

**The Alarms Professional plug-in  
PRINTED MANUAL**

# Alarms Professional plug-in

© 2016-2017 AGG Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 8/7/2017

## **Publisher**

*AGG Software*

## **Production**

© 2016-2017 AGG Software

*<http://www.aggsoft.com>*

---

# Table of Contents

<b>Part 1 Introduction</b>	<b>1</b>
<b>Part 2 System requirements</b>	<b>1</b>
<b>Part 3 Installing Alarms Professional</b>	<b>1</b>
<b>Part 4 Glossary</b>	<b>2</b>
<b>Part 5 Configuring the module</b>	<b>3</b>
1 Common .....	3
2 Groups .....	4
3 Alarms .....	10
4 Substitute variables .....	13
<b>Part 6 Troubles?</b>	<b>14</b>
1 Possible problems .....	14

## 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 an 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. A 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.

### Notes for Microsoft Vista and above:

Since our software saves data to the registry and installs to the Program Files folder, the following requirements must be met:

1. You need Administrator rights to run and install our software
2. The shortcut icon of our software will be located on the desktop;
3. Windows Vista will ask for your confirmation to continue the installation.

NOTE: You can configure the user account only once in order not to see the above dialog box any more. Search Google for the solution of this problem.

## 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.

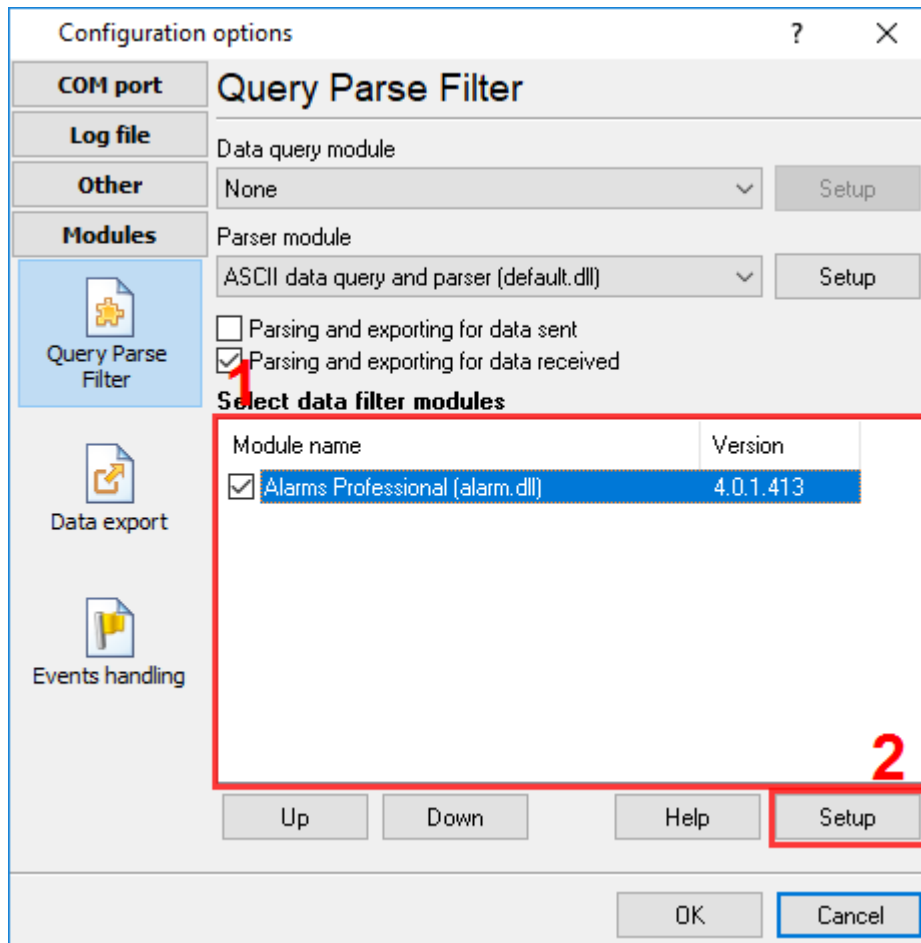


Fig.1. Example of installed module

## 4 Glossary

**Plug-in** - module

**Main program** – the program shell that uses this module. For example: Advanced Serial Data Logger

**Parser** – the module that processes the data flow singling out data packets from it and 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 is received from the parser. During the time between receiving two successive data packets, no alarms are analyzed or logged, and the module is "asleep."

