The Alarms Professional plug-in
PRINTED MANUAL

Alarms Professional plug-in

© 2016-2025 AGG Software

Printed: 3/27/2025

Table of Contents

1   Introduction

The Alarms Professional module for our loggers allows you to analyze the received data in real time and to log alarms based on the specified rules.

You can create rules for these alarm levels: Low Low, Low, High, and High High. The module can send alarms by email or write them to a database (MySQL, MS SQL, PostgreSQL, or ODBC) or a CSV file. It can also log the date and time of each alarm's ending.

The module provides many settings and is very flexible. You can configure it for any tasks.

2   System requirements

The following requirements must be met for "Alarms Professional" to be installed:

Operating system: Windows 2000 SP4 and above, including both x86 and x64 workstations and servers. The latest service pack for the corresponding OS is required.

Free disk space: Not less than 5 MB of free disk space is recommended.

Special access requirements: You should log on as a user with Administrator rights in order to install this module.

The main application (core) must be installed, for example, Advanced Serial Data Logger.

3   Installing Alarms Professional

1. Close the main application (for example, Advanced Serial Data Logger) if it is running;
2. Copy the program to your hard drive;
3. Run the module installation file with a double click on the file name in Windows Explorer;
4. Follow the instructions of the installation software. Usually, it is enough just to click the "Next" button several times;
5. Start the main application. The name of the module will appear on the "Modules" tab of the "Settings" window if it is successfully installed.

If the module is compatible with the program, its name and version will be displayed in the module list. You can see examples of installed modules on fig.1-2. Some types of modules require additional configuration. To do it, just select a module from the list and click the "Setup" button next to the list. The configuration of the module is described below.

You can see some types of modules on the "Log file" tab. To configure such a module, you should select it from the "File type" list and click the "Advanced" button.
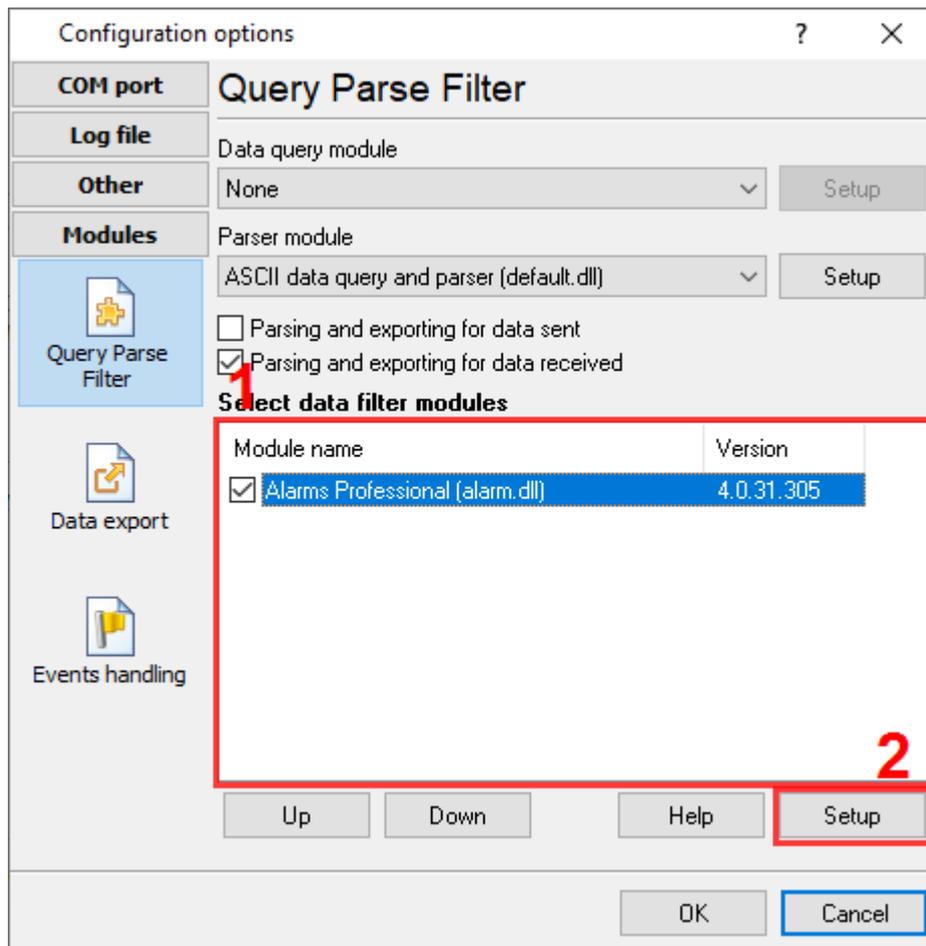
Fig. 1. Example of installed module

