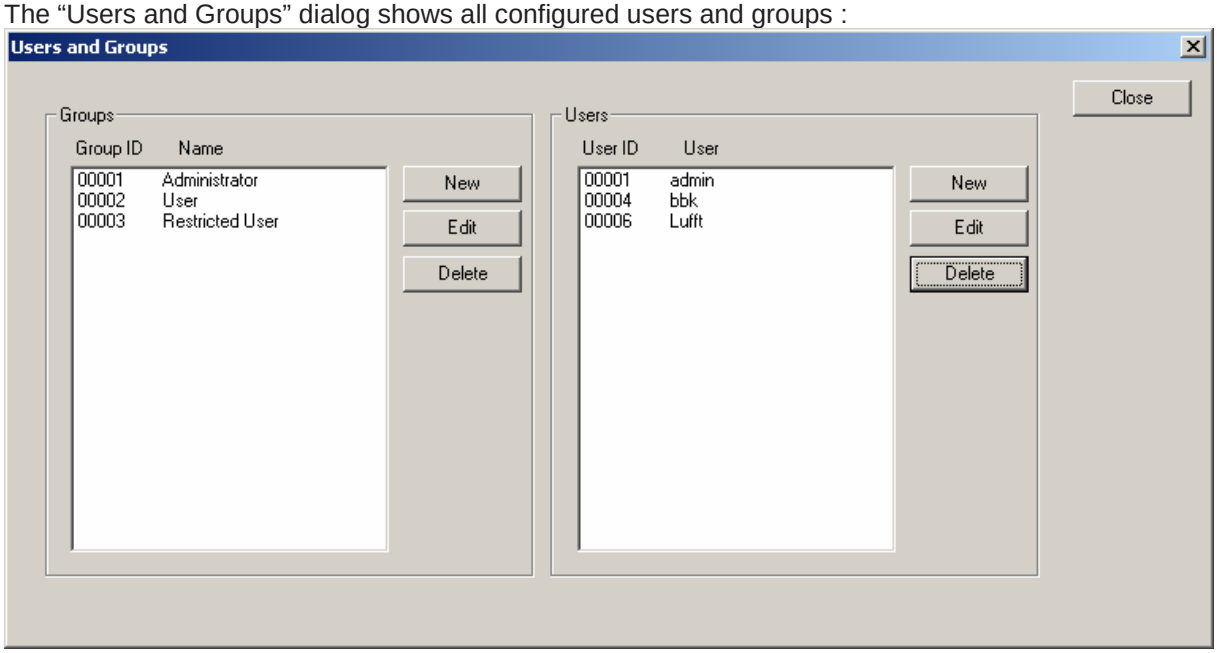
Group privileges are only used to apply default privileges to new users that are created - they are used as a “privilege template” for new users.

A “normal” user can only change its own user settings - i.e. the name and password etc. (not the access privileges).

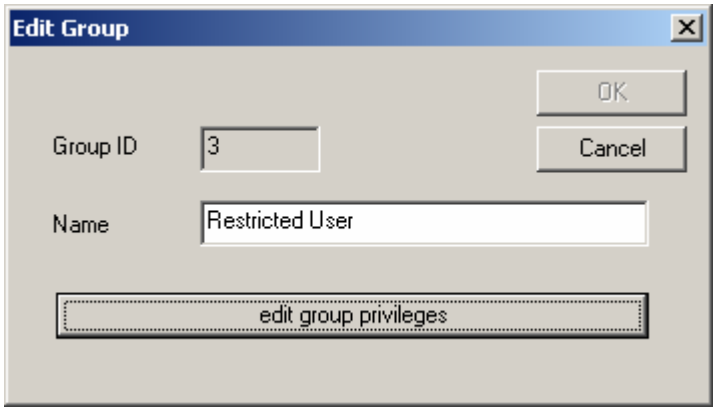
The following groups are created per default :
Administrator : access to everything
User : access to station configuration, site configuration and all (html) pages
Restricted user : view access to station configuration, site configuration and (html) pages

Note : access to pages on a generated web site relies on “page access” privileges only

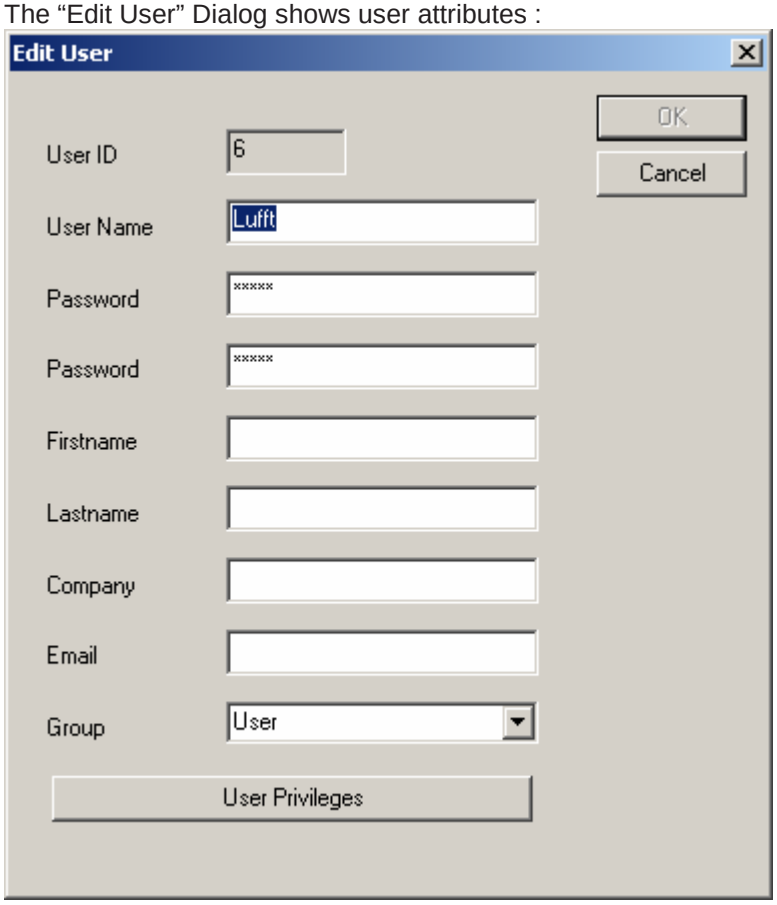
The “Users and Groups” dialog shows all configured users and groups :




The “Edit Group” dialog shows group attributes, i.e. the group id and name :

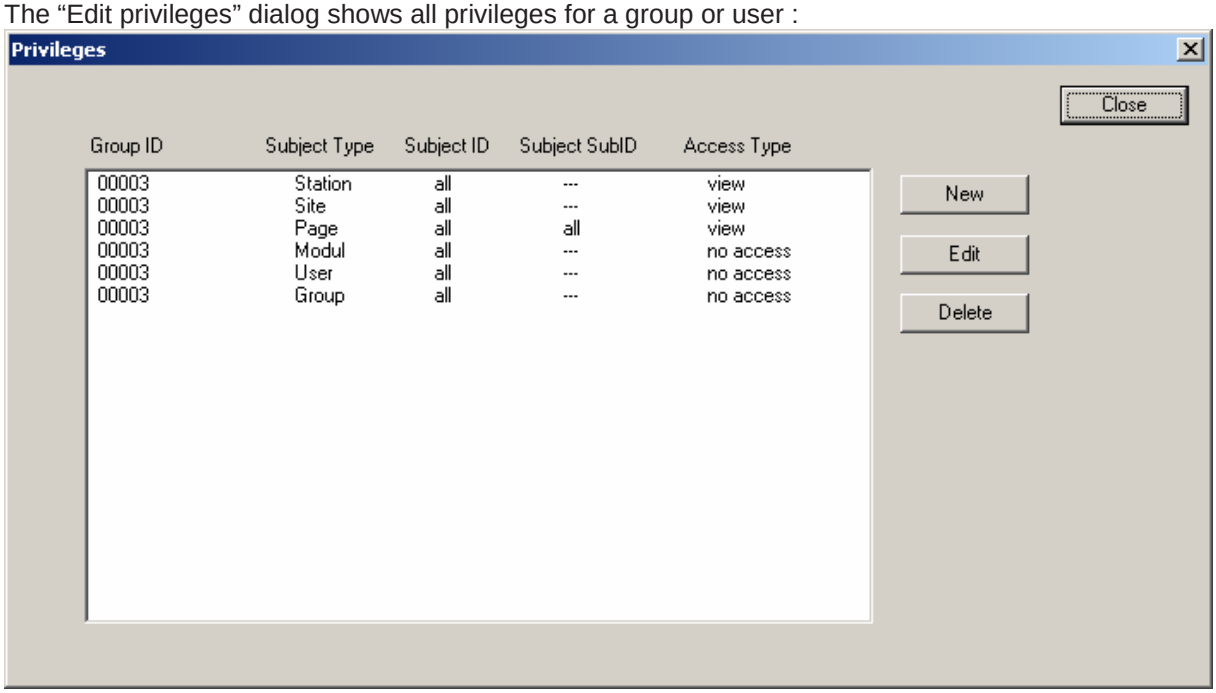


The “Edit User” Dialog shows user attributes :

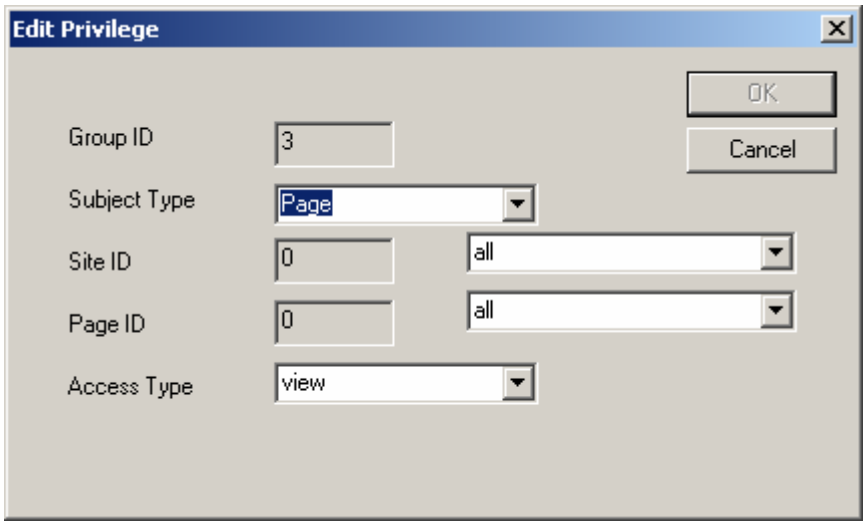



Changing a user’s group will apply the group’s privileges to the user.

The “Edit privileges” dialog shows all privileges for a group or user :



The Edit privilege button allows to edit (or create) a new privilege :




The “subject id’s” for privileges are shown depending on the privilege Subject (Page, Site, Station etc..)

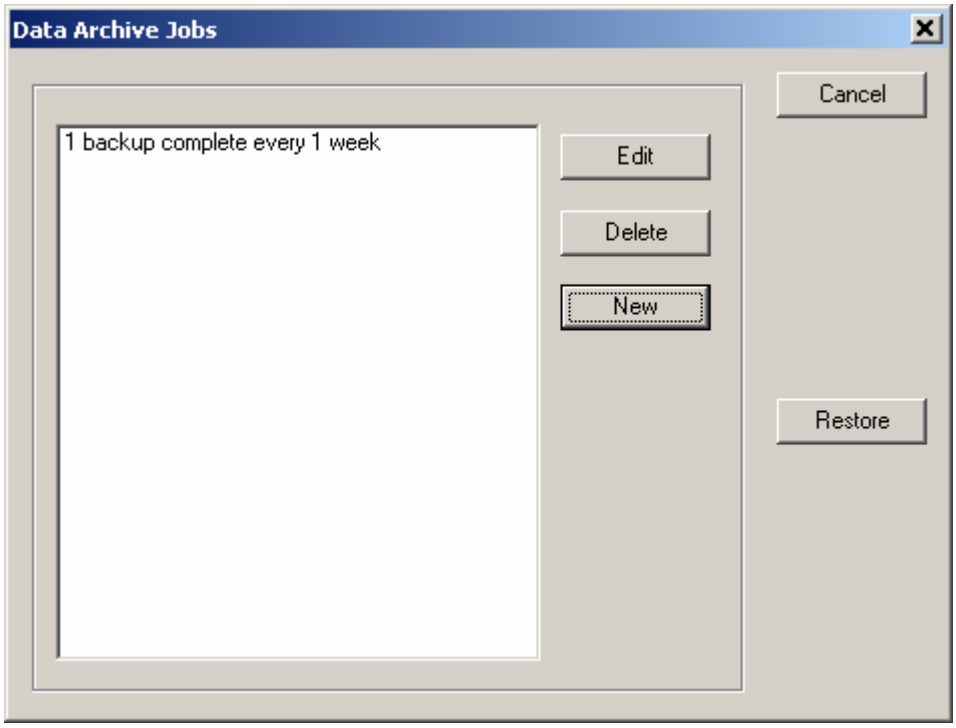
Privileges are always calculated “bottom-up”, i.e. if a user has a privilege for a specific page - this privilege is applied. If not the privilege for all pages of a site is applied if it exists. If it doesn’t exist, the privilege for all sites/pages is applied.

