



**The File requests plugin
PRINTED MANUAL**

File requests plugin

© 2004-2024 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: 11/2/2024

Publisher

AGG Software

Production

© 2004-2024 AGG Software

<http://www.aggsoft.com>

Table of Contents

Part 1 Description	1
Part 2 System requirements	1
Part 3 Installing File requests	1
Part 4 Glossary	2
Part 5 User Manual	3
1 How the module works	3
2 Parameters	5
3 Cron time format	9
Part 6 Troubles?	11
1 Possible problems	11

1 Description

This module allows you to send data via the interface being used after the data has been read from the specified file. Unlike other data query modules (ASCII Data Query in particular), this module can send queries of a large size and queries containing any characters. Besides, you can control the queries being sent remotely without changing the configuration of the program. The module can monitor if there are any files in the specified folder and send their contents. You can use the built-in scheduler to schedule sending data from the file accurate within one minute.

2 System requirements

The following requirements must be met for "File requests" 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 File requests

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.

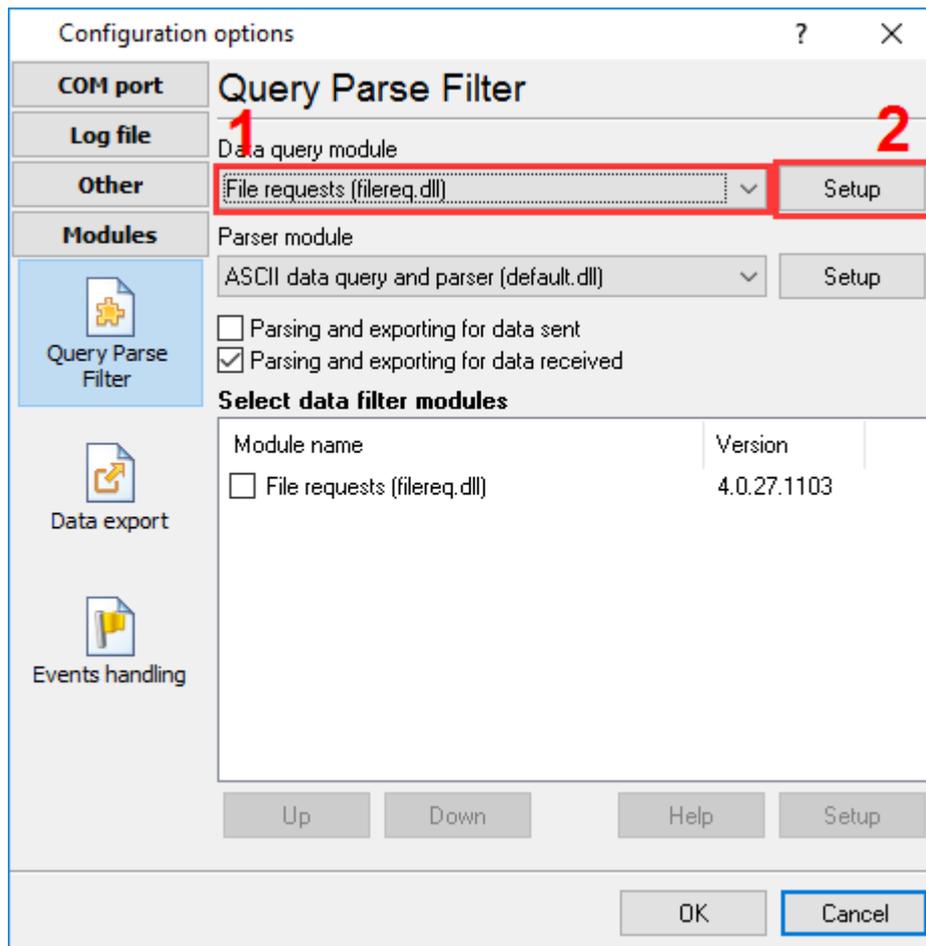


Fig. 1. Example of installed module

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 User Manual

5.1 How the module works

The module is loaded right when the program starts. After the module is loaded, it checks what mode is active (enabled). If all modes are disabled, the module stops its operation till the module settings are changed. If at least one mode is enabled, the module starts its operation. In case of the first two modes, the module checks the existence of files meeting the conditions set by the timer whose triggering interval is specified in the "[Scan interval](#)" parameter. In case of the scheduler mode, the module waits till the task is triggered.