4 Glossary

Main program - it is the main executable of the application, for example, Advanced Serial Data Logger and asdlog.exe. It allows you to create several configurations with different settings and use different plugins.

Plugin - it is the additional plugin module for the main program. The plugin module extends the functionality of the main program.

Parser - it is the plugin module that processes the data flow, singling out data packets from it, and then variables from data packets. These variables are used in data export modules after that.

Core - see "Main program."

5　Configuring the module

5.1　Common

To configure the module, you need to do the following:

1. Create one or more groups. In the general case, a group is an alarm handling method. You can configure each group separately. For example, one group can send alarms by email, and another group can write them to a database.

2. Create one or more alarm handlers. For each alarm handler, you can specify its own alarm levels (Low Low, Low, High, and High High). For each alarm level, you can specify its own group. This way, you can individually specify an alarm handling method, all the way to the alarm level.

Please keep the following in mind:

1. The module works with parser variables. It compares them with the specified limits or uses them in expressions. So you need to select and configure a parser in the logger.
2. Alarms are analyzed and logged only when a new data packet or event are received from the parser. During the time between receiving two successive data packets, no alarms are analyzed or logged, and the module is "asleep."
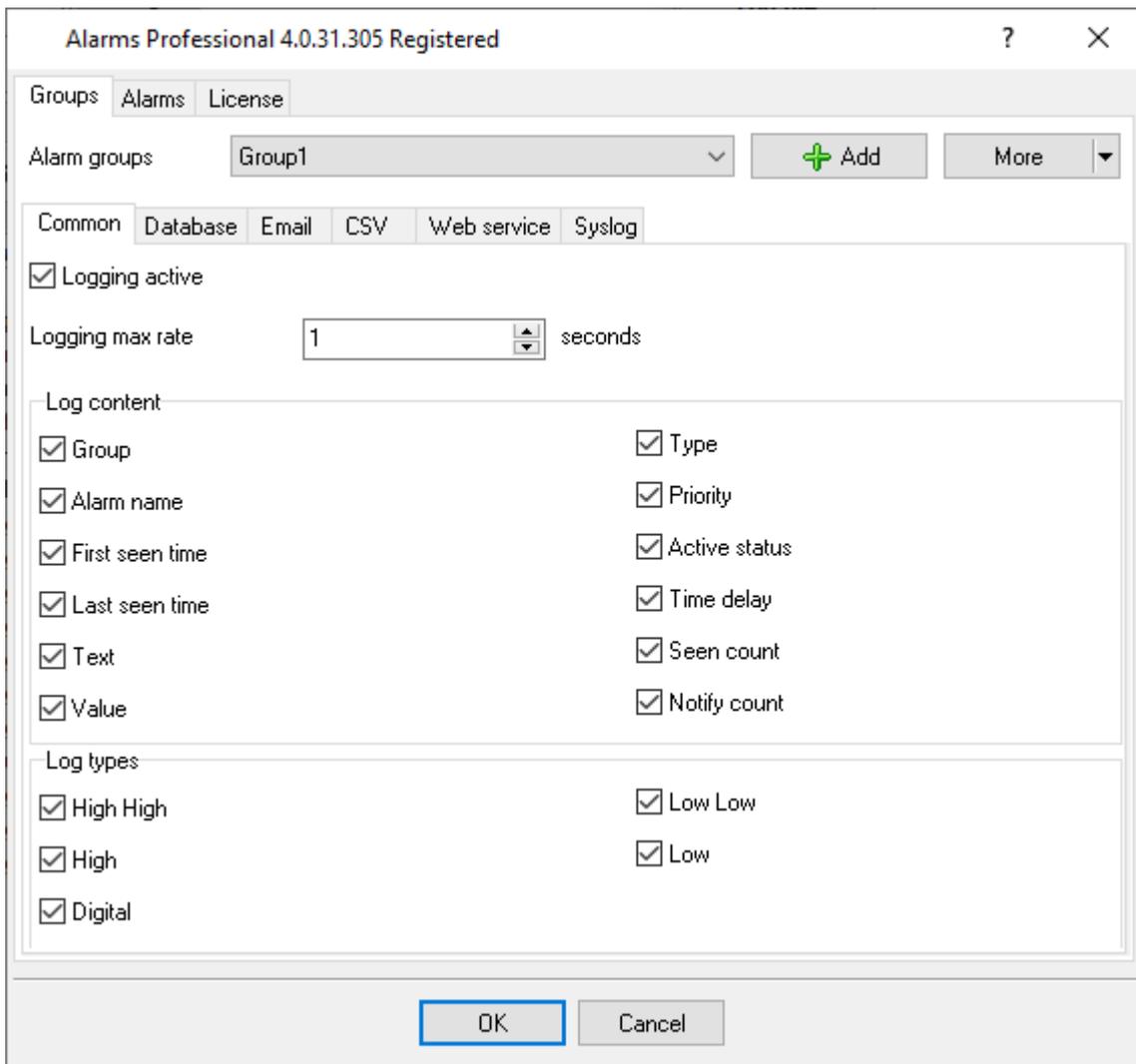
Figure 2: The module window