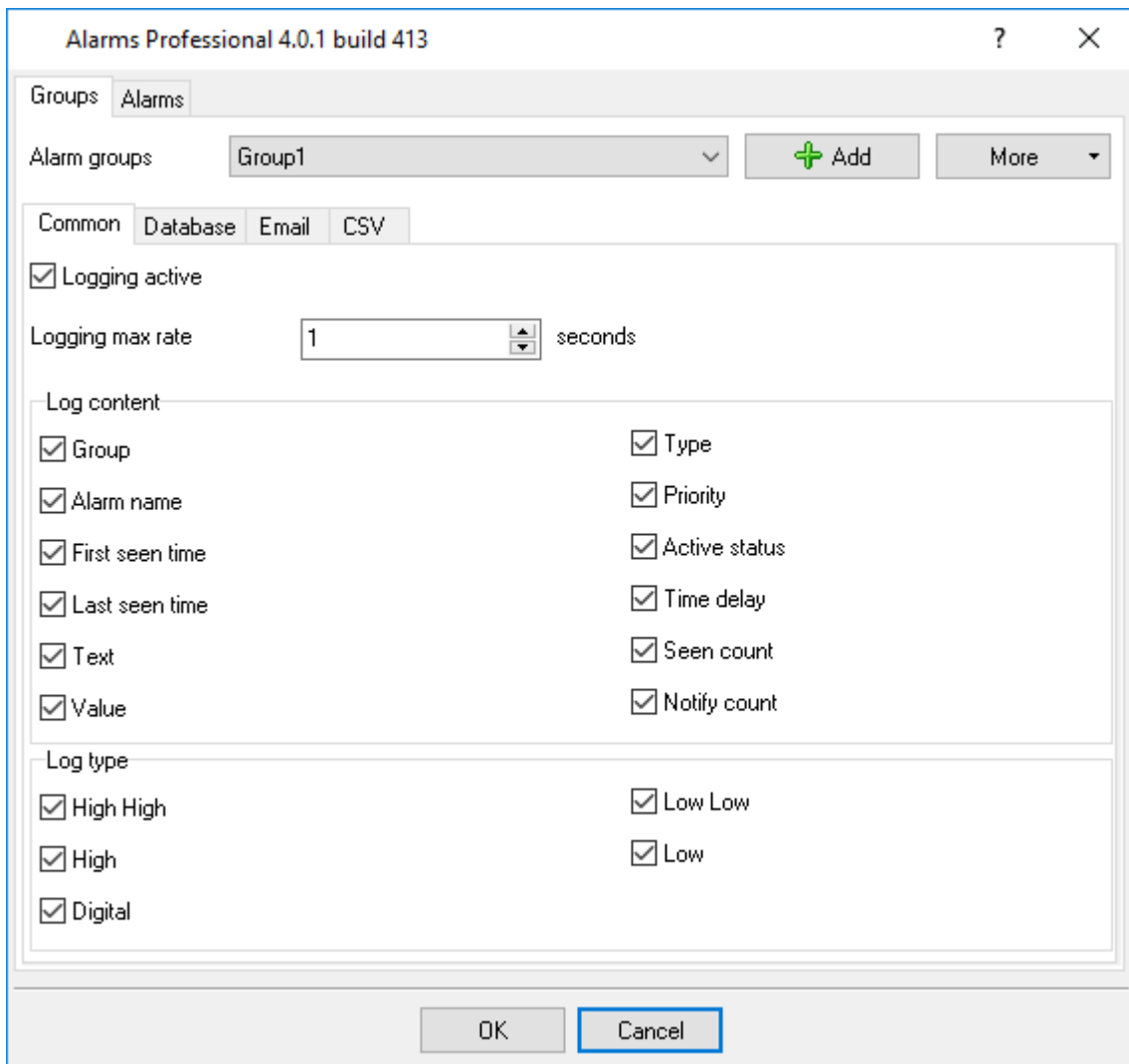


Figure 1: The module window

## 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 1) to configure the content to be logged and the data logging mode that is common for all logging methods.

Figure 1: 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 occurrence of an alarm.

**Last seen time** – The date and time of the last occurrence 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 appropriate 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 2) 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).

