

**The Scheduler & Hotkeys plugin  
PRINTED MANUAL**

# Scheduler & Hotkeys plugin

© 1999-2024 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**

© 1999-2024 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 Scheduler &amp; Hotkeys</b>	<b>1</b>
<b>Part 4 Glossary</b>	<b>2</b>
<b>Part 5 Configuring the module</b>	<b>3</b>
<b>Part 6 Cron time format</b>	<b>6</b>
<b>Part 7 Troubles?</b>	<b>8</b>
1 Possible problems .....	8

## 1 Introduction

The "Scheduler & Hotkey" module for our loggers (for instance, Advanced Serial Data Logger or Advanced TCP/IP Data Logger) is designed to perform tasks according to the predefined schedule or when the hotkey is pressed. For example, it allows you to collect data only at certain hours or when the operator needs it. The list of supported tasks includes opening and closing data sources (e.g. COM or TCP ports), sending data, generating internal program events, or creating a new log file.

## 2 System requirements

The following requirements must be met for "Scheduler & Hotkeys" 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 Scheduler & Hotkeys

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.

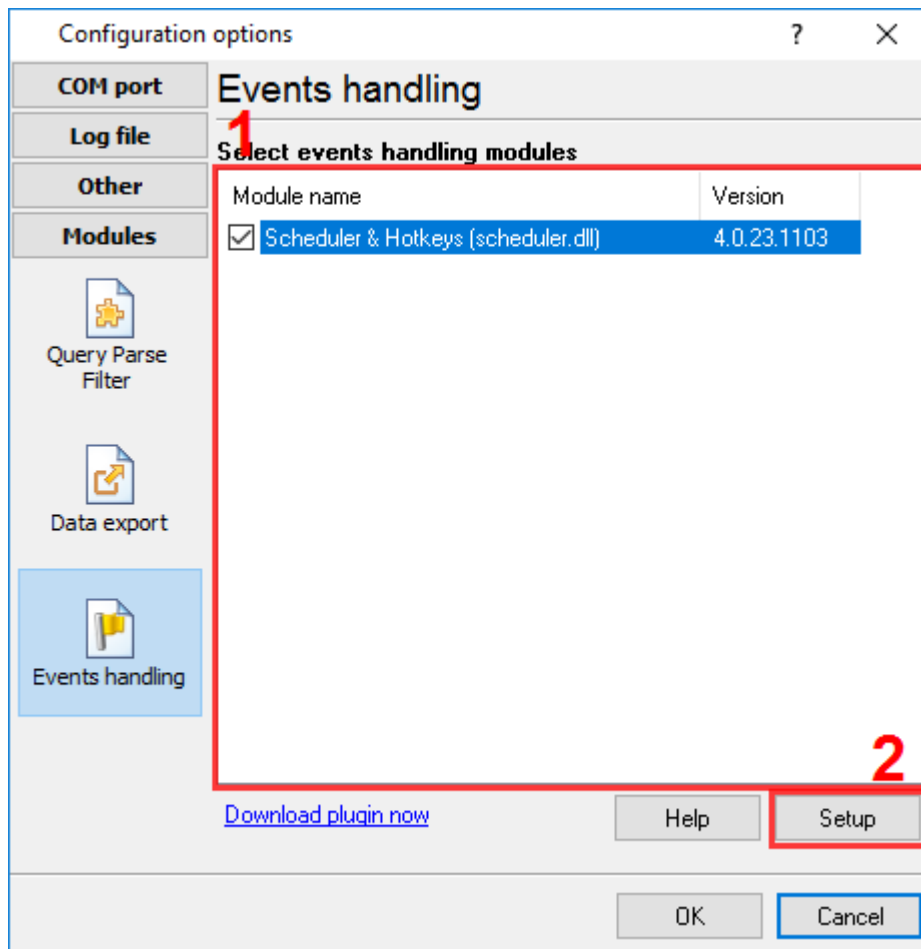


Fig. 1. Example of installed module

## 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

It is very easy to configure the module (fig. 1). You should define one or more tasks and specify when they should be performed. When the module is activated, the program will monitor the list of tasks and perform them according to the schedule.