5.2 Groups

A group allows you to configure alarm handling. You can specify the content to be logged for each alarm and specify the recipient for alarm sending or logging.

To create a new group, click the "Add" button next to the groups drop-down list. The program will prompt you for the name of the group.

To delete, rename, or copy a group, click the "More" button and select the desired action from the drop-down menu.

Common

You can use the "Common" tab (figure 3) to configure the content to be logged and the data logging mode that is common for all logging methods.

Figure 3: Common group parameters

Logging active – You can enable or disable a group. If the group is disabled, alarms will not be logged.

Logging max rate – You can specify the minimum allowed interval between two consecutive alarms. Setting this parameter to zero will disable this filter. For example, if you set this parameter to 60 seconds, an alarm will be logged only once every 60 seconds, even if alarms occur more frequently.

Log content – You can specify the content to be logged (for example, written to a database or sent by email). If you want to write one set of data to the database and send another set by email when handling one alarm, you need to create two groups with their own parameters and two alarm handlers, and then "bind" them to different groups.

Group – The name of the group.
Alarm name – The name of the alarm.
First seen time – The date and time of the first ocurrence of an alarm.
Last seen time – The date and time of the last ocurrence of an alarm.
Value – The parser variable value or the expression value that triggered the alarm.
Type – The alarm type.
Active status – This parameter defines whether the alarm is active at the moment.
Seen count – The number of times an alarm was triggered, which depends on the number of packets received.
Notify count – The number of times an alarm was logged, which depends on the filters configured.

The following data correspond to the parameters of the alarm handler of the approrpiate type:

Text – The description of the alarm.
Priority – The priority of the alarm.
Time delay – The delay of the alarm.

Log type – You can select alarm levels for which you want to log alarms by using this group.

Database

Use this tab (figure 4) to specify the database connection settings and select the data write mode.

Log to database – You can enable or disable logging to the database.

Type – The database type. The ODBC data type allows you to use any database for which an ODBC driver exists.