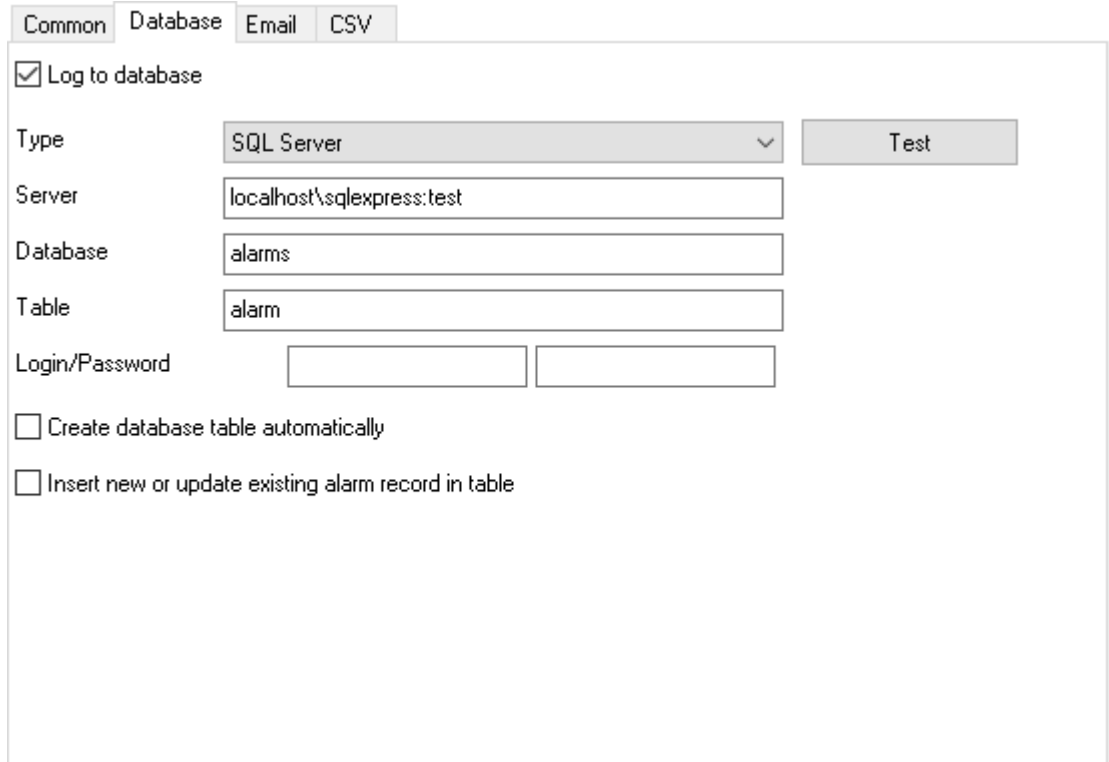
**Database** – The name of the database on the server.

**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.”



The screenshot shows a dialog box with four tabs: 'Common', 'Database', 'Email', and 'CSV'. The 'Database' tab is selected. It contains the following fields and options:

- Log to database
- Type: SQL Server (dropdown menu)
- Server: localhost\sqlexpress:test
- Database: alarms
- Table: alarm
- Login/Password: (two empty text boxes)
- Create database table automatically
- Insert new or update existing alarm record in table
- Test (button)

Figure 2: 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 3) to configure the email sending settings.

The screenshot shows a configuration window with four tabs: 'Common', 'Database', 'Email', and 'CSV'. The 'Email' tab is active. At the top, there is a checkbox labeled 'Send email' which is checked. Below this is a section titled 'Email settings' containing several fields: 'SMTP host' with the value 'smtp.gmail.com:465' and a 'Test' button to its right; a checked checkbox 'Server requires authentication' with a dropdown menu set to 'TLS'; 'Login/Password' with the value 'aggsoft@gmail.com' and a masked password field '\*\*\*\*\*'; and 'From name/address' with the value 'AGG Software' and 'aggsoft@gmail.com'. Below 'Email settings' is a section titled 'Send options' with a 'To address' field containing 'my@email.com'. At the bottom is a section titled 'Resend options' with a 'To address' field containing 'my@email.com' and a 'Resend after' field with a spinner set to '60' and the unit 'seconds'.

Figure 3: 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 4) to specify the parameters for writing alarms to an CSV file. By the way, you can easily open such a file in MS Excel.

The screenshot shows a dialog box with four tabs: 'Common', 'Database', 'Email', and 'CSV'. The 'CSV' tab is active. It contains the following settings:

- Log to CSV file
- CSV file name: c:\Logs\temp.csv
- Write column headers
- Column separator: ;

Figure 4: 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.

## 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 1) to select a parser variable or specify an expression to use its result to analyze and log alarms.

