



**The InfluxDB & HTTP Export plugin
PRINTED MANUAL**

InfluxDB & HTTP Export plugin

© 1999-2021 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: 3/29/2021

Publisher

AGG Software

Production

© 1999-2021 AGG Software

<http://www.aggsoft.com>

Table of Contents

Part 1 Introduction	1
Part 2 System requirements	1
Part 3 Installing InfluxDB & HTTP Export	2
Part 4 Glossary	3
Part 5 Connection settings	3
1 InfluxDB	3
Tags and values	4
2 HTTP POST	6
3 HTTP GET	7
4 Error handling	7
Part 6 Troubles?	9
1 Possible problems	9

1 Introduction

The "InfluxDB HTTP Export" data publishing module for our loggers allows you to write data to an InfluxDB database by using that database's HTTP API.

InfluxDB is a database intended to store time-series data. It was specially developed to handle a lot of read and write requests.

You don't need to know SQL to write data to an InfluxDB database.

The module can also export data using HTTP GET and HTTP POST requests.

The module can create a backup copy of data if a database is offline, or if an attempt to write data to the database has failed. Later, when the connection to the database is reestablished, all data will be restored from the backup copy.

2 System requirements

The following requirements must be met for "InfluxDB & HTTP Export" 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.

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 to this problem.

3 Installing InfluxDB & HTTP Export

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.

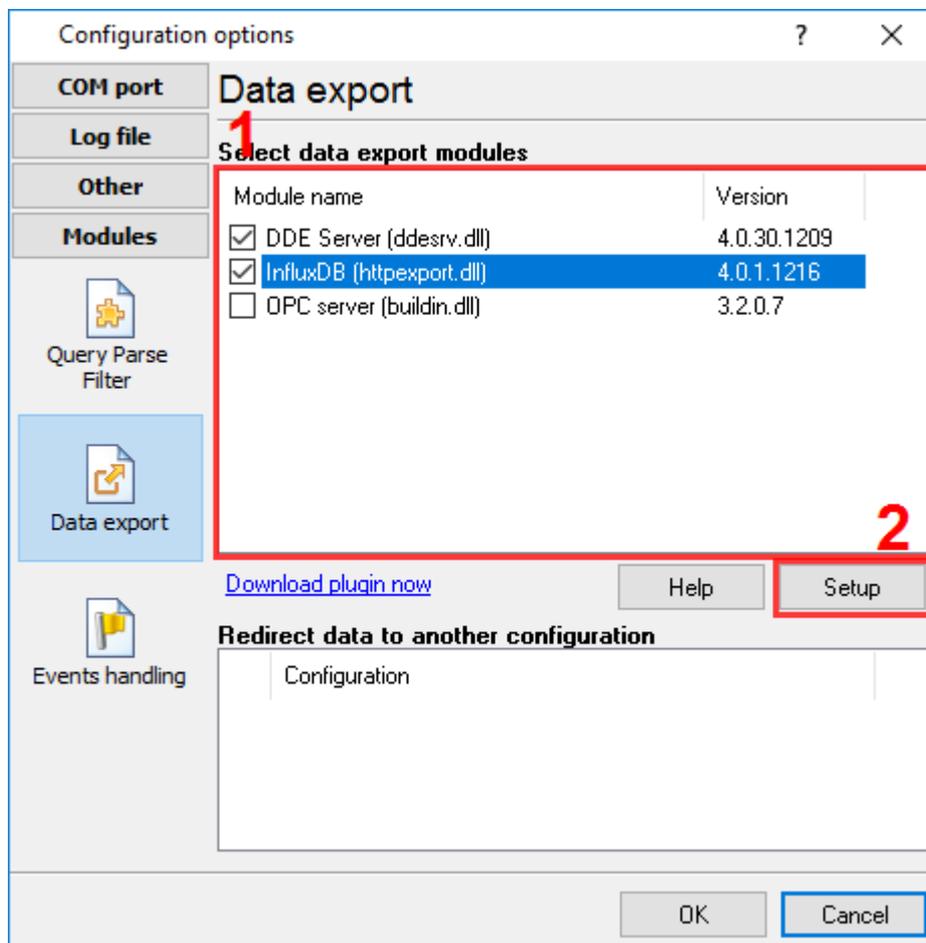


Fig.1. Example of installed module

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 InfluxDB

The module prompts you to configure the following connection parameters:

Temporarily disable – This checkbox allows you to temporarily disable a module (for example, to debug or test the program).