ATTENTION: Make sure you read the warning about the "[Folder](#)" parameter.

Simple scan mode

In this mode, all files in the folder specified in the "[Folder](#)" parameter are checked one after another. The module compares every file name with the value specified in the "[File](#)" parameter. If the file name coincides the value (taking the mask into account or not), this file will be processed (see below).

Time scan mode

All files in the folder specified in the "[Folder](#)" parameter are checked one after another. The module compares every file name with the value specified in the "File name prefix" and "File extension" parameters. If the prefix and file extension coincide with the value, the numbers coming after the prefix in the file name are considered the date and time when the data from this file will be sent. For example, if the file name is data200901011200.bin, the file will be sent on January 1, 2009 at 12:00. ????????

Task scheduler mode

In this mode, the module keeps comparing the current date and time with rules specified in the cron format all the time. If any of the rules is true, all files in the folder specified in the "[Folder](#)" parameter are checked one after another. The module compares every file name with the value specified in the "[File](#)" parameter. If the file name coincides the value (taking the mask into account or not), this file will be processed (see below).

Processing the file

1. The module checks the chunk size specified in the "[Chunk size](#)" parameter. If the chunk size equals 0, it writes the corresponding message in the log file and does not process the file;
 2. The module checks if the file exists. If the file does not exist, it writes the corresponding message in the log file and does not process the file;
 3. The module checks the file size. If the file size equals 0, it writes the corresponding message in the log file and does not process the file;
 4. The module reads the data chunk of the size specified in the parameters:
- It opens file in the read-only mode. If the file won't open, it writes the corresponding message in the log file and does not process the file;

-
- It checks the read position of the file for the "Read and don't change" mode. If the file is completely read and the current read position is smaller than the one obtained earlier (the file got smaller), the module writes the corresponding message in the log file, closes the file and stops processing it. Otherwise it sets the saved position in order to read from this position later on;
 - It reads data from the file.
 - If the "Decode" checkbox is selected and the file being processed is a text one, the data is decoded;
 - The data is sent via the interface that is being used;
5. The module closes the processed file;
 6. The module processes the data read mode:
 - If the mode is "Read and delete", the file is deleted. If the file is not deleted, the module writes the corresponding message in the log file and stops processing the file;
 - If the mode is "Read and truncate", the read data is removed from the file. If the data is not removed, the module writes the corresponding message in the log file and stops processing the file;
 - If the mode is "Read and don't change", nothing happens.

