



**The MQTT plugin
PRINTED MANUAL**

MQTT 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

Part 1 Introduction	1
Part 2 System requirements	1
Part 3 Installing MQTT	1
Part 4 Glossary	2
Part 5 Connection settings	3
1 Receiving data	3
2 Data publishing	7

1 Introduction

You can use the "MQTT" module with our loggers to receive and publish data using the MQTT protocol. MQTT (Message Queue Telemetry Transport) is a simplified network protocol that runs over TCP/IP. It is used to exchange messages between devices based on the publish-subscribe pattern.

Our logger acts as a client, so it requires a broker server running on your local network or the Internet (for example, a cross-platform Mosquitto server). You can subscribe to topics and receive data from them using the "MQTT" module only if you use it together with our [Data Logger Suite](#). Usually, data are sent in the text format, so the logger can convert text values to another data type to simplify their further handling. For example, you can log the data received to a file or a database.

You can publish data using any logger that supports data export. The program can also convert data types. If you use the "MQTT" module together with our Data Logger Suite, you can automatically group data from different sources and simultaneously publish them at another location.

2 System requirements

The following requirements must be met for "MQTT" 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 MQTT

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.

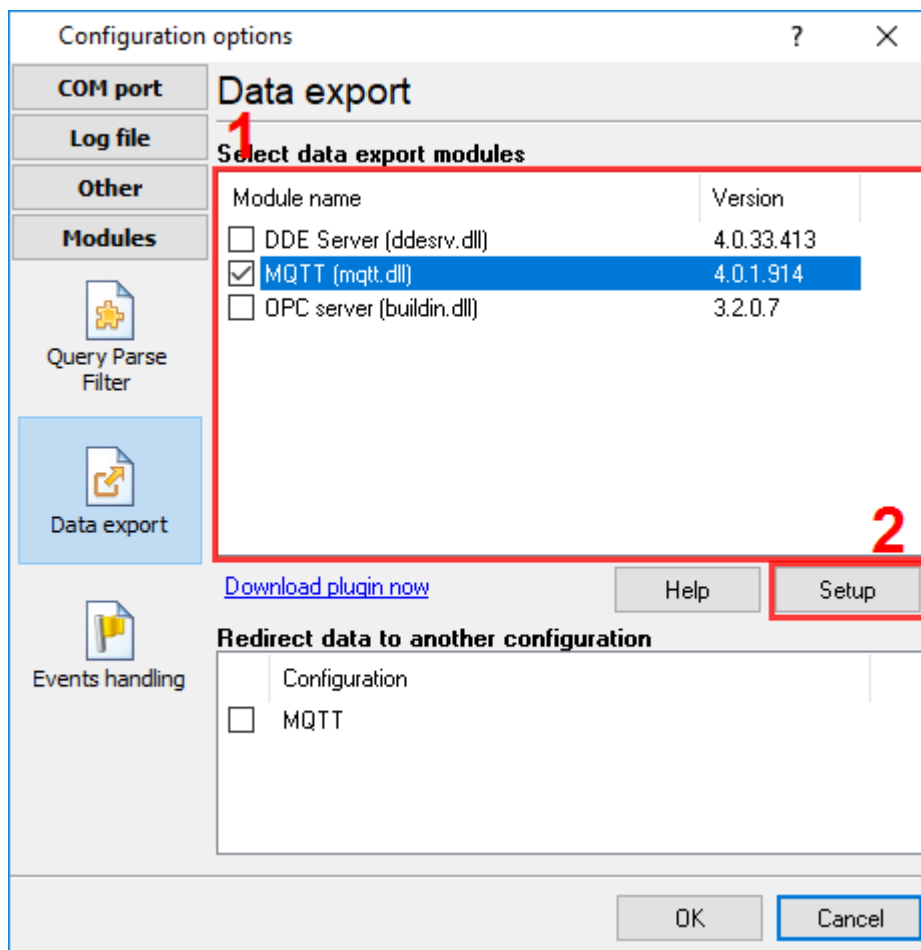


Fig. 1. Example of installed module

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 Connection settings

5.1 Receiving data

Add a data source

You can only receive data via MQTT if you use [Data Logger Suite](#). First, create a new configuration and select the data source type (figures 2 and 3). Then click "OK" and set up the broker connection.

