



Data Logger Suite

Trust in Confidence!

PRINTED MANUAL

Data Logger Suite

© 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: 29.11.2016

Publisher

AGG Software

Production

© 1999-2016 AGG Software

<http://www.aggsoft.com>

Table of Contents

Part 1 Introduction	1
1 About Data Logger Suite	1
2 Glossary	2
Part 2 License, Registration and technical support	4
1 License	4
2 Limitations	5
3 How to register	6
4 Support	6
Part 3 Installation	6
1 System requirements	6
2 Installation process	6
Part 4 Program use	7
1 Getting started	7
2 Introduction	8
3 Data flow diagram	10
4 Work complete	11
5 Useful advices	11
6 Configuration	12
Serial port	12
Serial (COM) port.....	12
Network connection	17
TCP/IP settings.....	17
Modem settings	24
Modem settings	24
Task schedule.....	30
OPC settings	41
OPC settings.....	41
SNMP settings	50
SNMP data source.....	50
Group and variable tree.....	50
Group properties.....	52
Item properties.....	54
File settings	55
File data source.....	55
Folders settings.....	57
Files settings	59
Schedule	61
DDE settings	62

Description.....	62
Groups, servers and items.....	62
Advanced parameters.....	67
Step-by-step instructions.....	68
Writing DDE to MSSQL.....	68
Writing a lot of DDE items	68
Writing several DDE items to separate columns.....	75
Writing DDE to MySQL.....	79
Writing DDE items to a database.....	81
Writing DDE data to Excel.....	82
SQL data	83
Introduction.....	83
Connection parameters	84
SQL	87
USB HID	88
Selecting USB devices	88
HID data log options.....	89
Samples	90
Additional parameters	91
Data view change.....	91
Date/time configuration.....	92
Name and security.....	93
Log files	95
Log rotation.....	95
Log file access.....	97
Log deletion.....	98
CSV file logging.....	100
Modules	100
Introduction & setup.....	100
OPC server.....	104
7 Program options	107
Window view	107
Date/time stamp view	109
Protocol and errors handling	110
Service mode on Windows 2000+	111
Configuration.....	111
Windows Vista+ notes.....	115
Restart & Security	116
Part 5 Serial communications overview	0
Part 6 Having problems?	117
1 Program doesn't run or work	117
2 FAQ	118

1 Introduction

1.1 About Data Logger Suite

Data Logger Suite is used to monitor and record data transferred through various data interfaces. The one program can combine data from multiple data sources and write it to files, database or export it in real-time to other applications.

What problems can be solved with Data Logger Suite?

By using a logger, an user may monitor the data flow in real-time, and respond appropriately to any anomalies or any other particular data that may arise. This level of control is crucial in some applications making use of RS232, TCP/IP and other data interfaces, most notably in scientific and industrial equipment. The data logger allows users to view, record, and even manipulate the data being transferred over a RS232, TCP/IP and other data interfaces.

Key features of Data Logger Suite is:

- Can collect data from multiple different data interfaces (RS232, RS422, RS485, TCP/IP, UDP, SNMP, OPC and others);
- Extended logging features. Can output received data to formatted log files. Supports date/time stamping;
- Fast multi-threaded, optimized and efficient architecture;
- Data filters. Allow you filter, format, aggregate your data. You may define simple rules or use powerful regular expressions;
- Real-time export capabilities. Data Logger Suite can run as a DDE server export OPC data to old applications;
- MS Excel. Data export to ready-to-use MS Excel files;
- Databases support. Exporting data to MSSQL, MySQL, ODBC-compatible databases (Firebird, Interbase and others);
- Plug-ins. Many free plug-in modules that are extending program features;
- Simple, dialog-driven step by step set-up Programming is not required to configure the software to collect data;
- Supporting various operating systems. It runs under all versions starting from Windows 2000, including 32 and 64-bit systems..
- Windows service mode. Unlike most other serial logging applications, Data Logger Suite can run as a service so that it starts as soon as the operating system starts and doesn't require a user to log in and run it. It will continue to run even as users logon and logoff the workstation.

It is extremely easy to use! The configuration process is fully menu driven and has complete, context sensitive, on-line help. you can easily customize all input to your exact specifications. Once you see how easy it is to use Data Logger Suite, you will never again take data readings by hand!

Company home page: <http://www.aggsoft.com/>

Software home page: <http://www.dataloggersuite.com/>

1.2 Glossary

ASCII - An acronym for American Standard Code for Information Interchange. ASCII files are plain, unformatted text files that are understood by virtually any computer. Windows Notepad and virtually any word processor can read and create ASCII files. ASCII files usually have the extension .TXT (e.g., README.TXT).

Binary File - A file that contains data or program instructions written in ASCII and extended ASCII characters.

Bit - Binary digit in the binary numbering system. Its value can be 0 or 1. In an 8-bit character scheme, it takes 8 bits to make a byte (character) of data.

Bytes - A collection of eight bits that represent a character, letter or punctuation mark.

Cable - Transmission medium of copper wire or optical fiber wrapped in a protective cover.

Client/Server - A networking system in which one or more file servers (Server) provide services; such as network management, application and centralized data storage for workstations (Clients).

COM port - Short for a serial communication port. Most serial communication software communicate with a computer through a communication port, and most IBM and IBM-compatible computers support up to four serial ports COM1, COM2, COM3 and COM4. Additional ports can be added by adding additional hardware.

Data bits - A group of bits (1's and 0's) that represent a single character or byte. Typically, there are seven or eight data bits. During an asynchronous communication (e.g., BitCom connecting to CompuServe), each side must agree on the number of data bits. Data bits are preceded by a start bit and followed by an optional parity bit and one or more stop bits.

DA – data access.

DNS (Domain Name System) - A DNS server lets you locate computers on a network or the Internet (TCP/IP network) by domain name. The DNS server maintains a database of domain names (host names) and their corresponding IP addresses. PC aggsoft's IP address, 198.63.211.24, corresponds to the DNS name www.aggsoft.com.

Flow control - A method of controlling the amount of data that two devices exchange. In data communications, flow control prevents one modem from "flooding" the other with data. If data comes in faster than it can be processed, the receiving side stores the data in a buffer. When the buffer is nearly full, the receiving side signals the sending side to stop until the buffer has space again. Between hardware (such as your modem and your computer), hardware flow control is used; between modems, software flow control is used.

Handshaking - Is the way in which the data flow between computers/hardware is regulated and controlled. Two distinct kinds of handshaking are described: Software Handshaking and Hardware Handshaking. An important distinction between the kinds of signals of the interface is between data signals and control signals. Data signals are simply the pins which actually transmit and receive the characters, while control signals are everything else.

Internet - A global network of networks used to exchange information using the TCP/IP protocol. It allows for electronic mail and the accessing and retrieval of information from remote sources.

IP, Internet Protocol - The Internet Protocol, usually referred to as the TCP/IP protocol stack, allows computers residing on different networks to connect across gateways on wide-area networks. Each node on an IP network is assigned an IP address, typically expressed as 'xx.xx.xx.xx'.

IP address (Internet Protocol address) - The address of a computer attached to a TCP/IP network. Every client and server station must have a unique IP address. Client workstations have either a permanent address or one that is dynamically assigned to them each dial-up session. IP addresses are written as four sets of numbers separated by periods; for example, 198.63.211.24.

LAN (Local Area Network) - A network connecting computers in a relatively small area such as a building.

NIC, Network Interface Card - A card containing the circuitry necessary to connect a computer to a particular network media. Typically, the NIC plugs into the computer's accessory bus, (PCI, USB, etc.) and provides a network connection such as 10baseFL (fiber ethernet), thin-net, AUI, etc.

OPC (OLE for Process Control) – a set of universally accepted specifications providing a universal data exchange mechanism in control and management systems.

OPC Alarms and Events - the OPC interface for access to alarm and event data.

OPC Data Access - the OPC interface for access to real-time data.

OPC DA - see OPC Data Access.

OPC Historical Data Access - the OPC interface for access to archived data.

OPC HDA - see OPC Historical Data Access.

PC - abbreviation for a Personal Computer.

Parity - In data communications, parity is a simple procedure of checking the integrity of transmitted data. The most common type of parity is Even, in which the number of 1's in a byte of data add up to an even number, and None, in which a parity bit is not added.

Ports - A connection point for a cable.

Protocol -A formal description of a set of rules and conventions that govern how devices on a network exchange information.

RS232, RS423, RS422 AND RS485 - The Electronics Industry Association (EIA) has produced standards for RS232, RS423, RS422, and RS485 that deal with data communications. EIA standards were previously marked with the prefix "RS" to indicate the recommended standard. Presently, the standards are now generally indicated as "EIA" standards to identify the standards organization.

Electronic data communications will generally fall into two broad categories: single-ended and differential. RS232 (single-ended) was introduced in 1962. RS232 has remained widely used, especially with CNC control builders. The specification allows for data transmission from one transmitter to one receiver at relatively slow data rates (up to 20K bits/second) and short distances

(up to 50' @ the maximum data rate). This 50' limitation can usually be exceeded to distances of 200' or more by using low capacitance cable and keeping the data rates down to 9600 baud and lower.

RTS/CTS Hardware handshaking - uses additional wires to tell a sending device when to stop or start sending data. DTR and RTS refer to these Hardware handshaking lines. you can select whether you need to use DTR or RTS individually, or use both lines for hardware handshaking. See also Xon/Xoff.

TCP/IP, Transport Control Protocol / Internet Protocol - TCP and IP are communications protocols, that is, structured languages in which data is communicated between one process and another, and between one network and another. TCP/IP is implemented in a multi-level layered structure.

TCP/IP is the 'glue' that that ties together the many heterogeneous networks that make up the Internet.

Stop bits - In data communication, one or two bits used to mark the end of a byte (or character). At least one stop bit is always sent.

2 License, Registration and technical support

2.1 License

Copyright © 1999-2016 AGG Software.
All Rights Reserved

SOFTWARE LICENSE

Trial Limited Version

The trial limited version of this software may be used for evaluation purposes at the user's own risk for a trial period. At the end of the trial period, the user must either purchase a license to continue using the software, or remove it from his/her system.

The trial limited version may be freely distributed, provided the distribution package is not modified. No person or company may charge a fee for the distribution of Data Logger Suite without written permission from the copyright holder.

Licensed Version

On payment of the appropriate license fee, the user is granted a non-exclusive license to use Data Logger Suite on one computer (i.e. a single CPU), for any legal purpose, at a time. The registered software may not be rented or leased, but may be permanently transferred, if the person receiving it agrees to terms of this license. If the software is an update, the transfer must include the update and all previous versions.

Registered customer are entitled to free updates during one year from the date of purchase. It

means that during one year you can download and install the latest registered versions of Data Logger Suite from our site. If you don't want to purchase an updates, you can use the program forever; it will never expire, but you won't be able to use the latest version. If you purchased the software more than one year ago, you are no longer entitled to free upgrade and technical support; however, you can purchase an updates to the latest version at a special, greatly discounted price, and this updates will allow you to have free updates and technical support for another year. The type of update license must match the type of your existing license.

Whilst every care has been taken in the construction and testing of this software, it is supplied subject to the condition that the user undertakes to evaluate the suitability of the control for his/her purposes. AGG Software makes no representation of the software's suitability for any purpose, and the user agrees that AGG Software has no responsibility for any loss or damage occasioned by the use of this software.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" AND AGG SOFTWARE DISCLAIMS ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, CONFORMANCE WITH DESCRIPTION, TITLE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL AGG SOFTWARE BE LIABLE FOR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL OR EXEMPLARY DAMAGES OR LOST PROFITS WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE PRODUCT, EVEN IF AGG SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, AGG SOFTWARE'S CUMULATIVE AND ENTIRE LIABILITY TO YOU OR ANY OTHER PARTY FOR ANY LOSS OR DAMAGES RESULTING FROM ANY CLAIMS, DEMANDS OR ACTIONS ARISING OUT OF OR RELATING TO THIS AGREEMENT SHALL NOT EXCEED THE PURCHASE PRICE PAID FOR THIS LICENSE.

Should any term of these terms and conditions be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms hereof.

If you do not agree to these conditions you should not install this software.

2.2 Limitations

Program is distributed on shareware terms. This means limited and unavailable secondary program possibilities, which become valuable or available after program registration. To register the program read [here](#) ⁶.

In trial version of our program are the following limits:

- Trial period is limited by 21 days. After that time program won't work until it is registered.
- Continuous program work time is limited. After set period a message will be displayed and program stops its work;
- All data export modules can handle first 100 records only;

2.3 How to register

The program is distributed on shareware terms. This signifies limited or unavailable many features of the program, getting of full value or available after program registration.

If you'd like to be a registered user, to get information about the release of new versions, to use technical support and, at last, to get access to disabled functions of the program, register your copy. For registration, please, read [license agreement](#)⁴.

If you want to buy a program through the Internet visit the [registration page](#) of our site. On this page you can get the newest information about the registration process, and also find an order link. After you've have the form of order registration. Enter your personal information and choose the most convenient payment method for you. Further, you'll get notification and follow the notes in it.

More information about services, registration documents, payment means you can get on our [registration page](#) of our site.

2.4 Support

Technical questions	support@aggsoft.com
Common questions	info@aggsoft.com
Sales questions	sales@aggsoft.com

3 Installation

3.1 System requirements

Windows 2000 Professional - Windowsa 8.1, including x64 and x86 OS, Workstation and Server OS.

3.2 Installation process

If any beta-version was installed on your computer, remove it.

Quit of the working Data Logger Suite on installation time.

Run an installation file.

By default, Data Logger Suite will be installed to the directory "/Programs Files/Data Logger Suite" of your system disk, but you can change this path.

In the standard distributive of Data Logger Suite are no additional modules files, which you can download from our [site](#).

4 Program use

4.1 Getting started

After you have successfully installed Data Logger Suite, use the following simple steps to configure and run it.

Open the Data Logger Suite program from the Start Menu.

At program run you get into the main program window (fig. 1.1.1), main elements of which are the main menu, the data window, the program messages list and the status bar. In the data window will be viewed formatted data processing. In the messages list are logged information, warning and error messages. The status bar shows current state of the selected data source, interface error messages and a number of bytes processed. Through the main menu, placed above the data window, you can get access to [program settings](#) ("Options/Program settings...") and from the menu "File" (fig. 1.1.2) can open a current log-file or clean the data window.

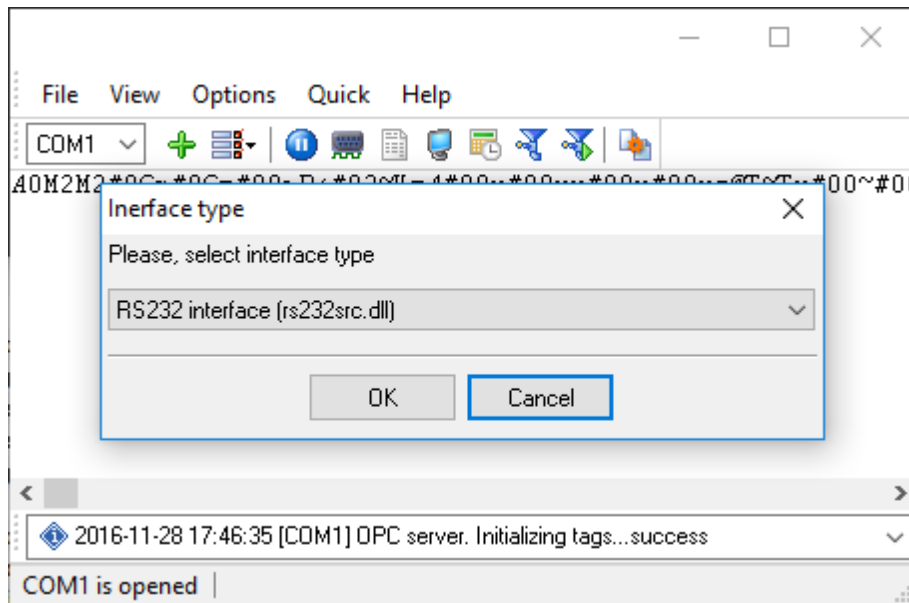


Fig. 1.1.1 Main program window

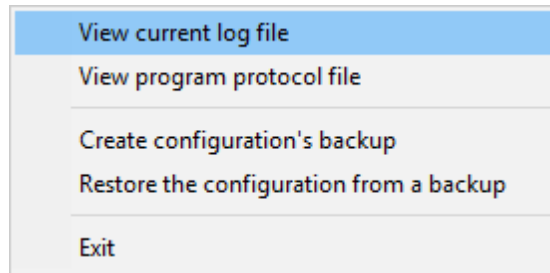


Fig. 1.1.2. "File" menu item

By default (after installation), the program has not any data sources configured. If the list of data sources on the toolbar is empty, then the program will ask you to add new configuration. Otherwise, the program will fill in the list of data sources and try to start logging of data sources configured. Yes, of course, all your settings are being saved while exiting from the program and loaded while the program start.

Set-Up is as Easy as 1-2-3

Step 1. Configure one or more data sources.

Click the "Add configuration" button on the toolbar with big green plus and choose . The "Data source" tab of the "Configuration options" dialog lets you configure your settings.

Step 2. Configure log file.

Select the "Log file" header in the configuration dialog window and enable logging for a necessary data direction.

Step 3. Define how you want the your data to be filtered and exported [parsed and translated](#)^[100].