5.2 Parameters

Common parameters

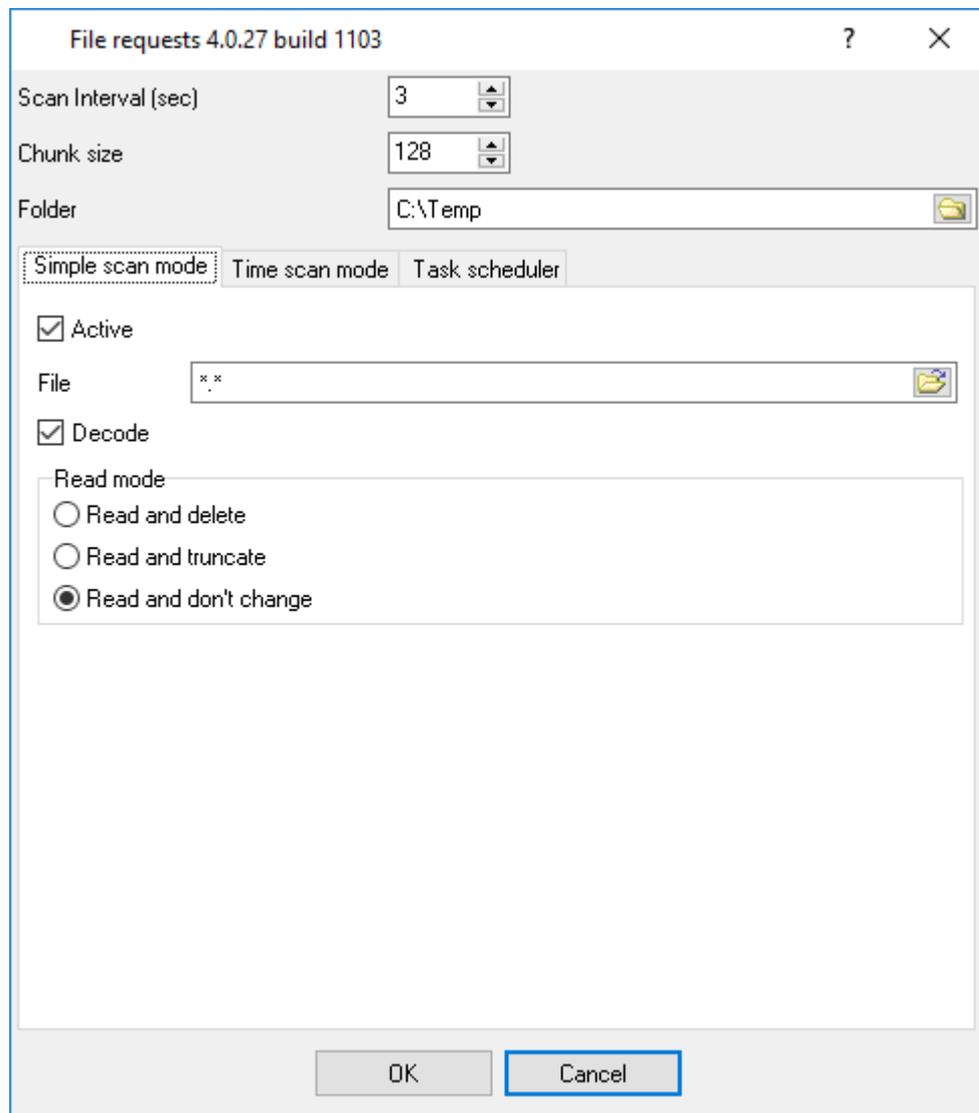


Fig. 1. Common parameters

Scan interval (sec) – the time interval at which the specified folder is scanned for new or modified data. The default is 3 seconds.

Chunk size – the number of bytes read from the file at a time. The default is 128 bytes.

Folder – the folder scanned for files. The default is C:\.

ATTENTION: If the selected folder contains critical files (for example, the root directory of the C:\ disk may contain system files the operating system cannot work or start without) and you set the read mode to "Read and delete" or "Read and truncate" (see below) in any scan mode, the file will be permanently deleted after it is processed. It may lead to failures both

in the operating system and in other software. It is recommended to change the folder right after you open the parameters dialog box for the first time.

