SmartView3

Database Tables

Version 2.3.1
June 2014

Contents

SmartView3 Database – Table Description 4
Table “sensor_values” 4
Table “stations” 5
Table “opus_stations” 6
Table “ppp_stations” 6
Table “sensors” 7
Table “units” 7
Table “station_sensors” 7
Table “station_opuses” 9
Table “station_files” 9
Table “file_transfers” 9
Table “params” 10
Table “users“ 10
Table “user_group” 10
Table “user_privileges” 10
Table “group_privileges” 10
Table “sites” 11
Table “site_stations” 12
Table “pages” 12
Table “page_params” 12
Table “page_elements” 13
Table “page_element_sensors” 14
Table “archive_pages” 14
Table “export_job” 14
Table “export_params” 16
Table “export_sensors” 16
Table “import_sensors” 17
Table “station_schedule” 17
Table “alarm_group” 18
Table “alarm_recipient” 18
Table “data_archive_jobs” 18
Table “value_map” 19
Table “value_map_table” 19
Table “config_change” 19
Table “status_map” 19
Table “status_map_entry” 20
Table “pictogram_bitmap” 20
Table “calc_channel” 20
Table “calc_channel_sensor” 20
Table “calc_channel_param” 21
Table “last_sensor_values” 21
Table “sensor_type_template” 21
Table “working_hours” 21
Table “working_hours_entry” 22
Table “alarm_event” 22
Table “sent_alarm” 23
Table “tls_stations” 23
Table “error_mapping” 23
Table “error_map_entry” 23
Table “ntcip_station_metadata” 23
Table “station_groups” 24
Table “smartweb_ftp” 24
Table “mobile_sensor_values” 24
Table “last_mobile_sensor_values” 24
Table “last_mobile_station_position” 24
Table “station_for_template_export” 25
Table “sensor_alarm_settings” 25
Table “alarm_status” 25
Table “kml_history_entries” 26

Table “last_kml_history_entry” 26

SmartView3 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/key” between “sensor_values” and “station_sensors” is column “station_id”,”sub_id” and “channel”). Table “sensors” holds general information about the device specific sensor types (name of sensor type, min/max values etc.). Table “units” holds the unit description for a sensor/value. All other tables contain mainly configuration information used by the collector software.

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: for OpusII and Opus200 type statione, depending on the device configuration, a single sensor channel may provide up to three different value types (average, minimum, maximum, sum). 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	1=min,2=max,3=avg,4=sum, 5=imp (imported), 6 = calc(ulated), 7 = act(ual) , 8 = vect avg (vectorial average), 9 = modal
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 device sensor type (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).

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)~~ longblob	descriptive name for station **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
model	~~varchar(255)~~ longblob	model of station (e.g. Lufft Opus 200) **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
station_type	int	type of station. Changed from Version 2.1 upwards (will be converted automatically when upgrading). For Versions earlier than 2.1 : 1 = Opus 200/RS232, 2 = Opus 200/Modem. From Version 2.1 on : 10 = Opus200, 12 = Axis
connection_type	int	Type of connection to station (new in Version 2.1) 1 = RS232, 2 = TAPI/Modem, 3 = TCP/IP (only Axis Server at the moment) 4 = RAS/PPP (only Axis Server)
location	~~varchar(255)~~ longblob	location (e.g. city/street etc.) **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
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) – OBSOLETE !
poll_winter	unsigned int	poll interval in secs (if params:operations_mode is winter) – OBSOLETE !

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	Timestamp (UTC) for last time data was imported for station
from_time	Int unsigned	Timestamp (UTC) - time for reading logger data
modem_pool_id	Int unsigned	Id for modem pool to be used for this station
display_time_offset	int	Time offset for displaying data in SmartWeb (relative to server time)
tz_name	Longblob	Time zone name (information only)
mssi_station_id	int unsigned	MSSI Station-Id for station
mssi_cam_id	int unsigned	MSSI Cam ID for (camera) station
station_group_id	int unsigned	Group id for station
support_prognosis	int unsigned	(bool) station supports prognosis values (for LCom via MSSI with prognosis module active)
mobile_id	bigInt unsigned	ID (mac address) of mobile NIRS station

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

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)
snmp_rw_community	varchar(255)	Community name with write access für snmp connection (NTCIP)
ntcip_cam_idx	int unsigned	Index for NTCIP camera in camera table

Table "sensors"

This table holds information about the device sensor types.