Server – The server connection string. Usually it is the IP address of the server or the name of the computer. For a local MS SQL Server database, you may need to specify the database instance (for example, COMPUTER\SQLEXPRESS).

Database – The name of the database on the server.

Table – The name of the database table to log messages to. The table has a specific structure that cannot be changed.

Login/Password – The database user name and password. If you have turned the "Create database table automatically" option on, the database user must have the appropriate privileges.

Create database table automatically – The program will check if a table with the specified name exists in the database; if the table doesn't exist, the program will create a new table.

Insert new or update existing alarm record in table – If this option is turned on, the program will search for an entry of the alarm in the database. If such an entry exists, it will be updated; otherwise a new entry will be added. When searching for the entry in the table, all of these fields must match: "Group name," "Alarm name," "Priority," and "Time delay."

Figure 4: Database connection settings

You can use special substitute variables in the database name and in the table name. This way, you can log alarms to different databases and different tables depending on input data or alarm handling settings. For example, you can specify alarm{PRIORITY} as a table name. {PRIORITY} is a substitute variable for the Priority parameter. Before writing data to the database, the program will replace {PRIORITY} with the priority value, so the table name will become, for example, "alarm1" in the end. You can find the list of all substitute variables in the following section.

Email

Use this tab (figure 5) to configure the email sending settings.
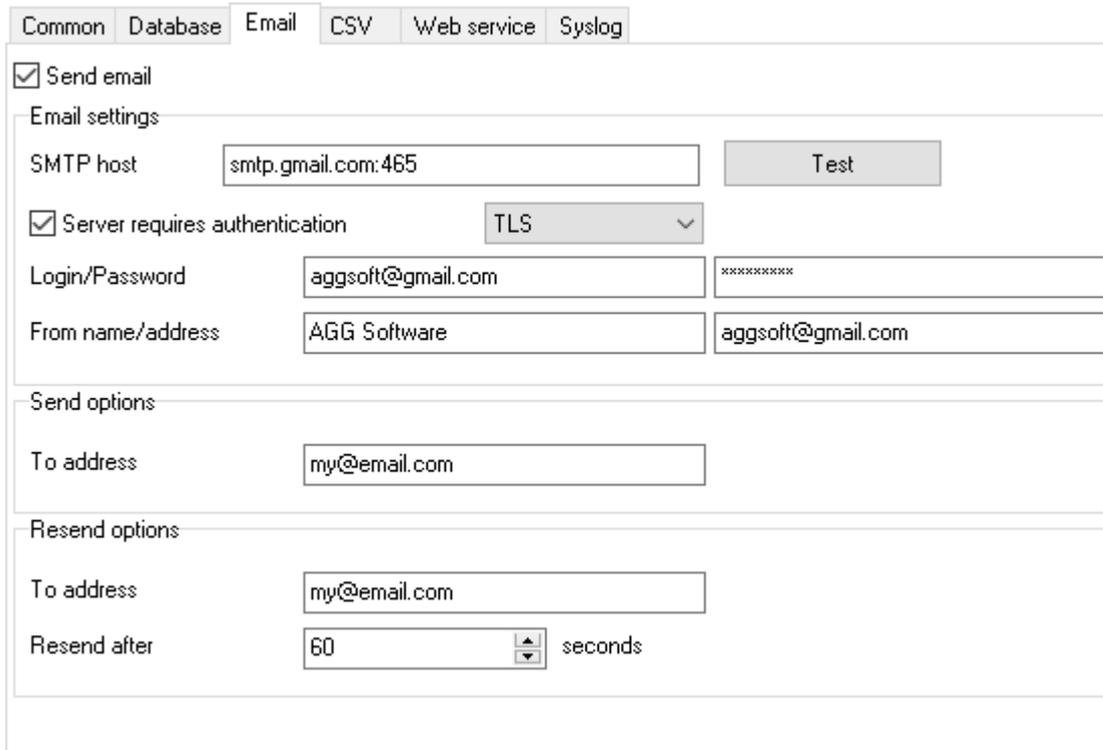
Figure 5: Email sending settings

Send email – You can enable or disable email sending.