# 12 Backup/Archive Data

The Backup/Archive Data module can be used to automatically backup or compress the data in the database. This function utilizes the “mysqldump” utility and the mysql command line utility. It requires that the “mysql program path” is included in the Windows PATH environment variable (see “Installation”).

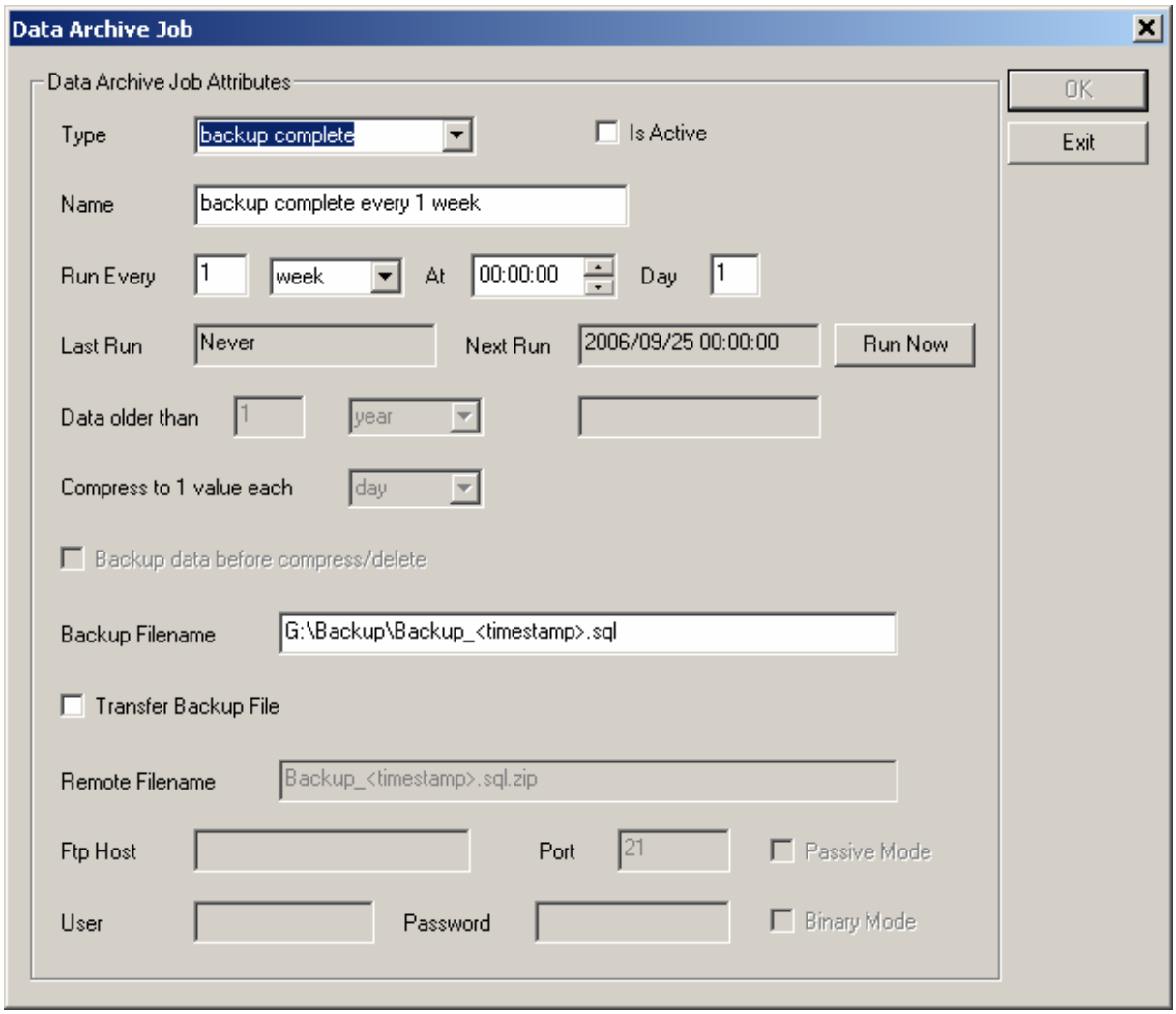
Backup and data compression can be set up through the main menu “Setup”, menu item “Backup/Archive Data”.

A dialog is displayed that lists all jobs configured to the module :



## 12.1 Configure Backup/Archive Jobs

On “Edit” or “New”, the Data Archive Job dialog is displayed :




- **Type:** can be “delete”, “delete pictures”, “compress”, “backup” or “backup complete”:
  ***delete**:* delete measurement data older than the configured time. The deleted data can be backed up automatically before it is deleted.
  ***compress:*** compress measurement data older than the configured time to 1 value each configured interval. The deleted data can be backed up automatically before it is compressed
  ***backup:*** backup all tables. The backup file will only contain “insert” statements for all SmartView3 database tables for all data in the tables, but no “CREATE TABLE” statements will be part of the backup. Existing entries in the database will not be overwritten, e.g. changes made to the database contents after the backup file was created (other than deleted entries) are not overwritten.
  ***backup complete:*** backup all tables with “DROP/CREATE TABLE” statements. This backup can be used to completely re-create the database. Note: as existing tables are dropped before they are re-created, all changes to the database after the backup file was created are lost.
  ***delete pictures***: delete picture data (CAM FTP/HTTP stations) from the database and from the harddrive (if appropriate). The deleted data can be backed up automatically before it is deleted.

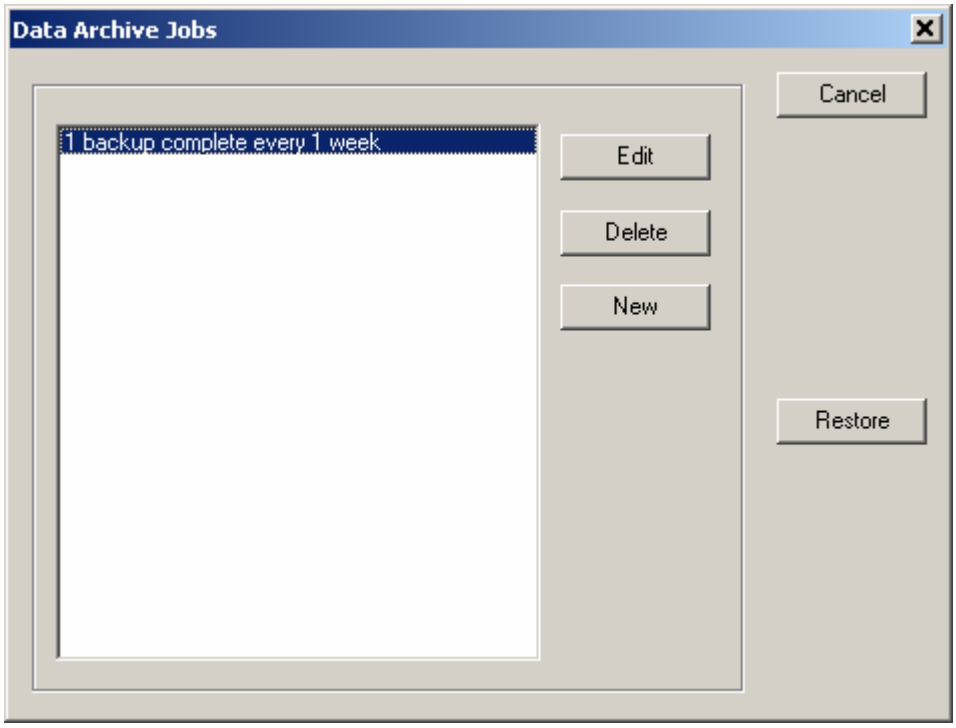
The backup files created are text files containing sql statements. They can be rather large in size, so please make sure there is sufficient disc space for the file(s) to be written.

All backup files created can be manually re-loaded to the database (or the database can be re-created manually for “backup complete” type backups) using the “mysql command line” utility and the “source <filename” command.
Besides the “.sql” file that contains the actual sql statements, a “.sql.info” file is created that contains information about the backup file that is used by SmartView3 when restoring a backup file.

- **Name:** name for the data archive job
- **Run every:** schedule information for this archive job
- **Last Run / Next Run:** time of last run and next run of the job
- **Data older than:** configure time offset for “delete” and “compress” type jobs
- **Compress to 1 value each**: for “compress” type jobs configure the “target” interval for compressed data
- **Backup data before compress/delete**: for “delete” and “compress” type jobs : create a backup of the data before deleting/compressing the data
- **Backup Filename:** name of the backup file. The filename may contain the special tags “<date>” or “<timestamp>” which will be expanded to the appropriate value when the backup file is created.
  ***Note:*** the filename should always end to .sql in order for the build in restore function to recognize the file.
  ***Note:*** as the backup file is a text file containing sql statements, it does take up a lot of disc space. Please ensure there is enough space on your drive where you write the backup file !
- **Transfer Backup File:** automatically transfer the backup file to an ftp host
- **Remote Filename:** filename on the ftp host (usually the same name as the Backup file, but you may include a path on the host)
- **Ftp Host:** the FTP host (name or ip address) the file is to be transferred to
- **Port:** the TCP/IP port for ftp on the server
- **Passive Mode:** use ftp passive mode
- **User:** the ftp user
- **Password:** the ftp password
- **Binary Mode:** use binary mode to transfer the file

## 12.2 Restoring Backup Files

There is a “Restore” button on the “Data Archive Job” Dialog, that can be used to restore a backup file created by SmartView3.




Clicking the restore button, a File Open dialog is displayed that shows all “.sql.info” files in the selected folder.

If the backup was created as part of a “compress” type job, the compressed values are removed from the database before the original values are restored by processing the backup file.

The restore function utilizes the mysql command line utility.

# 13 Styles.css

The CSS File serves as a central template file for all generated html files. Note that displaying the html files without corresponding CSS file will result in improperly formatted pages with incorrect font assignments.

The default “styles.css” file provided with SmartView3 can be saved to a file, modified, and loaded back into the database using the Edit Site Element dialog - see Edit Site Element.

Very little information regarding formatting (font style, size etc) is embedded in the html files. Most of it is done via the CSS file. For a detailed explanation of available html/CSS tags see the broadly available documentation, e.g. www.selfhtml.org.

Under normal circumstances, it will only be necessary to make very few modifications to the CSS file, e.g. to adapt the colours of the different html elements to your needs.

The relevant sections will be described later in detail.

SmartView3 will probably - in a later versions - include a built in editor, so that "manual" modifications per editor will no longer be needed. If you prefer to run SmartWeb without a site logo or with a logo of different size, some further changes will be needed.

Each CSS File consists of different sections:
CSS files group the formats by name. You will find a section named "divid_main" which holds the format specification for all ordinary text.

## *13.1 CSS sections used by SmartWeb*

divid_main:
this is the section where tables and diagrams are placed

divid_header:
area where the site's header is placed (usually top left)

divid_logo:
Section for the logo space (usually top right)
may be unused if logo is disabled (see below)

divid_navigation:
area where the navigation elements are placed (usually placed middle/left)

divid_time:
this is used fot the "generation" time on the bottom of each page.
If you want to make the time invisible, set the colour to the corresponding background colour

divid_print:
this format is used for all "Print Preview" pages

d_ta,d_th,d_td:
these sections are used for the data tables of the application
modify elements whenever modifications to the table's format are needed

p_ta,p_th,p_td:
same as above, used for print preview pages

body:
ordinary html text area - rarely used / unused within SmartWeb

link/visited/hover:

the link tags in web pages (where the html file points to another file) can be formatted individually

- link: an ordinary link
- hover: used if mouse cursor is over the link
- visited: used for a link that has been visited/used

Be careful when modifying the CSS file, errors in the file will cause unpredictable results in the way the html pages are displayed. In case of major malfunctions revert to the CSS version supplied with SmartWeb's installation package and carefully (step by step) repeat the changes made.