Column	Type	Description
sensor_id	int	unique id for a sensor
name	~~varchar(255)~~ longblob	descriptive name for sensor (read from device if supported byt the device type) **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
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 for device type
irs21_sensortype	int	IRS21 sensor type/channel or device sensor „sub type"
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
na_value_active	int	(bool) the N/A value is active/set
na_value	double	The N/A value (shown on web page as „N/A")
is_datetime	int unsigned	(bool) sensor's measure value is a datetime value

Table "units"

This table holds information about units

Column	Type	Description
unit_id	int	unique id for a unit
name	~~varchar(255)~~ longblob	descriptive name for unit (e.g. "temperature") **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
unit	~~varchar(255)~~ longblob	Unit string (e.g. °C, m, m/s, %, mm, imp/min, °) **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
opus_unit	char(255)	Unit string as defined in Opus (2 chars in Opus200, 6 chars in Opus208)

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 device sensor type (refers to entry in table “sensors”)
sensortype	int	sensor type as configured in Opus (obsolete – see table sensors colum „opus_sensortype“)
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)~~ longblob	Name for Sensor **NOTE: from version 1.6.14 on this column will be created as or converted to longblob, and the string is stored as 16 bit unicode in hex!**
alarm_active	int	Alarm for sensor is active (0 = false, != 0 = true)
~~alarm_min~~	~~float~~	~~Lower alarm limit/boundary~~ **NOTE: from version 1.11.0 on this column is OBSOLETE and (for new databases) no longer created (-> see table sensor_alarm_settings)**
~~alarm_max~~	~~float~~	~~Upper alarm limit/boundary~~ **NOTE: from version 1.11.0 on this column is OBSOLETE and (for new databases) no longer created (-> see table sensor_alarm_settings)**
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)~~ **NOTE: from version 1.11.0 on this column is OBSOLETE and (for new databases) no longer created (-> see table sensor_alarm_settings)**
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”~~ **NOTE: from version 1.11.0 on this column is OBSOLETE and (for new databases) no longer created (-> see table sensor_alarm_settings)**
sum_channel	Int unsigned	(bool) channel is treated as „Sum“ channel even if channel_mode does not indicate that it is a „sum“ channel (for reports).
error_map_id	int unsigned	Error mapping id for his sensor
is_prognosis	int unsigned	(bool) sensor channel is a prognosis sensor channel (causes Collector to remove all “old”

		values when storing new values)
souce_channel_sensor_nr	int unsigned	Sensor nr for source channel (NA Value mapping)

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)

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".

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	Timestamp (UTC) when 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

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

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

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

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

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

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
use_popup_menu	int unsigned	(bool) use popup menu instead of plain html links
show_cam_icon	Int unsigned	(bool) show cam icon on station icon in map
sort_stations_by_name	int unsigned	(bool) sort stations in lists (station page list, diagram page list) by name
site_url	Varchar(255)	URL for site in internet (for google maps)
enable_alarm_popup	int unsigned	(bool) open popup window on alarms

enable_audio_alarm	int unsigned	(bool) enable audio on alarms

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

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

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)

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
alarm_limit_color	int unsigned	Color for alarm limits in analog gauge (vert. bar diagram and round gauge)
show_alarm_status	int unsigned (bool)	Show alarm status in analog gauges (border around gauge in alarm color)

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.
calc_tendency_iv	int	(enum) time interval to check for last data when calculating tendency
calc_tendency_delta	double	Delta between values to calculate tendency (up/down..)
show_alarm_max_limit	int unsigned	(bool) show upper alarm limit in analog gauge or line diagram
show_alarm_min_limit	int unsigned	(bool) show lower alarm limit in analog gauge or line diagram

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)

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 a) transfer the export file for jobs of type EXPORT or of type EXPORT AND IMPORT after the export file has been written 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
import_delete_old_data	int unsigned	(bool) delete all (old) data for import sensors before writing new data to database (for prognosis/forecast)
thread_id	int unsigned	Internal id for thread
run_in_collector	int unsigned	(bool) run export job from collector instead of smartcom
export_add_utc_timestamp	int unsigned	(bool) add an additional UTC timestamp in first column of CSV Export

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)

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)
export_id1	int unsigned	Export type specific id 1 for sensor (e.g. FG for TLS type export)
export_id2	int unsigned	Export type specific id 2 for sensor (e.g. DE channel for TLS type export)
export_id3	int unsigned	Export type specific id 3 for sensor (e.g. DE type for TLS type export)
status_map_id	Int	ID for status map (export output will be text from status mapping)

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_import	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

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

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)

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
num_sms	int unsigned	Number of SMS sent to this recipient
price_sms	double	Price for sending one SMS

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

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	double	Value used to scale
offset	double	Offset for calculating scaled values
template	Int unsigned	Internal enum for type of mapping (used for "program known" mapping types only)

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	double	Lower limit of value interval
val_max	double	Upper limit of value interval
map_val	double	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

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…)

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)

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

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)

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)
iv_time_offset	int	Time offset in seconds for calc_interval

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)

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)

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).