URL to post data – Enter a full URL address for posting data. You can see an example of the URL address in Figure 1. Make sure that the URL address specifies the protocol (http or https) and the port number for accessing the HTTP API.

Login and password – The user's login and password to be used to connect to the database and write data to it. If user authentication is disabled in the database settings when using the HTTP API, leave these fields empty.

Database – The database name. The database must already exist, and the user must have privileges to write to the database.

Measurement – Speaking in terms of the traditional SQL database, it is a table name.

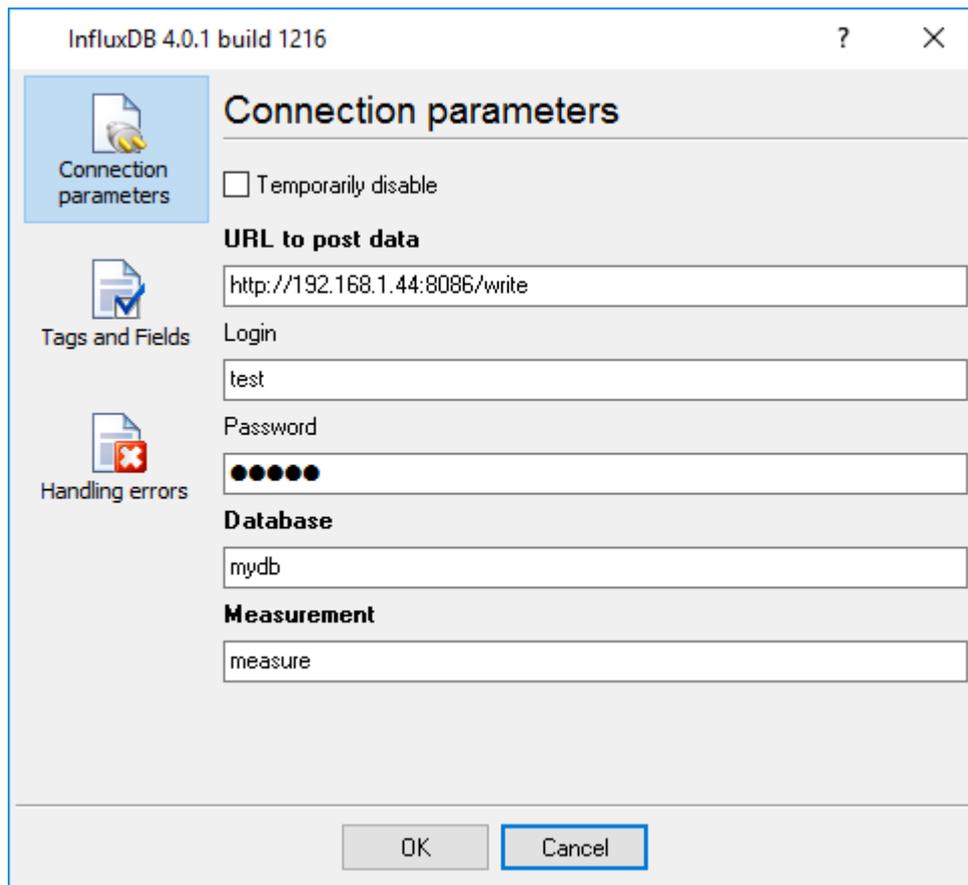


Figure 1: Connection settings

5.1.1 Tags and values

Speaking in terms of the traditional SQL database, a tag is an indexed table field, and a field is a non-indexed table field. Both tags and fields can store values of certain types.