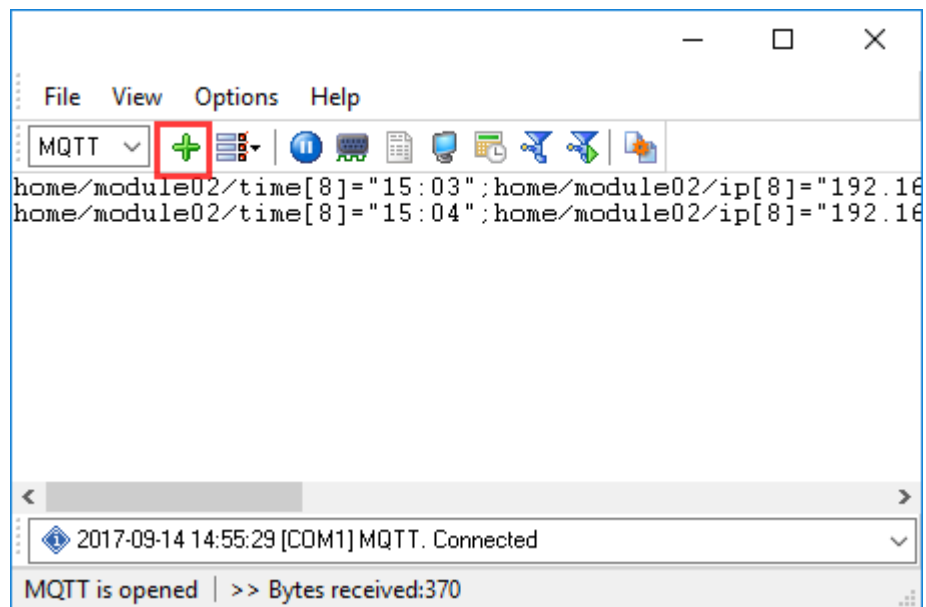


Fig. 2. Adding a new configuration

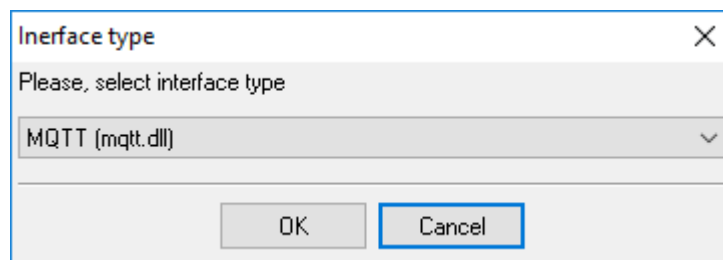


Fig. 3. Selecting the data source type

Connection settings

You need to specify the standard parameters for the connection:

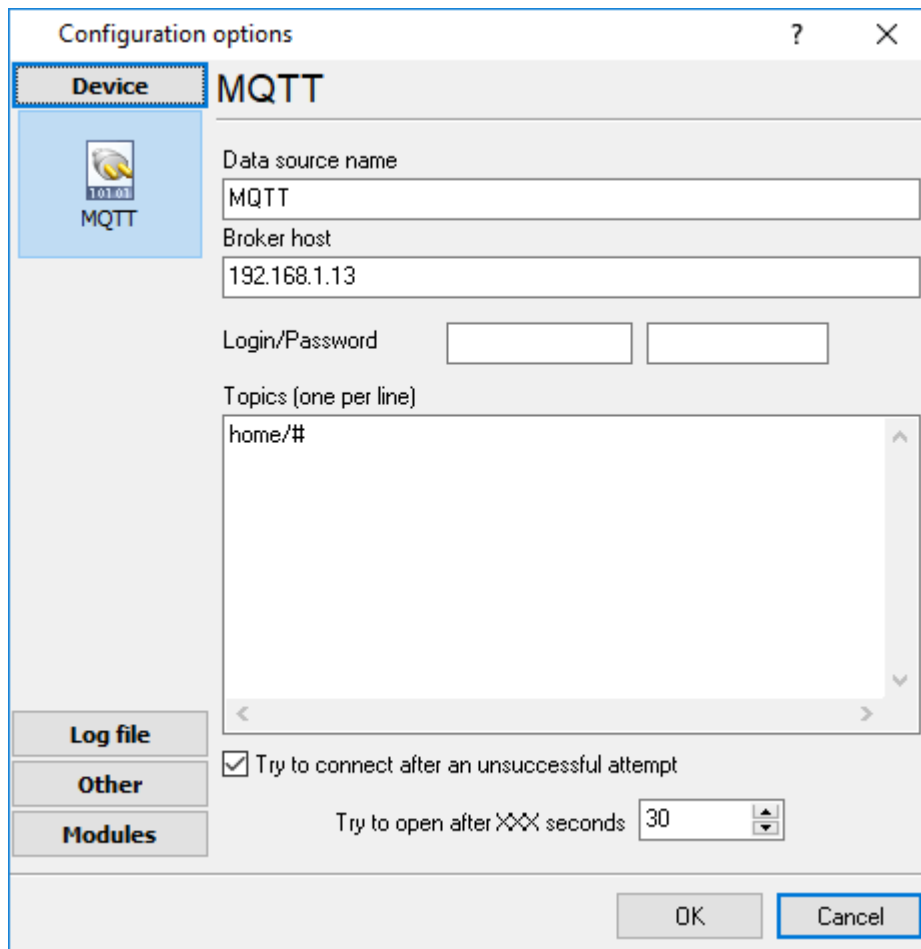


Fig. 4. Connection settings

Data source - The data source name that will be displayed in the program's main window. It will also be used as a client ID when connecting to the broker.