**Attention:**
Special care must be taken if the padding/position data are altered as they are crucial for the page layout.

## 13.2 Change colours or fonts

You can adjust the following tags freely to suit your needs. Modifications to those tags should be sufficient to adapt the pages to your needs and integrate it into a given corporate design.

The names of the tags should be self-explanatory: font-size font-weight, font-style , font-family, color, background-color, border-style, border-color, border-width (for the tables)

**Notes:**
font sizes always require a unit, either px (pixel) or pt (point)

colours are entered as hexadecimal rgb triple
examples:
000000 black
808080 grey
ffffff white
ff0000 red
00ff00 green
0000ff blue
There are some predefined text constants for colours, eg black, white, cyan etc that can be uses alternatively (see selfhtml.org)

## 13.3 Change logo size

The site is - out of the box - designed to work with a logo of 175*65 pixel.
whenever a logo changes in size, the following values need to be adjusted

divid_header:
margin 5,LOGO_WIDTH+5,5,5
height: LOGO_HEIGHT+5

divid_navigation:
top: set to LOGO_HEIGHT +10

If a setup without logo is required, simply decide what height the header section shall have - e.g. depending on its font size and content... (take this height as LOGO_HEIGHT) and apply the modifications describe above - with a LOGO_WIDTH of 0

Adjustment of the width of the navigation "window" :
modify these values

divid_navigation: width
divid_main margin set the last value to width + xxx

**Note:**
Whenever special national characters are to be used, make sure the font you select does supply them. Microsoft supplies a font named "Arial Unicode MS" that contains even japanese, korean and simplified chinese characters together with virtually all European special characters in one big font file.

## *13.4 Examples*

Default table, with background of section “divid_main” set to grey:

Table1

| Time | Temperature °C | Temperature °C | generic (Ohm) Resistance Ohm | generic (Ohm) Resistance Ohm | generic (Ohm) Resistance Ohm | Temperature °C | generic (Ohm) Resistance Ohm |
|---|---|---|---|---|---|---|---|
| 2005/09/26 00:05 | 24.49 | 52.86 | 100.18 | 199.51 | 298.63 | 24.77 | 428.66 |
| 2005/09/26 00:25 | 24.42 | 52.86 | 100.18 | 199.54 | 298.63 | 24.69 | 428.69 |

An example with different table options, thus resulting in the following table appearance:

Table1

| Time | Temperature °C | Temperature °C | generic (Ohm) Resistance Ohm | generic (Ohm) Resistance Ohm | generic (Ohm) Resistance Ohm | Temperature °C | generic (Ohm) Resistance Ohm | generic (Ohm) Resistance Ohm |
|---|---|---|---|---|---|---|---|---|
| 2005/09/26 00:05 | 24.49 | 52.86 | 100.18 | 199.51 | 298.63 | 24.77 | 428.66 | 509.71 |
| 2005/09/26 00:25 | 24.42 | 52.86 | 100.18 | 199.54 | 298.63 | 24.69 | 428.69 | 509.71 |

```
.d_ta
{
  font-size: 8pt;
  color: 0xFF0000;
  text-align:right;
  border-collapse:collapse;
  border-width:3px;
  border-style:solid;
  border-color:blue;

}

.d_th
{
  color: cyan;
  font-weight: bold;
  width:100px;
  border-top-width:1px solid black;
  border-collapse:collapse;
}
```

```
.d_td
{
  color: yellow;
  font-weight: normal;
  border-top-width:1px;
  border-top-style:dashed;
  border-color:white;
  padding-left:2.5em;
}
```

# 14 Appendix

## *14.1 MySql Database*

### 14.1.1 Parameters for MySQL

There are many parameters for MySQL that can be used to customize the operation of MySQL. These parameters are described in the MySQL Manual in detail. Please refer to the MySQL Manual for a complete description of all parameters.
The My.ini configuration file is stored in the MySQL directory (usually c:\Programs\MySql\MySql Server x.x\).

If you plan to use SmartView3 / SmartWeb to create a web site for visualization of your measure values, there is one parameter that should be set in the My.ini file :
**max_allowed_packet**
This parameter controls the maximum allowed length of an sql statement that may be passed to MySQL. With SmartView3, all data for a site, including logos and bitmaps, are stored in the MySql database (as hex strings in „blobs“).
The default value for this parameter is 1048576 byte (1 MByte). This will probably be not sufficient to store a bitmap (as we store this data as hex string, it needs double the amount of bytes than the original file).
We recommend to set this parameter to 16MB.
Section :
**[mysqld]**
**max_allowed_packet=16M**

other parameters of interest :
**datadir=**... specifies the path where MySql keeps the database. If you plan to keep a large number of measure values, you may want to keep the data on a separate hard drive that has sufficient space (see requirements). If you change the datadir parameter, please copy/move all data from the old path to the new path,

If you plan to optimize your database for space, and the hard drive or partition where you keep your database has not enough space left (you need at least as much free space as the database uses, as during optimization MySQL creates a copy of the database files), you may want to specify a different path for the temp files created during optimization/sort using the
**tmpdir=**
parameter. It is recommended to use a different hard drive for tmpdir in this case, as this speeds up the optimization process.

**query_cache_size=1M**
As there are quite some queries that are always executed the same way by Collector, setting this parameter is recommended. The default value is 0 (disabled). We recommend to set it to 1M.

### 14.1.2 Backup

Backup of the MySQL Database can be done either using the mysqldump utility provided with MySQL (see the MySQL manual for a detailed description) - which will dump the database content into a text file which can be used to restore the database and its contents again, or by backing up the MySQL files in the MySQL data folder. The MySQL data folder contains folders for each database, with the name of that database - the collector database is named “collector” by default (if you didn’t change that in the “Global Settings” dialog). To back up the data files directly, it is best to stop the collector program and the MySQL system service before backing up the files. See chapter “Database Backups” in the MySQL Manual.

Since Version 1.1.0, there is an automatic backup function build in SmartView3, that utilizes the “mysqldump” utility. See above !

### 14.1.3 Optimization

Only the table “sensor_values” contains significant amount of rows. If you keep many values in this table, optimizing the table reduces the disc space used by this table.

The table can either be optimized using the “OPTIMIZE TABLE” command in the MySQL command line utility, or by using the “myisamchk” utility. Using the “myisamchk” utility is probably more efficient, as you can set several parameters to use all available memory for the optimization.

Please see the MySQL manual for a complete description of both commands.

Note that while the table is optimized, the table cannot be accessed by other programs, i.e. Collector and other applications using the table may not run while the optimization is in progress.
The optimization may take quite some time, depending on the size of the sensor_values table and the speed of the CPU, and available memory.

If you use the myisamchk utility, stop the applications using the database and the MySQL service for optimal speed. Start myisamchk with appropriate parameters for your system.
Note that - besides space for a temporary index file in the MySQL data folder, there must be an equal amount of free space in the temporary folder (as configured for Windows). You may supply a specific folder to hold the temporary file to myisamchk. An example (assuming the MySQL bin folder is in the search path for programs, and you are in the MySQL data folder when entering the command) for running the utility is :

myisamchk.exe -v -r -S -O key_buffer=128M -O sort_buffer=128M -O read_buffer=2M -O write_buffer=2M -t h:\temp sensor_values

Processing this command on a sensor_values table with apox. 500 000 000 sensor values took on a Pentium 4 2 GHz system with 1 GByte Memory approx 6 hours.

Note: if the file size of the sensor_value table data and index file exceeds 4 GByte, the MySQL data folder and the temporary folder must reside on NTFS partitions (FAT32 partitions only support file sizes up to 4 GByte).

It is recommended to use a different hard drive to keep the temporary data (param -t to myisamchk.exe) to speed up the process.

## *14.2 Database Table Description*

The database will be created (or upgraded if appropriate) when SmartView3 is run the first time.

Programmers note : as we may add functionality to the software later, which may reqire changes to the database (e.g. adding additional columns/tables), we strongly recommend to address the specific columns needed in SQL queries (i.e. not using “select * from..” statements….), so changes to the database will not require changes to your query/software.

An application that retrieves data from this database analyze or view the measurement data will typically read information from the tables “sensor_values”, “station_sensors”, “stations”, “sensors” and “units” (Example see below).

The table “stations” provides information about the time a station was polled successfully, i.e. the column “last_data” is updated when data was retrieved successfully from a station. The table “sensor_values” holds the measured sensor values. Table “station_sensors” holds information about sensors attached to a station (the “link” between “sensor_values” and “station_sensors” is column “station_id”,”sub_id” and “channel”). Table “sensors” holds general information about the sensor types (name of sensor etc.). Table “units” holds the unit description for a sensor/value. All other tables contain mainly configuration information used by the collector software.

### 14.2.1 Table “sensor_values”

This table holds the sensor/measurement values retrieved from stations. For each sensor value read from a station a row is inserted. Note: depending on Opus setup, a single sensor may provide up to three different value types (average, minimum, maximum, sum). For every value type, a entry is generated. The value type is indicated by the value of column “value_type”.

| Column | Type | Description |
|---|---|---|
| measure_time | unsigned int | UTC timestamp when measurement was taken (will be the same as poll_time for “online data”, but different for historical data that was stored at the station..) |
| station_id | int | unique id for a station (refers to entry in table “stations” and “station_sensors”) |
| sub_id | int | identifier for “sub-station”, e.g. Opus id (refers to table “station_sensors” |
| channel | int | channel where sensor is attached to (refers to table “station_sensors” |
| value_type | int | 0=undefined,1=min,2=max,3=avg,4=sum |
| sensor_nr | int | sensor nr of station (refers to “sensor_nr” in table “station_sensors”) |
| poll_time | unsigned int | UTC timestamp when station was polled (station was connected). All values read from a station in a poll use the same value. |
| sensor_id | Int | id for the sensor (refers to entry in table “sensors”) |
| value | Float | the measured sensor value |
| error | int | error code for sample (0= no error) |

The primary key for this table is measure_time, station_id, sub_id, channel, value_type. This primary key ensures (as it is a unique key) that there is only one sensor value per given measure time and sensor and value type, and allows fast access to measurement values using measure_time, station_id, sub_id, channel, value_type as keys (or part of that list).

An index (station_id) is created on station_id, sub_id, channel, measure_time to allow fast access to (last) measure time for a sensor.

Another index (station_id_2) is created on station_id, sensor_nr and measure_time to allow fast access to measurement data using sensor_nr instead of station_id and sub_id (as provided by the first index station_id).

### 14.2.2 Table “stations”

This table holds information about stations. Some of the columns in this table are updated on every poll of the station. Note: depending on the type of station, an additional configuration entry in either table “opus_stations” or table “ppp_stations” is created.