Tags can store only string values, and fields can store string, integer, float, or boolean values. For more details on data types, please refer to the database manual.

In Figure 2, you can see a list of tags and fields, which also contains parser variables linked to them.

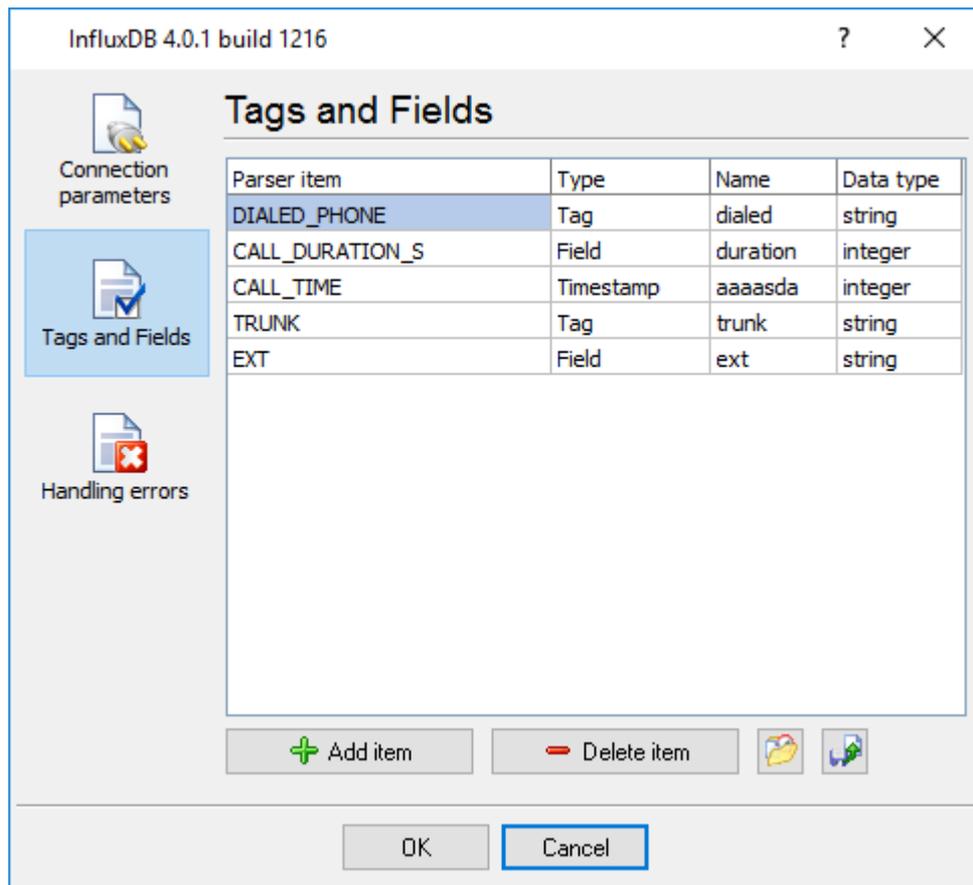


Figure 2: Tags and values

Parser variable name – The name of the parser variable whose value should be written to the database. You can select a variable name in the drop-down list, or enter the name manually.

You need to link variables for each tag and field, and also for each timestamp.

Note: Before doing that, you need to configure the parser. If the parser is not working properly, no data will be written to the database. For more details about configuring the parser, please refer to the parser manual or see tutorials on our website.

Note that you cannot write empty values to the database. If a parser variable with the specified name is unavailable when data are being exported, or if its value is empty, it will be ignored.

If values are missing for all specified fields when data are being exported, such a request will be ignored, too.

Type – Defines the parameter type (tag, field, or timestamp). The list can contain an unlimited number of tags and fields. Only one timestamp variable is allowed. If no timestamp variable is specified, a database timestamp will be used when writing data to the database. You don't need to specify the data type and the field name for the timestamp variable.

