



**The Bacnet/IP plugin  
PRINTED MANUAL**

# Bacnet/IP plugin

© 1999-2016 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: 28.11.2016

## **Publisher**

*AGG Software*

## **Production**

© 1999-2016 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 Bacnet/IP</b>	<b>1</b>
<b>Part 4 Glossary</b>	<b>2</b>
<b>Part 5 User manual</b>	<b>3</b>
<b>Part 6 How to configure the logger</b>	<b>5</b>
<b>Part 7 Troubles?</b>	<b>8</b>
1 Possible problems .....	8

## 1 Introduction

The Bacnet/IP module is designed to read data from BACnet compatible devices in the local network. The module can poll all listed devices and their sensors at regular intervals. The received values are passed to the main program, where they can be, for example, logged or written to a database.

## 2 System requirements

The following requirements must be met for "Bacnet/IP" 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 Bacnet/IP

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.

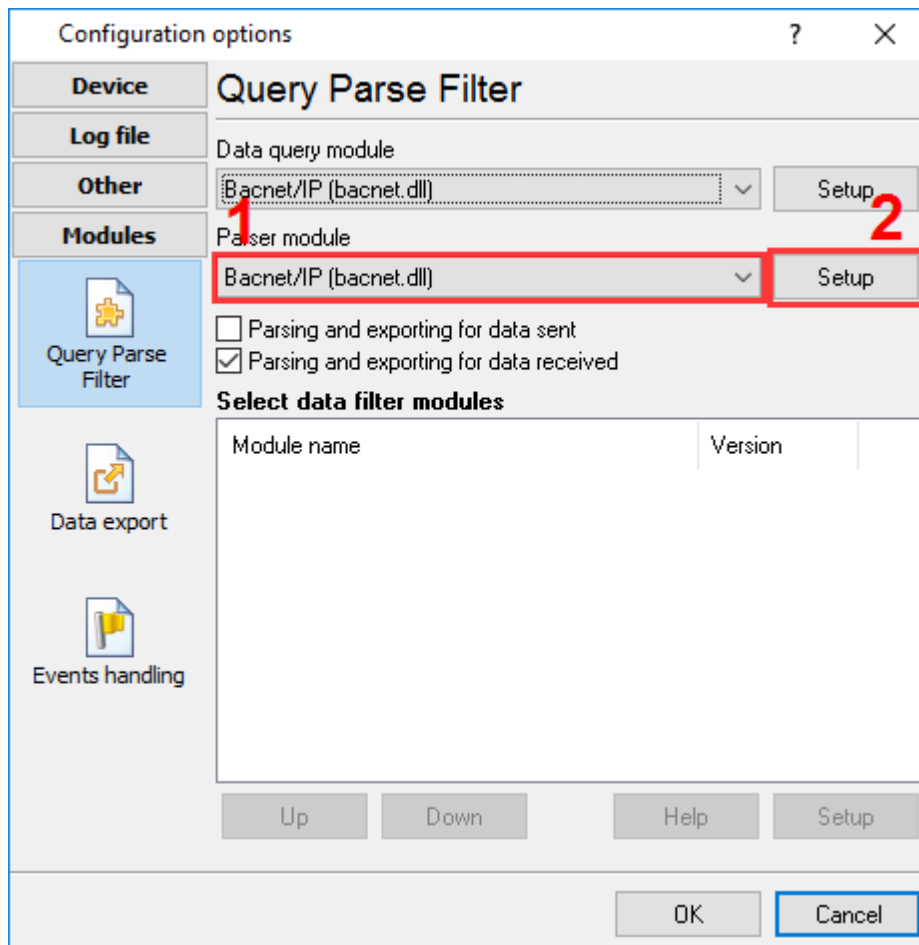


Fig.1. Example of installed module

## 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 User manual

For the module to work, you need to make a request list. Each request may be addressed to one or more devices. All requests in the queue are performed one by one, top to bottom.

- **Simultaneous:** A request is sent once, either when the program starts or when the module's configuration is changed;
- **Polling:** A request is sent regularly at the specified time interval.

To add a new request, click the "Action" button and select "Add a request." A dialog box will open (Figure 1). Enter a request description, which can include any characters, and click the "OK" button.