SMTP host – The program will use this SMTP server to send messages. You can enter the server's IP address or domain name. You can also specify the port number on the server by using the semicolon as a separator.

Server requires authentication – Some SMTP servers require authentication for email sending. If that is the case, turn this option on.

Login – The SMTP server user name.

Password – The SMTP server user password.

From (address) – When generating an email message, this address will be used for the From field. This is a mandatory field.

From (name) – When generating an email message, this name will be added to the From field. This is an optional field.

To address – The address of the recipient of the first alarm message.

Resend options – The program can send another email message for the same alarm.

To address – The additional email message will be sent to the specified address.

Resend after – If the alarm repeats, the next message about it will be sent only after the specified number of seconds.

> Please note that you can specify multiple email recipients by using the semicolon as a separator.

The alarm text will be used for the Subject of the email message; you can also use substitute variables.

The email message text has a fixed format and depends on the parameters specified on the "Common" tab.

CSV

Use this tab (figure 6) to specify the parameters for writing alarms to an CSV file. By the way, you can easily open such a file in MS Excel.

Figure 6: CSV logging settings

Log to CSV file – You can enable or disable logging to an CSV file.

CSV file name – The CSV file pathname.

Write column headers – If you turn this option on, the program will add a line containing column names at the beginning of the new file.

Column separator – You can specify an arbitrary column separator for the CSV file. Typically, the period (dot) or the comma is used. You can also use the #09 sequence, which stands for the TAB character.

Syslog

Logging active – You can enable or disable logging to a Syslog server.

Server - the IP address of the Syslog server.

Web service

Logging active – You can enable or disable logging to the selected web service.

Type - the list of all supported web services, where you can select one variant. If you would like to log data to multiple web services, you can create a new group.

Server - the full URL or IP address of the selected web service.

API key - a special API key is used to call the selected web service. You should obtain it from your web service provider.

Table - the destination resource name which is used in some services.
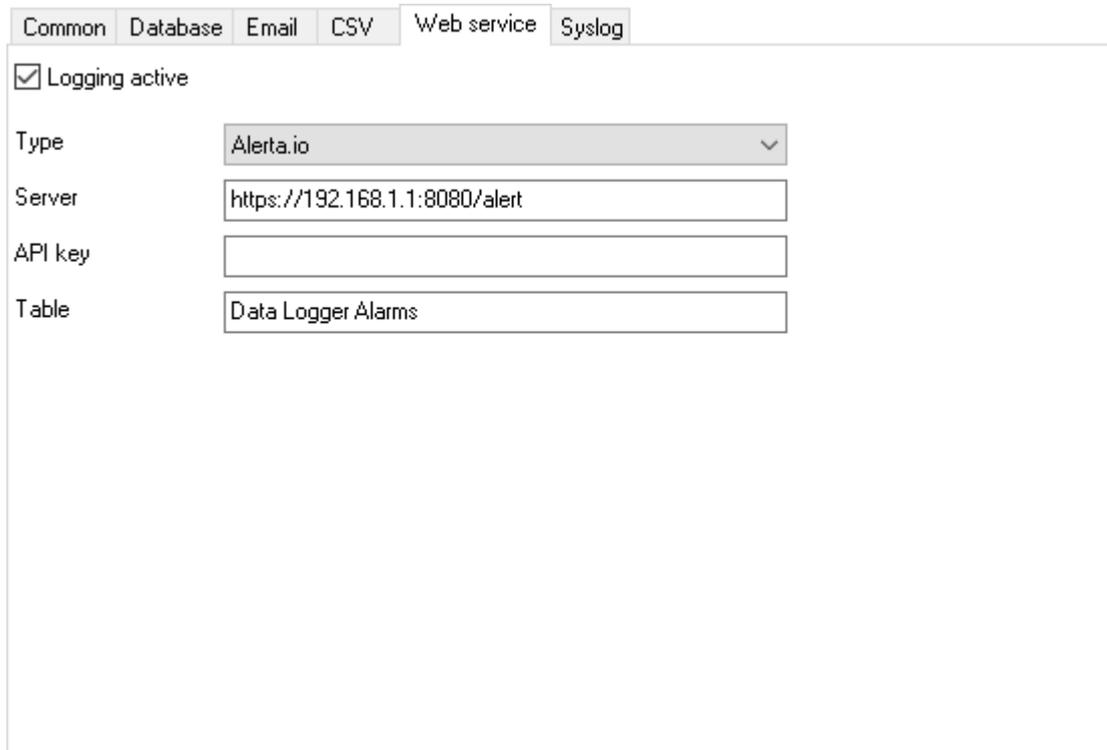
Figure 7: Web service logging settings