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for a station |
| collector_id | int | id for collector program that polls this station. |
| name | varchar(255) | descriptive name for station |
| model | varchar(255) | model of station (e.g. Lufft Opus 200) |
| station_type | int | type of station.<br>Changed from Version 2.1 upwards (will be converted automatically when upgrading).<br>For Versions earlier than 2.1 :<br>1 = Opus 200/RS232,<br>2 = Opus 200/Modem.<br>From Version 2.1 on :<br>10 = Opus200,<br>12 = Axis |

| | | |
|---|---|---|
| connection_type | int | Type of connection to station (new in Version 2.1)<br>1 = RS232,<br>2 = TAPI/Modem,<br>3 = TCP/IP (only Axis Server at the moment)<br>4 = RAS/PPP (only Axis Server) |
| Location | varchar(255) | location (e.g. city/street etc.) |
| Altitude | Int | Altitude of location (information only) |
| Latitude | Float | Latitude of location (information only) |
| longitude | Float | Longitude of location (information only) |
| time_offset | Int | time offset/timezone for station |
| set_station_time | Int | set stations clock after reading data (0=no/false, 1=yes/true) |
| last_clock_sync | int unsigned | Last time when stations clock was set (UTC Timestamp). |
| is_active | int | station is active (is polled..) (0=no/false, 1=yes/true) |
| read_stored_data | int | read stations data buffer (0=no/false, 1=yes/true, 2=read all stored data) [not supported yet] |
| reset_station | int | 0 = do not reset station, 1=reset station on next connect (will be set to 0 by collector if reset has been done) |
| poll_summer | unsigned int | poll interval in secs (if params:operations_mode is summer) |
| poll_winter | unsigned int | poll interval in secs (if params:operations_mode is winter) |
| next_poll | unsigned int | UTC timestamp : time for next poll of station; is updated on every poll attempt |
| last_data | unsigned int | UTC timestamp : last time station was polled successfully (start of dial-out); is updated on every successful poll |
| transfers_ok | int | number of successful polls (statistical information) |
| transfers_err | int | number of failed polls (statistical information) |
| last_error | int | error code for last poll (0=OK…) |
| Deleted | int | Bool value indicating that station has been deleted (0=false - not deleted; != 0 = true - deleted) |
| alarm_active | int | Bool value indicating if alarm for station is active or not |
| alarm_group | int | Id of alarm group for station (key for table “alarm_group”) |
| time_type | int | (internal enum) time type of station (UTC/Local Time/ fixed offset/local time with DST) |
| device_timeout | int unsigned | Timeout for device I/O (short commands) |
| device_timeout_long | int unsigned | Timeout for device I/O (long commands) |
| last_import | Int unsigned | Last time data was imported for station |
| from_time | Int unsigned | Time offset for reading logger data |
| Modem_pool_id | Int unsigned | Id for modem pool to be used for this station |

### 14.2.3 Table “opus_stations”

This table holds additional configuration information for opus type stations

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for station (refers to entry in table “stations”) |
| phone | varchar(255) | phone number of Opus station (if attached to a modem) |
| com_port | int | COM (RS232) Port number if Opus is connected via RS232 |
| com_params | varchar(255) | COM Port param string , e.g. “baud=19200 parity=N data=8 stop=1” |
| wake_up | int | send wake_up command to opus (needed if Opus is in “sleep” mode) |
| check_config | int | check station configuration on every poll (0=no/false, 1=yes/true) |

| | | |
|---|---|---|
| config_change | unsigned int | UTC timestamp of last configuration change (not used) |
| cam_stationid | Int | Station id of associated AXIS station for cam pic |

### 14.2.4 Table “ppp_stations”

This table holds additional configuration information for stations that provide tcpip/ppp dial in service, e.g. AXIS stations (and maybe NTCIP type stations later)

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for station (refers to entry in table “stations”) |
| phone | varchar(255) | phone number for station |
| tcpip | varchar(255) | TCP-IP address of station (for TCP/IP connections) |
| port | int | TCP-IP port (for TCP/IP connections) |
| ppp_user | varchar(255) | user for ppp connection |
| ppp_passwd | varchar(255) | password for ppp connection |
| snmp_community | varchar(255) | community name for snmp connection (NTCIP) |
| ftp_passive_mode | int | Use ftp passive mode (0 = no, != 0 = yes) |

### 14.2.5 Table “sensors”

This table holds information about the sensors.

| Column | Type | Description |
|---|---|---|
| sensor_id | int | unique id for a sensor |
| name | varchar(255) | descriptive name for sensor (opus channel description !) |
| unit_id | int | id for unit of sensor. refers to entry in table “units” |
| scale | float | scale of sensor value |
| opus_sensortype | int | sensor type as defined in OPUS (used to detect sensor) |
| irs21_sensortype | int | IRS21 sensor type/channel |
| snmp_oid | varchar(255) | SNMP OID of a sensor (used for “NTCIP” stations only) |
| station_type | Int | Station/device type for sensor type |
| template_id | Int | Sensor Type Template (log. Sensor type) for this sensor |

### 14.2.6 Table “units”

This table holds information about units

| Column | Type | Description |
|---|---|---|
| unit_id | int | unique id for a unit |
| name | varchar(255) | descriptive name for unit (e.g. “temperature”) |
| unit | varchar(255) | Unit string (e.g. ℃, m, m/s, %, mm, imp/min, °) |
| opus_unit | char(255) | Unit string as defined in Opus (2 chars in Opus200, 6 chars in Opus208) |

### 14.2.7 Table “station_sensors”

This table holds information about the sensors attached to a station. The sensors defined for a station in this table will be polled, and an entry in table “sensor_values” is created for every sensor poll.

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for station (refers to entry in table “stations”) |
| sub_id | int | identifier for “sub-station”, e.g. Opus id |
| channel | int | channel where sensor is attached to |

| | | |
|---|---|---|
| sensor_nr | int | unique number for sensor per station (optional sensor mapping info for visualization) |
| is_active | int | channel is active 0=false,1=true |
| sensor_id | int | id for sensor (refers to entry in table “sensors”) |
| sensortype | int | sensor type as configured in Opus |
| channel_mode | int | Opus channel mode (MIN/MAX/AVG/SUM) |
| value_min | float | Opus value scaling information |
| value_max | float | Opus value scaling information |
| sensor_interval | int unsigned | Opus internal sensor interval (ms) |
| name | varchar(255) | Name for Sensor |
| alarm_active | int | Alarm for sensor is active (0 = false, != 0 = true) |
| alarm_min | float | Lower alarm limit/boundary |
| alarm_max | float | Upper alarm limit/boundary |
| is_import | int | (bool) indicates if this sensor is an “import” sensor or not |
| alarm_group | int | Alarm recipient group for sensor (key for table alarm_group) |
| data_type | int | data type of sensor (for UMB devices) |
| last_error | int | Last error for sensor – indicates if there was an error or alarm reading sensor values for this sensor. |
| value_map_id | Int unsigned | Id for value mapping which is applied to values when they are read from the device |
| sample_int | Int unsigned | Sample interval for sensor |
| template_id | int unsigned | Sensor Type Template ID (log. Sensor type) for this sensor |
| alarm_status_map_id | int unsigned | Status map id for “alarm on status” |

### 14.2.8 Table “station_opuses”

This table holds information about the opus attached to a station. This opus configuration information is needed by the collector program .

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for station (refers to entry in table “stations”) |
| opus_id | int | Opus (CAN) id (refers to “sub_id” in table “station_sensors” |
| name | varchar(255) | Name (as configured in Opus) |
| opus_type | int | Opus type (as configured in Opus) |
| e2_crc | int unsigned | Opus configuration checksum |
| irs21_mode | int | Opus EEProm 367 (IRS21 config info) |
| Irs21_diag | int | IRS21 diag mode (0=no diag, 1=read diag data) |
| version | int | Opus Firmware Version |
| Description | longblob | Decription/type of device (Unicode string as hex) |

### 14.2.9 Table “station_files”

This table holds information about ftp file transfers (e.g. cam picture) for stations that provide ftp server services (Axis type stations).

| Column | Type | Description |
|---|---|---|
| Station_id | Int | unique id for station (refers to entry in table “stations”) |
| filename | varchar(255) | remote file name |
| ftp_user | varchar(255) | ftp user name |
| ftp_passwd | varchar(255) | ftp password |
| local_file_tpl | varchar(255) | template string for local filename (see below) |
| transfer_type | Int | type of ftp transfer. 0=binary, 1=ASCII (CR/LF translation) |
| keep_num_cam_pics | int | Number of cam pics to keep in database for this station |

The tag (see below) that may be part of the local file name template (column local_file_tpl) is expanded before the local file is created (the filetransfer is started). The resulting filename is stored in table “file_transfers” .
If local_file_tpl contains no tag (eg “c:\pics_station1\pic.jpg”) each transfer will overwrite the last file transferred.

List of supported tags:
<date> : expands to current date format “yyyymmdd”
<timestamp> : expands to timestamp (localtime) format “yyyymmddhhmmss”

e.g. c:\pictures\pics_station1\pic<timestamp>.jpg

***Note:***
Due to some limitations in MySQL command interpreter, the backslash needs to be masked, i.e. you need to write “c:\\pictures\\pic_station1\\pic<timestamp>.jpg” to this column to get the filename template “c:\pictures\pic_station1\pic<timestamp>.jpg”.

### 14.2.10 Table “file_transfers”

This table holds information about files that have been transferred from a station (see table “station_files”.

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for station (refers to entry in table “stations”) |
| poll_time | unsigned int | time the station was connected for poll/file transfer |
| filename | varchar(255) | local filename of transferred file (expanded filename template, see table “station_files”). |
| error | int | error code if transfer did fail. |
| file_data | longblob | File data as Hex-String in blob |

### 14.2.11 Table “params”

This table is used to store global parameters

| Column | Type | Description |
|---|---|---|
| param_name | varchar(255) | Name of the param (unique) |
| param_type | int | Type of param (internal enum) |
| param_value | longblob | Value of the param stored as hex string |

### 14.2.12 Table “users“

This table stores user information.

| Column | Type | Description |
|---|---|---|
| user_id | int | Unique id for user |
| user | blob | The user name (Unicode) stored as hex string |
| password | blob | The password stored as hex string |
| first_name | blob | First name of user (Unicode) stored as hex string |
| last_name | blob | Last name of user (Unicode) stored as hex string |
| company | blob | Company (Unicode) stored as hex string |
| email | varchar(255) | Email address of user |
| group_id | int | Group ID for user |

### 14.2.13 Table “user_group”

This table stores user group information.

| Column | Type | Description |
|---|---|---|
| group_id | Int | Unique ID for group |
| group_name | Blob | Name for group (Unicode) stored as blob |

### 14.2.14 Table “user_privileges”

Privileges for a user. These privileges are copied from the group privileges when a user is created. Changing group privileges applies to newly created users only.

| Column | Type | Description |
|---|---|---|
| user_id | int | User that the privilege applies to |
| subject_type | int | Internal enum for subject of privilege (Site, Page, Station….) |
| subject_id | int | Id for subject of privilege (Site, Station…) |
| subject_subid | int | 2nd level id for subject (e.g. page id -> site id is subject id..) |
| privilege_type | int | Type of privilege (internal enum), e.g. NONE; VIEW, EDIT, CREATE |

### 14.2.15 Table “group_privileges”

Privileges for a group. These are only used as “template” when a new user is created.

| Column | Type | Description |
|---|---|---|
| group_id | int | Group that the privilege applies to |
| subject_type | int | Internal enum for subject of privilege (Site, Page, Station….) |
| subject_id | int | Id for subject of privilege (Site, Station…) |
| subject_subid | int | 2nd level id for subject (e.g. page id -> site id is subject id..) |
| privilege_type | int | Type of privilege (internal enum), e.g. NONE; VIEW, EDIT, CREATE |

### 14.2.16 Table “sites”

This table holds information about sites.

| Column | Type | Description |
|---|---|---|
| site_id | int | Unique identifier for site |
| name | blob | Name (Unicode) of site stored as hex string |
| restrict_access | int | Access to site/pages is restricted by .htaccess. 0 = false, != 0 = true |
| start_page_id | int | Page id of start (home) page for site |
| use_ftp | int | Use ftp to transfer files to web server. 0 = false, != 0 = true, |
| ftp_host | varchar(255) | ftp host (IP address or host name) for web server |
| ftp_user | varchar(255) | ftp user for web server |
| ftp_password | varchar(255) | ftp password for web server |
| local_path | varchar(255) | Local path to store generated html files |
| remote_path | varchar(255) | Remote path on web server to store generated files |
| site_header | blob | Site header text (Unicode) - is used for every page of the site |
| site_footer | blob | Site footer text (Unicode) - is used for every page of the site |
| menu_right | int | 0 = false -> menu on left side; != 0 = true -> menu on right side |
| dia_width | int | Width of bar and line diagrams for this site |
| line_dia_height | int | Height of line diagrams for this site |
| bar_dia_height | int | Height of bar diagrams for this site |
| num_sens_in_dia | int | Maximum number of sensors/curves in a diagram |
| num_sens_in_tab | int | Maximum number of sensors in a table |
| htpasswd_path | varchar(255) | Path to the htpasswd file |
| csv_download | int | (bool) include data as .csv file for download on data |

| | | |
|---|---|---|
| | | pages |
| print_view | int | (bool) include a print view of pages (white background, no menus etc.) |
| table_separate_page | int | (bool) place tables for data pages on a separate page |
| page_interval | int | (internal enum) default time interval for data pages |
| ftp_port | Int | ftp port |
| ftp_passive | Int | (bool) use ftp passive mode |
| is_active | Int unsigned | (bool) site is active (or inactive) |
| auto_delete_pages | Int unsigned | (bool) automatically delete older data (archive) pages |
| auto_delete_iv_counter | Int unsigned | Number of “auto_delete_iv” to calculate which old data pages to delete |
| auto_delete_iv | Int unsigned | (internal enum) interval (weeks/months/years) to calculate which old data pages to delete |