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
default_status_map	int unsigned	status maping id for sensor type

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

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

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	double	Sensor value that caused the alarm event
value_error	int	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)
sensor_nr	int unsigned	Sensor nr for this alarm event

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)

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)

Table "error_mapping"

This table holds information for error mapping tables.

Column	Type	Description
error_map_id	int unsigned	unique id for error mapping table
Name	blob	Name for error mapping table (unicode string stored as hex string

Table "error_map_entry"

This table holds information for error mapping table entries.

Column	Type	Description
error_map_id	int unsigned	unique id for error mapping table
error_code	int	error code to be mapped
error_map_type	Int unsigned	(enum) type of error mapping
map_value	double	value to replace error value
error_text	Blob	special error text to be displayed in SmartWeb pages (unicode string stored as hex string)
new_error_code	Int	special error code to replace original error code

Table "ntcip_station_metadata"

This table holds meta data information for NTCIP stations

Column	Type	Description
station_id	int unsigned	unique id for station
param_name	varchar(255)	string representation for OID of data entry
param_type	int unsigned	Type of parameter (internal enum)

param_value	longblob	value for data entry – stored as hex string
asn_type	int unsigned	ASN (snmp) data type for OID
value_time	bigint unsigned	UTC timestamp when param value was read last
read_from_station	int unsigned	(bool) flag indicating if this OID is actually read from station (or preconfigured)

Table "station_groups"

This table contains data for station groups

Column	Type	Description
group_id	int unsigned	unique id for group
group_name	longblob	Name of group (Unicode string stored as hex string)
parent_group_id	int unsigned	Id for parent group of group (0 for top level groups)

Table "smartweb_ftp"

This table contains the queue data for smartweb ftp transfers

Column	Type	Description
site_id	int unsigned	Smartweb site id
local_file	varchar(255)	Local filename (without path)
needs_transfer	int unsigned	(bool) flag if file needs to be transferred
creation_time	bigint unsigned	UTC Timestamp file was last written (local)
remote_file	varchar(255)	Remote filename (without path)
transfer_time	bigint unsigned	UTC Timestamp file was last transferred
transfer_type	int unsigned	(enum) type of transfer (put, delete..)

Table "mobile_sensor_values"

This table contains the sensor values for mobile stations/sensors

Column	Type	Description
measure_time	int unsigned	UTC timestamp
station_id	Int unsigned	The station id
sensor_nr	Int unsigned	Sensor nr of this sensor
longitude	double	The longitude
latitude	double	The latitude
altitude	double	The altitude
speed	double	The speed for the station
course	double	The course/heading
horizontal_accuracy	double	The horizontal accuracy (for latitude/longitude)
value	double	The measure value for this time/location/sensor
error	double	Error indicator for this measure value

Table "last_mobile_sensor_values"

This table is used to cache the respective last sensor value for a mobile statio/sensor. See "mobile_sensor_value".

Table "last_mobile_station_position"

This table contains the last known position of a mobile station

Column	Type	Description
station_id	int unsigned	The station id
measure_time	Int unsigned	UTC timestamp

longitude	double	The longitude
latitude	double	The latitude
altitude	double	The altitude
speed	double	The speed
course	double	The corse/heading
horizontal_accuracy	double	The horizontal accuracy (for latitude/longitude)

Table "station_for_template_export"

This table is used to configure the stations to be used for template export

Column	Type	Description
job_template_id	Int unsigned	The template job id
station_id	int unsigned	The station id

Table "sensor_alarm_settings"

This table is used to configure the alarm parameters for a station sensor channel. See chapter Sensor Alarm in SmartView3 manual for details.