5.3   Alarms

On this tab, you can create one or more alarm handlers. You can configure multiple handlers for any event and specify different alarm handling modes for them by using different groups.

To create a new handler, click the "Add" button next to the drop-down list. The program will prompt you for the name of the handler.

To delete, rename, or copy a handler, click the "More" button and select the desired action from the drop-down menu.

Value

You can use the "Value" tab (figure 8) to select a parser variable or specify an expression to use its result to analyze and log alarms.

Figure 8: The value is a parser variable

Description – A textual description of the event handler.

Data type – The data type of the parser variables or the data type of the expression result. The type of data of variables or the result of an expression. Depending on the data type, different criteria for evaluating alarms can be applied. For example, for strings and boolean values, only "Digital" type alarms are available, where only strict value matching is allowed, and for numerical data, all types of alarms are available.

Data source – You can specify the data source type. If you have selected "Parser item" (parser variable), you can select a variable name from the drop-down list or type it in. If you have selected "Expression," you need to enter the expression into the field below.

Expression – If this option is enabled, the plugin can apply arithmetic or logical operations, formulas, or functions to incoming data and use the result (the result of the last expression in the list) in alarm rules. Moreover, you can use several parser variables from one data packet in expressions.

> Please note that the alarms will not be triggered if the expression contains any errors, or if the parser variable referenced by the expression doesn't exist or has an invalid value.
>
> Also, expressions can change the data type of values. For example, in an expression, you compare two numbers, but the data type of the result will be "Boolean".

Examples of expressions:

VALUE1 + 10 – Add 10 to the parser variable value VALUE1, and then use the result when checking the alarm triggering conditions.
VALUE2 - VALUE2 – The difference of two parser variable values.
VALUE2 > VALUE2 – The comparison of two parser variable values.

You can find the description of some additional mathematical or logical functions in the documentation for the Expressions module.

Event - an event can be generated by another module or by the program itself. You can specify the ID of this event in the "Expression" field. When the specified event occurs, the configured alarm handler will be executed. The best option for processing such events is the "Digital" alarm handler with the "true" value in the "Limit" field.

Example:

MY-EVENT

Note: if you specify the "MY-EVENT" identifier in the settings, but the program receives an event with the "MY-EVENT-CLOSE" identifier (the "-CLOSE" suffix is added), the program interprets this event as an alarm completion signal for the "MY-EVENT" event.

Additionally, you may check a context of the received event using expressions, and filter our some signals using the "IGNORE_EVENT_IF" function.

Control alarms for data sources separately - this option enables you to manage alarms for multiple data sources individually. For instance, if you are polling multiple devices within a single configuration, a parsed data packet will include the device address, which can be utilized to distinguish data for each device. In such cases, you define the variable name (e.g., DEVICE_ID), and the plugin will commence monitoring alarms for these devices individually. Furthermore, you can differentiate between devices using a complex value containing multiple variables by employing one or more placeholders such as: {DATA_SOURCE_NAME}-{DEVICE_ID}.

Alarm types

You can set a few alarm levels for each handler: Low Low, Low, High, High High, or Digital.