The "Plug-in" button on the toolbar in the main window or "Modules" tab in the dialog window lets you specify how to parse, filter and format your data to the fit the exact format required by your application. It also lets you pre-define automatic output strings to be sent to an external device.

Now, the program process and exports data from one or multiple data sources.

4.2 Introduction

Program can work with any kind of serial devices. Before configuring our software, the following conditions should be executed:

- Device should have a RS-232 serial port interface (can be also used a RS-485 interface with an additional hardware converter);
- Device is configured to send data to serial port with or without requests from a PC side;
- You know all information about serial port parameters of your device (If your device uses hardware or software flow control (please, read your device's data sheet), then you should know about flow control type);

- Device's serial port is connected with computer serial port with a cable (null-modem or other special cable);
- Computer's COM port, to which your device is connected is not busy, for example by mouse software driver.

How to configure port parameters, you can read in the next chapter "[Serial port settings](#)"^[12].

Program can work with any kind of network interface cards (NIC). Before configuring our software, the following conditions should be executed:

- If your PC has more than one network interface card (NIC) then Data Logger Suite will display a list of all the IP addresses for each NIC installed in your system so that you can select the IP Address that you want to use. In order for Data Logger Suite to act as a server, the PC that it is running on must have at least one network interface card with an IP address assigned to it;
- If Data Logger Suite will work as a server and your PC receives the IP address dynamically from a DHCP server, then you should ask your network administrator to assign a static IP address to your PC.

How to configure port parameters, you can read in the next chapter "[IP settings](#)"^[17].

Data Logger Suite can save data to a log file(s) without any changes (i.e. create raw binary log files) or write to log files depending on the parser module selected. In the first case you can view the log file with any hex editor and use this data for further analysis and remaking. In the second case you can view data with any text editor. You can find more information about log files in the "[Log rotation](#)"^[95] chapter.

You can watch the data in the data window ([fig. 1.1.1](#)^[7]). The data view is fully customizable. You can watch data in decimal, hexadecimal or your own format. How to customize data view you can read in the "[Data view](#)"^[91] chapter and how to customize program view you can read in the "[Window view](#)"^[107] chapter.

The data can be exported or transferred to one or more targets. Most simple way is to configure the log file rotation. But it is small part of all features of Data Logger Suite. Data Logger Suite has many [additional modules](#)^[100] (so-called plug-ins), that are appreciably extending possibilities of the logging software. You can download and install any module supported. Most modules are free of charge for our customers. How to install and configure modules you can read in the "[Modules](#)"^[100] chapter.

The program and their plug-ins generates many messages and writes they to the list in the main window ([fig. 1.1.1](#)^[7]) and a protocol file, that you can use for administration of the software. You can configure types of system messages. More information about it you can read in the "[Protocol and errors handling](#)"^[110] chapter.

4.3 Data flow diagram

This diagram may help you to understand the flow of data within our software and a place of each module. All modules are described in following chapters.

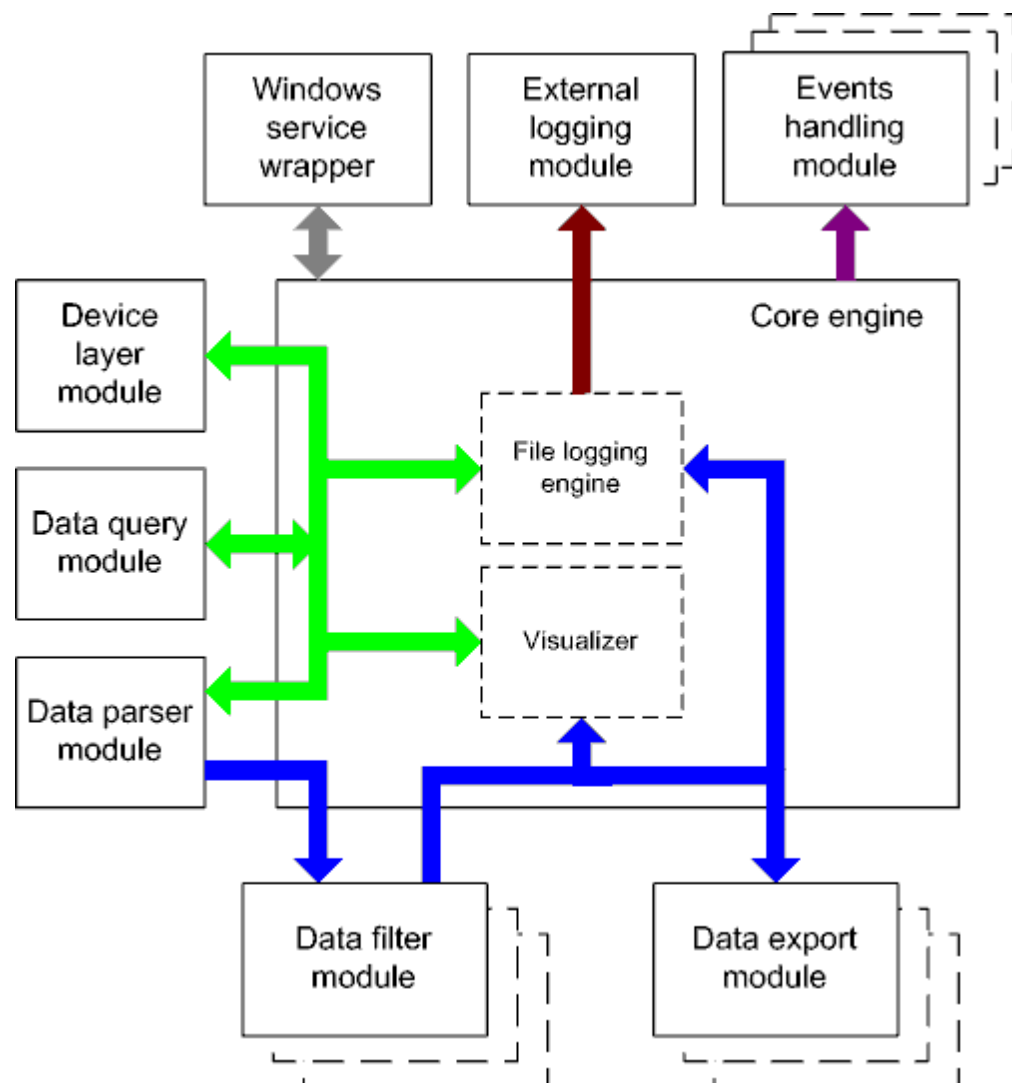


Fig. 1.2.1 Data flow diagram

History:

█ - Binary flow of data (RAW, unformatted data).

█ - Parsed data (formatted data). The data flow was been separated to data packets and variables. Each data packet can be interpreted as a row, and each variable can be interpreted as a column.

Wires with other colors mark other relations with unstructured data flow.

4.4 Work complete

After program work stop all program settings will be saved in Windows registry. Opened for reading/writing data sources will be automatically closed and will be available for other applications.

4.5 Useful advices

1. Look through hint helps on all window elements - this will help you to get a picture of this element's function.
2. You can change all program settings without restarting the program. To transfer settings to another computer you can do the following:
 1. Create a configuration backup from the "File" menu and restore it using the same menu.
 2. Or export the registry node with all program settings. Start regedit.exe and export the following registry node:

on Windows x64

HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\AGG Software\Data Logger Suite

on Windows x32

HKEY_LOCAL_MACHINE\SOFTWARE\AGG Software\Data Logger Suite

3. On another computer import settings to the Windows registry.

Many main window elements have "hot" keys for quick access to its functions.

- Ctrl+S - analogues to click on "Start/Pause" button on toolbar.
- Ctrl+C - analogues to click on "Clear" button on toolbar.
- Ctrl+P - call options window with configuration settings.
- Ctrl+L - call options window with log file settings.
- With Ctrl+W hot key You can configure window view.
- Ctrl+R - show window with program's restart settings.
- Ctrl+E - Windows NT/2000/XP service settings is available here.
- Ctrl+M - here you can configure queries, parsers and other modules.

4. You can look at the summary statistic that contains summary about sent and received data, created files etc (View - Statistics)
5. You can save program settings to an INI file. It may help to install and use several copies of the program. You can make your choice from the "Options" menu.
6. The program window can display only last 10 message. The full program log file (if activated) you can open using the "File - View program protocol file" menu item.

4.6 Configuration

4.6.1 Serial port

4.6.1.1 Serial (COM) port

COM port is short for a serial communication port. Most serial communication software communicate with a computer through a communication port, and most IBM and IBM-compatible computers support up to four serial ports COM1, COM2, COM3 and COM4. Additional ports can be added by adding additional hardware.

Data Logger Suite can manipulate with many serial ports in the same time (up to 255 serial ports).

You can open serial ports in Data Logger Suite software in two modes:

1. **Spy mode.** In this mode the program monitor data flow on ports selected. In this mode Data Logger Suite intercept all data exchange between any Windows application and external device;
2. **Standard.** In this mode the program opens a serial port through Windows API functions, and read/write data from/to a serial port as a regular Windows application. In this mode opens a serial port with exclusive rights and other application will not have access to a serial port.

If one or more port are configured already, then Data Logger Suite is opening these ports and starting logging. If the port is opened successful, then the status bar in the main window displays a status of this port ([fig. 1.1.1](#)⁷). But, before you should configure serial port parameters. For minimization of configuration we combined serial ports with same settings to the "Configuration". The configuration can include one or more serial ports with **identical** settings. For example, if you have many identical devices, that connected to different serial ports, then you can specify port numbers in one configuration only. But, if you want to use serial port with different settings, then you should create more than one configurations.

You can create the new configuration by clicking the "Plus" button in the main window ([fig. 1.1.1](#)⁷) or through the "Options" menu. After you clicked the "Plus" button, the dialog window will be opened ([fig. 2.1.2](#)). The dialog window contains few sections with parameters. The "COM port" section is described in this chapter.

You can manage the configuration created with a drop down menu near the "Plus" button ([fig. 2.1.1](#)).

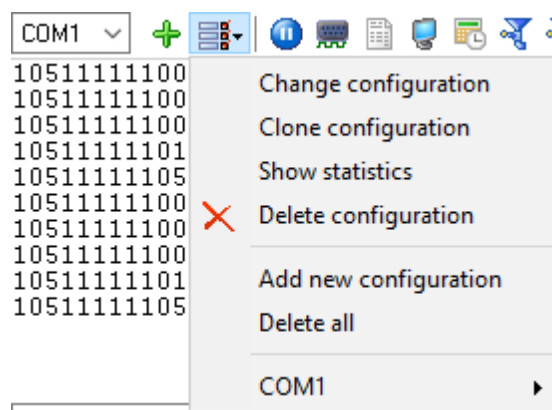


Fig. 2.1.1 Access to the port configuration

The "COM port settings" tab contains indispensable settings of any serial port: baud rate, data bits, etc. You should configure it with same values, that your external device uses for data exchange.

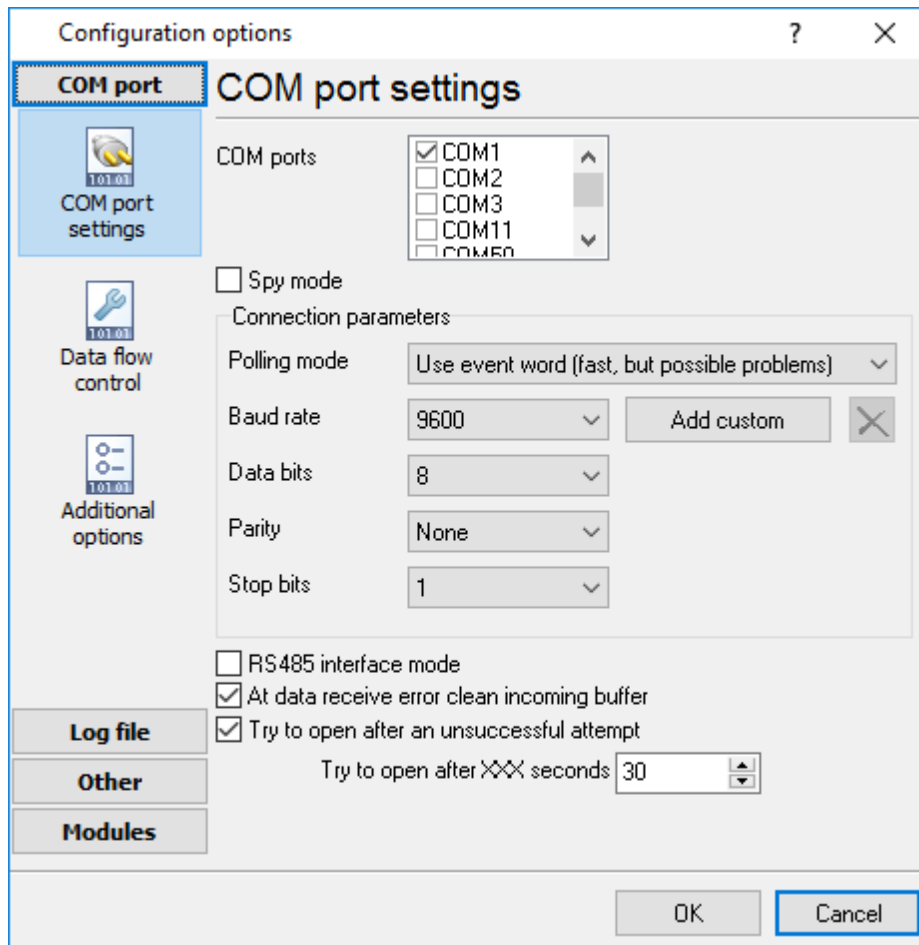


Fig. 2.1.2 COM port parameters

If you are logging data over RS-485 with an additional hardware converter and your converter doesn't support data direction auto detection, then specify **"RS485 interface mode"**. This option instructs Data Logger Suite to set the RTS line at low level while data receiving and vice versa. The serial port driver can detect errors while data receiving (for example, bad quality of a connection line). You can specify with the **"At data receive error clean incoming buffer"** option to ignore data blocks, that contain errors and clean an incoming buffer.

In some cases the program can't open a serial port while starting (for example, the port is already used by other application). With the **"Try to open after an unsuccessful attempt"** option you can specify to try to open the serial port again after the interval specified. The program will try to open the serial port until an attempt will be successful.

Check line status mode

The Windows communication API provides two methods to check for received data and line/modem status changes: API calls (polling) and an event word. The event word is maintained by the Windows communications driver. As data is received or line/modem status changes occur, the driver sets bits in the event word. The application can check the bits to determine if any communication events

occurred. If so, the application can make the appropriate API call to clear the event word and retrieve the data or the new line/modem status values.

Windows also provides API calls to retrieve the same status information provided by the event word but the API calls are slower. Data Logger Suite uses the event word by default for the fastest possible performance. Unfortunately, there is at least one communication driver (WRPI.DRV, included with some U.S. Robotics modems) that doesn't appear to support the event word. For this and similar drivers, select other mode before Data Logger Suite will receive data.

To rise data transmit adequacy you can use hardware and/or software data flow control (fig. 2.1.3). When using hardware data flow control are used some lines (wires) of connecting cable. Depending on used lines, you must setup checks against corresponding fields.

Hardware flow control

When the hardware flow control options are an empty, as they are by default, there is no hardware flow control. The options can be combined to enable hardware flow control.

"Receive flow control" stops a remote device from transmitting while the local input buffer is too full. "Transmit flow control" stops the local device from transmitting while the remote input buffer is too full.

Receive flow control is enabled by including the "Use RTS" and/or "Use DTR" elements in the options. When enabled, the corresponding modem control signals (RTS and/or DTR) are lowered when the input buffer reaches the 90% size of the buffer. The remote must recognize these signals and stop sending data while they are held low.

As the application processes received characters, buffer usage eventually drops below the 10% size of the buffer. At that point, the corresponding modem control signals are raised again. The remote must recognize these signals and start sending data again.

Transmit flow control is enabled by including the "Require CTS" and/or "Require DSR" elements in the options. With one or both of these options enabled, the Windows communications driver doesn't transmit data unless the remote device is providing the corresponding modem status signal (CTS and/or DSR). The remote must raise and lower these signals when needed to control the flow of transmitted characters.

Note that flow control using RTS and CTS is much more common than flow control using DTR and DSR.

Software flow control

This routine turns on one or both aspects of automatic software flow control based on the value assigned to the property.

"Receive flow control" stops a remote device from transmitting while the local receive buffer is too full. "Transmit flow control" stops the local device from transmitting while the remote receive buffer is too full.

Receive flow control is enabled by assigning "On receiving" or "Both" to the "Type" property. When enabled, an XOff character is sent when the input buffer reaches the level 10% size of the buffer. The remote must recognize this character and stop sending data after it is received.

As the application processes received characters, buffer usage eventually drops below the level 10% of the buffer. At that point, an XOn character is sent. The remote must recognize this character and start sending data again.

Transmit flow control is enabled by assigning "On transmitting" or "Both" to the "Type" property. The 10% and 90% size of the buffer are not used in this case. When transmit flow control is enabled, the communications driver stops transmitting whenever it receives an XOff character. The driver does not start transmitting again until it receives an XOn character or the user sets software flow control to "None".

Software data flow control can be setup on receive, transmit or both modes, but so as the great number of device doesn't need data sending, select only control mode "On receive". In case of activation of data transmit control remote object (in our case your device) can send special codes, signaling about data transmit stop or start. On default, received from device character 0x11 Hex signalizes to COM port driver to start data receive and character 0x13 Hex - to stop data receive from device.