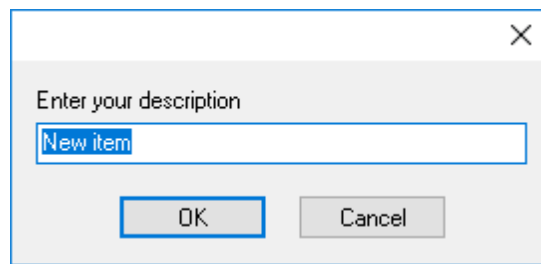


Figure 1: Request description entry dialog box

The new request will appear in the request list (Figure 2). Each request includes a few important options that need to be configured:

- **Send requests, otherwise only process responses:** Turn on this option if you want the module to poll devices at regular intervals.
- **Endpoint:** Specify the IP address and port of the device to be polled in the local network. The BACnet protocol allows you to automatically find BACnet devices in the local network. To do that, just specify a broadcast IP address like 192.168.1.255, where 255 is the mask of the subnetwork to which the request will be sent. All devices that have replied to that request will be automatically added to the request queue.
- **Instance:** A unique identifier of a BACnet device in the local network. To specify multiple addresses, use the semicolon as a delimiter (for example, 1345;1346). You can also use the \* character (see Figure 2), which would force the module to poll all devices found in the network;
- **Request timeout (ms):** The time to wait for a device's response after sending a request. If this time expires, the module will automatically cancel the current request and proceed to the next request in the queue. If any data are received from that device after the time has expired, they will be ignored. Response waiting time depends on the network and on the performance of the devices.

### Request method

- **Once, on program startup:** The program will send a request only once, when it starts.
- **Polling:** The program will send requests regularly at the specified time interval. Because the module performs requests in the queue one by one, the actual interval between requests also depends on the time of execution of the previous requests. If the specified time interval is too small, the program might not be able to perform each request in the queue within the specified interval.

**Request queue**

Property	Value
<b>Request #1</b>	
<input checked="" type="checkbox"/> Send requests, otherwise parse response only	
Endpoint	192.168.1.255:47808
Instance	*
Request timeout (ms)	1300
<b>Request method</b>	
<input type="radio"/> Once, on program startup	
<input checked="" type="radio"/> Polling	
Interval (ms)	1000
Interval units	
<b>Response items</b>	
<b>Item #1</b>	
Name	ITEM1
Data type	Analog Input
Instance	0
<b>Item #2</b>	
Name	ITEM2
Data type	Analog Input
Instance	1

Action   Read basic device properties

Minimal interval between data packets (ms)

**Figure 2: Request queue**

If you have added several requests to the queue, you can move them up or down. To do it, select a request, click the “Action” button, and select an action (“Move up” or “Move down”).

You can also click this button to change the description of a request, or to remove a request from the queue.

You can also perform the same actions by using the context menu that pops up when you right-click items in the request tree.

## Response elements

This list contains polled values.

To add more variables to a request, click the “Action” button and select “Add a response.” Before doing that, you need to select the request from the queue. The new variable will appear in the “Response elements” group (Figure 2).

Each response element has the following options:

- **Name:** The name of the variable. The value of the variable is passed under that name to the main program and used in the data export modules. You must give a unique name to each response element in the request. However, variable names in different requests may coincide.
- **Data type:** The type of the value to be read from a BACnet device. The BACnet protocol defines a set of standard data types. Some devices support only a limited set of data types. For example, a device may support analog values only;
- **Instance:** The sequence number of the value in the device's internal memory.

## 6 How to configure the logger

This module implements data communication based on the BACnet/IP protocol in an Ethernet network. For this reason, the module can be used with the following products only:

- Advanced TCP/IP Data Logger
- Data Logger Suite

To configure a network connection, you need to create a new configuration in the program by clicking the “green plus” button in the main window. In the network connection settings, select the “UDP” protocol type and the “Server” mode (Figure 1). The standard port number used for the BACnet protocol is 47808, but in practice any other port number can be used, too. The IP address 0.0.0.0 allows the program to use any available network interfaces of the computer.