When checking the alarm level, the program will compare the initial value of the variable or the result of the expression with the specified limit.

The first four levels can be used for numerical values only and allow you to specify alarm level gradation. The >= or <= rule is used when making a comparison.

The Digital alarm level means that the value must exactly match the limit. You can also use this alarm type to handle textual or logical values. When comparing logical values, it is assumed that True is 1 and False is 0.

Please note what will happen if you have enabled all alarm levels and a few alarms of different levels are triggered at the same time: In that case, the program will generate an event for the highest-priority level only. The priority of alarm levels, from the most important to the least important one, is as follows: Digital, Low Low, High High, Low, High.

All alarm levels have the same parameters (figure 9).

Figure 9: Alarm settings

Enable – You can enable of disable the handler for this alarm level.

Group – The alarm handling group.

Limit – this is the limit value that depends on the data type of the source value. For numbers, the limit can be any numerical value. For strings, only the "Digital" alarm type can be used, and the text

that triggers the alarm is specified. For boolean values, 1 or "true" (without quotes) is specified for "True," and 0 or "false" is specified for "False."

Deadband - the plugin will not generate a new event until the newly received numerical value goes outside the "dead zone" relative to the previous value.

Priority – The priority number. You can use this parameter to further categorize events in the database.

Delay time – The event will be logged with the specified delay (in seconds). If the alarm ends within that time, it will not be logged. But if no other data is received during the delay time, the alarm will be logged anyway, so that later you can see whether it had ended.

Disable for date range – You can specify a date range during which the handler will be disabled.

Disable daily for time range – You can specify a daily time range during which the handler will be disabled.

Alarm text – The alarm text that you need to be logged. You can use substitute variables in the text. For example, it can be handy for saving the values of the parser variables used in the expression as text.

Logging max rate – You can specify the minimum allowed interval between two consecutive alarms of this type. If this parameter is set to zero, the alarms will not be filtered. This parameter works independently of the similar parameter for the group. If a larger rate is specified for the group, that rate will be effectively used when logging alarms.

5.4　Substitute variables

The special substitute variables allow you to insert parser variable values or other parameters' values into some textual values.

For example, you can specify alarm{PRIORITY} as a table name. {PRIORITY} is a substitute variable for the Priority parameter. Before writing data to the database, the program will replace {PRIORITY} with the priority value, so the table name will become, for example, "alarm1" in the end.

Each substitute variable is embraced in curly brackets and includes a parameter name or a parser variable name.

You can use the names of the following alarm parameters in substitute variables:

ALARM – The alarm name.
VALUE – The value of a parser variable (if it was used when checking the alarm status).
PREV_VALUE – The previous value of a parser variable.
TIMESTAMP – The event logging date and time stamp.
FIRST_SEEN – The date and time stamp of the first ocurrence of an alarm.
LAST_SEEN – The date and time stamp of the last ocurrence of an alarm.
SEEN_COUNT – The number of times an alarm was triggered, which depends on the number of packets received.
NOTIFY_COUNT – The number of times an alarm was logged, which depends on the filters configured.

PRIORITY – The priority.
TEXT – The textual description of the alarm for the type of the alarm triggered.
TYPE - The alarm type (HighHigh, High, LowLow, Low, Digital).
DESC – The textual description of the alarm handler.
GROUP – The group name.
ACTIVE – The current alert state (Y - active or N - inactive).

You can use parser variables as substitute variables in a similar manner.

Please note that a parser variable name can include only English letters (a–z), digits, and the underscore character. It cannot include the space character.

Boolean values are converted to 1 (True) or 0 (False). After the substitute variable name, you can specify an additional format identifier: {NAME:S}. In that case, the "True" or "False" value will be inserted.

For values of the "Date" type, you can specify the date and time format: {NAME:YYYY-MM-DD}. The result will look, for example, as follows: 2017-01-01. You can also use the time format: {NAME:HH:NN:SS}. If the date format is not specified, the system format for date or time will be used.