The screenshot shows a configuration window for an alarm handler. At the top, there are five tabs: 'Value', 'High High', 'High', 'Low', 'Low Low', and 'Digital'. The 'Value' tab is selected. Below the tabs, there are three main sections: 'Data type', 'Data source', and 'Description'. The 'Data type' section has a dropdown menu with 'Integer' selected. The 'Data source' section has a dropdown menu with 'Parser item' selected, and a text input field containing 'VALUE1'. The 'Description' section has a text input field containing 'Alarm description'.

Figure 1: The value is a parser variable

**Data type** – The data type of the parser variables or the data type of the result of the expression.

**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** – The expression, which can contain arithmetic or logical operations, formulas, or functions (figure 2). You can use parser variables in the expression.



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.

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](#).

**Description** – A textual description of the event handler.

Value	High High	High	Low	Low Low	Digital
Data type	Integer				
Data source	Expression		VALUE1		
	[VALUE1 - VALUE2] > 10				
Description	Alarm! {VALUE1} is higher {VALUE2}				

Figure 2: The value is an expression

## 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  $\geq$  or  $\leq$  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 3).

The screenshot shows the 'Alarm settings' configuration window for the 'Digital' alarm level. The window has several tabs: 'Value', 'High High', 'High', 'Low', 'Low Low', and 'Digital'. The 'Digital' tab is currently selected. The settings are as follows:

- Enable:** A checked checkbox.
- Alarm group:** A dropdown menu showing 'Group1'.
- Alarm priority:** A numeric spinner set to '0'.
- Limit:** A numeric input field containing '90'.
- Deadband:** A numeric spinner set to '0.00'.
- Time delay:** A numeric spinner set to '0'.
- Disable for date range:** An unchecked checkbox with two date pickers set to '..' and a time picker set to '0:00:00'.
- Disable daily for time range:** An unchecked checkbox with two time pickers set to '0:00:00'.
- Alarm text:** A text input field containing 'Alarm text'.
- Logging max rate:** A numeric spinner set to '0' followed by the text 'seconds'.

Figure 3: Alarm settings

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

**Group** – The alarm handling [group](#).

**Limit** – The limit.

**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 occurrence of an alarm.

**LAST\_SEEN** – The date and time stamp of the last occurrence 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.

**DESC** – The textual description of the alarm handler.

**GROUP** – The group name.

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.

## 6 Troubles?

### 6.1 Possible problems

**No data for publication/exporting** – no data is passed for exporting. Solution: configure the parser, make sure that one or more variables are declared in the parser.

**Error on binding variable with name %s [%s]** – the error usually occurs if data does not correspond to the specified format. For example, the date and time format does not correspond to the data.

**Unable to disconnect from the database [%s]** and **Unable to connect to a database [%s]** – it is impossible to connect/disconnect to/from the database. You should check the parameters of the database connection. The analysis of the additional information will help you locate the error.

**Database access error [%s]. Stop operations with the database?** – the message appears if an error occurs during an attempt to execute an SQL query if the second variant of reacting to errors is selected. The message implies a "Yes" or "No" answer. The analysis of the additional information will help you locate the error.

**Unable to verify your SQL script [%s]** – the message appears when an attempt to analyze your SQL query fails. Check if the syntax of your SQL query is correct.

**Tested successfully** – the message appears if your database connection is successfully tested. It requires no additional actions.

**Database isn't used** – the message appears if the module is temporarily disabled (the "Temporarily disabled" check box is selected) or the database name field is empty. Check the connection parameters.

**Database isn't selected** - the message appears if the database type is not selected. Check the connection parameters.

**Database: %s** – %s contains the database name. The message appears if the database connection is successful. Usually, you see it when you call the module for the first time. It requires no additional actions.

**Invalid data block length (columns=%d,length=%d)** – an internal application error. It means that the data sent by the parser is in an invalid format. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

**The time of connection is not due yet (%d,%d)** – the message appears during an attempt to connect to the database after the connection to it has been lost and the "Reconnect after" option is enabled. No additional actions are required.

**Invalid procedure call. Bad arguments** –an attempt to call the module using invalid parameters. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

**Writing to the database is complete** - the message appears if your queue of SQL queries is successfully executed. It requires no additional actions.

**Writing to the database is complete with errors** – the message appears if the executing your queue of SQL queries was interrupted by an error. It requires no additional actions.

**Your SQL is empty. Please, specify some SQL text first** – the message appears if you do not enter the text for your SQL query. Check if the options on the "SQL queue" tab are configured correctly.

**Invalid temporary path** – the path to the temporary file specified by you does not exist. Enter a new path in the "Temporary folder" field on the "Errors handling" tab.

%s, %d – will be replaced by additional information.