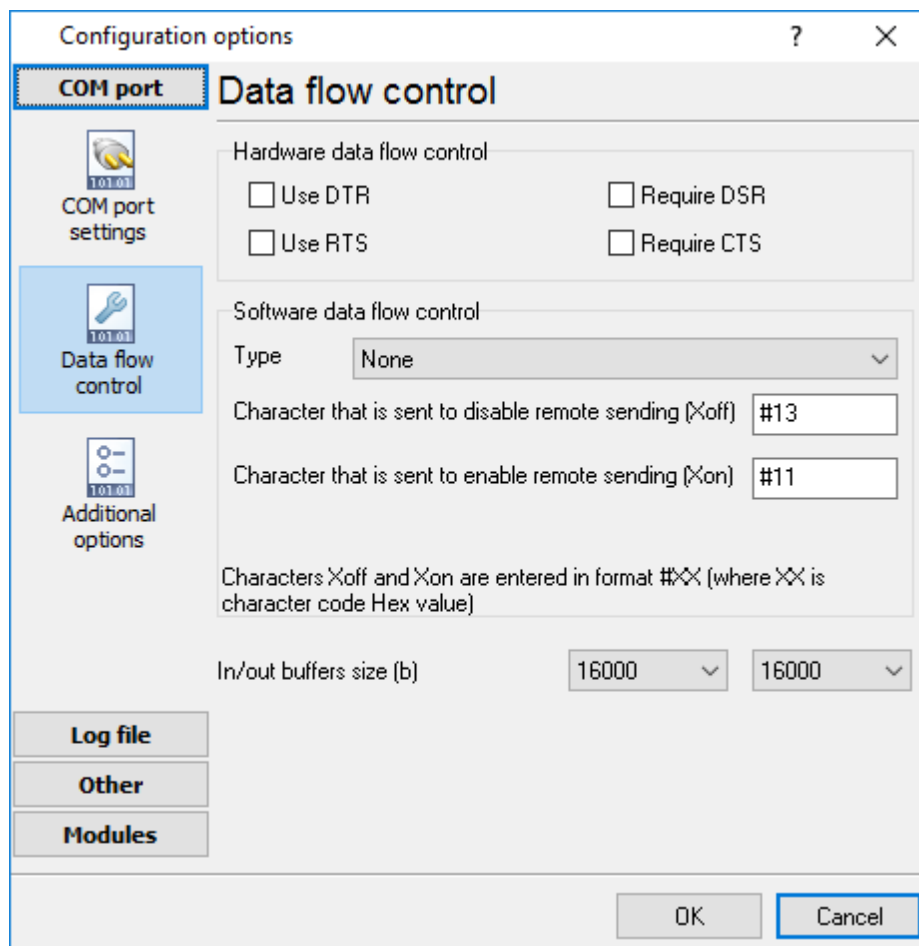


Fig. 2.1.3 Data flow control

Spy mode

In this mode Data Logger Suite doesn't send and receive any data, and only spies data exchange,

made by other programs.

To spy received and sent data open COM port before running the given program. If the given program receives data over COM port, the data exchange process will be displayed in data receive window. Don't forget to set up check box "Spy mode" to spy data receive by the given program (if necessary).

To exit Data Logger Suite close the given program or stop data exchange over COM port in it.

you must close, which data exchange you spies, *before* closing Data Logger Suite.

Serial data transfer errors

Line errors can occur during data exchange and displayed in the main program window in the status bar.

UART receiver parity error - occurs if you configured invalid parity type;

UART receiver overrun,

UART receiver framing error - occurs if you configured invalid number of stop or data bits;

transmit timeout waiting for CTS,

transmit timeout waiting for DSR,

transmit timeout waiting for RLSD - occurs if you configured invalid hardware flow control or your serial interface cable isn't wired for hardware flow control

transmit queue is full - occurs if Data Logger Suite can't send data to remote device;

break condition received

Port restart

You can also set the program to initiate the serial interface at the specified time. On some old versions of the Windows NT operating system it could help to avoid the loss of data when the program has been working for a long time without restarting. Please use the "Additional options" tab (fig. 2.1.4)

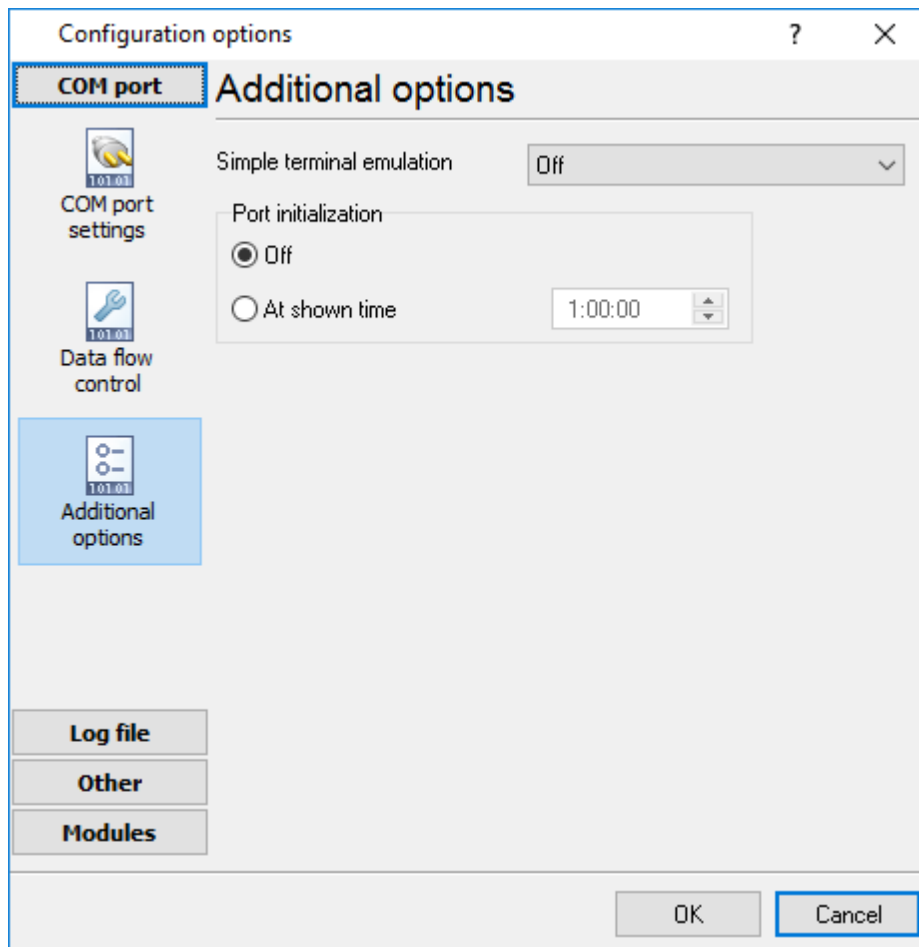


Fig. 2.1.4 Additional options

Here you can also select the terminal emulation mode. In this mode the program will remove or interpret some special terminal sequences automatically.

4.6.2 Network connection

4.6.2.1 TCP/IP settings

UDP vs. TCP

The most commonly used network protocols today are TCP (Transport Control Protocol) and UDP (User Datagram Protocol). TCP is a proven and reliable protocol, and probably the most widely implemented protocol in use on IP networks today. However, TCP has a lot of overhead and payload issues, and can sometimes be 'too-reliable' or robust for many applications. In fact, when used as transport, for many serial based applications TCP can actually hinder reliable communications. In contrast, UDP is a much simpler protocol, and is being used more frequently today - particularly in areas where bandwidth or throughput is constrained. An example is the predominant use of UDP for transport of wireless data applications.

UDP is first a connectionless protocol. Like TCP, UDP runs on top of IP networks. But unlike TCP, UDP does little to help with transport delivery or error recovery. Instead it offers a direct way to send

and receive packets, letting the software application manage things like error recovery and data retransmission. Once primarily used for broadcasting small messages, UDP is now used for everything from browsers to Instant Messaging, Video, and Voice over IP applications.

While a powerful tool, the downside to using UDP is that there is not 'connection' report to know that you have end-to-end connectivity. This often makes detecting whether or not a packet is 'making it' from one place to another quite a hassle.

Client vs Server

Data Logger Suite can be configured to log data from as many ports that you like simultaneously on a single PC. The program uses a multi configurations. Each configuration may contain different settings for each TCP/IP port. Each configuration has a set of TCP/IP parameters that are described below.

Each port configuration (i.e. TCP/IP connection) in Data Logger Suite can act as:

1. **Client.** You will need to specify the remote host IP address and the port number for the TCP/IP server that you want to connect to. The IP address that you specify in Data Logger Suite when configuring it as a client may also be either a URL or the name of a computer located on your network. For example, if you want to connect to a computer named "Plant1", you can simply enter "Plant1" for the IP address instead of the actual IP address. If you are configuring Data Logger Suite as a client and your network is set up to assign IP addresses dynamically to each individual workstation, then you may need to use the name of the PC that you want to connect to instead of an actual IP address in order to guarantee a connection;
2. **Server.** In this mode you should specify the IP address of the local PC will be used and you only need to specify the port number that you would like to use. If your PC has more than one network interface card (NIC) then Data Logger Suite will display a list of all the IP addresses for each NIC installed in your system so that you can select the IP Address that you want to use. In order for Data Logger Suite to act as a server, the PC that it is running on must have at least one network interface card with an IP address assigned to it. In Microsoft Windows, the TCP/IP protocol can be configured to automatically obtain an IP address from a host computer. This means that your PC may not have an IP address until it is connected to a network server or a host computer. You may need to contact your network administrator to assign an IP address to your PC if you wish to configure a TCP/IP server connection. This is done in the network settings for the TCP/IP protocol in your control panel.

After you enter the parameters that you would like to use, you must click the "OK" button to establish a connection between Data Logger Suite and the TCP/IP port. If the current port configuration is set up as a client, it will immediately try to establish a connection to the specified remote server. If the server is not available, Data Logger Suite will continually try to establish the connection until it is successful. If the port configuration is set up as a server, it will listen the specified port until a client establishes a connection to it.

If one or more port are configured already, then Data Logger Suite is opening these ports and starting logging. If the port is opened successful, then the status bar in the main window displays a status of this port ([fig. 1.1.1](#)⁷). But, before you should configure port parameters that are described below.

You can create the new configuration by clicking the "Plus" button in the main window ([fig. 1.1.1](#)⁷) or through the "Options" menu. After you clicked the "Plus" button, the dialog window will be opened ([fig. 2.2.2](#)). The dialog window contains few sections with parameters. The "IP settings" section is

described in this chapter.

To log data from more than one TCP/IP connection you would create and configure multiple port configurations. You can manage the configuration created with a drop down menu near the "Plus" button (fig. 2.2.1).

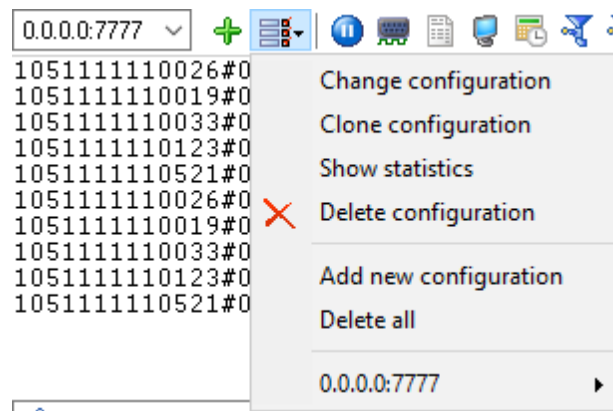


Fig. 2.2.1 Access to the port configuration

The "IP settings" tab contains indispensable settings of any TCP/IP port: IP address and port.

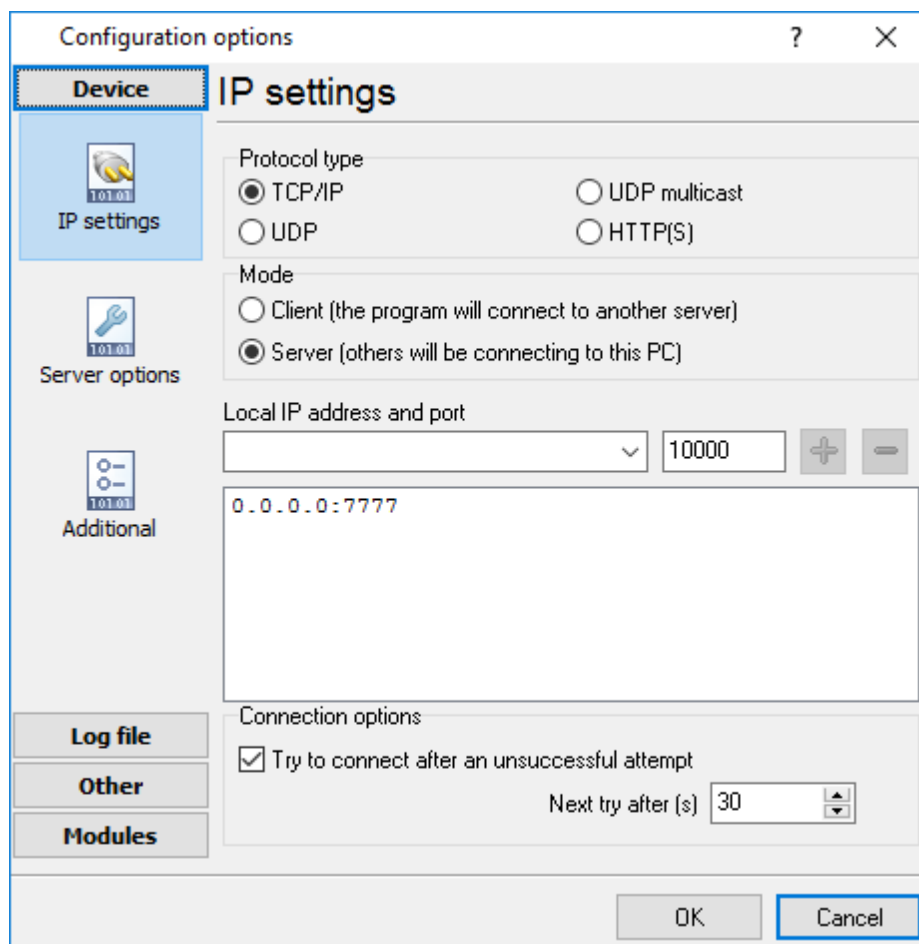


Fig. 2.2.2 TCP/IP parameters

Port

In addition to IP address, you should specify how to connect to a remote machine. Our software can be thought of as a trunk line with thousands of individual lines (the ports) which are used to connect machines. Some ports are considered well-known ports. For example, the port typically used for network mail systems (SMTP) is port 25, the telnet port is port 23, the network news server port (NNTP) is typically port 119, and so on. To see a list of well-known ports, inspect the SERVICES file in the Windows directory (for Windows NT it is in the WINNT\SYSTEM32\DRIVERS\ETC directory). The SERVICES file is a text file used by Data Logger Suite to perform port lookups (which return the service name for the specified port) and port name lookups (which return the port number for the specified service name). You can open this file in any text editor to see a list of port numbers and their corresponding service names. While these well-known ports are not set in stone, they are traditional and their use should be reserved for the service which they represent. When writing network applications, you should select a port number that is not likely to be duplicated by other applications on your network. In most cases you can choose a port number other than any of the well-known port numbers.

The IP address and port number are used in combination to create a socket. A socket is first created and then is used to establish connection between two computers. How the socket is used depends on whether the application is a client or a server. If an application is a server, it creates the socket, opens it, and then listens on that socket for computers trying to establish a connection. At this point the server is in a polling loop listening and waiting for a possible connection. A client application, on the other hand, creates a socket using the IP address of a particular server and the port number that the server is known to be listening on. The client then uses the socket to attempt to connect to the server. When the server hears the connection attempt, it wakes up and decides whether or not to accept the connection. Usually this is done by examining the IP address of the client and comparing it to a list of known IP addresses (some servers don't discriminate and accept all connections). If the connection is accepted, the client and server begin communicating and data is transmitted.

Connection options

If the remote server (in the client mode) or local network interface (in the server mode) is not available and the **"Try to connect after unsuccessful attempt"** options is True, then Data Logger Suite will continually try to establish the connection until it is successful. The program will try to establish the connection each N seconds that you can specify in the **"Next try after XXX seconds"** field.

Allowed IP addresses

This option is active in the server mode and allows you to enter one or more IP addresses that have access to the server. IP addresses that are not listed in this fields will be refused by the server. This options is very useful if you transfer your data over Internet connection or your server PC is connected to a big corporate network. You can specify multiple addresses - one per row. If you'll not specify any address here, then Data Logger Suite will accept connections from all IP addresses.

Firewall settings

After you install Microsoft Windows XP Service Pack 2 (SP2), our Data Logger Suite may not seem to work. Windows Firewall, enabled by default, blocks unsolicited access to your computer via the network and may be blocking the normal operation of the program. To provide increased security to

Windows XP users, Windows Firewall blocks unsolicited connections to your computer. When Windows Firewall detects incoming network traffic that it does not recognize, a Security Alert dialog box appears. The security alert dialog box looks like this:



Fig. 2.2.3 Firewall alert

The dialog box includes the following buttons:

- Unblock this program;
- Keep Blocking this program;
- Keep blocking this program, but Ask Me Later.