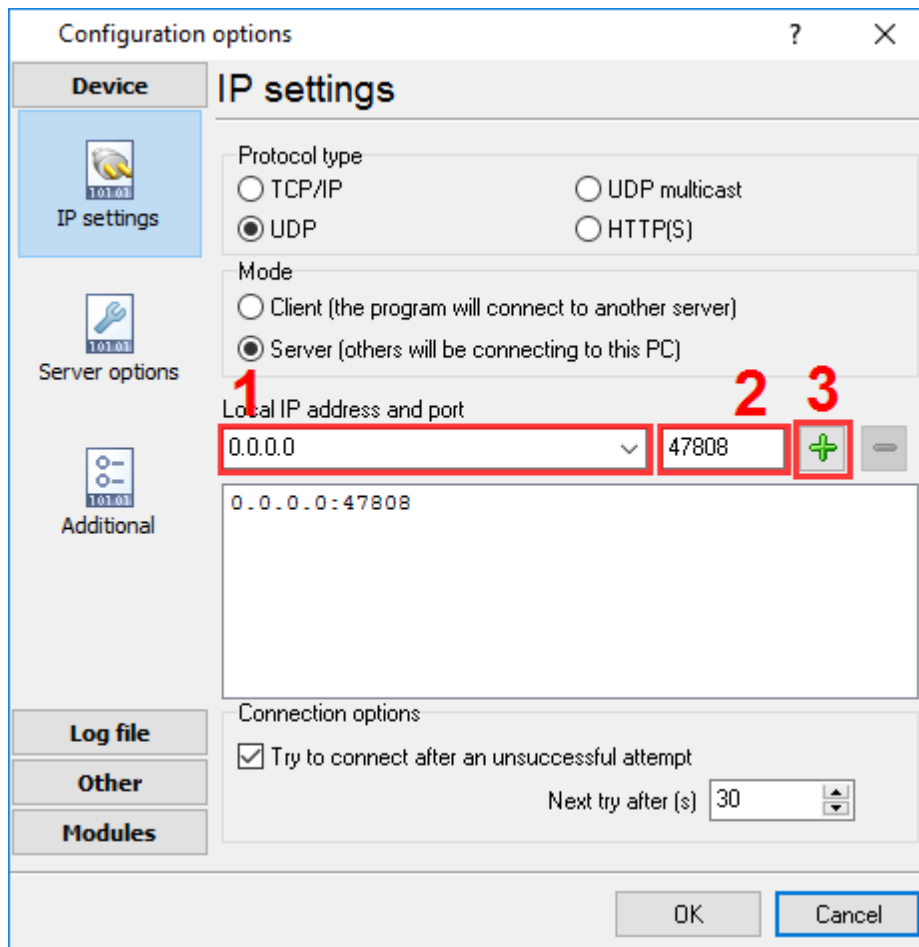


Figure 1: Configuring the network connection

After that, you can select the recommended data display mode (Figure 2), so that the received values are displayed in the main window.