Name – The name of a tag or field in the database to write data to.

Data type – Defines the data type in the database for the specified parameter. Depending on the specified data type, the module will convert data from the parser variable to the necessary data type in the database.

The parameter order in the list doesn't matter. When generating a request to the database, the module will sort tags and fields alphabetically, and the timestamp will always be in the end.

5.2 HTTP POST

The HTTP Export plugin can also export data using the simple HTTP POST request.

URL to post data – Enter a full URL address for posting data. You can see an example of the URL address in Figure 1. Make sure that the URL address specifies the protocol (http or https) and the port number for accessing the HTTP API. The URL may contain a special placeholder like {PARSER_VALUE}. The plugin replaces this placeholder using the corresponding parser's variable value before each request.

API key (optional) - It allows you to check requests validity. The plugin adds the "apikey" value to the request. It is applicable for the 'application/x-www-form-urlencoded' encoding only. You may use customer headers for other encoding.

Successful processing sign (optional) - if it is specified, the plugin checks a server response, and mark the request invalid if the server response does not include the specified sign. The plugin tries to repeat unsuccessful requests five times.

Data encoding

By default, the plugin uses the 'application/x-www-form-urlencoded; charset=utf-8' content type encoding in the HTTP request.

String value - the plugin encodes all string to the UTF-8 encoding.

Decimal numbers - the plugin converts decimal number to a string without any separators (example, 123456789).

Float numbers - the plugin converts float number to a string with the dot delimiter (example, 123.456789).

Datetime - the plugin uses the ISO format YYYY-MM-DD HH:NN:SS (example, 2019-01-01 00:00:01).

Date - YYYY-MM-DD (example, 2019-01-01).

Time - HH:NN:SS (example, 00:00:01).

Boolean values - The plugin encodes the logical "true" to "TRUE", and logical "false" to "FALSE" (without quotes).

The plugin encodes all other data types as strings.

The "Advanced" tab allows you to specify extra options of the HTTP request:

Encoding - you may use a custom encoding type if you send data in JSON or SOAP formats.

```
application/json  
application/soap  
text/plain  
application/binary
```

If you select the custom encoding, you must prepare and set the "HTTP_DATA" variable. For example, you can do it using the "Data encode" and "Expressions" plugins.

Below is the example expressions for the "Expressions" plugin.

```
HTTP_DATA='{\"value\": \"12345\", \"value2\": 2}'
```

Custom headers

You may add custom HTTP headers for all your HTTP requests. You should add one header value per line.

Example:

```
My-Name: My-Value
```

Please note, you should use semi-colon and space as a separator between the header name and value.

5.3 HTTP GET

The HTTP GET setup is mostly identical to [HTTP POST](#), but with some differences:

1. The HTTP GET request size is limited to 1024 bytes.
2. If a data packet contains the "HTTP_DATA" variable, the plugin will encode to a URL-compatible string and append the variable value to the request URL. The plugin will ignore all other variables.
3. If a data packet contains the "HTTP_DATA_RAW" variable, the plugin will append that value without any encoding to the request URL. The plugin will ignore all other variables.

5.4 Error handling

During the module's operation, some errors may occur when the program interacts with the database (for example, no connection to the database, temporary issues with the database, etc.). The "Handling errors" tab allows you to configure the error handling settings (Figure 3).