Column	Type	Description
station_id	int unsigned	Station id for sensor
sensor_nr	int unsigned	Sensor nr for sensor (station_id and sensor_nr are primary key)
alarm_mode	int unsigned	(enum) the alarm calculation mode
alarm_group	int unsigned	The recipient group id for alarm recipents
alarm_min	double	Lower alarm threshold
alarm_max	double	Upper alarm threshold
alarm_status_map_id	int unsigned	The status mapping id used for alarm condition calculation
alarm_minimum_threshold_violation_period	int unsigned	Minimum time period (in seconds)
alarm_minimum_threshold_violation_sample_count	int unsigned	Minimum number of samples in period
alarm_value_quality_sensor_nr	int unsigned	Sensor nr for quality sensor
alarm_value_minimum_quality_value	int unsigned	Minimum quality value
alarm_value_maximum_quality_value	int unsigned	Maximum quality value
alarm_email_subject_1	longblob	Alarm email subject text (Unicode stored as hex string)
alarm_email_subject_2	longblob	Alarm email subject text (Unicode stored as hex string) – for future use
alarm_text_1	longblob	Alarm text (Unicode stored as hex string)
alarm_text_2	longblob	Alarm text (Unicode stored as hex string) – for future use
alarm_text_3	longblob	Alarm text (Unicode stored as hex string) – for future use

Table "alarm_status"

This table is used to store alarm status details (current/last alarm status)

Column	Type	Description
station_id	int unsigned	Station id for sensor
sensor_nr	int unsigned	Sensor nr for sensor (station_id and sensor_nr are primary key)
export_job_id	Int unsigned	export job id

is_alarm	int	(Boolean) is an alarm value
alarm_reason	int unsigned	(Enum) reason for alarm
status_creation_time	BIGINT	UTC Timestamp – creation time of status
alarm_measure_time	BIGINT	UTC Timestamp – value causing the alarm
alarm_measure_value	double	Value causing the alarm
alarm_value_error	int	Error indicator for alarm value

Table "kml_history_entries"

This table is used to store the kml file history used with google maps

Column	Type	Description
last_value_time	BigInt	UTC Timestamp – time of last measurement in file
site_id	Int	Site id of map page
page_id	Int	Page id of map page
filename	Longblob	The file name for the kml file, in ansi, stored as hex blob
kml_data	Longblob	The kml data (content) of the file, stored as hex string

Table "last_kml_history_entry"

This table is used to store the last (most current) kml data for a map page

Column	Type	Description
site_id	Int	Site id of map mape
page_id	Int	Page id of map page
last_value_time	BigInt	UTC Timestamp – time of last measurement in file
filename	Longblob	The file name for the kml file, in ansi, stored as hex blob
kml_data	Longblob	The kml data (content) of the file, stored as hex string

Document History

September 2005	P. Rau	Version 1.0
September 2006	P. Rau	Version 1.1.0 • New table “data_archive_jobs” • New columns “export_date_header” and “export_time_header” for table “export_jobs” • New tables “value_map” and “value_map_table” New column “value_map_id” for table “station_sensors” New column “value_map_id” for table “export_sensors” New column “value_map_id” for table “import_sensors” • New column “sample_int” for table “station_sensors” • New table “config_changes” • New column “modem_pool_id” for table “stations”
November 2006	P. Rau	Version 1.1.1 • New columns table “export_job” : colum “is_active”, “import_only_new_files”, “ftp_delete_source”, “import_delete_after”, “export_no_values” Table “sites” : new column “is_active”
November 2006	P. Rau	Version 1.1.3 • Changes for MySQL Version 5.0.27 in create table statements • New columns for table “sites” : “auto_delete_pages”, “auto_delete_iv_counter” and “auto_delete_iv”
Feb. 2007	P. Rau	Version 1.2.0 • New column “sequence” and “status_map_id” in table “page_element_sensors” • 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” • New tables “status_map”, “status_map_entry” and “pictogram_bitmap”
May 2007	P. Rau	Version 1.2.7 • new tables “calc_channel” “calc_channel_sensor” and “calc_channel_param” • new column “time_iv” for table “page_elements”
November 2007	P. Rau	Version 1.2.11 • New columns “plausi_mode, plausi_type and plausi_value” for table “value_map”
November 2007	P. Rau	Version 1.2.12 • New table “last_sensor_values”
December 2007	P. Rau	Version 1.3.0 • New column “auto_refresh_iv” on table “pages” and “archive_pages” • New column “report_sum_channel” for table “page_element_sensors”
January 2008	P. Rau	Version 1.3.1 • New column “sequence” for table “pages”
March 2008	P. Rau	Version 1.3.6 • New column “keep_num_cam_pics” for table “station_files”
April 2008	P. Rau	Version 1.4.0 • New table “sensor_type_template” • New column “template_id” to table “station_sensors” • New column “template_id” to table “sensors” • New columns “is_template”,”job_template_id” and “template_type_id” to table “export_job” • New columns “sensor_template_id” and “unit name” to table “export_sensors” • New column “sensor_template_id” to table “import_sensors” • New columns “is_template” and “template_type_id” to table pages • New columns “is_template” and “template_id” to table page_elements • New columns “is_template” and “template_id” to table “page_element_sensors”
August 2008	P. Rau	Version 1.4.7 • new column “alarm_level” on table “status_map_entry” • new column “alarm_status_map_id” on table “station_sensors” • new columns “user_id”, “is_active”, “alarm_reasons” and “working_hours_id” on table “alarm_recipient” • new table “working_hours_entry” • new table “working_hours” • new table “alarm_event” • new table “sent_alarm”
August 2008	P. Rau	Version 1.4.8 • New Table “tls_stations”
September 2008	P. Rau	Version 1.4.9 • New columns “display_time_offset” and “tz_name” for table “stations”
October 2008	P. Rau	Version 1.4.11 • new columns “num_sms” and “price_sms” for table “alarm-recipient”