Simple scan mode parameters

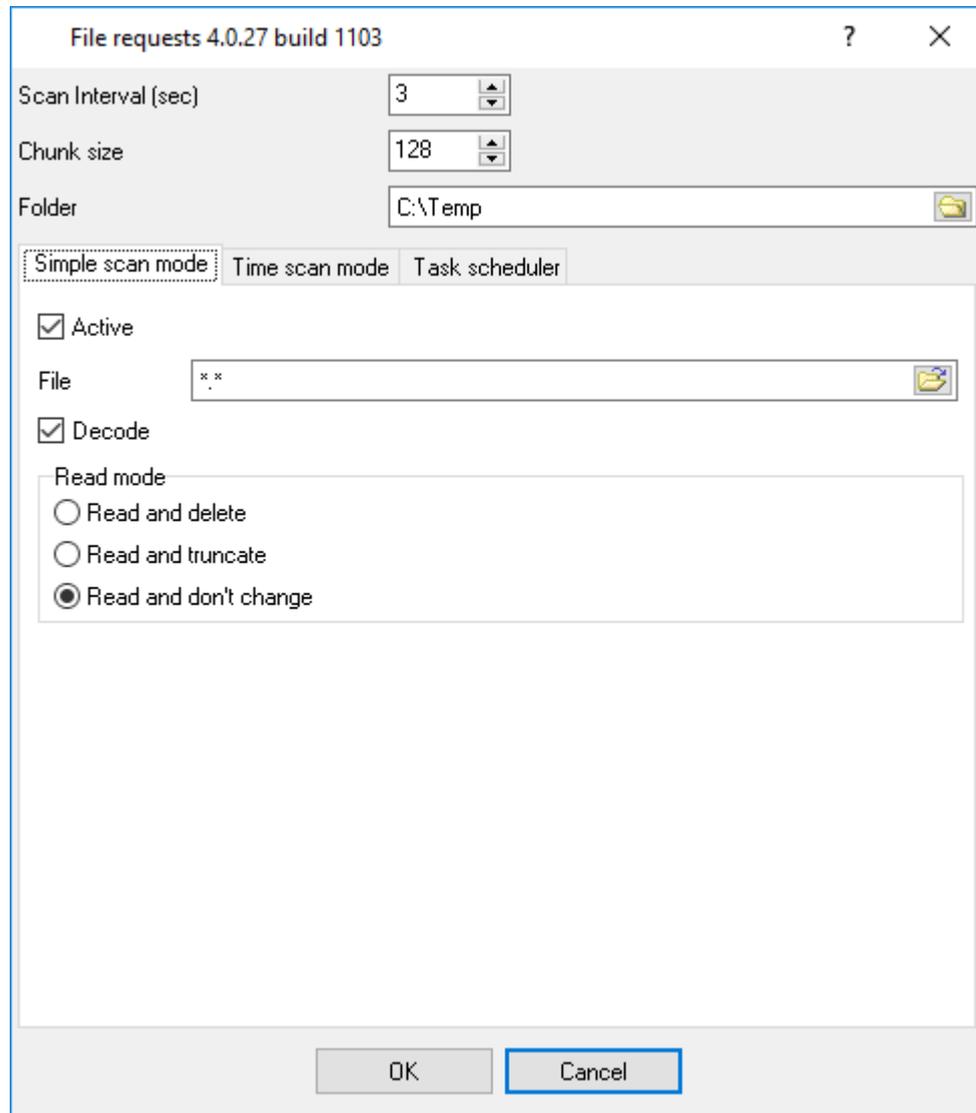


Fig. 2. Simple scan mode parameters

Enabled - enable/disable the mode. It is disabled by default.

File – the mask of files that will be processed. You can use the "*" and "?" characters to select several files. Example: *.* - process all files in the folder specified in the "[Folder](#)" parameter. The default is *.*.

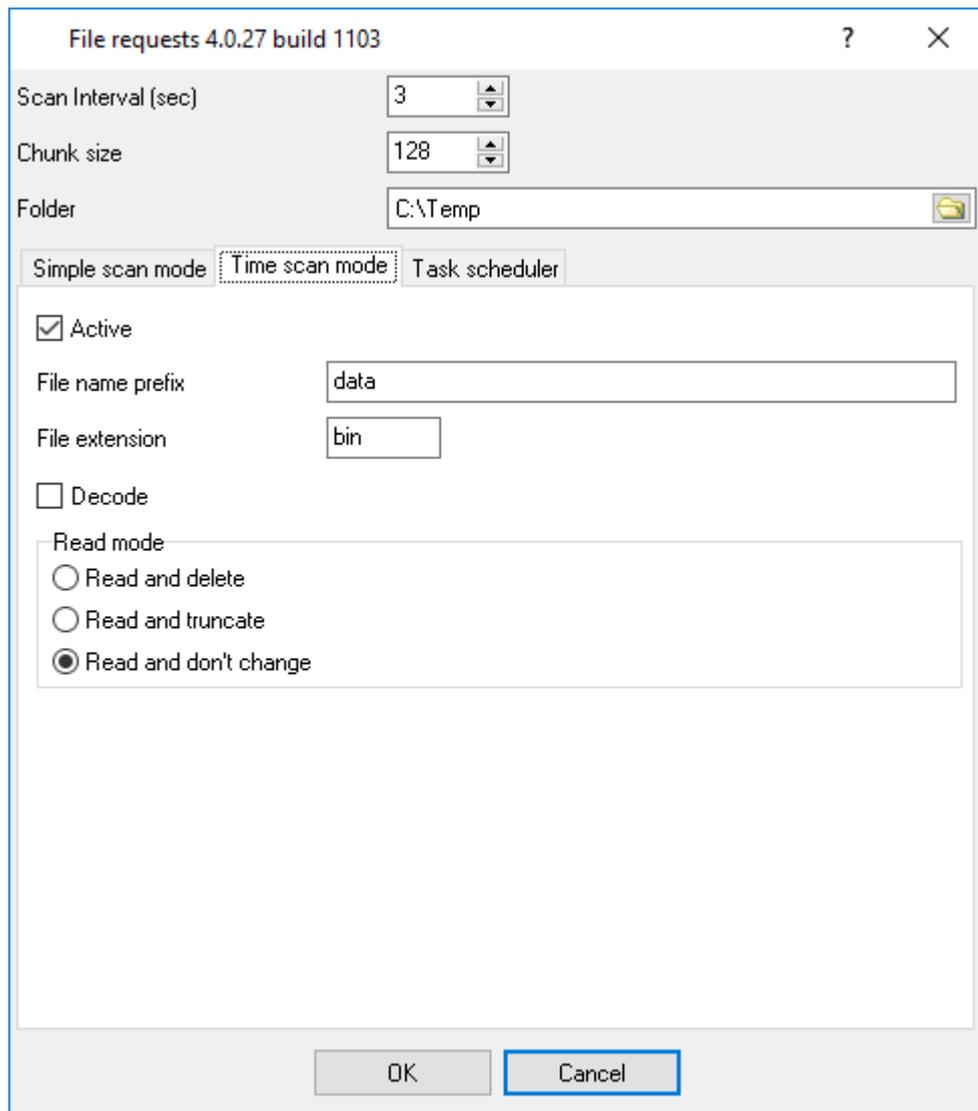
Decode – if it is enabled, characters like #XX (where XX is a hexadecimal character code) are converted into one character. It works only for text files. It is enabled by default.