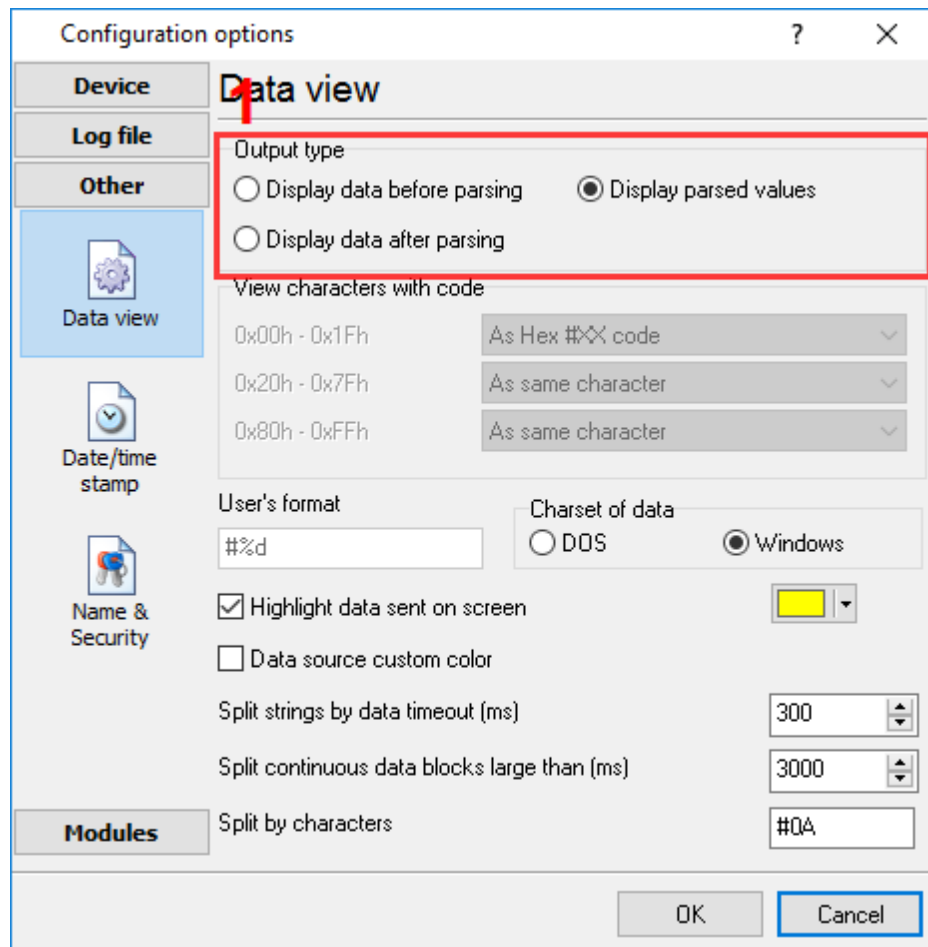


Figure 2: Data display mode

On the data request modules tab, select “BACnet/IP” as shown in Figure 3. The availability of active filter modules and export modules depends on the specific task.

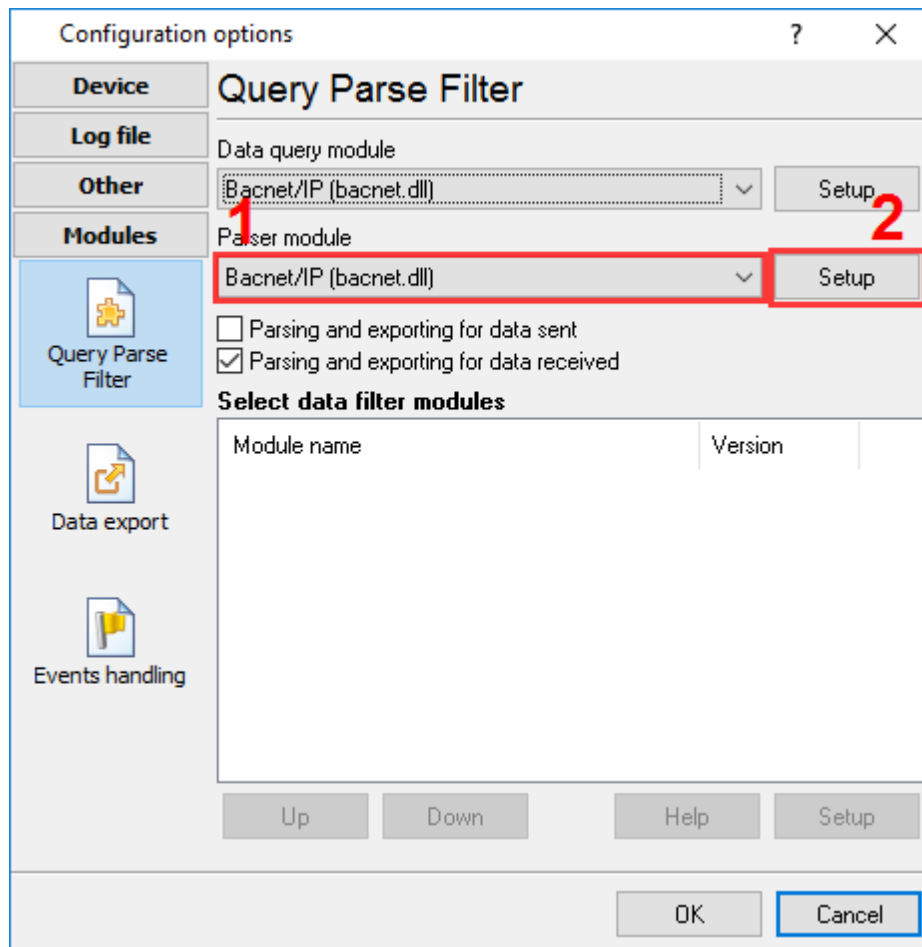


Figure 3: Selecting modules

## 7 Troubles?

### 7.1 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.