### 14.2.17 Table “site_stations”

This table holds information about stations that are displayed in a site.

| Column | Type | Description |
|---|---|---|
| site_id | int | Id of the site |
| station_id | int | Id of the station |

### 14.2.18 Table “pages”

This table holds information about pages of a site

| Column | Type | Description |
|---|---|---|
| site_id | int | Id of site |
| page_id | int | Id of page |
| page_nr | int | Sequence number of page (for data pages only) |
| page_type | int | Type of page (internal enum) |
| table_separate_page | int | Table(s) with sensor data are written to a separate page (data pages only) |
| page_title | longblob | Title for page (Unicode) stored as hex string |
| page_subtitle | longblob | Sub-Title for page (Unicode) stored as hex string |
| page_footer | longblob | Footer for page (Unicode) stored as hex string |
| page_interval | int | Time interval for page (internal enum, for data pages only) |
| start_time | unsigned int | Start time (UTC Timestamp) for data page |
| station_id | int | Station id for station associated with this page (if appropriate) |
| data_as_csv | int | Bool (0 = false, != 0 = true) - include sensor data as .csv file for download |
| is_generated | int | Bool (0 = false, != 0 = true) -indicates that a data or archive_list page has been generated by SmartView |
| last_generated | unsigned int | Time (UTC Timestamp) when page was last generated/written |
| data_pageid | int | Id of associated data page (for archive_list pages) |
| print_view | int | (bool) include a print view for the page (no background color, no menu etc.) |
| update_interval | int unsigned | Minimum update interval for page (for later use) |
| template_id | int | Id for page template |
| auto_refresh_iv | int unsigned | Timer (in sec) for auto refresh of html page |
| sequence | int unsigned | Sequence for page – used to determine order of station/data/archive list pages in the appropriate “menu” page (station list, data list and archive menu page) |
| is_template | int | (Bool) indicates that page is a template page |
| template_id | Int unsigned | ID for template page |

### 14.2.19 Table “page_params”

This table is used to store parameters for sites and pages

| Column | Type | Description |
|---|---|---|
| site_id | int | Id of site |
| page_id | int | Id of page |
| param_name | varchar(255) | Name of param |
| param_type | Int | Type of param (internal enum) |
| param_value | longblob | Value of param (stored as hex string) |

### 14.2.20 Table “page_elements”

This table holds information about elements of a page

| Column | Type | Description |
|---|---|---|
| site_id | int | Id of site |
| page_id | int | Id of page |
| element_id | int | Id of element |
| element_type | int | Type of element (internal enum), e.g. BAR_DIAGRAM or LINE_DIAGRAM |
| station_id | int | Id of station associated with this element (if appropriate) |
| link_page_id | int | Id of a page associated with this element (if appropriate) |
| x | int | Horizontal sequence or position (pixel) of element |
| y | int | Vertical sequence or position (pixel) of element |
| width | int | Width of element in pixel |
| height | int | Height of element in pixel |
| ax1_scale_min | float | Minimum value for 1st y-axis of a line diagram |
| ax1_scale_max | float | Maximum value for 1st y-axis of a line diagram |
| ax1_auto_scale | int | Bool (0 = false, != 0 = true) - automatic scale for 1st y-axis of a line diagram |
| ax2_scale_min | float | Minimum value for 2nd y-axis of a line diagram |
| ax2_scale_max | float | Maximum value for 2nd y-axis of a line diagram |
| ax2_auto_scale | int | Bool (0 = false, != 0 = true) - automatic scale for 2nd y-axis of a line diagram |
| ax3_scale_min | float | Minimum value for 3rd y-axis of a line diagram |
| ax3_scale_max | float | Maximum value for 3rd y-axis of a line diagram |
| ax3_auto_scale | int | Bool (0 = false, != 0 = true) - automatic scale for 3rd y-axis of a line diagram |
| ax4_scale_min | float | Minimum value for 4th y-axis of a line diagram |
| ax4_scale_max | float | Maximum value for 4th y-axis of a line diagram |
| ax4_auto_scale | int | Bool (0 = false, != 0 = true) - automatic scale for 4th y-axis of a line diagram |
| name | blob | Name of page element |
| data_name | varchar(255) | Name of data (file) of element (if data is attached) |
| element_data | longblob | Data (file) of element (e.g. bitmap) - stored as hex string |
| text | longblob | Text for Element (text element) - Unicode stored as hex string |
| title_font_size | int | Size for diagram title text |
| title_text_color | Int unsigned | Color (rgb) for diagram title |
| legend_font_size | int | Size for diagram legend text |
| legend_font_color | int unsigned | Color (rgb) for legend text |
| scale_font_size | int | Size for scale/axis text |
| scale_font_color | int unsigned | Color (rgb) for scale/axis text |
| datetime_font_size | int | Size for datetime range text |
| datetime_font_color | int unsigned | Color (rgb) for datetime range text |
| background_color | int unsigned | Background color for diagram |
| axis_color | int unsigned | Color (rgb) for axis/grid/scale |
| pointer_color | int unsigned | Color (rgb) for pointer (analog gauge) |
| time_iv | int unsigned | Time interval (internal enum) for report page elements |
| is_template | int unsigned | (bool) indicates that page element is a template |

| | | |
|---|---|---|
| | | element |
| template_id | int unsigned | ID for element template |

### 14.2.21 Table “page_element_sensors”

This table holds information about sensors for a page or page element

| Column | Type | Description |
|---|---|---|
| site_id | int | Id of site |
| page_id | int | Id of page |
| element_id | int | Id of element |
| station_id | int | Id of station |
| sensor_nr | int | Sensor nr of sensor in station |
| value_type | int | Value type of this sensor (internal enum) - MIN/MAX/AVG/SUM |
| bar_value_min | float | Lower boundary for color interval for bar diagram (obsolete) |
| bar_value_max | float | Upper boundary for color interval for bar diagram (obsolete) |
| color | int | Color (rgb) for sensor in line diagram |
| width | int | Width of line in line diagram |
| pen_style | int | Style of pen (solid/dash/dot…) for line in line diagram |
| bar_diagram_legend | blob | Legend for bar diagram color/value interval |
| last_data_page | int | Last data page nr where data was available for this sensor |
| sequence | int unsigned | sequence of sensor (for tables and line diagrams) |
| status_map_id | int unsigned | Id of “status map” for sensor (horiz. Bar diagram, table, color map, pictogram) |
| report_sum_channel | int unsigned | (bool) treat sensor as “sum” channel in reports |
| is_template | int unsigned | (bool) indicates a template sensor |
| template_id | Int unsigned | Sensor Type Template (log. Sensor Type) for this Sensor. |

### 14.2.22 Table “archive_pages”

This table holds information about pages of type “Archive”, i.e. “old” Data pages. The structure is the same as for table “pages” (only the primary key is different)

### 14.2.23 Table “export_job”

This table holds information about export jobs

| Column | Type | Description |
|---|---|---|
| job_id | int | Id of job |
| job_name | blob | Name of job (Unicode, stored as hex string) |
| job_type | int | (internal enum) type of job (Export / Import or both) |
| run_type | int | (internal enum) job is run manual/on new data or on different time intervals |
| run_every | int | Counter for run_type interval |
| run_hour | int | Time of day job is run |
| run_day | int | Day of week or month job is run |
| export_format | int | (internal enum) format for export (CSV,XML,…) |
| data_start | int | (internal enum) specifies time for start of data (all available, fixed, or start of a specific interval like last week or last month…) |
| data_iv_counter | int | Number of data_intervals |
| data_iv | int | (internal enum) interval for data (all data, fixed end time, or interval like hour, week, month, year) |
| export_filename | varchar(255) | Name of export file, may contain tags <date> or |

| | | |
|---|---|---|
| | | <timestamp> |
| export_decimal_point | char(1) | Character to be used as decimal point (“.” or “,”) |
| export_error_value | varchar(255) | String to indicate an error value |
| export_csv_separator | varchar(255) | Character/String to be used as csv separator (“,” or “;” or TAB) |
| export_values_in_quotes | int | (bool) flag if csv export values are quoted or not |
| export_include_header | int | (bool) flag if header line should be written |
| export_append_data | int | (bool) flag if data is appended to export file, or if export file is overwritten |
| export_time_is_local | int | (bool) flag if timestamps are in local time or in UTC time |
| export_time_sep_column | int | (bool) flag if date and time are to be written in separate columns |
| export_date_str | varchar(255) | Format string for date (yyyy = year, mm = month, dd=day, default yyyy/mm/dd) |
| export_time_str | varchar(255) | Format string for time (hh = hour, mm= minute, ss=second, default hh:mm:ss) |
| export_last_column_with_sep | int | (bool) flag if last column is terminated by csv separator or not |
| last_run | int unsigned | Timestamp (UTC) of last run of job |
| next_run | int unsigned | Timestamp (UTC) of next run of job |
| import_f ilename | varchar(255) | Name of import file, may contain tags <date> or <timestamp> |
| import_format | int | (internal enum) Format of import file (currently only CSV is supported) |
| import_skip_header | int | (bool) skip first line of import file (because it is a header line) |
| import_decimal_point | char(1) | Character used as decimal point (“.” or “,”) |
| import_error_value | varchar(255) | String that indicates an error value |
| import_error_column | int | Not used (column that indicates error value) |
| import_no_val_is_error | int | (bool) if true, a missing value is treaded as an error value |
| import_csv_separartor | varchar(255) | Character/string used as separator in CSV file (“,” or “;” or TAB) |
| import_values_in_quotes | int | (bool) flag that indicates if values are quoted |
| import_time_is_local | int | (bool) flag that indicates if time values are in local time or in UTC |
| import_time_in_sep_column | int | (bool) flag that indicates if date and time are in separate columns or not |
| import_date_str | varchar(255) | Format string for date (yyyy = year, mm = month, dd=day, default yyyy/mm/dd) |
| import_time_str | varchar(255) | Format string for time (hh = hour, mm= minute, ss=second, default hh:mm:ss) |
| exe_filename | varchar(255) | Name of (external) program to be run |
| exe_params | varchar(255) | Params for (external) program. May contain tags #export_file and #import_file (which are replaced by the appropriate values) |
| run_error | int | Exit code of (external) program |
| use_ftp | Int | (bool) flag that indicates if ftp should be used to<br>a) transfer the export file for jobs of type EXPORT or of type EXPORT AND IMPORT after the export file has been written<br>b) transfer the import file for jobs of type IMPORT before the import file is processed |
| ftp_host | varchar(255) | Name or IP of ftp host |
| ftp_port | int | Port number to be used for ftp (default : 21) |
| remote_filename | varchar(255) | Name of remote file |
| ftp_user | varchar(255) | User name for ftp |

| | | |
|---|---|---|
| ftp_password | varchar(255) | Password for ftp user |
| ftp_binary | int | (bool) use ftp binary transfer |
| ftp_passive | int | (bool) use ftp passive mode |
| station_id | int | Station associated with this job |
| fixed_data_start | int unsigned | Timestamp (UTC) for data in export job if a fixed date is assigned |
| fixed_data_end | int unsigned | Timestamp (UTC) for data in export job if a fixed date is assigned |
| export_date_header | Blob | Header for date column in export file (Unicode stored as hex string) |
| export_time_header | Blob | Header for time column in export file (Unicode stored as hex string) |
| is_active | int unsigned | (bool) export job is active |
| Import_only_new_files | Int unsigned | (bool) only import files that have been “last modified” after job was run last time |
| ftp_delete_source | Int unsigned | (bool) delete source file after ftp transfer |
| Import_delete_after | Int unsigned | (bool) delete import file after import was successful |
| export_no_values | Int unsigned | (bool) create export file (empty or with header only) even if there is no data to export |
| is_template | int unsigned | (bool) indicates that job is a template job |
| job_template_id | int unsigned | ID for Export Job Template |
| template_type_id | int unsigned | ID for pre-defined export job template |

### 14.2.24 Table “export_params”

This table holds additional parameter (e.g. parameter for XML export) for export jobs.

| Column | Type | Description |
|---|---|---|
| job_id | int | Job id |
| param_name | varchar(255) | Name of param |
| param_type | int | Type of param (internal enum) |
| param_value | longblob | Value of param (stored as hex string) |