In order for our program to function properly, you must unblock the program by clicking the Unblock button. Unblocking allows Windows XP SP2 to allow the program to continue to work by adding it as an exception to your Windows Firewall configuration. Exceptions are specific programs and processes that you allow to bypass the firewall. After you add a program as an exception, you no longer receive the security alert. If you choose to continue blocking the program certain functions will be disabled.

Note: If you are using another firewall software, then please, refer to a firewall manual for corresponding settings.

Limitations

The specific limit of connections is dependent on how much physical memory your server has, and how busy the connections are:

The Memory Factor: According to Microsoft, the WinNT and successor kernels allocate sockets out of the non-paged memory pool. (That is, memory that cannot be swapped to the page file by the

virtual memory subsystem.) The size of this pool is necessarily fixed, and is dependent on the amount of physical memory in the system. On Intel x86 machines, the non-paged memory pool stops growing at 1/8 the size of physical memory, with a hard maximum of 128 megabytes for Windows NT 4.0, and 256 megabytes for Windows 2000. Thus for NT 4, the size of the non-paged pool stops increasing once the machine has 1 GB of physical memory. On Win2K, you hit the wall at 2 GB.

The "Busy-ness" Factor: The amount of data associated with each socket varies depending on how that socket's used, but the minimum size is around 2 KB. Overlapped I/O buffers also eat into the non-paged pool, in blocks of 4 KB. (4 KB is the x86's memory management unit's page size.) Thus a simplistic application that's regularly sending and receiving on a socket will tie up at least 10 KB of non-pageable memory.

The Win32 event mechanism (e.g. `WaitForMultipleObjects()`) can only wait on 64 event objects at a time. Winsock 2 provides the `WSAEventSelect()` function which lets you use Win32's event mechanism to wait for events on sockets. Because it uses Win32's event mechanism, you can only wait for events on 64 sockets at a time. If you want to wait on more than 64 Winsock event objects at a time, you need to use multiple threads, each waiting on no more than 64 of the sockets.

If you have more than 64 connection at a time, then we recommend to create multiple configuration in our software (the green plus button). Each configuration will use different port number and will run in a different thread. This change will allow to decrease an influence of Windows limitations.

Additional parameters

The "Additional" tab contains additional settings of a TCP/IP or UDP connection (fig. 2.2.4).

Simple terminal emulation - the program realizes simple implementation of some terminal protocols. If this emulation is enabled then the program will process some special commands and character sequences.

Following options are effective only in the "TCP/IP server" mode:

Limit of simultaneous connections - you can define number of clients that can connect to the server at the same time. It allows to optimize a server load with large number of TCP clients.

Disconnect inactive clients after (s) - if a client is connected, but didn't send or receive any data within the specified time, then the connection with this client will be closed. If you will specify "-1" then the clients will not be disconnected.

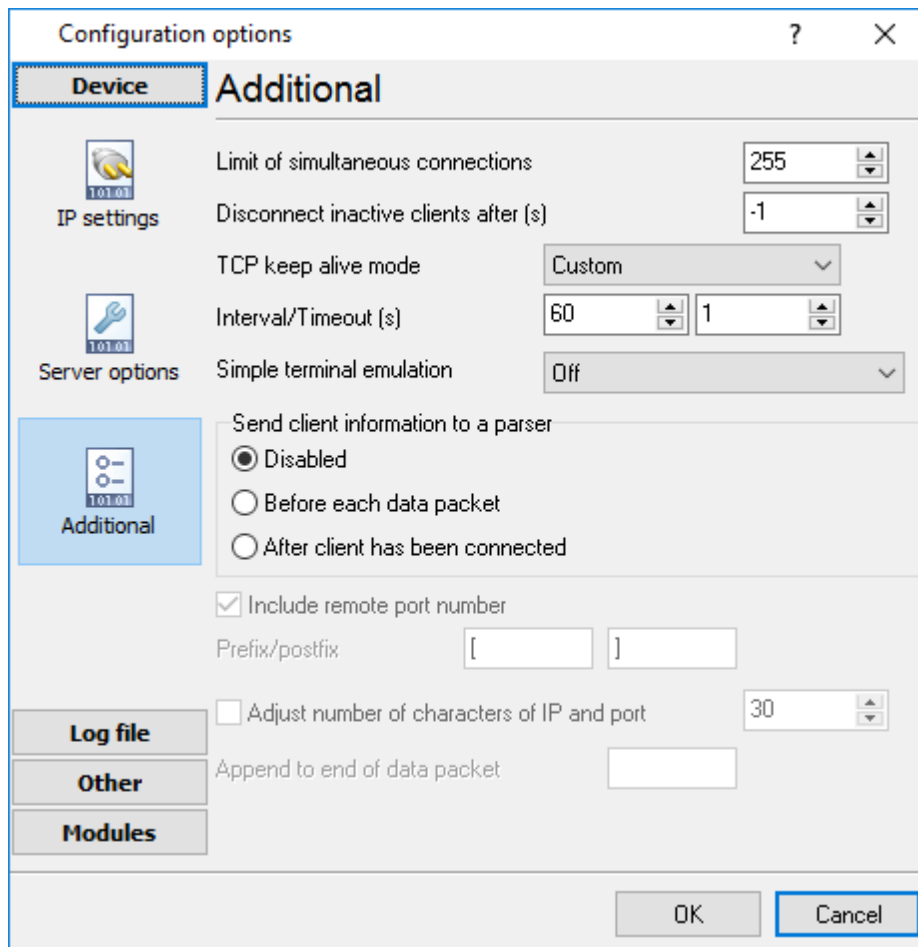


Fig. 2.2.4. Additional parameters

Following options are effective only in the TCP/IP server or client modes:

TCP keep alive mode

A TCP keep-alive packet is a short packet which is sent periodically by the OS to keep the connection alive. The connection stay alive because those packets and their replies generate small traffic on the connection when the application is idle.

Keep-alives can be used to verify that the computer at the remote end of a connection is still available.

It is simply an ACK with the sequence number set to one less than the current sequence number for the connection. A host receiving one of these ACKs responds with an ACK for the current sequence number.

TCP keep-alives can be sent once every KeepAliveTime (defaults to 7,200,000 milliseconds or two hours) if no other data or higher-level keep-alives have been carried over the TCP connection. If there is no response to a keep-alive, it is repeated once every KeepAliveInterval seconds. KeepAliveInterval defaults to 1 second. Some (buggy) routers may not handle keep-alive packets properly.

Our software supports three modes of keep alive (fig 2.2.4):

1. **Off** - the program doesn't use keep alive at all. You can disable keep alive if your network is very stable or your routers doesn't support it.