Read mode – select one of the three read modes:

1. **Read and delete** – data is read from the file that is deleted afterwards;
2. **Read and truncate** – data is read from the file and the read data is removed from it afterwards;
3. **Read and don't change** – data is read from the file and the file is not modified in any way after that. In this mode, the current position is saved after the data chunk is read and data will be read from the last position the next time. If the file size changes, the error message is displayed and the data is read from the beginning of the file.

Note: data is read in chunks whose size you can specify in the [common parameters](#).

Time scan mode parameters



The screenshot shows a dialog box titled "File requests 4.0.27 build 1103". It has three tabs: "Simple scan mode", "Time scan mode" (which is selected), and "Task scheduler". The "Time scan mode" tab contains the following settings:

- Scan Interval (sec): 3
- Chunk size: 128
- Folder: C:\Temp
- Active
- File name prefix: data
- File extension: bin
- Decode
- Read mode:
 - Read and delete
 - Read and truncate
 - Read and don't change

At the bottom of the dialog are "OK" and "Cancel" buttons.

Fig. 3. Time scan mode parameters

Enabled – analogous to the simple scan mode. It is disabled by default.

File name prefix – the characters at the beginning of the file name. If the module finds a file that has these characters at the beginning of its file name, it processes the file. Otherwise the file is

skipped. Example: Test – all files with names stating with Test are processed. By default the string is empty.

File extension – the filter for types of files to process. You can use the wildcard characters, like in the [file name](#) in the simple mode. Example: * - process all files, DLL – process only dynamic link library, EXE – process only applications. The default is *.

Decode and **Read mode** – these parameters are described above.

Task scheduler mode

File requests 4.0.27 build 1103

Scan Interval (sec) 3

Chunk size 128

Folder C:\Temp

Simple scan mode Time scan mode **Task scheduler**

Active

+ - X

Time	Mode
<input type="checkbox"/> * * * * 15	Read and don't change

Time ("Cron" format)

WD * * * * 15 MIN 15

Decode

File c:\Logs\data.bin

Read mode

Read and delete Read and don't change

Read and truncate Read from beginning

OK Cancel

Fig. 4. Task scheduler mode parameters

Enabled - enable/disable the scheduler. It is disabled by default.

Rules – the time and parameters of the task:

1. **Time** – the time when data from the file is read specified in the cron format. You can find the detailed information about this format and see examples in the "Cron time format" section. The default is `* * * * *`, which means "every minute of every hour of every day of every month of every week";
2. The rest of the parameters are similar to those of the simple mode.