### 14.2.25 Table “export_sensors”

This table holds information about sensors in a specific export job (of type EXPORT or EXPORT AND IMPORT)

| Column | Type | Description |
|---|---|---|
| job_id | int | Id of job |
| station_id | int | Station id to identify sensor |
| sensor_nr | int | Sensor nr to identify sensor |
| value_type | int | (internal enum) value type to identify sensor |
| convert_type | int | (internal enum) specifies the conversion that is to be done on value when exported (none, to int, or to bool) |
| sequence | int | Sequence of sensor in export |
| last_export | int unsigned | Timestamp (UTC) of last exported value for this sensor |
| compare_operator | int | (internal enum) compare operation to be performend when value is converted to bool (<,<=, ==, >=, > , !=) |
| compare_value | float | Value for comparison when value is converted to bool |
| export_name | blob | Name of Column (Unicode) for header in export file |
| export_scale | float | Scale factor for export |
| value_map_id | Int unsigned | Id of value map used when data is exported |
| sensor_template_id | int unsigned | Sensor Type Template ID (log. Sensor Type) for this sensor |

| | | |
|---|---|---|
| unit_name | Blob | Expected unit name for template job sensor (UNICODE) |

### 14.2.26 Table “import_sensors”

This table holds information about sensors in a specific export job (of type IMPORT or EXPORT AND IMPORT)

| Column | Type | Description |
|---|---|---|
| job_id | int | Id of job |
| station_id | int | Station id to identify sensor |
| sensor_nr | int | Sensor nr to identify sensor |
| value_type | int | (internal enum) value type to identify sensor |
| sequence | int | Sequence of sensor in import file |
| last_export | int unsigned | Timestamp (UTC) of last imported value for this sensor |
| value_format | int | (internal enum) format of value (float, int, bool) |
| import_scale | float | Scale factor for import of value |
| calc_daybreak_val | int | (bool) calculate daybreak value for import sensor (yes/no) |
| calc_daybreak_type | int | (enum) type of daybreak value calculation (linear interpolation, last value, next value) |
| value_map_id | Int unsigned | Id for value mapping when data is imported |
| sensor_template_id | int unsigned | Sensor Type Template id (log. Sensor type) for this sensor |

### 14.2.27 Table “station_schedule”

This table holds schedule information for a station

| Column | Type | Description |
|---|---|---|
| station_id | int | Station ID |
| schedule_id | int | Id of schedule entry |
| Type | int | Type of entry (exclude/run) |
| op_mode | int | Operations mode for entry (all/summer/winter) |
| Every | int | Poll every xx iv_type …… |
| iv_type | int | Interval type (minute/hour/days….) |
| Hour | int | Offset/hour |
| Day | int | Day of week or month |
| exclude_from_hour | int | Start of exclude interval |
| exclude_to_hour | int | End of exclude interval |

### 14.2.28 Table “alarm_group”

This table configures “alarm groups” which receive alarms

| Column | Type | Description |
|---|---|---|
| alarm_group_id | int | Id of alarm group |
| name | blob | Name of alarm group (Unicode, as hex-string) |
| oce_per_op | int | Notify once per operation (not used) |

### 14.2.29 Table “alarm_recipient”

This table configures alarm recipients for an alarm group.

| Column | Type | Description |
|---|---|---|
| alarm_group_id | int | Id of alarm group |
| recipient_id | Iit | Id of alarm recipient |
| name | blob | Name of recipient |
| alarm_type | int | Indicated type of alarm (email/sms….) |
| address | blob | Address of recipient (Unicode, stored as hex string) |
| user_id | Int | Associated user id for this alarm recipient (for |

| | | |
|---|---|---|
| | | future use) |
| is_active | Int | (bool) recipient is active |
| alarm_reasons | Int | (coded) alarm reasons this recipient wants to receive (for future use) |
| working_hours_id | int | Working hours (shift) associated with this user |

### 14.2.30 Table "data_archive_jobs"

This table holds configuration data for data compression and backup jobs

| Column | Type | Description |
|---|---|---|
| job_id | Int unsigned | Id of the job |
| name | Blob | Name (Unicode string stored as hex string) |
| job_type | Int unsigned | Internal enum that codes the type of data compression/backup job |
| run_every | Int unsigned | Run job every xxx run_iv |
| run_iv | Int unsigned | Internal enum for run interval (hour/day/week…) |
| run_day | Int unsigned | Day (of week or month) to run job |
| run_hour | Int unsigned | Offset (seconds) for job (if interval is >= day ) |
| offset_iv | Int unsigned | Internal enum for Interval (day/week/…) to calculate offset for data to be compressed |
| offset_counter | Int unsigned | Compress data that is older than offset_counter offset_iv…. |
| compress_iv | Int unsigned | Internal enum – interval (minute, 10 minute, 30 minute, hour, day, …..) for compressed values |
| backup_before | Int unsigned | (bool) backup data before compressing/deleting data |
| zip_backup | Int unsigned | (bool) zip backup data |
| use_ftp | Int unsigned | (bool) transfer backup file using ftp to a host |
| ftp_host | Blob | Ftp host or ip address (Unicode stored as hex string) |
| ftp port | Int unsigned | TCP/IP Port for ftp transfer |
| ftp_passive | Int unsigned | (bool) use ftp passive mode |
| ftp_binary | Int unsigned | (bool) use ftp binary mode |
| ftp_user | Blob | User for ftp (Unicode stored as hex string) |
| ftp_passwd | Blob | Password for ftp (Unicode stored as hex string) |
| local_fname | Blob | Local filename for backup file (Unicode stored as hex string) |
| remote_fname | Blob | Remote filename on host for backup file (Unicode stored as hex string) |
| last_run | Int unsigned | UTC Timestamp when job was executed last time |
| next_run | Int unsigned | UTC Timestamp when job will be executed next time |
| is_active | Int unsigned | (bool) job is active or inactive |

### 14.2.31 Table "value_map"

This table holds configuration data for value mapping

| Column | Type | Description |
|---|---|---|
| map_id | Int unsigned | Id for value mapping |
| name | Blob | Name for value mapping (Unicode stored as hex string) |
| type | Int unsigned | Internal enum for type of mapping (scale/table) |
| scale | Float | Value used to scale |
| offset | Float | Offset for calculating scaled values |
| template | Int unsigned | Internal enum for type of mapping (used for "program known" mapping types only) |

### 14.2.32 Table “value_map_table”

This table holds configuration data for value mapping of type “table”

| Column | Type | Description |
|---|---|---|
| map_id | Int unsigned | Id for value mapping |
| entry_id | Int unsigned | Id for value map table entry |
| val_min | Float | Lower limit of value interval |
| val_max | Float | Upper limit of value interval |
| map_val | Float | Value that “input” value will be mapped to if input value is within interval (val_min <= input value < val_max) |
| plausi_mode | int unsigned | Internal enum for plausibility checking mode. |
| plausi_type | int unsigned | Internal enum for plausibility checking type |
| plausi_value | Double | Internal enum for plausibility checking value |

### 14.2.33 Table “config_change”

This table holds (temporary) data for configuration changes to a device.

| Column | Type | Description |
|---|---|---|
| station_id | Int unsigned | Station id |
| device_id | Int unsigned | Device id |
| channel | Int unsigned | Channel |
| change_type | Int unsigned | Internal enum for type of change (name, mode, store interval, sample interval..) |
| param_type | Int unsigned | Internal enum for type of value (number/string…) |

### 14.2.34 Table “status_map”

This table holds configuration data for status mapping.

| Column | Type | Description |
|---|---|---|
| map_id | int unsigned | Status map id |
| template_id | int unsigned | Template id for pre-defined status maps |
| Name | blob | Name of status map (Unicode stored as hex string) |

### 14.2.35 Table “status_map_entry”

This table holds configuration data for status mapping.

| Column | Type | Description |
|---|---|---|
| map_id | int unsigned | Status map id |
| entry_id | int unsigned | ID for status map entry |
| val_min | float | Lower limit for value interval |
| val_max | float | Upper limit for value interval |
| description | Blob | Description for value interval |
| color | int unsigned | Color (rgb) for value interval |
| pictogram_id | int unsigned | Id of pictogram bitmap for value interval |
| alarm_level | int unsigned | Alarm level (none/warning/alarm) associated with this status map entry |

### 14.2.36 Table “pictogram_bitmap”

This table is used to store pictogram bitmaps.

| Column | Type | Description |
|---|---|---|
| pictogram_id | int unsigned | Id of pictogram bitmap |
| name | blob | Name of pictogram (Unicode stored as hex string) |
| filename | blob | Filename of pictogram bitmap (Unicode stored as hex string) |
| width | int unsigned | Width of pictogram bitmap in pixel |

| | | |
|---|---|---|
| height | int unsigned | Height of pictogram bitmap in pixel |
| pic_format | int unsigned | Format of pictogram bitmap (for future use) |
| data | longblob | Bitmap data (stored as hex string) |

### 14.2.37 Table "calc_channel"

This table is used to store information for calculated channels

| Column | Type | Description |
|---|---|---|
| calc_channel_id | int unsigned | Id of calculated channel |
| station_id | int unsigned | Station id to identify channel |
| sensor_nr | int unsigned | Sensor nr to identify channel |
| calc_type | int unsigned | (internal enum) type of calculation (min/max/avg/sum/value mapping….) |
| value_map_id | int unsigned | Value mapping id (for type value mapping) |
| calc_interval | int unsigned | (internal enum) calculation interval for interval based calculations (min/max/avg/sum) |
| last_calc_time | int unsigned | UTC timestamp of last calculation (not used at the moment) |

### 14.2.38 Table "calc_channel_sensor"

This table is used to store information for "source sensors" of calculated channels

| Column | Type | Description |
|---|---|---|
| calc_channel_id | int unsigned | Id of calculated channel |
| calc_channel_sensor_id | int unsigned | Id of source sensor for calculated channel |
| station_id | int unsigned | Station id to identify source sensor |
| sensor_nr | int unsigned | Sensor nr to identify source sensor |
| value_type | int unsigned | Sensor value type to identify source sensor |
| last_calc_time | int unsigned | UTC timestamp of last calculation (not used at the moment) |

### 14.2.39 Table "calc_channel_param"

This table is used to store information for params of calculated channels (params for specific calculations – for future use)

| Column | Type | Description |
|---|---|---|
| calc_channel_id | int unsigned | Id of calculated channel |
| param_name | varchar(255) | Name of param |
| param_type | int | Type of param (internal enum) |
| param_value | longblob | Value of param (stored as hex string) |

### 14.2.40 Table "last_sensor_values"

This table holds the last sensor/measurement values retrieved from stations for performance reasons.

| Column | Type | Description |
|---|---|---|
| station_id | int | unique id for a station (refers to entry in table "stations" and "station_sensors") |
| sub_id | int | identifier for "sub-station", e.g. Opus id (refers to table "station_sensors" |
| channel | int | channel where sensor is attached to (refers to table "station_sensors" |
| value_type | int | 0=undefined,1=min,2=max,3=avg,4=sum |
| sensor_nr | int | sensor nr of station (refers to "sensor_nr" in table "station_sensors") |
| poll_time | unsigned int | UTC timestamp when station was polled (station was connected). All values read from a station in a poll use the same value. |
| sensor_id | Int | id for the sensor type (refers to entry in table "sensors") |

| | | |
|---|---|---|
| measure_time | unsigned int | UTC timestamp when measurement was taken (will be the same as poll_time for “online data”, but different for historical data that was stored at the station..) |
| value | Float | the measured sensor value |
| error | int | error code for sample (0= no error) |

The primary key for this table is station_id, sub_id, channel, value_type. This primary key ensures (as it is a unique key) that there is only one sensor value per given measure time and sensor and value type, and allows fast access to measurement values using measure_time, station_id, sub_id, channel, value_type as keys (or part of that list).