Broker address - The IP address or domain name of the server hosting the broker. By default, the port 1883 is used when connecting to the server. If the server uses a nonstandard port, specify it using the colon as a separator: 192.168.1.1:2222.

Login and password - The user's login and password for connecting to the broker.

If connection fails, try to reconnect - Enabling this option will make the program try to reconnect to the broker if the connection is lost for any reason (network problem, broker shutdown, etc.).

Topic - The list of topics, to which the program has to subscribe. Put each topic name on a separate line. You can use the standard MQTT wildcard characters in topic names.

/ (slash) — The topic level separator, which helps you represent topics as a tree on the server.

(hash) - This wildcard character matches any subtopic at any nesting level. For example, you can specify the topic name "home/#" to get data from each subtopic of "home"; for example:

home/temperature

home/temperature/hall
home/humidity

You can use the # wildcard character only after the topic level separator:

Correct: home/#
Incorrect: home#

Don't use any names after the # character:

Incorrect: home/#/hall

+ (plus sign) - This wildcard character matches any subtopic, but only at the next nesting level. For example, you can specify the topic name "home/+" to get data from the following subtopics:

home/temperature
home/humidity

In this case, you will not get any data from "home/temperature/hall"

Change prefix of received values

The plugin uses the topic name as a part of an exported variable name. You may change that prefix by adding the new prefix to the topic in the following format:

home/temperature->temperature1
home/humidity->humidity1
home/#->data
home/#->

The last row in the example above removes the prefix at all.

Receiving data

If you set up the connection and save the settings, the program will connect to the server, subscribe to the specified topics, and wait for any data from the broker. When the data is received, the program will group all the values within 500 milliseconds. It means that all the values received during that time will be exported in one line. You can watch data receiving in the main window of the program (fig. 5).

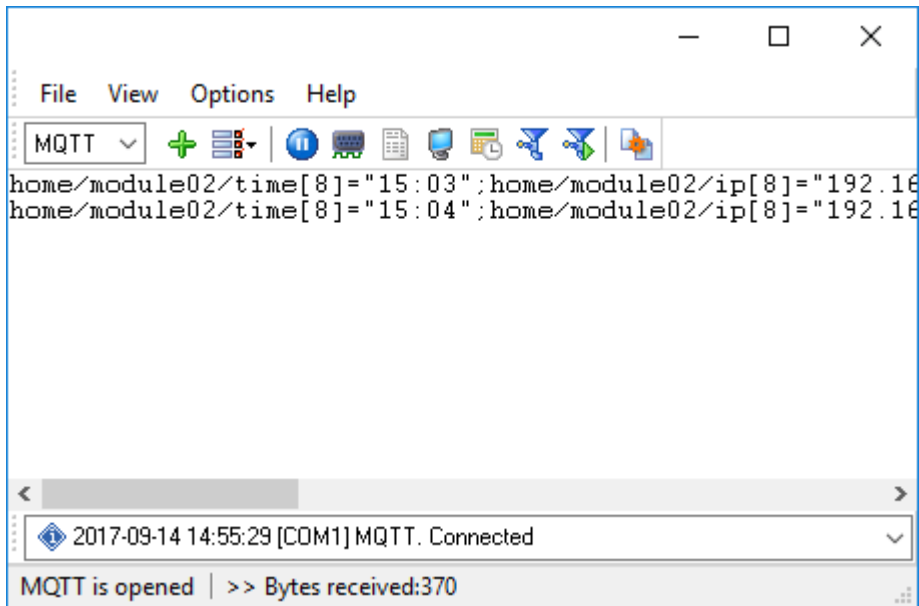


Fig. 5. Received data

JSON in received data

The plugin automatically detects JSON data in incoming payloads. JSON data should begin and end with "{" and "}" characters without additional spaces before.

In this case, the plugin extracts all values from JSON and exports it as separate variables.