List operations:

- Add – add new time to the list;
- Remove – remove the current time from the list. Unavailable if the list is empty;
- Remove All – remove all times from the list; Unavailable if the list is empty.

5.3 Cron time format

The CRON format is a simple yet powerful way to describe time and operation periodicity. The traditional (inherited from the Unix world) CRON format consists of five fields separated with spaces:

<Second> <Minutes> <Hours> <Month days> <Months> <Weekdays>

Any of the five fields can contain the `*` (asterisk) character as its value. It stands for the entire range of possible values. For example, every minute, every hour and so on. In the first four fields, you can also use the proprietary `"?"` (w/o quotes) character. See its description below.

Any field can contain a list of comma-separated values (for example, `1,3,7`) or an interval (subrange) of values defined by a hyphen (for example, `1-5`).

You can use the `/` character after the asterisk (`*`) or after an interval to specify the value increment. For example, you can use `0-23/2` in the "Hours" field to specify that the operation should be carried out every two hours (old version analog: `0,2,4,6,8,10,12,14,16,18,20,22`). The value `*/4` in the "Minutes" field means that the operations must be carried out every four minutes. `1-30/3` is the same as `1,4,7,10,13,16,19,22,25,28`.

You can use three-word abbreviations in the "Months" (Jan, Feb, ..., Dec) and "Weekdays" (Mon, Tue, ..., Sun) fields instead of numbers.

Examples

Note: the <Second> field equal 0 in all examples

Format	Description
<code>* * * * *</code>	every minute
<code>59 23 31 12 5</code>	one minute before the end of the year if the last day in the year is Friday
<code>59 23 31 Dec Fri</code>	one minute before the end of the year if the last day in the year is Friday (one more variant)
<code>45 17 7 6 *</code>	every year on the 7th of June at 17:45
<code>0,15,30,45 0,6,12,18 1,15,31 * 1-5 *</code>	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays

* /15 * /6 1,15,31 * 1-5	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays (one more variant)
0 12 * * 1-5 (0 12 * * Mon-Fri)	at noon on workdays
* * * 1,3,5,7,9,11 *	every minute in January, March, May, July, September, and November
1,2,3,5,20-25,30-35,59 23 31 12 *	on the last day of the year at 23:01, 23:02, 23:03, 23:05, 23:20, 23:21, 23:22, 23:23, 23:24, 23:25, 23:30, 23:31, 23:32, 23:33, 23:34, 23:35, 23:59
0 9 1-7 * 1	on the first Monday of every month at 9 in the morning
0 0 1 * *	at midnight on the 1st of every month
* 0-11 * *	every minute till noon
* * * 1,2,3 *	every minute in January, February, and March
* * * Jan, Feb, Mar *	every minute in January, February, and March
0 0 * * *	every day at midnight
0 0 * * 3	every Wednesday at midnight

You can use the proprietary "?" character in the first four fields of the CRON format. It stands for the start time, i.e., the question mark will be replaced with the start time during the field processing: minute for the minute field, hour for the "Hours" field, month day for the month day field, and month for the month field.

For example, if you specify:

? ? * * *

The task will be run at the moment of startup and will continue being run simultaneously (if the user does not restart the program again, of course) – the question marks are replaced with the time the program was started at. For example, if you start the program at 8:25, the questions marks will be replaced like this:

25 8 * * * *

Here are some more examples:

- ? ? ? ? * - run only at startup;
- ? * * * * - run at startup (for example, at 10:15) and continue being run in exactly one hour: at 11:15, 12:15, 13:15 and so on;
- * ? * * * - run every minute during the startup hour;
- */5 ? * * * - run on the next day (if CRON is not restarted) at the same hour every minute and so on every day, once in five minutes, during the startup hour.

6 Troubles?

6.1 Possible problems

No data for publication/exporting - no data is passed for export. Solution: configure the parser and make sure that one or more variables are defined 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 match source data.

Unable to disconnect from the database [%s] and **Unable to connect to a database [%s]** - it is impossible to connect or disconnect 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 the "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 execution of 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.