### 14.2.41 Table “sensor_type_template”

This table holds sensor type template (logical sensor type) information. Sensor type templates are used to match sensors for export/import and page/page element templates.

| Column | Type | Description |
|---|---|---|
| template_id | int | unique id for sensor type template |
| template_type_id | int | (Internal enum) internal identifier for pre-defined sensor template types |
| name | longblob | Name/description for sensor type template |

### 14.2.42 Table “working_hours”

This table holds working hours information for alarm recipients. Times (on weekday basis) are stored in table “working_hours_entry”.

| Column | Type | Description |
|---|---|---|
| working_hours_id | int unsigned | unique id for working hours |
| Name | longblob | Name/description for working hours |
| is_private | int unsigned | (bool) is a private definition (for a specific user) – for future use |

### 14.2.43 Table “working_hours_entry”

This table holds times (on weekday basis) for a working hours definition.

| Column | Type | Description |
|---|---|---|
| working_hours_id | int unsigned | unique id for working hours |
| entry_id | int unsigned | unique id for this entry |
| weekday | int unsigned | weekday for this entry |
| from_time | int unsigned | (seconds from midnight) “start of duty” timestamp |
| to_time | int unsigned | (seconds from midnight) “end of duty” timestamp |

### 14.2.44 Table “alarm_event”

This table holds alarm_events. When Collector polls a station, the received data is analysed and one (or more) alarm_events are written to this table. SmartCom reads, processes, and deletes these events. For the moment, only the last event that was actually sent for a specific station/device/channel and alarm reason is kept in this table (no alarm history is kept at the moment).

| Column | Type | Description |
|---|---|---|
| alarm_id | int unsigned | unique id for alarm event |

| | | |
|---|---|---|
| station_id | int unsigned | Station id for this alarm event |
| device_id | int unsigned | Device id for this alarm event |
| channel | int unsigned | Channel for this alarm event |
| export_job_id | int unsigned | Export job id for this alarm event |
| alarm_reason | int unsigned | (enum) reason for alarm (sensor error/alarm value/alarm status/warning status/ok…) |
| is_processed | int unsigned | (bool) flag indicating if this alarm has been processed by the AlarmHandler in SmartCom |
| alarm_time | int unsigned | Unix (UTC) timestamp alarm was created |
| alarm_reason_time | int unsigned | Unix (UTC) timestamp (i.e. time associated with the measurement value) associated with the reason for the alarm |
| is_preserved | int unsigned | (bool) flag indicating if this alarm is preserved (i.e. the alarm condition did not change from the last time the station was polled) |
| is_sent | int unsigned | (bool) flag indicating that this alarm has been sent to the (active) recipients. |
| alarm_group | int unsigned | Alarm (recipient) group associated with this alarm event |
| sensor_value | int unsigned | Sensor value that caused the alarm event |
| value_error | int unsigned | Error condition that caused the alarm event |
| is_confirmed | Int unsigned | (bool) Alarm has been confirmed (for future use) |
| confirm_time | Int unsigned | Unix (UTC) timestamp when alarm has been confirmed (for future use) |
| confirm_user_id | Int unsigned | User id for user that confirmed this alarm (for future use) |

### 14.2.45 Table “sent_alarm”

This table holds information for alarms sent to recipients. Note: at the moment, only entries associated with the last sent alarm for a specific station/device/channel/export job are kept (no historical data is kept in database)

| Column | Type | Description |
|---|---|---|
| alarm_id | int unsigned | unique id for alarm event |
| sent_time | int unsigned | Unix (UTC) timestamp when alarm was sent |
| recipient_id | Int unsigned | Recipient id for alarm recipient the alarm was sent to |
| sent_ok | Int unsigned | (bool) flag indicating if alarm could be sent |
| is_confirmed | Int unsigned | (bool) flag indicating that recipient did confirm the alarm (for future use) |
| confirm_time | Int unsigned | Unix (UTC) timestamp when alarm was confirmed by recipient (for future use) |

### 14.2.46 Table “tls_stations”

This table holds information for TLS (over IP) stations.

| Column | Type | Description |
|---|---|---|
| station_id | int unsigned | unique id for station |
| osi7_addr | int unsigned | OSI7 address for TLS station – must be unique for each TLS based station |
| osi2_addr | int unsigned | OSI2 address for TLS station (for future use) |

## 14.3 Document History

| | | |
|---|---|---|
| September 2005 | P. Rau | Version 1.0 |
| October 2005 | P. Rau | Version 1.0.1:<br>• Some bug fixes and general enhancements<br>• SmartWeb and SmartCom now show status screen |
| October 2005 | P. Rau | Version 1.0.2<br>• Fixed memory leak<br>• Support for CAM picture on station page<br>• CSV Separator and decimal point can be configured |
| November 2005 | P. Rau | Version 1.0.3<br>• Refined exception handling in SmartWeb<br>• Bug Fix Diagram creating Diagrams<br>• Hide inactive channels<br>• Added Legend for bar diagrams<br>• Fixed problems with MySQL Version 5 |
| November 2005 | P. Rau | Version 1.0.4<br>• Changed handling of data for csv export and html tables for sample interval < 1 minute<br>• Added param for FTP (active/passive mode)<br>• Added param for “Error value” in csv export files<br>• Added “manual” export to SmartView3 |
| December 2005 | P. Rau | Version 1.0.5<br>• Allow selection of multiple sensors in “Sensor Select Dialog”<br>• Bug Fix SmartCom |
| January 2006 | P. Rau | Version 1.0.6<br>• Added automatic Export/Import module (SmartWeb) with support for calculated sensors<br>• Changed “manual” export to export/import (SmartView3)<br>• Added configuration for Export/Import (SmartView3)<br>• Changed “Create Data Pages for Stations” behavior. If user now answers “NO” no to question “re-create all generated data pages for this site”, existing data pages for stations are not deleted. Only pages for stations that do not have a data page assigned are created. Number of created pages is reported. |
| April 2006 | P. Rau | Version 1.0.7<br>• Fixed problem in html tables with “space only” cell content<br>• Integrated Setup with SmartView3<br>• Integration of Collector in manual<br>• Collector SmartView3 with GUI Interface – Collector is now integral part of SmartView3<br>• SmartCom and SmartWeb with GUI Interface<br>• Alarms via Email<br>• Support for OpusII/OpusII GMA, HP100 and UMB devices<br>• Support for different Clock/Time settings for Stations<br>• Enhanced scheduling for stations |
| May 2006 | P. Rau | Version 1.0.8<br>• Timeouts can now be configured on a “per station” basis (needed for UMB devices) instead of a global timeout parameter.<br>• Added columns “device_timeout” and “device_timeout_long” to table “stations”<br>• Fixed problem with RS232 connection timeout in release build<br>• Fixed problem with Opus2 in release build<br>• Fixed problem with TCP/IP Connection<br>• Fixed problem with time offset configuration for stations<br>• Fixed problem with Alarm Settings (Treeview) dialog |
| May 2006 | P. Rau | Version 1.0.9<br>• Fixed “desktop refresh” effect<br>• Improved archive page handling<br>• Added alarm params for export jobs<br>• Added scaling to export sensors<br>• Added scaling to import sensors<br>• Added “calc daybreak value” to import sensors<br>• Added “expires” html meta tag to force reload of data pages<br>• Fixed reading online data for Opus208 disabled channel (don’t try to read online data if channel is inactive)<br>• Improved error handling on scan_ids for Opus200 network<br>• Adjusted bar diagram width and line diagram width to match within a page (depending on maximum number of used y-axis in line diagrams) |
| | P. Rau | Version 1.0.10<br>• Extended Site configuration for FTP (Port/Passive Mode) |

| | | |
|---|---|---|
| | | • Enhanced page update trigger for import jobs<br>• Added maintenance functions “Remove Station” and “Abort Transfer”<br>• Added time offset specification for reading Opus200 logger data |
| June 2006 | P. Rau | Version 1.0.11<br>• Enhanced handling of “lost database connection” error (MySQL 5 drops database connection if inactive for some time)<br>• Corrected time offset for reading Opus200 logger data for station time settings<br>• Added param “OFFLINE-NO-MINUTE-OFFSET” for collector<br>• Corrected retry handling for reading logger data |
| June 2006 | P. Rau | Version 1.0.12<br>• Changed commands to read Opus200 config (not reading specific addresses anymore, but reading complete EEProm data – is faster and works with older firmware versions as well)<br>• Changed command to read UMB values to read multiple values with one command<br>• Disable deletion of last/single schedule entry for station<br>• Default for “set station time” changed to false<br>• Disabled “Set station time” for UMB devices.<br>• Disabled “Read stored data” for UMB devices<br>• Changed default for time offset “read all stored data” to start of current day<br>• Show station type on station list in SmartView3<br>• Fixed Bug with thread synchronization in Collector |
| September 2006 | P. Rau | Version 1.1.0<br>• Improved TAPI handling for TAPI device changes (changes to TAPI devices don’t cause error that requires reboot of PC anymore)<br>• Added “data archive” functions to automatically compress/delete measurement data from database, and backup data<br>New table “data_archive_jobs”<br>• Added start times to Export (2/3/4 days ago, “since last export” and “since last export day”)<br>• New Export Job types “Export + Run”, “Run and Import”<br>• Column header for date/time column can be configured now<br>New columns “export_date_header” and “export_time_header” for table “export_jobs”<br>• Improved logging and error handling for UMB devices<br>• Separate parameters for connection retries and communication retries<br>• Improved site edit interface<br>• Fixed problem with first read of Opus200 data when station time not UTC (for older firmware versions)<br>• New page type “external link” for links in main menu<br>• Removed Cam Stations from selection for web site<br>• Cam Station selection for “data” station now via combo box (instead of editing station id)<br>• “Value Mapping” for sensor values read from device, export and import of data<br>New tables “value_map” and “value_map_table”<br>New column “value_map_id” for table “station_sensors”<br>New column “value_map_id” for table “export_sensors”<br>New column “value_map_id” for table “import_sensors”<br>• Configuration change for Opus200 devices (channel name, channel mode, sample and store interval)<br>New column “sample_int” for table “station_sensors”<br>• New table “config_changes”<br>• “Modem Pools” / Assign specific Modem(s) to a station<br>New column “mpdem_pool_id” for table “stations”<br>• Bug Fix “config change” for UMB device |
| November 2006 | P. Rau | Version 1.1.1<br>• Some bug fixes<br>• Export/Import type “FTP” with wildcard support to transfer multiple files from server<br>• Export/Import – Import with wildcard support and new option “new files only”<br>• UTC-TIMESTAMP as new time format for export/import jobs<br>• Export/Import: only create “empty” export file if configured<br>• Export/Import: new option “delete import file” after import was successful<br>• Export/Import: new option “delete source” for ftp transfer (delete source file after successful transfer)<br>• Export/Import: new option “import only new files” – only import files that have a last_modification timestamp newer than the job’s “last_run” timestamp. |

| | | |
|---|---|---|
| | | • Dewpoint calculation as external program for export/import<br>• Automatic delete of (older) cam pictures<br>• "Sites" can be set active/inactive<br>• Export/Import jobs can be set active/inactive<br>• New parameters for Collector regarding UMB Stations : UMB Master Address (UMB Address used by collector), Max Class ID (maximum Class ID used when scanning for UMB devices), and "Skip scan class after xxx timeouts.<br>• New columns table "export_job" : colum "is_active", "import_only_new_files", "ftp_delete_source", "import_delete_after", "export_no_values"<br>Table "sites" : new column "is_active" |
| November 2006 | P. Rau | Version 1.1.2<br>• Bug Fix display channel mode on station page<br>• "PinCheck" to set pin for GSM Modem SIM Card now internal<br>• Enhanced Exception Handling for reading station list in Collector |
| November 2006 | P. Rau | Version 1.1.3<br>Changes for MySQL Version 5.0.27 in create table statements<br>• New columns for table "sites" : "auto_delete_pages", "auto_delete_iv_counter" and "auto_delete_iv" |
| Feb. 2007 | P. Rau | Version 1.2.0<br>• Bug fixes and minor enhancements.<br>• Automatically delete data (archive) pages that are older than a specified time<br>• Display last sensor values for selectable sensors on mouse "scroll over" station icon on map display<br>• Display last sensor values on station page as "analog gauge" or "pictogram".<br>• Configure which sensors to display on station page, and in which order (in table).<br>• Configure sequence of elements shown on data or station page<br>• Configure sequence of sensors in tables and diagrams<br>• Configure text size and colors for diagrams<br>• "Status Mapping" is now used to configure horiz. Bar Diagrams, Color Code diagrams and Pictogram diagrams.<br>• New column "sequence" and "status_map_id" in table "page_element_sensors"<br>• New columns "title_font_size", "title_text_color", "legend_font_size", "legend_text_color", "scale_font_size", "scale_text_color", "datetime_font_size", "datetime_text_color", "background_color", "axis_color" and "pointer_color" in table "page_elements"<br>• New tables "status_map", "status_map_entry" and "pictogram_bitmap" |
| March 2007 | P. Rau | Version 1.2.1<br>• Bug Fix Scheduling Export/Import for "Run Only" and "Run+Import" Jobs.<br>• Bug Fix Export/Import with "blanks in external program name/path"<br>• Retries on file I/O errors in SmartWeb. |
| March 2007 | P. Rau | Version 1.2.2<br>• UMB sensors now all set to "inactive" when new device configuration is read<br>• For UMB device sensors: multiple selection of sensors in station sensor list dialog and pop-up menu to set selected sensors active/inactive<br>• New Param for SmartWeb: max. number of analog gauges per line (for station pages)<br>• New page element for station page: "New line" to separate analog gauges manually<br>• Changed timing for communication with Opus2/HP100 |
| March 2007 | P. Rau | Version 1.2.3<br>• Changed timing for communication with Opus2/HP100 (fix for problem setting time)<br>• Bug fix "new station schedule entry"<br>• Bug fix "DST change" Timestamp problem |
| April 2007 | P. Rau | Version 1.2.4<br>• Tests / Installation note for Microsoft ® Vista © systems<br>• Changed "external program start" from "system()" call to "CreateProcess()" call, so no command window is opened<br>• New Export/Import job types: "Export+Import" (without run) and "Run with external Export" (for external programs that access database directly)<br>• Bug Fix for Timestamp problem SmartCom/Export<br>• Improved Thread synchronization |
| May 2007 | P. Rau | Version 1.2.5<br>• Bug Fix "scaling analog gauges"<br>• optimized memory usage |