Received data handling

When any data are received, the logger converts them to a simple text format, so that you can immediately log them to a text file. However, if you want to use the data received in a data filtering or data export module, you need to let the parser process the data (fig. 6). Just select the "OPC parser" module from the list. You don't need to set up the parser.

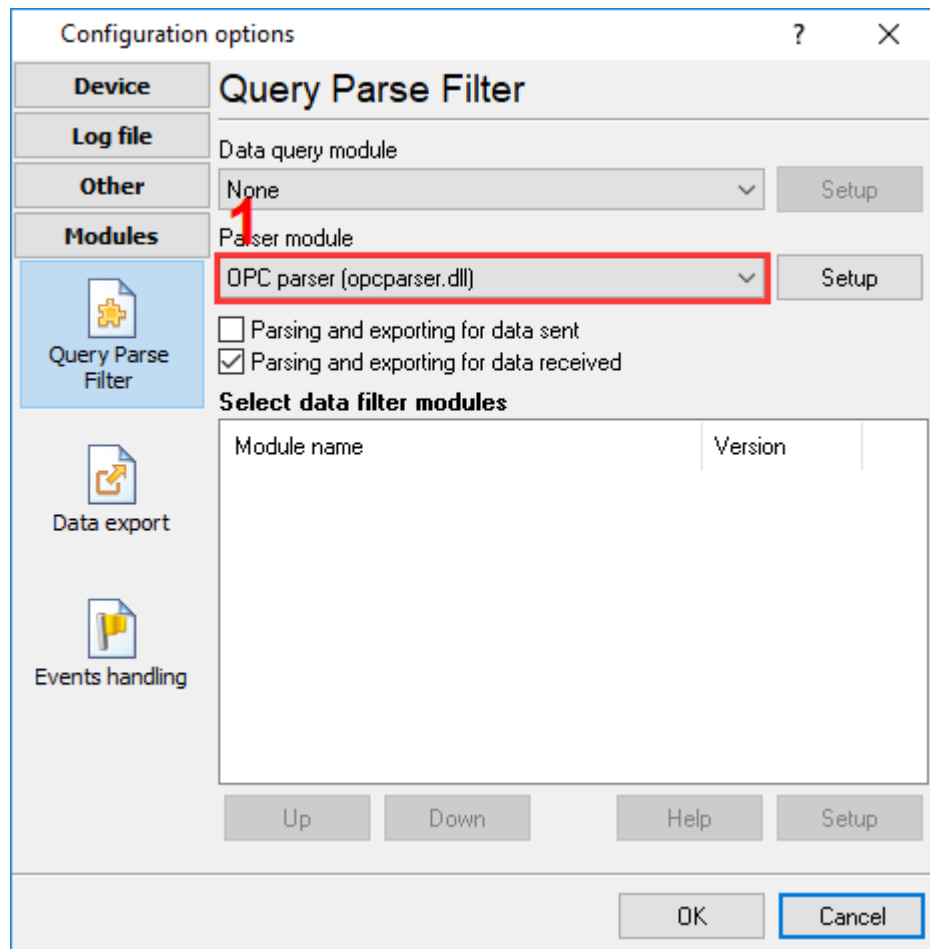


Fig. 6. The data parser

5.2 Data publishing

When publishing data, the logger acts as a publisher and passes them to the broker. You can get the input data from any data source supported by our loggers (COM, TCP, OPC, etc.). Use the parser to prepare the variables for exporting. The "MQTT" data export module will get the variables from the parser and publish them.

To enable the "MQTT" module, check the checkbox to the left of its name in the data export modules list (fig. 7).