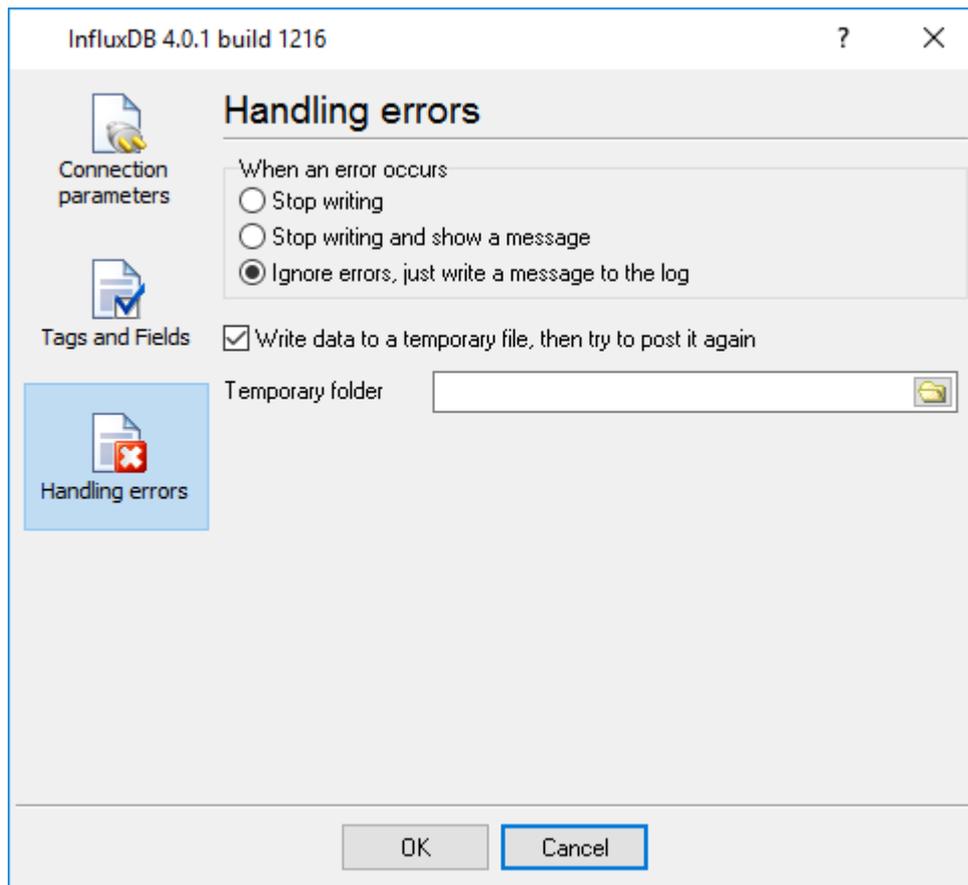


Figure 3: Error handling settings

The following options of the module's response to an error are available:

1. **Stop writing** – If an error occurs, the program will generate an error message and set the internal status "**Temporarily disable**"; that is, any further data writing operations will be stopped until the module's settings are changed. Any data received for posting after the error has occurred will NOT be written to a temporary file or to the database. This option may be useful when you are configuring the module's settings.
2. **Stop writing and show a message** – This setting is similar to the previous one. The only difference is that an error message will be displayed.
3. **Ignore errors, just write a message to the log** – If an error occurs, the module will write an error message to the log, but will continue its operation.

In the last two cases, if you want to avoid the loss of data intended for posting, you can save them to a temporary file, which will be created in the **Temporary folder**. The data will be written to a temporary file only if the "**Write data to a temporary file**" option is turned on. If an error occurs when the program tries to post any data, the data will be written to that file.

Later the module will take five attempts to write data from the temporary file to the database. If all the five attempts to write data to the database fail, the data will be deleted from the temporary file (that is, the data will be lost).

Note that if you have not properly configured writing to the database, multiple attempts to restore data from the backup copy will significantly increase the server load and the use of network resources.

However, you can be assured that if you fix the error, all data from the temporary file will be written to the database.

Note that when specifying the **Temporary folder**, you need to select a folder to which the user who has launched the program has both read and write privileges. Otherwise, the module will be unable to create a temporary file, and the data will be lost. If you leave the "**Temporary folder**" field empty, the program will create a temporary file in the module installation folder.

When restoring data from a temporary file, the module can simultaneously use up to five data streams to transfer data to the server. Keep that in mind when configuring a firewall for the database server.

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.