| | | |
|---|---|---|
| May 2007 | P. Rau | Version 1.2.6<br>• enhanced logging/error handling SmartWeb |
| May 2007 | P. Rau | Version 1.2.7<br>• Bug fix wrong attributes for archive pages (when re-generating archive pages)<br>• “Calculated channels “<br>• Reports with min/max/avg/sum on day/month/year per station in SmartView3<br>• Reports as “page element” for station and data page<br>• New station schedule times – every 5/10/12/20/30 minute to the full hour<br>• new tables “calc_channel” “calc_channel_sensor” and “calc_channel_param”<br>• new column “time_iv” for table “page_elements” |
| June 2007 | P. Rau | Version 1.2.8<br>• Bug Fix “inconsistent horiz. Bar diagrams” when sensor values are not in regular intervals<br>• Optimized station page update |
| September 2007 | P. Rau | Version 1.2.9<br>• Wake-Up message for Opus20x after establishing connection |
| October 2007 | P. Rau | Version 1.2.10<br>• HTTP transfer for CAM picture<br>• Improved error handling for reading UMB device configuration |
| November 2007 | P. Rau | Version 1.2.11<br>• New “Calc Channel” type “DIFF” to calculate precipitation from absolute value<br>• New “Value Map” type “Plausibility Check” for UMB Stations<br>• New columns “plausi_mode, plausi_type and plausi_value” for table “value_map” |
| November 2007 | P. Rau | Version 1.2.12<br>• New table “last_sensor_values”<br>• New Params “Autostart Collector” and “Autostart SmartWeb” for SmartView3<br>• Fixed bug caused by rounding error in “DIFF” calculation<br>• Function to delete sensor values from database |
| December 2007 | P. Rau | Version 1.3.0<br>• For new installation sensor values are now stored as “double” instead of “float”. For older installations this can be changed manually in the database by changing the appropriate columns in “sensor_values” and “last_sensor_values”.<br>• Fix for firmware bug in UMB device class 7 (error when reading device configuration)<br>• New param “only keep last picture” for collector -> only the last picture data is stored in table file_transfers.<br>• Size of “station icon” now configurable via station icon element width/height<br>• Station display (“Edit/Stations”) in SmartView can now be sorted by Station-ID, Name, Type, Next-Poll, Last Data or Status by clicking on list column header, and list view is re-sized with window size<br>• Auto-Refresh for “separate table” page of data page now does not switch to diagram page anymore.<br>• Auto Refresh timer can now be configured for pages with auto refresh (map page, station page and data pages)<br>• New column “auto_refresh_iv” on table “pages” and “archive_pages”<br>• Sensors for Reports on Pages can now be configured as “Sum” sensor if they default to “min/max/avg” sensor<br>• New column “report_sum_channel” for table “page_element_sensors”<br>• New function to clear logger memory for Opus200/Opus2 and HP100 devices |
| January 2008 | P. Rau | Version 1.3.1<br>• Bug fix “delete report” (reports could not be deleted from pages)<br>• Bug fix “disable delete calc sensor” for non-admins (button was not disabled)<br>• New Status Map entry for “RC Lufft Defautl” -> 7 = Critical<br>• Bug fix “Enable edit status map” when called from main menu<br>• Improved error and timeout handling for http cam picture transfers<br>• Improved timeout handling for ftp transfers in SmartWeb<br>• Improved semaphore (critical section) handling<br>• New parameter “EXPORT-EXEC-TIMEOUT” and “DATA-ARCHIVE-EXEC-TIMEOUT” for SmartCom<br>• Bug Fix size of displayed map on open of map edit dialog<br>• New function “Reset Page” on single page of a site (not only for all pages for a site) |

| | | |
|---|---|---|
| | | • SmartWeb: Optimized page generation on startup and site configuration changes::<br>Station pages are now only updated if there is new data for the station; menu and list pages are only updated if data pages were updated (because of new data). So if there was no new data and no configuration change since last startup of SmartWeb NO pages are generated at all.<br>• Text Elements can now be placed on Station and Data pages between diagrams etc.<br>• Improved handling of changes to export job sensors for "since last export" time (when new export sensor is added, time starts at first available data for this sensor, not at "all available data" now)<br>• Sequence for link entries in "station list", "data page list" and "archive page list" can now be changed<br>• New column "sequence" for table "pages"<br>• Sensor value type (min/max/avg….) is now shown in line diagram legend and table header.<br>• Modem-Pool is now displayed in "Station List"<br>• New station type "IMPORT" |
| February 2008 | P. Rau | Version 1.3.2<br>• Improved calc channel (diff) performance<br>• Fixed problem with combining log files at day break<br>• Fixed display of log file size in Collector setup dialog<br>• Improved timer handling (using TickCount() instead of absolute time whereever possible)<br>• Corrected error messages<br>• Reduced error messages for Opus200 on successful retries when reading logger data<br>• Limited "retries within retries" for reading Opus200 logger and EEProm data<br>• IRS21 Diag Sensors are now set "inactive" for Opus200 on default. Sensor values are still read from device if "diag" is active, but (inactive) sensors are not included in SmartWeb configuration. |
| February 2008 | P. Rau | Version 1.3.3<br>• Show (UMB) error code in sensor status and tables<br>• Improved error messages for export/import (include job id/name)<br>• Force a "close database" on station poll thread timeout, and on "collector inactivity timeout" (-> special error handling for customer problem application "hanging" in DB access) |
| February 2008 | P. Rau | Version 1.3.4<br>• Bug Fix: when saving last sensor value in separate table, now the latest timestamp from a list of values is used instead of the last value in the list (problem with un-sorted import, e.g. Venturia plant disease model)<br>• For UMB Stations: when reading device/sensor configuration from station for the first time, it is checked if a device of this type and version is already stored in the database. If so, the sensor configuration is copied from the database instead of read from the device.<br>• New station state "Saving Sensor Config" when saving the sensor configuration read from a station<br>• Optimized saving of sensor configuration for a station (assignment of sensor types etc..)<br>• Improved error handling on (remote) database connection problems<br>• Changed IPC host to "localhost" if all programs run on local computer (no network connection needed; IPC does not fail anymore if network connection goes down for whatever reason) |
| March 2008 | P. Rau | Version 1.3.5<br>• Bug Fix "Critical Section" for Windows2003 Server |
| March 2008 | P. Rau | Version 1.3.6<br>• Specify number of pictures to keep for each cam station<br>• New column "keep_num_cam_pics" for table "station_files"<br>• Bug fix "Critical Section Handling" |
| March 2008 | P. Rau | Version 1.3.7<br>• Bug Fix DB access for new Export Job staring "last export"<br>• Cam Picture Archive<br>• Improved error logging for critical sections |
| March 2008 | P. Rau | Version 1.3.8<br>• Additional/Changed error checking for critical sections |
| March 2008 | P. Rau | Version 1.3.9<br>• Bug-Fix "deleting archive pages on config change of page"<br>• Change station page and data page names for a station if station name changes |
| | | Version 1.3.10<br>• Consistency check site sensor configuration with active/configured station sensors |

| | | |
|---|---|---|
| | | • Improved Opus200 communication for bad line quality links<br>• Fixed bug calculating measurement times if "specific offset to UTC" is configured for station |
| March 2008 | P. Rau | Version 1.3.11<br>• Changed automatic re-generation of archive pages (if a new sensor/diagram is added or data for archive page is now available) not to re-generate all archive pages, but to start with archive page for which we actually have new data, i.e. if a new sensor is added to the station/data page, only the current page will be affected (and no archive pages will be re-generated) |
| April 2008 | P. Rau | Version 1.4.0<br>• Improved error handling for Opus200 station (reading logger data)<br>• Logical Sensor Types (Sensor Type Templates)<br>• New table "sensor_type_template"<br>• New column "template_id" to table "station_sensors"<br>• New column "template_id" to table "sensors"<br>• Templates for Export/Import Jobs<br>• New columns "is_template","job_template_id" and "template_type_id" to table "export_job"<br>• New columns "sensor_template_id" and "unit name" to table "export_sensors"<br>• New column "sensor_template_id" to table "import_sensors"<br>• Templates for station and data pages<br>• New columns "is_template" and "template_type_id" to table pages<br>• New columns "is_template" and "template_id" to table page_elements<br>• New columns "is_template" and "template_id" to table "page_element_sensors"<br>• Links to "data pages" for a station on station page<br>• Table with last sensor values for each station on "Overview" page<br>• Status Mapping color code shown in tables<br>• Sensors don't need to be maintained on "page" level anymore (only assigned to page elements now) |
| April 2008 | P. Rau | Version 1.4.1<br>• Dialog to assign logical sensor type added to station's sensor list dialog<br>• Fixed scrolling problem for dynamic dialogs when dialog size exceeds screen size |
| April 2008 | P. Rau | Version 1.4.2<br>• Bug Fix entries menu "Setup" when Edit/Stations or Edit/Web-Site is opened<br>• Bug Fix "Edit Log. Sensor Type" |
| April 2008 | P. Rau | Version 1.4.3<br>• Bug Fix Export "since last export" (saving export job might have changed last_export_date for sensor)<br>• Optimized database updates for export jobs (updating last_run/next_run for job and last_export for export sensors, last_import for import sensors) |
| May 2008 | P. Rau | Version 1.4.4<br>• New Export Data Start "last value only"<br>• Allow configuration of Import sensors for Export Job type "Run (external export)"<br>• Bug fix sensor error values in horiz. Bar Diagrams<br>• Don't display last sensor values on overview table if station could not be polled<br>• Display station name without additional "Station" text in overview table |
| June 2008 | P. Rau | Version 1.4.5<br>• Bug fix overview table<br>• Bug fix sensor status in station page table |
| June 2008 | P. Rau | Version 1.4.6<br>• When "re-calculating calc channel for all available data", the calculation is now broken down to daily intervals instead of reading/calculating/writing all available data for a sensor in one block to handle large amount of data for a sensor better<br>• Import can handle date/time values without leading zeros now |
| August 2008 | P. Rau | Version 1.4.7<br>• Bug fix Data Archive Job run every year<br>• Bug fix "Delete Pages older..:"<br>• Bug fix displaying last sensor status (error/alarm) on Station page<br>• Alarm via SMS<br>• Alarm on Status<br>• New Alarm handling – Repeated alarms only after configured time. Minimum time between alarms.<br>• Working hours for alarm recipients.<br>• Changed retry handling on reading Opus20x EEProm and Logger-Data (more "inner" retries on invalid blocks, no "inner" retries on no |

| | | |
|---|---|---|
| | | response)<br>• Changes to database:<br>- new column “alarm_level” on table “status_map_entry”<br>- new column “alarm_status_map_id” on table “station_sensors”<br>- new columns “user_id”, “is_active”, “alarm_reasons” and “working_hours_id” on table “alarm_recipient”<br>- new table “working_hours_entry”<br>- new table “working_hours”<br>- new table “alarm_event”<br>- new table “sent_alarm” |
| August 2008 | P. Rau | Version 1.4.8<br>• Support for TLSoIP stations<br>• New Table “tls_stations” |