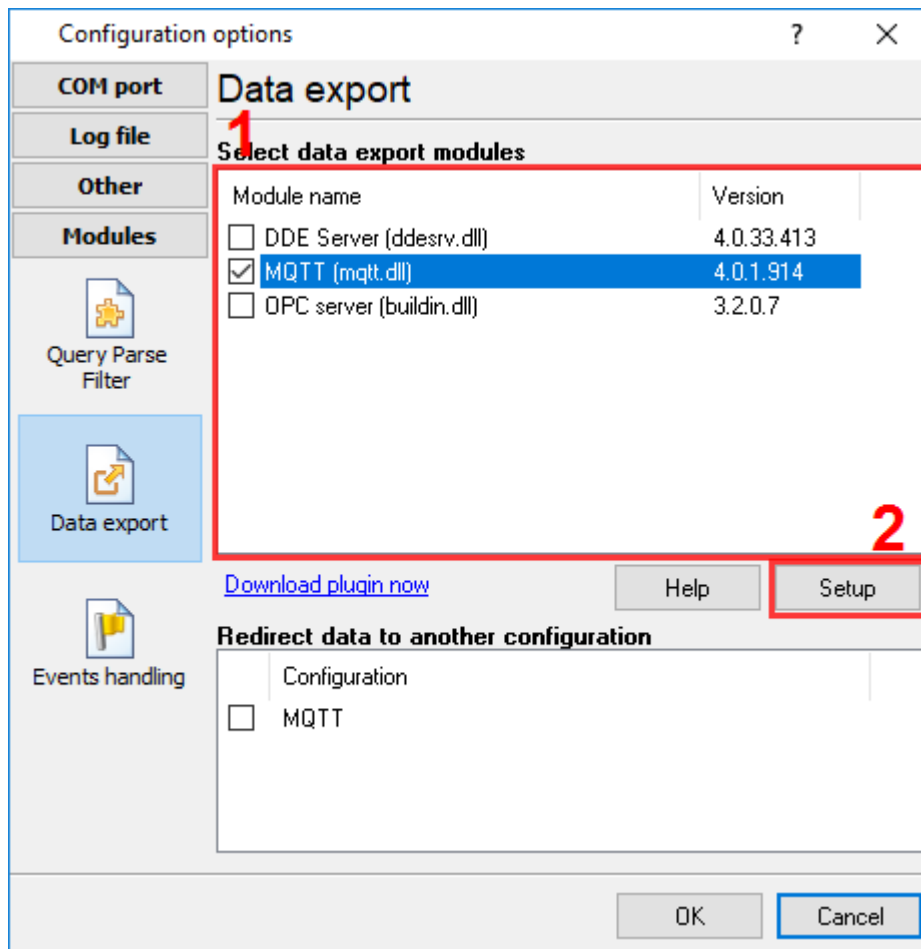
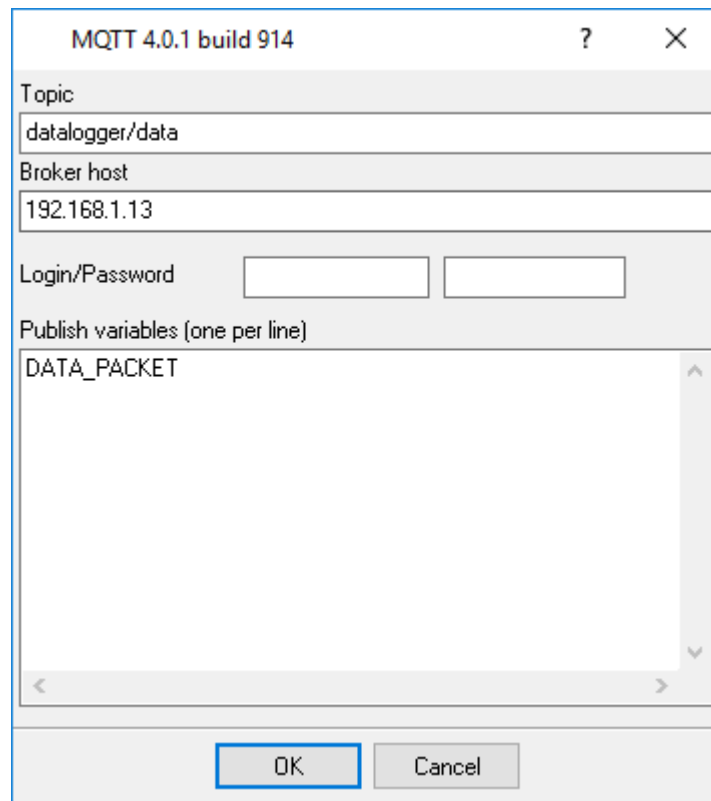


Fig. 7. Selecting the data export module

For more details about the broker connection settings, see the [previous section](#).

When publishing the data, you need to specify the basic topic for publishing them (fig. 8). The resulting topic name will be derived from the basic topic name and the parser variable name. For example, the basic topic name is “datalogger/data” and the parser variable name is “Temperature”; the value will be published in the topic named “datalogger/data/Temperature”

You can also specify a list of variables for publishing. If the parser also produces any other variables that are not specified in your list, they will not be published via MQTT. If your list is empty, all of the available data will be published.



The image shows a dialog box titled "MQTT 4.0.1 build 914" with a help icon (?) and a close icon (X). The dialog contains the following fields and controls:

- Topic:** A text input field containing "datalogger/data".
- Broker host:** A text input field containing "192.168.1.13".
- Login/Password:** Two empty text input fields.
- Publish variables (one per line):** A text area containing "DATA_PACKET".
- Buttons:** "OK" and "Cancel" buttons at the bottom.

Fig. 8. Data exporting settings