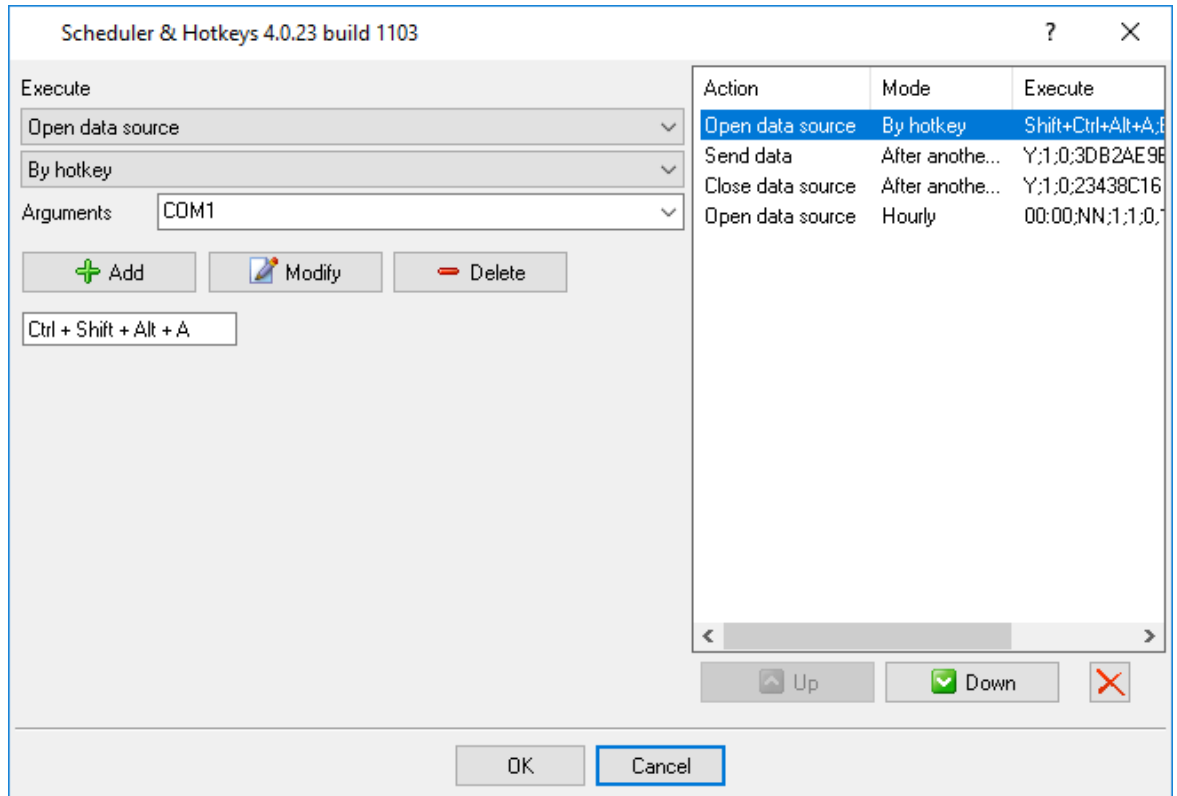


Fig.1. Module configuration dialog box

The left part of the module configuration dialog box contains the list of tasks that will be performed. Tasks in this list have top-down priority. If there are tasks due at the same time in the list, the upper task will be performed first. The "Up" and "Down" button allow you to change the order the tasks are performed in.

You can specify the parameters of each task, add, edit or remove it from the list in the right part of the dialog box.

### Adding a task

You should do the following:

1. Select the type of the task that will be performed;
2. Specify when the task will be performed;
3. Specify its arguments (if necessary for the selected task);
4. Specify the parameters of the task (date and time or hotkey);
5. Click the Add button.

## Editing a task

1. Select the task from the list on the left;
2. Change the parameters of the task;
3. Click the Apply button.

## Removing a task

1. Select the task from the list on the left;
2. Click the Remove button.

## Removing all tasks from the list

It is enough to click the button with the red cross next to the "Up" and "Down" buttons.

Note: All changes in the module configuration dialog box will be saved only after you click the OK button.

## Task types

**Open/close the data source** - the data source will be either opened or closed when this task is performed. It can be a COM port or a TCP port depending on the type of the logger. After the data source is opened, the logger starts receiving and processing data. When the data source is closed, this process is paused. Tasks of this type allow you to receive and process data only when it is needed. You should specify the name of the data source as an argument for this task, e.g. "COM1".

**Send data** - when this task is performed, data to all data sources defined in this configuration will be sent. The data is specified as a text string. If you need to send hexadecimal data, you should specify it in the #FF format. For instance, if you specify the string as "#02Data#03", the following 6 bytes will be sent: 0x02h, D, a, t, a, 0x03h. The data source must be opened before the data is sent.

**New log file** - a new log file will be created when this task is performed. At the same time, the program increases the internal counter that is later added to the name of the file. If a new log file is created according to the schedule specified in the log file settings of the program, the internal counter is reset.

**New event** - when this task is performed, a new internal program event will be generated. The name of the event is specified as the task argument. This event can be later processed by one of the event processing modules.

## Task mode

**Hourly** - you should use special buttons to select hours when the task will be performed. The task will be performed daily at the specified hours. You can specify minutes in the Time field (fig.2). Note: hours specified in the Time field are ignored.