		• New column “import_delete_old_data” on table “export_job”
January 2009	P. Rau	Version 1.5.0 • New column “sum_channel” in table “station_sensors”
February 2009	P. Rau	Version 1.5.3 • New column “error_map_id” in table “station_sensors” • New tables “error_mapping” and “error_map_entry”
June 2009	P.Rau	Version 1.6.0 • New columns “mssi_station_id”, “mssi_cam_id” and “station_group_id” for table “stations”
July 2009	P. Rau	Version 1.6.2 • New columns “calc_tendency_iv” and “calc_tendency_delta” for table page_element_sensors
October 2009	P. Rau	Version 1.6.7 • New columns “export_id1”, “export_id2” and “export_id3” for table export_sensors
November 2009	P. Rau	Version 1.6.12 • New columns “na_value_active” and “na_value” for table “sensors”
December 2009	P. Rau	Version 1.6.14 • Following columns will now be created as (or converted to) „longblob“ instead of „varchar“, and the string is stored as 16 bit unicode in hex**!** table „stations“ columns „name“ „model“ and „location“ table „sensors“ column „name“ table „units“ columns „name“, and „unit“ table „station_sensors“ column „name“
August 2010	P. Rau	Version 1.6.22 • New column “iv_time_offset” in table “calc_channel”
October 2010	P. Rau	Version 1.7.0 • New column “snmp_rw_community” in table “ppp_stations” • New column “ntcip_cam_idx” in table “ppp_stations”
June 2011	P. Rau	Version 1.7.8 • New column “support_prognosis” in table “stations” • New column “is_prognosis” in table “station_sensors” • New column “is_datetime” in table “sensors” • New table “ntcip_station_metadata”
August 2011	P. Rau	Version 1.8.0 • New column “thread id” and “run_in_collector” on table “export job” • New column “use_popup_menu” in table “sites • New column “show_cam_icon” in table sites • New column “default_status_map” in table “sensor_type_template”
October 2011	P. Rau	Version 1.8.2 • New table “station_groups” • New column “sort_stations_by_name” in table “sites”
October 2011	P. Rau	Version 1.8.3 • New columns “source_channel_sensor_nr” in table “station_sensors”
October 2011	P. Rau	Version 1.8.4 • New table “smartweb_ftp”
September 2012	P. Rau	Version 1.9.0 • New column “mobile id” for table “stations” • New column “site url” for table “sites” • New table “mobile_sensor_values” • New table “last_mobile_sensor_value” • New table “last_mobile_station_position”
December 2012	P. Rau	Version 1.10.0 • New column “export_add_utc_timestamp” for table “export_job” • New table “station_for_template_export”
January 2013	P. Rau	Version 1.11.0 • New column “sensor_nr” for table “alarm_event” • New table “sensor_alarm_settings” • OBSOLETE/REMOVED columns “alarm_group”, alarm_min, alarm_max, alarm_status_map_id from table station_sensors
January 2013	P. Rau	Version 1.12.0 • New Columns “alarm_limit_color” and “show_alarm_status” on table “page_elements” • New Column “show_alarm_max_limit” and “show_alarm_min_limit” on page_element_sensors • New columns “enable_alarm_popup” and “enable_audio_alarm” on table “sites” • New table “alarm_status”
March 2013	P. Rau	Version 2.0.0 • New Licensing model (no changes to database tables)
April 2013	P. Rau	Version 2.1.0 • New logical sensor types (no changes to tables)
October 2013	P. Rau	Version 2.1.6

		• Changed template for combined road condition
November 2013	P. Rau	Version 2.2.0 • New column “status_map_id” on table “export_sensors”
June 2014	P. Rau	Version 2.3.0 • New tables “kml_history_entries” and “last_kml_history_entry”
June 2014	P. Rau	Version 2.3.1 • Added site_id to new tables kml_history_entries and last_kml_history_entry