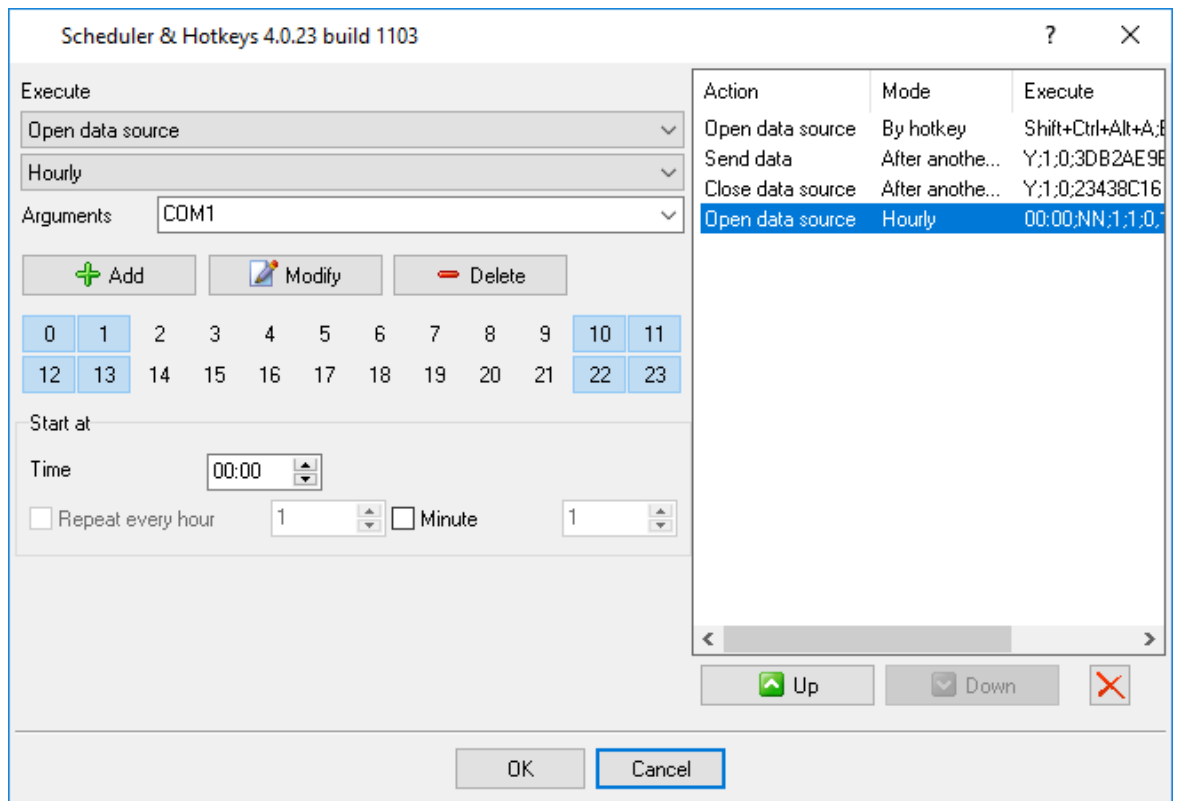


Fig.2. Task mode

**Daily** - the task will be performed every day at the specified time.

**Weekly** - the task will be performed on the specified days of the week at the specified time.

**By days** - the task will be performed on the specified days of the week but you should also specify the number of the day of the week in the month. For example, if you select "first Tuesday", the task will be performed only on every first Tuesday of every month.

**Monthly** - the task will be performed on the specified days of the month at the specified time.

**Hotkey** - the task will be performed every time you press the specified key combination. To specify the necessary key combination, you should activate the special field and press the necessary key combination. This key combination must not be used in other programs or in the operating system. This hotkey will be active independently of the main window state and will not be active when the program is running as a Windows service.

**After another task** - the task will be performed after some other task from the list will be performed. While specifying this task, you should select another task from the drop-down list. Using this mode, you can create sequences of tasks that will be performed one after another. In this mode, you can specify the time interval that should pass after the parent task is performed. If the parent task will be performed several times within this time interval, the child task will be performed only once after the parent task is performed for the last time.

## Repeating a task

You can specify how tasks will be repeated for tasks performed at a certain time.



**Every hour** - the task will be performed every number of hours specified in the corresponding field. If the number is 1, the task will be performed every hour, if it is 2, it will be performed every two hours, if it is 3, it will be performed every three hours, etc.

**Every minute** - the task will be performed every number of minutes specified in the corresponding field. If the number is 1, the task will be performed every minute, if it is 2, it will be performed every two minutes, if it is 3, it will be performed every three minutes, etc.

The program starts counting hours and minutes from the time of the task start. For instance, if the time is specified as 11:00, the task will not be performed before this time.

#### Example 1:

Mode: Daily  
Time: 18:00  
Repeat every hour: Enabled. Value: 2  
Repeat every minute: Enabled. Value: 10

All in all, the task will be performed every day at 18:00, 18:10, 18:20, 18:30, 18:40, 18:50, 20:00, 20:10, 20:20, 20:30, 20:40, 20:50, 22:00, 22:10, 2:20, 22:30, 22:40, 22:50.

#### Example 2:

Mode: Hourly. Selected hours: 12-14  
Time: 00:10  
Repeat every hour: unavailable (disabled)  
Repeat every minute: Enabled. Value: 30

All in all, the task will be performed every day at 12:10, 12:40, 13:10, 13:40, 14:10, 14:40.

## 6 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
* * * * *	every minute
59 23 31 12 5	one minute before the end of the year if the last day in the year is Friday
59 23 31 Dec Fri	one minute before the end of the year if the last day in the year is Friday (one more variant)
45 17 7 6 *	every year on the 7th of June at 17:45
0,15,30,45 0,6,12,18 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
*/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.

## 7 Troubles?

### 7.1 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.