2. **System** - the program will use keep alive, but use system values of KeepAliveTime and KeepAliveInterval. These values are stored in the following registry branch:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters]  
KeepAliveTime (32-bit number) = milliseconds  
KeepAliveInterval (32-bit number) = milliseconds
```

3. **Custom** - the program will use keep alive, but you can specify your values of KeepAliveTime and KeepAliveInterval, that are more applicable for your network and system. **Note:** in our software you should define these values in seconds.

Note: Some routers may not allow keep-alive TCP packets. In this case the "keep-alive" function will not work.

Following options are effective only in the server mode (TCP/IP or UDP):

Send client information to the a parser - if this option is activated then program will append a system header to a data packet about the remote client. It allows the parser to separate data packets from several clients that send data simultaneously.

4.6.3 Modem settings

4.6.3.1 Modem settings

Basic AT modem characteristics

Modems may have a lot of different characteristics: design, supported data transfer protocols, error correction protocols, voice and fax communication.

1. Modems may differ in their design (appearance, external or internal modems).
2. Modems also differ by types.
3. Modems differ in the data transfer rate as well. It is measured in bps (bit per second) and manufacturers set it to 2400, 9600, 14400, 16800, 19200, 28800, 33600, 56000 bps. The actual data transfer rate depends not only on the bps value. Such characteristics as error correction and data compression also influence it.
4. Hayes-compatible modems, asynchronous modems supporting a set of modem registers and commands, have become the de facto standard nowadays. This standard is based on the support of AT commands.

TAPI – General information

TAPI stands for Telephony Application Programming Interface. TAPI comprises a lot of functions allowing devices transferring data over telephone lines to be programmed in a way independent of a specific device allowing users to communicate by phone. TAPI supports voice and data transfer and provides for a lot of connection types and call management. TAPI allows applications to use the functions of a phone, for example, support conferences and transfer voice mail. Applications rely on existing telephony service providers that provide TSPI (Telephony Service Provider Interface). The

user can install any number of telephony service providers if only they do not access the same device at the same time. The user associates a device with a service provider during installation. Some providers can access several devices. Sometimes the user may need to install the driver of a device together with the provider.

How the program works

After the program is loaded, it analyzes the [task list](#)^[30]. If the task list has at least one active task in it and it is due, it starts performing it. Exception: a task of the "On start" type is activated right after the program is loaded. After the program establishes a connection, it receives/sends data using the established connection. If there is no data and the timeout specified in the "Break the connection" task with the "The module didn't send/receive the data" condition expires, the connection is broken. A connection can also be broken in case the interval specified in the "Break the connection" task with the "After the connection was established" condition expires. Note that the "Break the connection" task with the "The module didn't send/receive the data" condition resets the timeout each time the program receives or sends data. In other cases, a connection is established and broken according to the specified task parameters.

Data exchange is carried out according to a task schedule (see "[Schedule management](#)"^[30]). If there is no TAPI driver for your modem, you can use the AT mode. It is recommended to use the TAPI mode in other cases.

AT modem settings

Select the "Modem settings" tab in the Settings dialog box and select the "AT-modem" device type in the drop-down list on this tab. As a result, you will see AT modem settings. Use the AT settings drop-down list to select one of the following settings:

AT modem general settings:

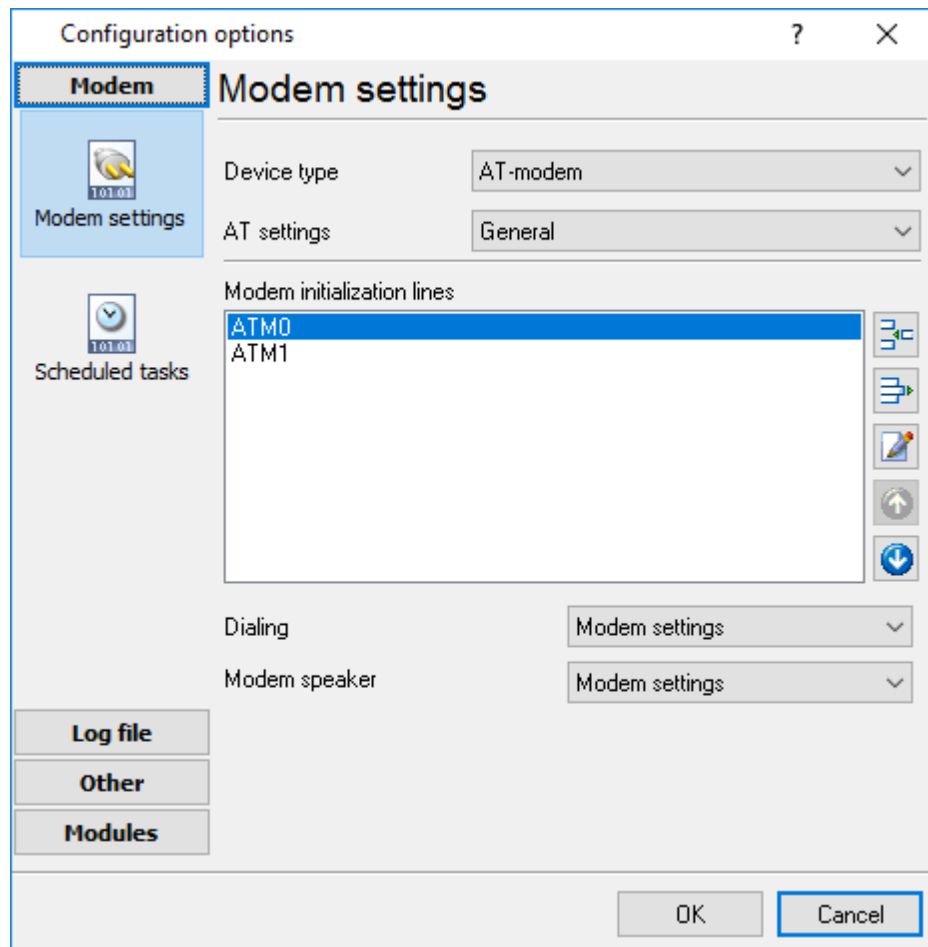


Fig. 2.3.1 AT modem general settings

Modem initialization lines – the list contains the modem initialize lines (a string of not more than 255 characters). The list allows you to add, remove, edit and move modem AT commands. These AT commands are used to initialize the modem before dialing a number. The following buttons for managing the lines are located to the right from the list:



- add a new initialization line to the list. After you click this button, the dialog box where you should enter the initialization line and click the <OK> button appears. If you enter an empty string or the initialization line is already in the list, the corresponding message will appear on the screen.



- remove the initialization line selected in the list. After you click this button, the confirmation dialog box appears. You should click the <Yes> button in this dialog box to remove the line or the <No> button to cancel the operation. If there are no initialization lines, this operation is unavailable.



- edit the initialization line selected in the list. After you click this button, the dialog box where you should edit the initialization line and click the <OK> button appears. If you enter an empty string or the initialization line is already in the list, the corresponding message will appear on the screen. If there are no initialization lines, this operation is unavailable.



- move the selected initialization line one position up. If there are no initialization lines, or if there is only one initialization line, or if the initialization line is the first one in the list, this operation is unavailable.



- move the selected initialization line one position down. If there are no initialization lines, or if there is only one initialization line, or if the initialization line is the last one in the list, this operation is unavailable.

Tone dialing – the checkbox enables/disables tone dialing. It is disabled by default.

Mute modem speaker – the checkbox enables/disables the internal modem speaker. It is disabled by default.

AT modem port settings:

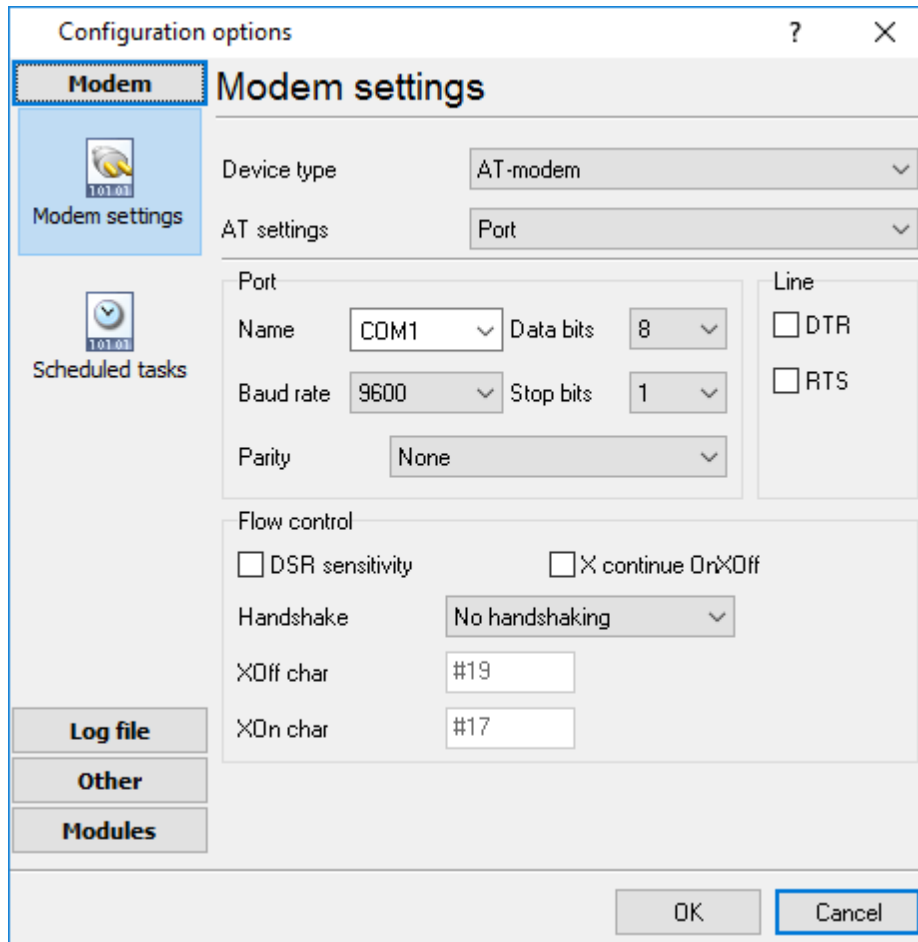


Fig. 2.3.2 AT modem port settings

Name – the list allows you to select the com port the modem uses to connect to the remote terminal. All ports available in the system are displayed in the drop-down list. If there are no ports in the system, the list will be empty. If the port currently selected in the program is removed from the system and the program is started again, the removed port will be highlighted in red. The default is the first port in the list.

Baud rate – the list allows you to select one of the standard data transfer baud rates via the RS232 interface. The default is 9600 baud/sec.

Data bits – the list allows you to select the number of bits in a byte used to receive/send data. Do not use the following combinations: data bits 5 – stop bits - 2; data bits 6, 7, 8 – stop bits 1.5. 8 is selected by default.

Stop bits – the list allows you to select the number of stop bits for one character. Do not use the following combinations: data bits 5 – stop bits - 2; data bits 6, 7, 8 – stop bits 1.5. 1 is selected by default.

Parity – the list allows you to select the parity check. There is no check by default.

DTR – the checkbox enables/disables the DTR signal. It is disabled by default.

RTS – the checkbox enables/disables the RTS signal. It is disabled by default.

DSR sensitivity – if the checkbox is selected, the program ignores all data received from the modem as long as the DSR signal is set. It is disabled by default.

X continue OnXOff – the checkbox enables/disables transferring data when the buffer is full and the driver has sent the XOffChar character. If the checkbox is selected, the transfer will continue when the number of data bytes in the incoming data buffer will be less than XOffLim and the driver sends the XOffChar character. If the checkbox is not selected, transferring data to the program will not be continued till the number of data bytes in the incoming data buffer is greater than or equal to XOffLim and the driver sends the XOnChar character for resuming the data transfer. It is disabled by default.

Handshake – the list allows you to select the flow control type that can be software (XON/XOFF) or hardware (RTS/CTS or DTR/DSR). There is no flow control by default.

XOff char – this field allows you to enter the hexadecimal code of the XOff character (the character of disabling the data transfer) for receiving and sending data. The character code is specified in the #XX format, where # is the code sign, XX is the character code. It is #19 by default.

XOn char – this field allows you to enter the hexadecimal code of the XOn character (the character of enabling the data transfer) for receiving and sending data. The character code is specified in the #XX format, where # is the code sign, XX is the character code. It is #17 by default.

TAPI device settings

Select the "Modem settings" tab in the Settings dialog box and select the "TAPI-device" device type in the drop-down list on this tab. As a result, you will see TAPI settings. Use the TAPI settings drop-down list to select one of the following settings:

TAPI device general settings:

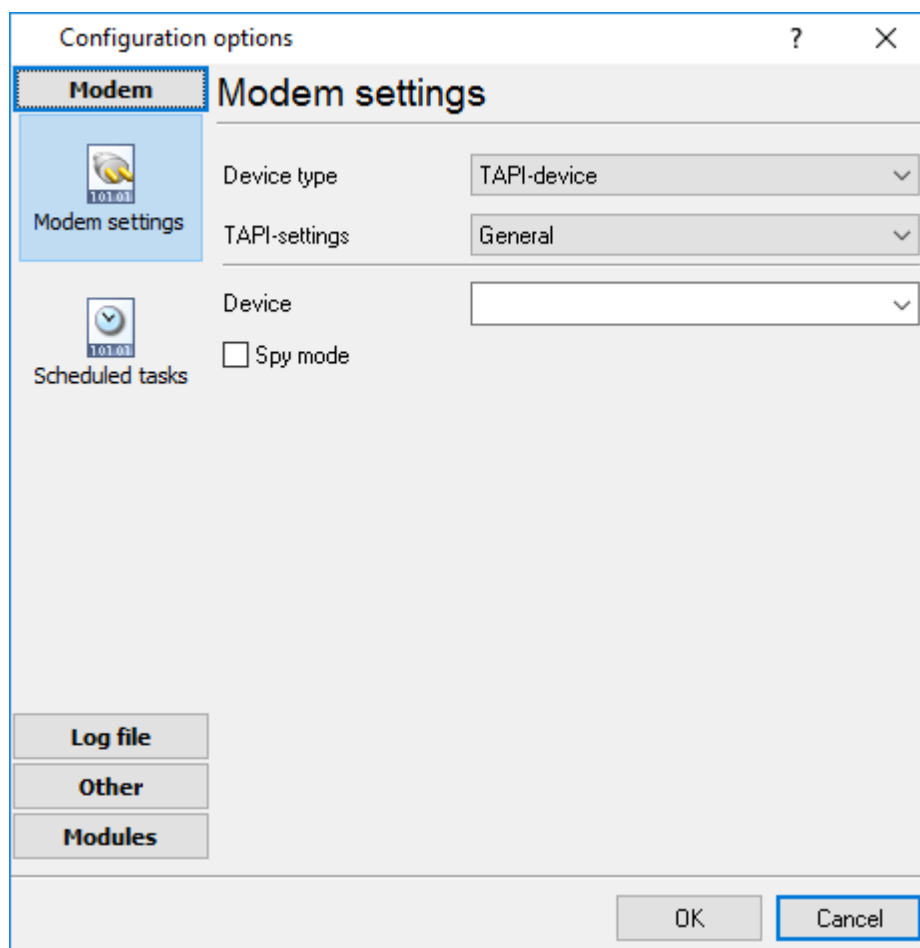


Fig. 2.3.3 TAPI device general settings

Device – the list allows you to select the TAPI device that is used to call the remote terminal. The list shows only devices supporting data transfer. The list may be empty if there are no such devices in the system. If the device currently selected in the program is removed from the system and the program is started again, the removed device will be highlighted in red. The default is the first device in the list.

TAPI device options:

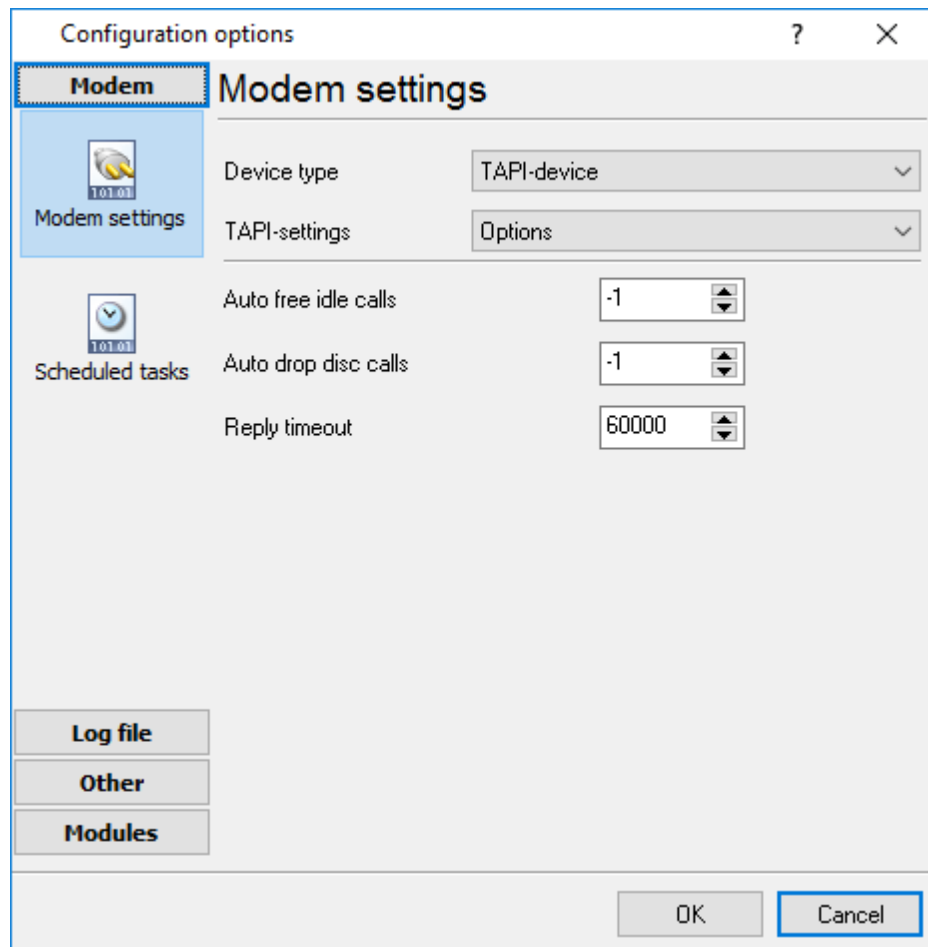


Fig. 2.2.4 TAPI device options

Auto free idle calls – the input field allows you to close connections that have no actions for them (data transfer, etc.) for the specified interval. It is set to -1 by default.

Auto drop disc calls – the input field enables/disables automatically closing a connection upon disconnection. When the connection is switched to the "Disconnected" status, it will be automatically closed after the specified timeout expires (in milliseconds). If the parameter is set to -1, this feature will not be used.

Reply timeout – since some TAPI functions (calling, dialing, etc.) may take quite a while, the program calls these functions in the asynchronous mode (i.e. it does not wait till these functions finish their work, but continues its work). However, the program uses this timeout to control the work of the called operations. If the requested operation (calling, dialing, etc.) is not finished within the specified timeout, it will be canceled and the program will display the corresponding error message.

4.6.3.2 Task schedule

It is possible to establish and break a connection at any time. The task scheduler is used for that. It is necessary to add at least one task establishing a connection for the program to be able to receive/send data. The schedule allows you to add, delete, edit, clone and move tasks. You can also change phone numbers in the task list. You can manage the task schedule on the Scheduled

tasks tab of the Settings dialog box.

Schedule management

Select the "Scheduled tasks" tab in the Settings dialog box, the task list will be displayed on the screen. It is empty by default.

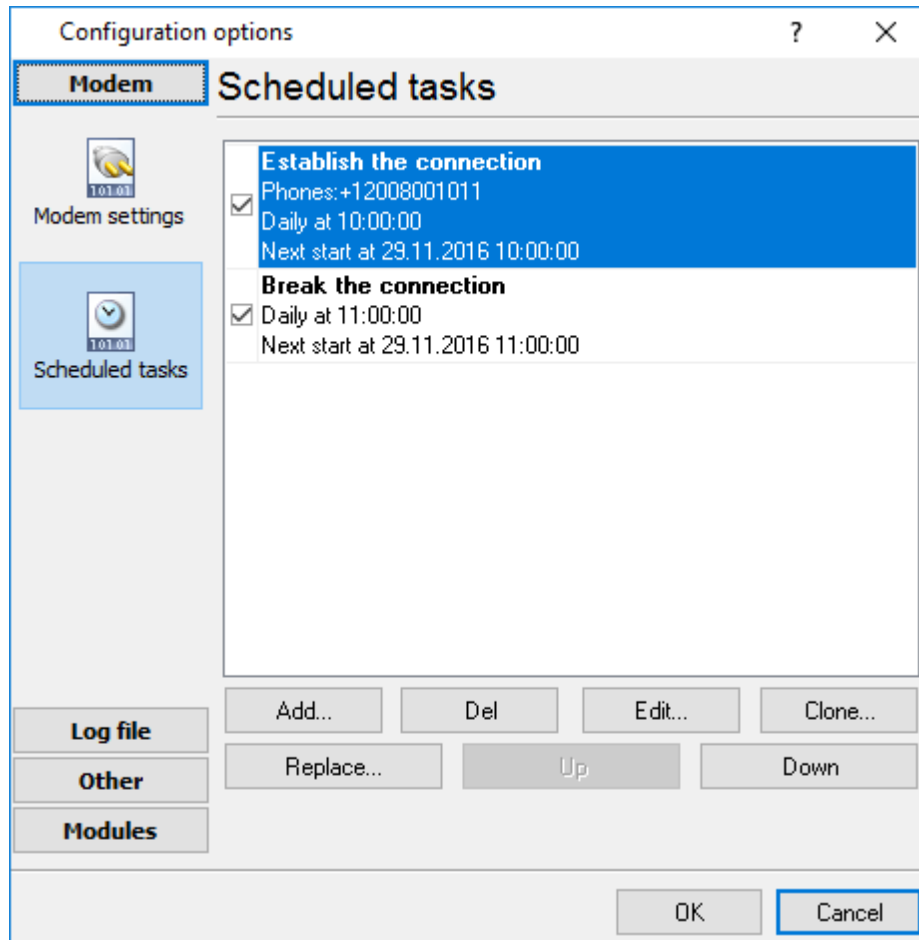


Fig. 2.3.5 Task list

Adding a tasks:

Click the <Add...> button on the Scheduled tasks tab or select the <Add...> item from the context menu of the task list (opened with a right click) and you will see the Task dialog box on the screen. By default, the "Establish the connection" type and the "Daily" condition are specified. Specify the necessary task parameters and click the <OK> button – the task will be added to the list. If you change your mind, click the <Cancel> button. Task parameters depend on the task type and the condition (see the "Task types" section).

Deleting a task:

Click the button on the Scheduled tasks tab or select the item from the context menu of the task list (opened with a right click) and you will see the confirmation dialog box on the screen. Click the <Yes> button and the task will be deleted from the list. If you change your mind, click the <No> button. If there are no tasks in the list, this operation is unavailable.

Editing a task:

Click the <Edit...> button on the Scheduled tasks tab or select the <Edit...> item from the context menu of the task list (opened with a right click) and you will see the Task dialog box on the screen. Change task parameters and click the <OK> button – the task will be saved in the list. If you change your mind, click the <Cancel> button. If there are no tasks in the list, this operation is unavailable.

Cloning a task:

Click the <Clone...> button on the Scheduled tasks tab or select the <Clone...> item from the context menu of the task list (opened with a right click) and you will see the Task dialog box with the parameters of the current task on the screen. Change task parameters and click the <OK> button – the task will be added to the list. If you change your mind, click the <Cancel> button. If there are no tasks in the list, this operation is unavailable.

Replacing task phone numbers:

Click the <Replace...> button on the Scheduled tasks tab or select the <Replace...> item from the context menu of the task list (opened with a right click) and you will see the Replace dialog box on the screen:

Enter the old phone number into the "Old phone" field and the new phone number into the "New phone" field and click the <OK> button – the phone number specified in the "Old phone" field will be replaced with the new phone number in all tasks. If you change your mind about replacing the phone number, click the <Cancel> button. If there are no tasks in the list, this operation is unavailable.

Moving a task up:

Click the <Up> button on the Scheduled tasks tab or select the <Up> item from the context menu of the task list (opened with a right click) and the task will be moved one position up. If the task is the first or only one in the list or if there are no tasks at all, this operation is unavailable.

Moving a task down:

Click the <Down> button on the Scheduled tasks tab or select the <Down> item from the context menu of the task list (opened with a right click) and the task will be moved one position down. If the task is the last or only one in the list or if there are no tasks at all, this operation is unavailable.

Note: you can add any number of tasks with different parameters. The order tasks are executed in depends on their positions in the list, i.e. the first task in the list is executed first, the second one is executed second, etc. The time the task is started next time and the activity state of the task also influence the order the tasks are executed in. If the checkbox of the task is not selected, it will not be executed. To receive/send data, you should add at least one task of the "Establish the connection" type. Tasks of the "Break the connection" type are executed only if there is a connection established, i.e. if at least one task of the "Establish the connection" has been executed. If there are no tasks of this type, but there is a task of the "Break the connection" type, it will not be executed even if it is active, i.e. its checkbox is selected. The user is completely responsible for making up a correct schedule so be attentive while adding new tasks.

Task types

There are two types of tasks implemented in the program: "Establish the connection" and "Break the connection". Each type has its set of conditions.

Conditions for **"Establish the connection"** tasks:

- Daily
- Once

- By week days
- By month days
- Periodically
- On start

Conditions for "**Break the connection**" tasks:

- Daily
- Once
- By week days
- By month days
- Periodically
- After the connection was established
- The program didn't send/receive the data

Parameters in the Task and Confirmation groups:

The screenshot shows a dialog box titled "Task" with a close button (X) and a help button (?). The dialog is divided into three main sections: "Task", "Condition", and "Confirmation".

- Task section:**
 - Task type:** A dropdown menu set to "Establish the connection".
 - Attempts count:** A numeric spinner set to 10.
 - Pause between attempts (sec):** A numeric spinner set to 15.
 - Phones:** A list box containing a single entry "+12008001011" with a checkmark to its left. To the right of the list are icons for adding, deleting, and moving items.
- Condition section:**
 - Condition:** A dropdown menu set to "Daily".
 - Start time:** A text field containing "10:00:00".
- Confirmation section:**
 - Ask confirmation before task execution:** A checked checkbox.
 - Timeout (sec):** A numeric spinner set to 20.

At the bottom of the dialog are two buttons: "OK" (with a green checkmark icon) and "Cancel" (with a red X icon).

Fig. 2.3.7 Task parameters

Task type – the list allows you to select one of the two task types: "Establish the connection" or "Break the connection". The default is "Establish the connection".

Attempts count – the field contains the number of attempts to execute the task. A new attempt starts when, for example, an error during calling, connecting, etc. occurs. The maximum number of attempts is 500. The default is 10.

Pause between attempts (sec) – specify the pause (in seconds) between the attempts. The maximum is 3600 seconds. The default is 15 seconds.

Phones – the list contains task phone numbers the program uses to dial the remote terminal. You can enter any number of phone numbers, but there should be at least one phone number in the list, otherwise the task will not be added to the list or the parameters of the task will not be updated. There is a checkbox next to each phone number for you to make it active. There should be at least one active phone number in the list, otherwise the dialog box will not be closed when you click the <OK> button. The program dials only active phone numbers from the list of the task being executed. The following buttons for managing phone numbers are on the right:



- add a new phone number to the list. After you click this button, the dialog box where you should enter the phone number and click the <OK> button appears. If you enter an empty string or the phone number is already in the list, the corresponding message will appear on the screen.



- remove the phone number selected in the list. After you click this button, the confirmation dialog box appears. You should click the <Yes> button in this dialog box to remove the phone number from the list or the <No> button to cancel the operation. If there are no phone numbers, this operation is unavailable.



- edit the phone number selected in the list. After you click this button, the dialog box where you should edit the phone number and click the <OK> button appears. If you enter an empty string or the phone number is already in the list, the corresponding message will appear on the screen. If there are no phone numbers, this operation is unavailable.



- move the selected phone number one position up. If there are no phone numbers or if the selected phone number is the only or first one in the list, this operation is unavailable.



- move the selected phone number one position down. If there are no phone numbers or if the selected phone number is the only or last one in the list, this operation is unavailable.

Ask confirmation before task execution – the checkbox enables/disables confirmation before the task is executed. If the checkbox is selected and the task is due, a dialog box will appear on the screen. You should click the <Yes> button to execute the task or the <No> button to cancel the task. If the Timeout (sec) field contains 0, the dialog box will be on the screen until the user makes the choice, otherwise the time left before the dialog box is closed is displayed on the title bar of the dialog box and if the user does not click the <No> button within this timeout, the task will be executed automatically. The checkbox is selected by default.

Timeout (sec) – the field contains the timeout (in seconds) for the program to wait for the answer from the user after the confirmation dialog box appears on the screen. If you set it to 0, the program waits till the user makes the choice, otherwise the dialog box will be closed after the timeout expires and the task will be executed. The default is 20 seconds.

Daily condition:

The screenshot shows a 'Task' dialog box with the following fields and controls:

- Task type:** A dropdown menu set to 'Establish the connection'.
- Attempts count:** A numeric input field set to '10'.
- Pause between attempts (sec):** A numeric input field set to '15'.
- Phones:** A list box containing a single entry '+12008001011' with a checkmark to its left. To the right of the list are four icons: a list with a plus, a list with a minus, a list with a pencil, and a list with a trash can.
- Condition:** A dropdown menu set to 'Daily'.
- Start time:** A time input field set to '10:00:00'.
- Confirmation:** A checkbox labeled 'Ask confirmation before task execution' which is checked.
- Timeout (sec):** A numeric input field set to '20'.
- Buttons:** 'OK' (with a green checkmark icon) and 'Cancel' (with a red X icon).

Fig. 2.3.8 Daily condition

Start time – the field contains the time when the task should be executed. If the time is specified incorrectly, an error message will appear on the screen. The default is the current time.

Note: the task is executed every day at Start time. The task must be active in the task list, otherwise it will not be executed.

Once condition:

Task

Task type: Establish the connection

Attempts count: 10 Pause between attempts (sec): 15

Phones: ☒ +12008001011

Condition

Condition: Once

Start time: 14:55:27 Date: 28.11.2016

Confirmation

☒ Ask confirmation before task execution Timeout (sec): 20

OK Cancel

Fig. 2.3.9 Once condition

Start time – the field contains the time when the task should be executed. If the time is specified incorrectly, an error message will appear on the screen. The default is the current time.

Date – the field contains the date when the task should be executed. If you specify an invalid date, you will see an error message on the screen after you click the <OK> button. The default is the current date.

Note: the task is executed once at Start time on Date. The task must be active in the task list, otherwise it will not be executed.

By week days condition:

The screenshot shows a 'Task' dialog box with the following fields and options:

- Task type:** A dropdown menu set to 'Establish the connection'.
- Attempts count:** A numeric spinner set to 10.
- Pause between attempts (sec):** A numeric spinner set to 15.
- Phones:** A list box containing a checked entry '+12008001011'.
- Condition:** A dropdown menu set to 'By week days'.
- Start time:** A text field containing '14:55:27'.
- Interval:** A numeric spinner set to 1, followed by a dropdown menu set to 'Hours'.
- Week days:** A list box with checkboxes for Monday (checked), Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Confirmation:** A checked checkbox labeled 'Ask confirmation before task execution'.
- Timeout (sec):** A numeric spinner set to 20.
- Buttons:** 'OK' (with a green checkmark icon) and 'Cancel' (with a red X icon).

Fig. 2.3.10 By week days condition

Start time – the field contains specify the time when the task should be executed. If the time is specified incorrectly, an error message will appear on the screen. The default is the current time.

Interval – the field contains the time in which the task will be executed again. The interval can be set to any value, but it must not be less than 0 and it is specified in hours and minutes. The default is 1 hour.

Week days – the list contains the weekdays the task is executed on. At least one weekday must be selected, otherwise the dialog box will not be closed when you click the <OK> button. No weekday is selected by default.

Note: the task is executed on Week days at Start time. The task will be executed next time after Interval expires. If the Start time is less than the system time, the task start time is counted from the Start time taking into account the specified Interval. The Interval is active within a day. The task must be active in the task list, otherwise it will not be executed.

By month days condition:

Task

Task type: Establish the connection

Attempts count: 10 Pause between attempts (sec): 15

Phones: ☒ +12008001011

Condition

Condition: By month days

Start time: 14:55:27 Month days: ☒ 1, ☐ 2, ☐ 3, ☐ 4, ☐ 5, ☐ 6, ☐ 7, ☐ 8, ☐ 9, ☐ 10

Interval: 1 Hours

Confirmation

☒ Ask confirmation before task execution Timeout (sec): 20

OK Cancel

Fig. 2.3.11 By month days condition

Start time – the field contains the time when the task should be executed. If the time is specified incorrectly, an error message will appear on the screen. The default is the current time.

Interval – the field contains the time in which the task will be executed again. The interval can be set to any value, but it must not be less than 0 and it is specified in hours and minutes. The default is 1 hour.

Month days – select the days the task is executed on. At least one day must be selected, otherwise the dialog box will not be closed when you click the <OK> button. No day is selected by default.

Note: the task is executed on Month days at Start time. The task will be executed next time after Interval expires. If the Start time is less than the system time, the task start time is counted from the Start time taking into account the specified Interval and selected Month days. The Interval is active within a day. The task must be active in the task list, otherwise it will not be executed.

Periodically condition:

The screenshot shows a 'Task' dialog box with the following settings:

- Task type:** Establish the connection
- Attempts count:** 10
- Pause between attempts (sec):** 15
- Phones:** A list containing '+12008001011' with a checkmark.
- Condition:** Periodically
- Start time:** 14:55:27
- Interval:** 1 Hours
- Confirmation:**
 - ☒ Ask confirmation before task execution
 - Timeout (sec):** 20

At the bottom, there are 'OK' and 'Cancel' buttons.

Fig. 2.3.12 Periodically condition

Start time – the field contains the time when the task should be executed. If the time is specified incorrectly, an error message will appear on the screen. The default is the current time.

Interval – the field contains the time in which the task will be executed again. The interval can be set to any value, but it must not be less than 0 and it is specified in hours and minutes. The default is 1 hour.

Note: the task is executed periodically during a day at Start time. The task will be executed next time after Interval expires. If the Start time is less than the system time, the task start time is counted from the Start time taking into account the specified Interval. The Interval is active within a day. The task must be active in the task list, otherwise it will not be executed.

On start condition:

This condition has no parameters.

Note: the task is executed when you start the program or when you click the <Start/Stop> button in the main window of the program. The task must be active in the task list, otherwise it will not be executed.

After the connection was established condition:

The screenshot shows a 'Task' configuration window. The 'Task' section includes a dropdown for 'Task type' (set to 'Break the connection'), a numeric spinner for 'Attempts count' (set to 10), and another numeric spinner for 'Pause between attempts (sec)' (set to 15). The 'Condition' section has a dropdown for 'Condition' (set to 'After the connection was established') and a unit selector for 'Interval' (set to 'Hours' with a value of 1). The 'Confirmation' section features a checked checkbox for 'Ask confirmation before task execution' and a numeric spinner for 'Timeout (sec)' (set to 20). The window concludes with 'OK' and 'Cancel' buttons.

Fig. 2.3.14 After the connection was established condition

Interval – specify the time in which the task will be executed after the connection is established. The interval can be set to any value, but it must not be less than 0 and it is specified in hours, minutes or seconds. The default is 1 hour.

Note: the task is executed when the Interval after a connection is established expires. The task must be active in the task list, otherwise it will not be executed.

The program didn't send/receive the data condition

Interval – specify the time in which the task will be executed after the connection is established. The interval can be set to any value, but it must not be less than 0 and it is specified in hours, minutes or seconds. The default is 1 hour.

Note: the task is executed after Interval after the connection is established expires if no data has been received or sent within this interval. The task must be active in the task list, otherwise it will not be executed.

4.6.4 OPC settings

4.6.4.1 OPC settings

How it works

After the program is loaded, it analyzes the configuration tree and items are added to the internal structures of the data program during this analysis. The existence of items on the server is also checked during this analysis. This process may take from several seconds to several minutes depending on the OPC server. If an item does not exist, the corresponding message is added to the log and this item will be highlighted in red when you open the configuration. If the OPC server is sending data, the program will immediately start receiving data from it provided that the configuration tree is not empty. Also, the program will send data received from the core to OPC servers for the corresponding items.

The following happens when data is being received from an OPC server:

1. If the **"Wait till all items are updated"** checkbox is selected in the group parameters, the item update flag is set first and the value of the item is changed. Then the items of the group the item (whose data is received) belongs to are checked for value updates. If all items in the group have been updated, the data is sent to data export module. The **"Export valid values only"** and **"Don't export NULL values"** checkboxes are taken into account during export, i.e. if an item has a valid value and the checkbox is selected or if it is not selected, the following check is performed: if the value of the item is different from NULL and the checkbox is selected or if the checkbox is not selected, the item is exported into the core provided it exists. After the data is exported, the item update flag is cleared.
2. If the **"Wait till all items are updated"** checkbox is not selected in the group parameters, items are exported. The **"Export valid values only"** and **"Don't export NULL values"** checkboxes are taken into account during export, i.e. if an item has a valid value and the checkbox is selected or if it is not selected, the following check is performed: if the value of the item is different from NULL and the checkbox is selected or if the checkbox is not selected, the item is exported into the core provided it exists.

When data is received, the server name will be added to the item name if the "Add server name to item name" checkbox is selected in the group properties. The <;>, <">, <['>, <]> characters in server and item names are replaced with the <_> character. If an item is of the string type, the <\>, <"> characters are replaced with <\\>, <\ ">, if a character has a code within the range \$00..\$1F, <\x> characters are added before the character.

The following happens when data is being sent to an OPC server (e.g. from a data query module):

1. The data is copied to the temporary buffer;
2. A line is singled out from the temporary buffer. Lines can be separated by the <CR> or <CR+LF> characters;
3. An item is singled out from the line. Items are separated by the <;> character;
4. The item is decoded and, as a result, you get the server name (may be absent), the item name, the item type (may be absent), the item value. If no item type is specified, it is determined automatically according to the following rules: if the value starts from quotation marks, the data type is "string", or if the value can be converted into an integer without any errors, the type is "integer", or if the value can be converted into a fraction without any errors, the type is "double". In all other cases, the program interprets the value as being the "string" data type;

- The item is sent to the core. If no server name is specified, the value is sent to all items specified in the configuration and having the corresponding item name.

Groups, servers and variables

Data can be sent and received at any time. The "OPC groups, servers and variables" tree is used for that. For the program to be able to receive/send data, you should add at least one group with one item. The tree allows you to add, remove, edit, view and assign other names to items and also move items and groups.

Select the "OPC settings" tab in the settings dialog box and you will see the group and item tree on the screen. It is empty by default.

OPC Settings.

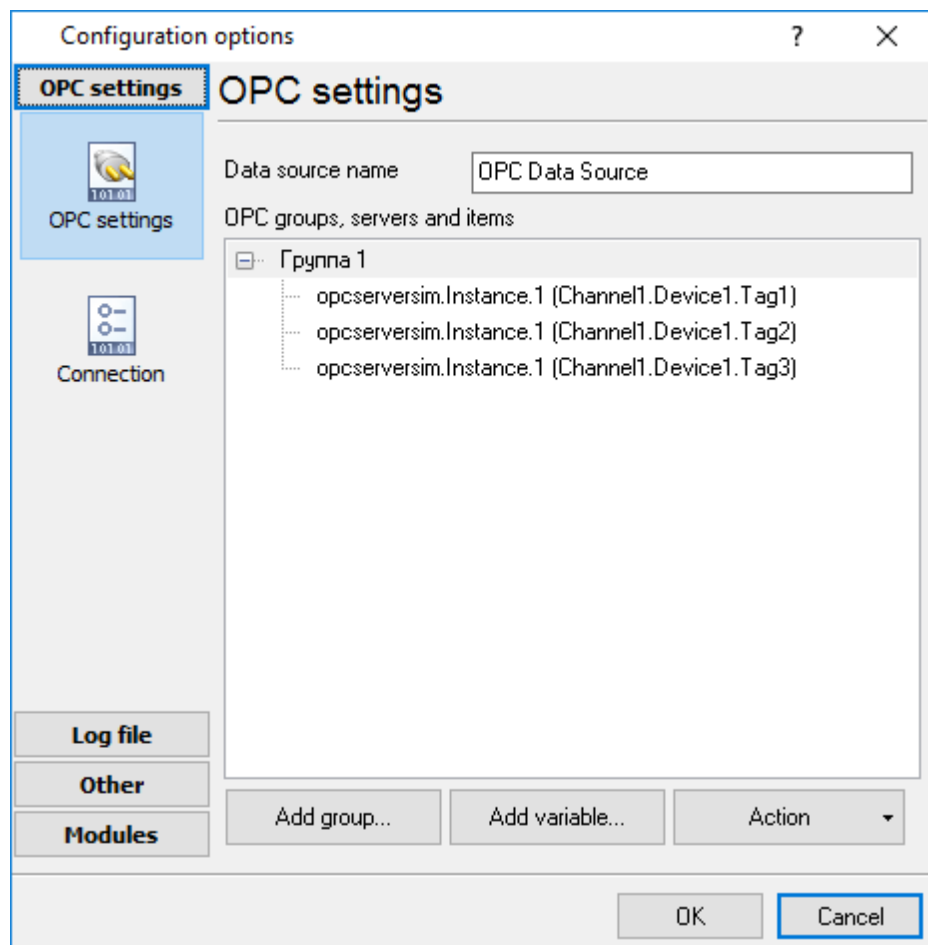


Fig. 2.4.1. OPC settings

Data source name – the field contains any name that will denote the operations performed with OPC servers.

OPC groups, servers and variables – the tree contains groups of servers and variables used to send/receive data.

Action – the button allows you to add groups and items from OPC servers to the configuration, remove groups and items, assign new names to items, view and edit data, move groups and items up/down the tree. You can right-click the tree instead of this button and you will see the same menu that you see when you click the "**Action**" button.

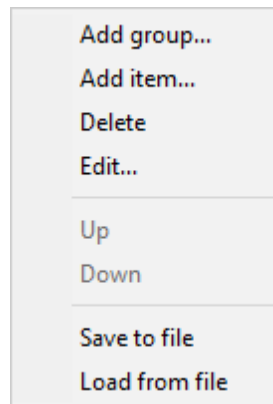


Fig. 2.4.2 The action menu after a right click on a group

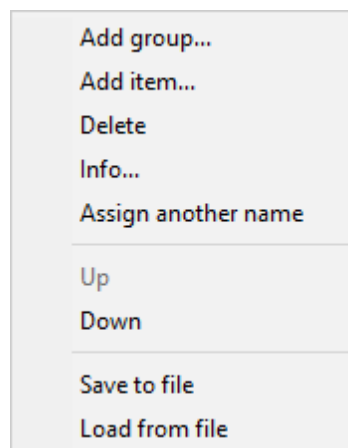


Fig. 2.4.3 The action menu after a right click on an item

Adding a group:

Click the "**Action**" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "Add group..." item and you will see the Group properties dialog box on the screen (Fig. 2.4.4).

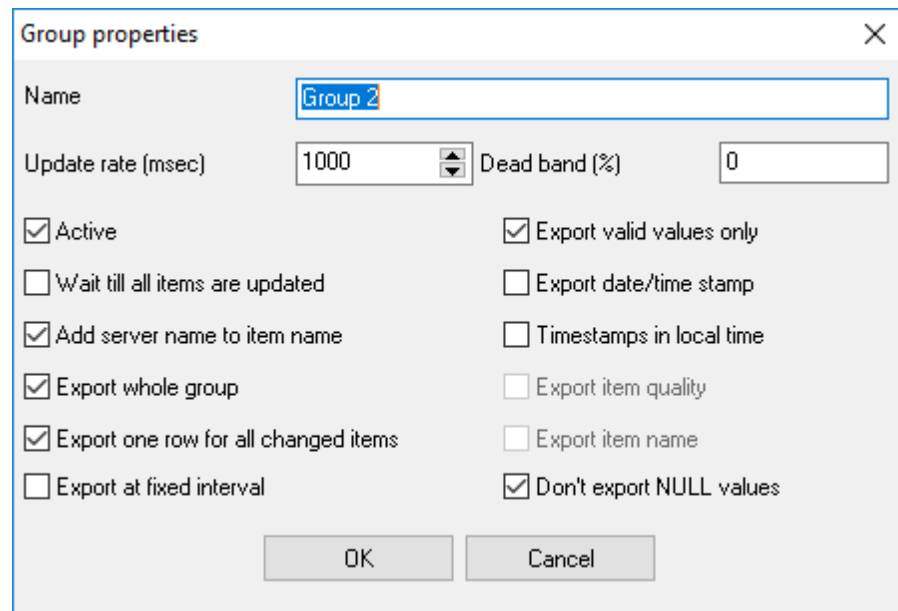


Fig. 2.4.4 Group properties

Name – the field contains the name of the group. The name may contain any characters. By default, it is Group XXX, where XXX is the number of the group.

Update rate (msec) – the field contains the update rate of items in the group when they are read from the OPC server. The value is specified in milliseconds. By default, 1000 milliseconds, i.e. 1 second.

Dead band (%) – the field contains the value of the dead band in percent. The default is 0 percent.

Active – the checkbox defines whether the group is active. If it is selected, items from this group will be exported (items will be read from servers), if it is not selected, they will not be exported (items will not be exported). By default, the checkbox is selected.

Export valid values only – the checkbox enabled/disabled exporting only valid values. If it is selected, only those items in the group whose value is OPC_QUALITY_GOOD will be exported, if it is not selected, the quality of the item value is not taken into account. By default, the checkbox is selected.

Export date/time stamp – the checkbox enabled/disabled exporting of update date and time of an OPC tag.

Export whole group - if this checkbox enabled then the program will export all tags in the group at once, otherwise each tag in the group will be export separately: one tag on a new line.

Timestamp in local time – if this checkbox enabled then the program will export the timestamp in local time zone. Otherwise the UTC time zone will be used.

Export item name, Export item quality – these options are available if the "Export whole group" option is disabled. In this case the program will add two special items to the output that will contain the tag name and tag quality. For example:


```
ITEM_NAME[8]="opcserver.sim.Instance.1.T1";ITEM_VALUE[3]=1606521099;ITEM_QUALITY[3]=192  
ITEM_NAME[8]="opcserver.sim.Instance.1.T2";ITEM_VALUE[3]=1170485188;ITEM_QUALITY[3]=192  
ITEM_NAME[8]="opcserver.sim.Instance.1.Val1";ITEM_VALUE[5]=4958.32440443337;ITEM_QUALITY[3]=192  
ITEM_NAME[8]="opcserver.sim.Instance.1.Val2";ITEM_VALUE[5]=4970.36790242419;ITEM_QUALITY[3]=192
```

Export at fixed interval – if this checkbox enabled then the program will export all tags in the group periodically, even through a tag value isn't changed.

Add server name to the item name – if this checkbox enabled then the program will append a server name before the tag name. It allows to distinguish identical tags from different servers.

Don't export NULL value – the checkbox allows you not to export items with null values. If it is selected, items in the group that have a null value will not be exported, if it is not selected, the null value of an item is not taken into account. By default, the checkbox is selected.

Wait till all items are updated – the checkbox enables/disables waiting till all items in the group are updated. If it is selected, items in this group will be exported only when all items in the group are updated (the "Export valid values only" and "Don't export NULL values" checkboxes are taken into account during export), if it is not selected, all items will be exported taking into account the "Export valid values only" and "Don't export NULL values" checkboxes. By default, the checkbox is selected.

Note: See the paragraph "[How it works](#)" for more details about the export algorithm.

After you specify the group properties, click the "OK" button to save the properties or the "Cancel" button to cancel it. The dialog box will be closed and a new group will appear in the tree.

Adding an item:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "**Add item...**" item and you will see the Item properties dialog box on the screen (Fig. 2.4.5).

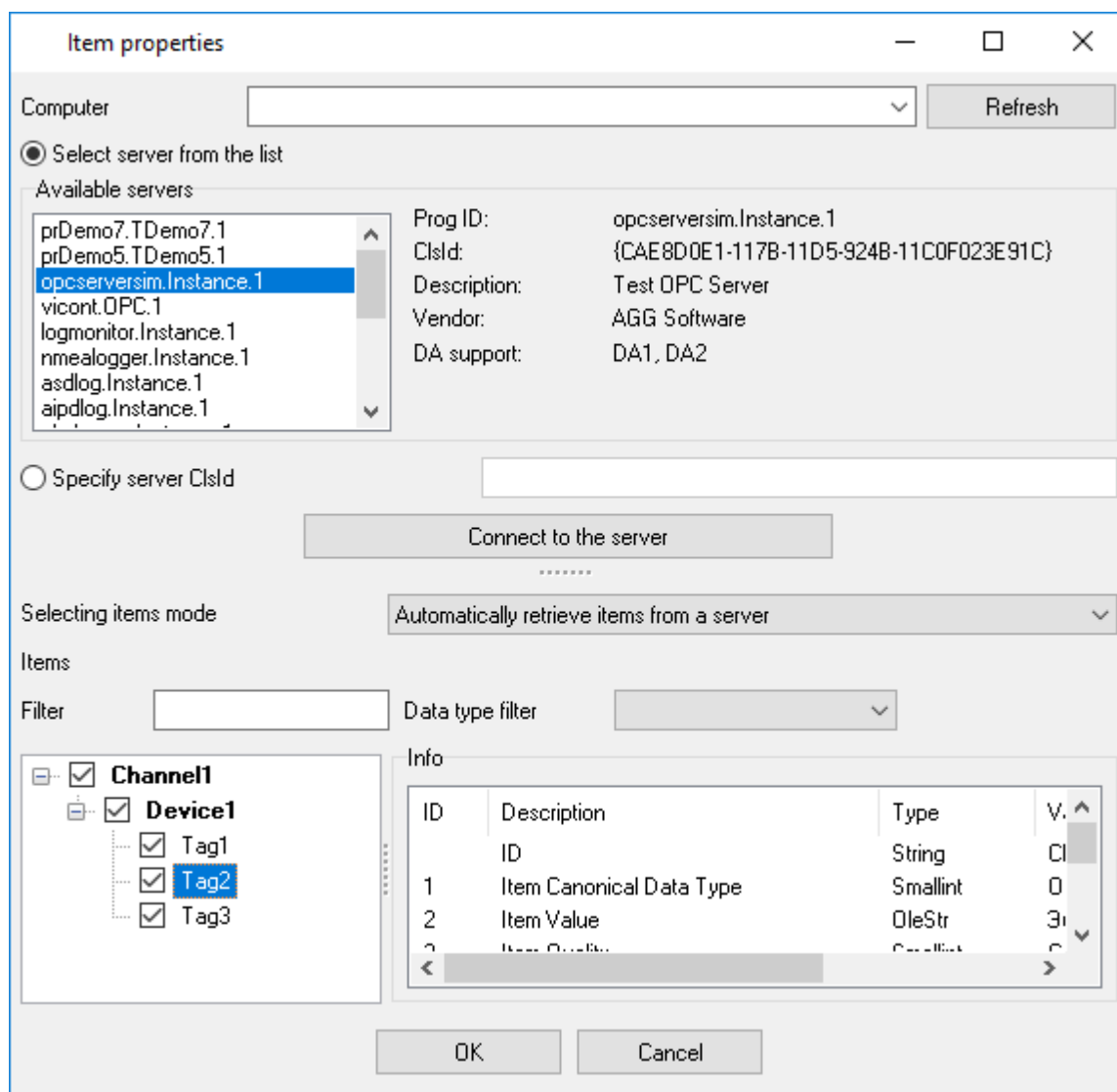


Fig. 2.4.5 Automatic mode item properties

Servers – the list contains the list of available OPC servers.

Info – the group shows brief information about the selected OPC server.

Prog ID – the field shows the program identifier (the name of the program).

Description – the fields shows the brief description of the server.

DA support – the fields shows the supported DA.

Vendor – the field shows the name of the vendor.

Connect – the button connects the program to the server and after that items (if there are any on the server) appear in the "Items" list. It is also possible to connect to the server with a double click on the server selected in the "Servers" list.

Selecting items mode – the list allows you to select one of the modes for item selection/input. There are two modes available: Automatically retrieve items from a server and Manual mode.

Items – the list contains items available on the selected server. There is a checkbox next to each item. It enables/disables adding the corresponding item to the tree (Fig. 2.4.1) after you click the "OK" button and close the dialog box. Once a connection is established to the server, the checkboxes of all items are selected. You can select or clear a checkbox by clicking it. Also, you can use the menu (Fig. 2.4.6) opened with a right click on the list.

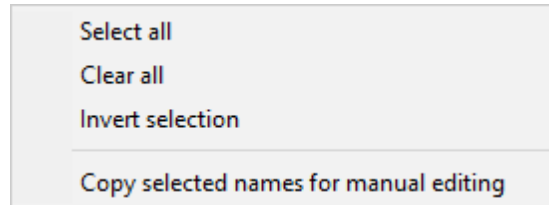


Fig. 2.4.6 Item menu

Select all (Fig. 2.4.6) – select the checkboxes of all items.

Clear all (Fig. 2.4.6) – clear the checkboxes of all items.

Invert selection (Fig. 2.4.6) – invert the checkboxes of all items.

Info – the group shows information about the selected item.

ID – the column shows the item identifier. It cannot be edited.

Description – the column shows the brief description of the item. It cannot be edited.

Type – the column shows the item type. It cannot be edited.

Value – the column shows the item value. It cannot be edited.

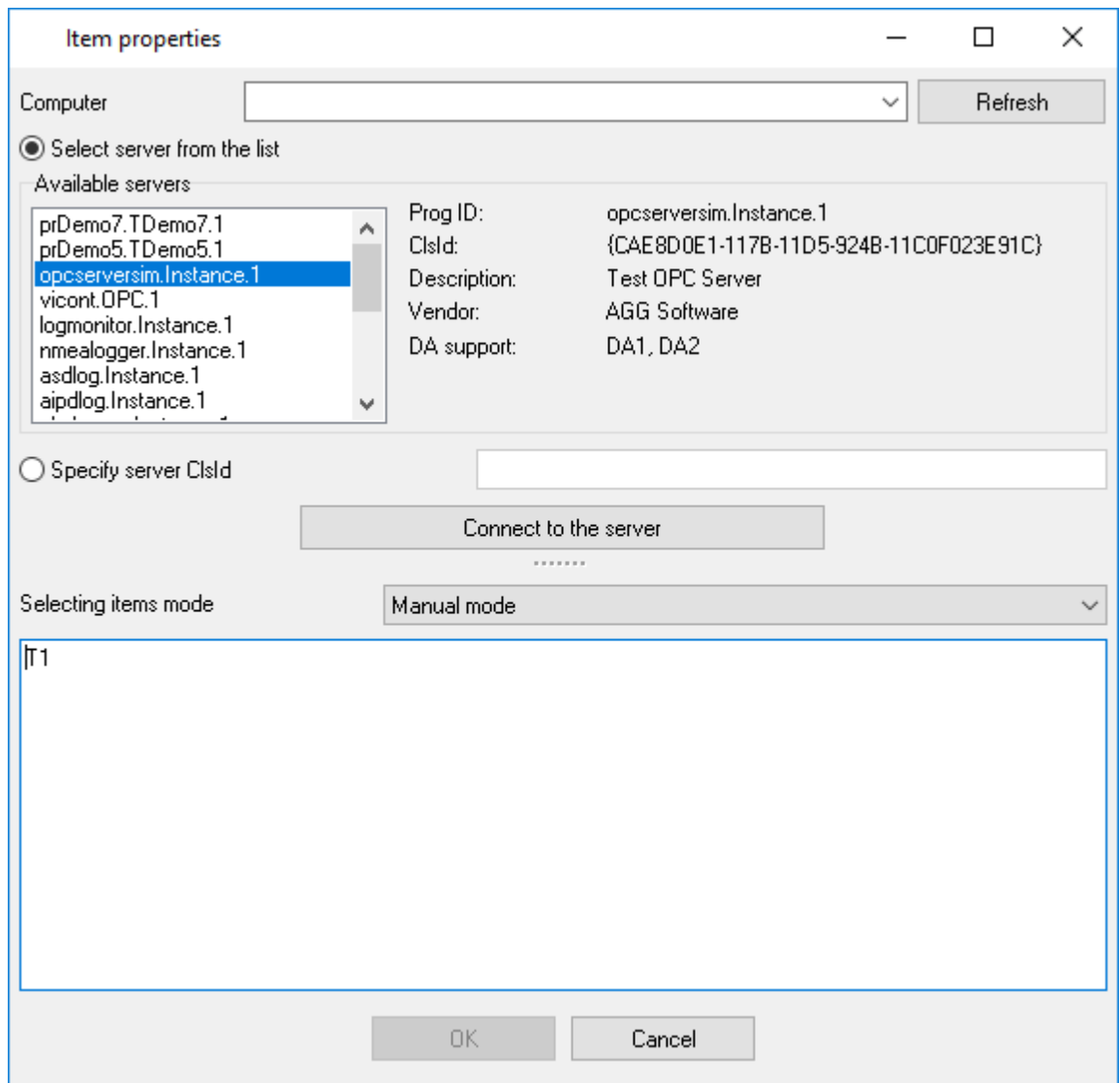


Fig. 2.4.7 Manual mode item properties

Items – the text field allows you to type items manually. The number of items is unlimited.

Removing an item:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the **"Delete"** menu item and you will see the confirmation dialog box on the screen. Click the "Yes" button in this dialog box and the group or the item will be removed from the tree after that. If you change your mind, click the "No" button. The "Delete" menu item is not available when the tree is empty.

Editing a group:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the **"Edit..."** menu item and you will see the Group properties dialog box on the screen (Fig. 2.4.4).

Note: Note that a group must be selected in the tree, otherwise you will not see this menu item.

See "Adding groups:" for the description of group properties.

Note: See the paragraph "[How it works](#)"^[41] for more details about the export algorithm.

After you specify the group properties, click the "OK" button to save the properties or the "Cancel" button to cancel it.

Viewing information about an item:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "Info..." menu item and you will see the Item properties dialog box on the screen

All item properties except the "Info" group will be unavailable. Click the "Close" button after you view the information.

Note: Note that an item must be selected in the tree, otherwise you will not see this menu item. See "Adding an item:" for the description of item properties.

Assigning another name to an item:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "Assign another name..." menu item and you will see the dialog box where you can type a new name for the item. Type a new name for the item and click the "OK" button – the new name of the item will appear in the tree. Note that the name must differ from those already existing in the tree, otherwise you will see the corresponding message on the screen and the item will not be renamed.

Note 1: It is this assigned item name that is used during export/import, not the one in the tree. By default, the assigned name has the same value as the value selected when the item was added.

Note 2: Note that an item must be selected in the tree, otherwise you will not see this menu item.

Moving a group or an item up:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "**Up**" menu item and the group or the item will move one position up. If a group or an item is the only one or if there are no groups and items at all, the menu item will be unavailable.

Moving a group or an item down:

Click the "Action" button or right-click the "OPC groups, servers and variables" tree and you will see the action menu (Fig. 2.4.2 or Fig. 2.4.3). Select the "**Down**" menu item and the group or the item will move one position down. If a group or an item is the only one or if there are no groups and items at all, the menu item will be unavailable.

Note: you can add any number of groups and items with different parameters.

4.6.5 SNMP settings

4.6.5.1 SNMP data source

After the program is started, its configuration is analyzed and the list of active groups is formed. After that the program is switched into the mode of reading variables from remote devices via the SNMP protocol. The following operations are carried out for every group:

1. If it is time to read variables, the program switches to step 2. Otherwise the program waits till its time to start a data query;
2. The program receives the value for every variable in the group. If the value is not received, an error messages is written to the log file;
3. If "[Wait till all items are updated](#)^[52]" is enabled the program checks if the values of the variables have changed. If the values of all variables have not changed, the program moves on to step 1. Otherwise it moves to step 4;
4. Variables are exported depend on [the group properties](#)^[52].

4.6.5.2 Group and variable tree

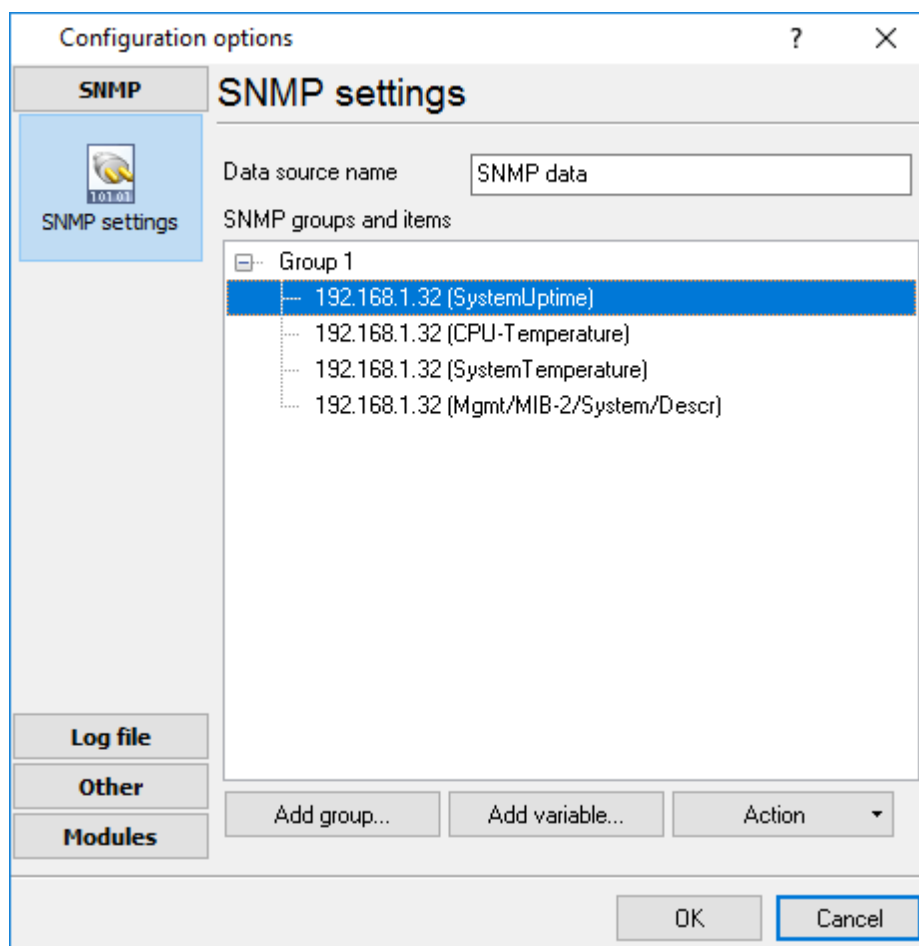


Fig. 2.5.1 Group and variable tree

You can see the group and variable tree in fig . 1. The program uses this information to query data. You can create any number of groups and any number of variables in each group. Note that you should add groups and variables with names that are not in the list yet.

Data source name – the name shown in the program. The name must be unique, otherwise you will not be able to save the settings. Specify any number of ASCII characters in this field. The default is Data source #<Number>.

Group and variable tree – contains groups with variables included in them which the program operates. The tree is empty by default.

Action – actions with the group and variable tree. A click on this button opens the menu shown in fig. 2.5.2 (a similar menu will appear if you right-click the tree).

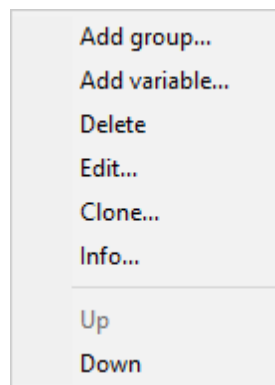


Fig. 2.5.2 Actions with groups and variables

Add group... – add a new group. If you select this item, you will see the dialog box with [the default group properties](#)^[52]. Fill out the necessary fields in this dialog box and click "OK". If a group with this name already exists you will see a warning on the screen and will not be able to save the group.

Add variable... – add a new variable to the current group. If you select this item, you will see the dialog box with [the default item properties](#)^[54]. Specify the necessary values in this dialog box and click "OK". If the new variable is unique, it will be saved and appear in the tree. Otherwise you will see a warning on the screen and you will not be able to save the variable.

Delete – delete the current group or variable. If you select a group or several groups, you will see the deletion confirmation dialog box on the screen. If you click "Yes", the selected groups together with variables included in them will be permanently deleted. If you select one or several variables, you will also see the deletion confirmation dialog box on the screen.

Edit... – edit the selected group or item (variable). Depending on what is currently selected in the tree, you will see the Group properties or Item properties dialog box. After you edit the necessary fields, click "OK" and the properties will be saved.

Clone... – copy the selected variable (item). You will see the same item properties dialog box as in the edit mode.

Info... – show information about the selected group or variable. If you select this item, you will see the same dialog box as the "Add group" or "Add item" dialog box. However, you will not be able to

edit any of the fields. It is used to view the properties and to protect them against accidental change. It is unavailable if the tree is empty.

Up – move the current group or variable one position up. It is unavailable if the current group or variable is at the beginning.

Down – move the current group or variable one position down. It is unavailable if the current group or variable is at the end.

Note: The program checks if this variable exists on the server when you launch the program. If the variable does not exist, it will be marked in the tree with help of a red exclamation mark.

4.6.5.3 Group properties

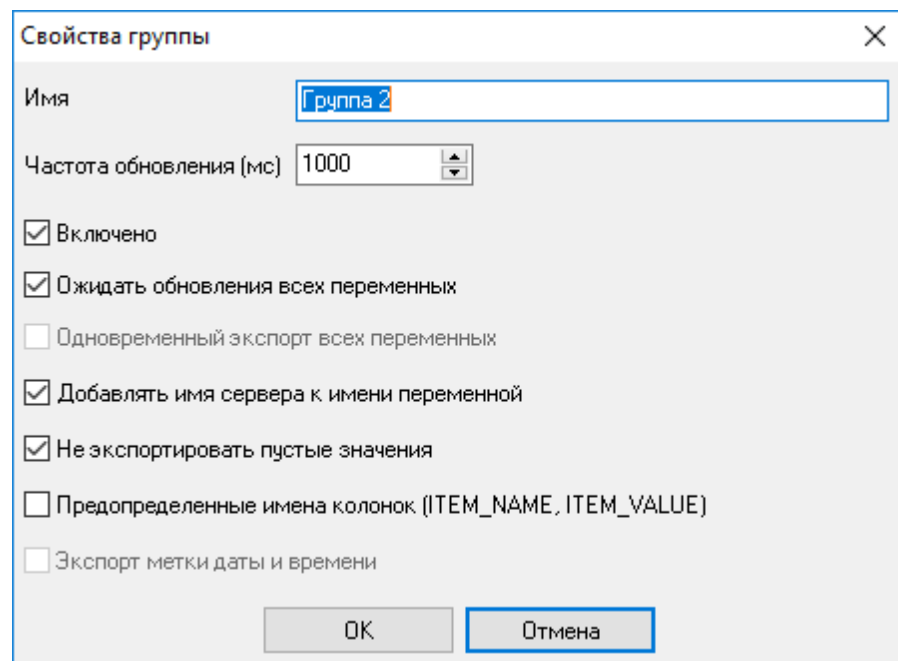


Fig. 2.5.3 Group properties

Name – a set of characters serving as the name of the group. **ATTENTION:** The group name must be unique or the properties will not be saved. The default is Group <Number>.

Update rate (msec) – this property defines how often the variables will be read from the SNMP device. The default is 1000, i.e. 1 second.

Active - enable/disable reading the values of the variable from the SNMP device. It is enabled by default.

Wait till all items are updated – if it is enabled, the program will export the variables from the group only if the values of all variables in the group have been updated since the data was read last time. For example, if there are two variables in the group, the program will read the values of the variables until the values of both variables change as compared to their previous value. In this case, the program exports the last received value (for example, if the value of the first variable changes at once but the program has to wait several seconds or minutes for the value of the second one to

change and the value of the first variable changes several times during this period). It is enabled by default.

Export all items at once – if this option is enabled, all variables are exported as one string of data. If this option is disabled, each value is exported in a separate string. With this option disabled, it is convenient to export a large number of values defined in the group.

Add server name to item name – if it is enabled, the server name is added to the variable name being exported. It is enabled by default.

Don't export NULL or empty values – the checkbox allows you not to export variables with null values. If it is enabled, variables that have null values in this group will not be exported. If it is disabled, the null value of a variable is not taken into account. It is enabled by default.

Fixed column names (ITEM_NAME, ITEM_VALUE) – if this option is enabled, the names and values being exported will be sent to the kernel in the form of the following strings: ITEM_NAME [<String type>] = "<Variable name>"; ITEM_VALUE [<Value type>] = "<Value>". It is disabled by default.

Example:

```
ITEM_NAME[256]=".iso.org.dod.internet.private.enterprises.2162.1.2.0";ITEM_VALUE[8]
="computerName"
ITEM_NAME[256]=".iso.org.dod.internet.private.enterprises.2162.1.3.0";ITEM_VALUE[8]="1"
```

This export method is convenient when you need to export a large number of values. For example, two columns in the database are enough to save the values of all variables in this case.

Export date/time stamp – if this option is enabled, the date and time is added before the string being exported as it is described in the previous property. It is unavailable if the previous option is not enabled. It is disabled by default.

Example:

```
ITEM_DATETIME_STAMP[7]=01.03.2010 15:52:00;ITEM_NAME[256]=".iso.org.dod.internet.private.
enterprises.2162.1.2.0";ITEM_VALUE[8]="computerName"
ITEM_DATETIME_STAMP[7]=01.03.2010 15:52:00;ITEM_NAME[256]=".iso.org.dod.internet.private.
enterprises.2162.1.3.0";ITEM_VALUE[8]="1"
```

Item properties

Host (name or IP) 192.168.1.111

Port 161 **Response timeout (ms)** 500

Community public **Version** SNMPv2c

Get list of variables from:

☒ **Remote device**

Base OID .1.3.6.1.2.1 **Get variables**

☐ **MIB file**

Sel	OID	Name	Data type	Export name
☐	.1.3.6.1.2.1	Mgmt	Undefined	Mgmt
☐	.1.3.6.1.2.1.1.1.0	Mgmt/MIB-2/	Octet string	Mgmt/MIB-2
☒	.1.3.6.1.2.1.1.2.0	Mgmt/MIB-2/	Object ID	Mgmt/MIB-2
☐	.1.3.6.1.2.1.1.3.0	Mgmt/MIB-2/	Time ticks	Mgmt/MIB-2
☐	.1.3.6.1.2.1.1.4.0	Mgmt/MIB-2/	Octet string	Mgmt/MIB-2

☐ **Specify OID manually**

OID .1.3.6.1.2.1.1.2.0

Data type Object ID

Export name Mgmt/MIB-2/System/ObjectID

OK **Cancel**

Fig. 2.5.4 Item properties

Host (name or IP) – the name or IP address of the device the values of the variables are read from. You can enter either URL (for example, www.aggssoft.com) or the IP address (for example, 192.168.1.1). The field stores the last 10 entered values so if the name is the same for different variables, you can just select it from the list. The default is localhost.

Port – the port the program uses to connect to the device (previous property). The default is 161.

Community – the group of variables on the remote SNMP device. The default is public. Note: The community with the "public" name is set on the remote device by default. The network administrator can change this name for any other name for security reasons. The field stores the last 10 entered values as well.

Get variables – refresh the variables in the "Variable" drop-down list (see below). If the remote device does not exist, if you enter an invalid port, or if you specify a non-existing community, you will see a warning on the screen.

Base OID – the base group of SNMP variables on the remote device. The program will start getting variable names for the "Variable" list from this group and below. The default is ".1.3.6.1.2.1". All **MIB-II** variable names begin with this value. More detailed information about MIB (Management Information Base) you can find [here](#). If you'll specify an invalid value here, then the program will return an empty list of variables.

Name – the name of the variable on the remote device. You can type it manually or select it from the list. To fill the list with variables, click the "Get variables" button. The field is empty by default. You can see sample variables in fig. 2.5.4. You can enter either a numeric path or a symbolic path for the standard (.iso.org.dod.internet.private.enterprises) variable location. You can also combine the symbolic and numeric methods. The should always begin with a dot symbol.

Data type – the type of data in the variable being read. If the "Variable" list is filled and you select a variable from the list, the data type of the selected variable will automatically appear in this field. The default is Null.

Export name – if you specify any name in this field, this value will be used during export instead of the variable name. By default the string is empty.

4.6.6 File settings

4.6.6.1 File data source

About log files

A log file is a file containing records about events in the chronological order.

Logging means the chronological recording of data with varied (customizable) level of details about systems events (errors, warnings, messages). Usually, the data is saved to a file.

Examining the contents of an error log file after failures often makes it possible to understand what causes them. Old hardware and software systems use log files to save and store data.

Purpose

The program is used to monitor folders (directories) with log files or separate log files in real time. Once the program detects new data in the log file, it can send notifications to the administrator or export and archive data from log files. The built-in script and filter tools allow you to single out only the events you are interested in from one or several log files. It considerably decreases the load on the administrator whose job is to maintain several web servers or data servers.

A good example of how to use the program is monitoring a computer where a lot of users can create, copy and edit a file. Log files allow you to track all changes and the administrator can control all operations using the log file. With our program, the administrator can create event types that are to be detected (for example, deleting a file or creating a file with a certain name) and receive an immediate notification about this event as a desktop or e-mail message. The program becomes even more useful if you need to control several servers simultaneously.

How the program works

Once started, the program analyzes the list of folders specified in the configuration checking if these folders exist. If a folder exists, it is added to the scan list, otherwise it is skipped. Once the data source is started, the file list is also filled with the initial file size values. Then the program is switched into one of the scan modes specified in the configuration:

1. "Simple" – simple scan mode. In this mode, the program just analyzes the folders and subfolders specified in the configuration for changes in the file.
2. "Shell" – the shell mode uses operating system events when files and folders are created or modified.

If the program is configured to read data at startup, after the initial processing it will read data from the files of the corresponding folders and subfolders and prepare this data for exporting, archiving or sending notifications.

The program monitors new files and folders and starts processing them in both modes. The program also monitors deleted files and folders and stops processing them.

If a file has just been created or the size of an existing file has changed, the program reads this data from the file and passes them on for further processing. The parameters specified in the configuration are used to read data (see "File settings"). If there is a delay set in the configuration before reading the file and the file changes during this delay, the program delays reading the file for the time of the delay specified in the configuration. If the file changes again, the reading will be delayed again and so on till the file stops changing. After the program reads the file, it performs one of the following three operations with it:

1. The file is deleted;
2. The file is cleared, i.e. the file size becomes equal to zero;
3. The file is not modified.

The program takes into account the file mask specified in the configuration, i.e. if "*.txt" is specified in the configuration, the program will scan and read data only from text file – it will ignore all other files. The program also uses the minimum file size value from the configuration during the scanning process. If the file size is less than the specified value, the program does not read the file till the file size is equal to or larger than the specified value.

The program reads data from the file by blocks whose size is specified in the configuration. The larger the block is, the fewer times the program accesses the disk and the better the performance is, but the requirements to the computer performance are higher.

4.6.6.2 Folders settings

Select the "**Folders**" tab in the "Configuration options" window and you will see the list of folders on the screen (fig. 2.6.1).

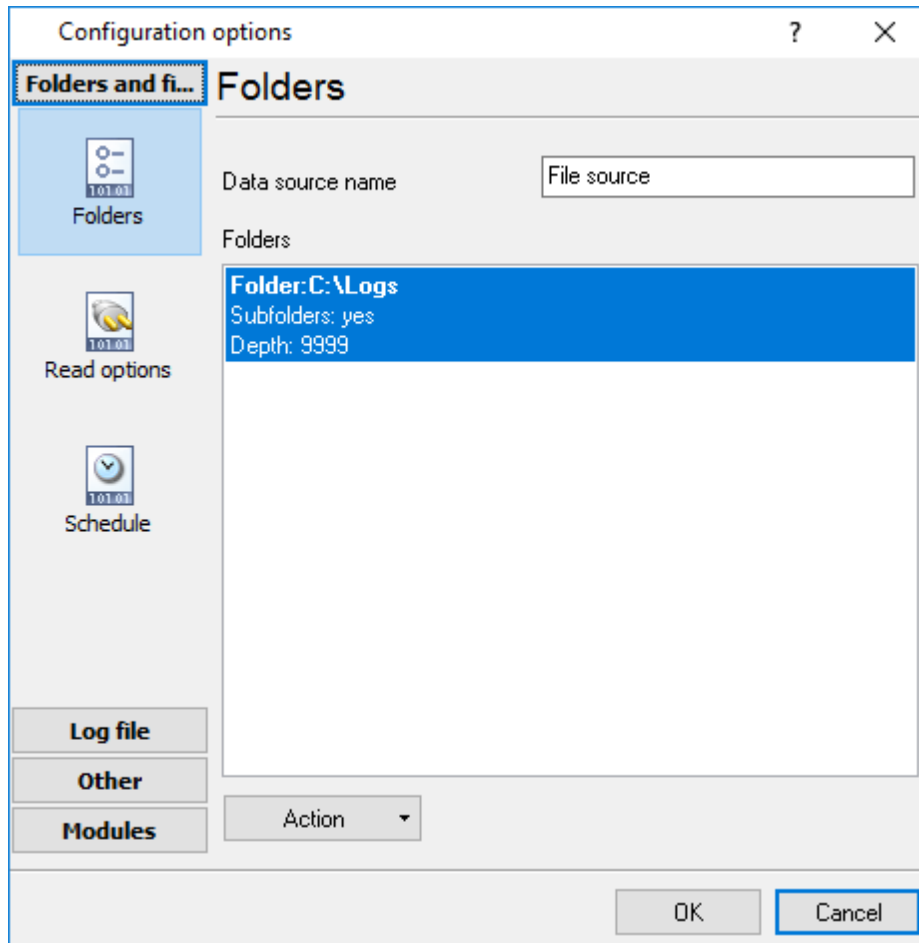


Fig. 2.6.1 Folders settings

Data source name – the field contains any name that will denote the operations performed with files. You will see this name in the drop-down list in the main window of the program. By default, it is Data source #XXX, where XXX is the number of the data source.

By default, the list of folders for scanning is empty. Click the "**Action**" button or right-click the empty list and you will see the action menu (fig. 2.6.2).

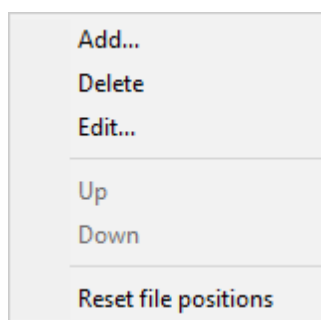


Fig. 2.6.2 Action menu

Add... - the item allows you to add a new folder to the list. After you select this item, you will see the Folder Properties dialog box on the screen (fig. 2.6.3). Specify the folder properties, click the OK button and the folder will appear in the list. If you change your mind, click Cancel.

Delete - the item removes the selected folder. Before the deletion, the confirmation dialog box will be displayed on the screen when you will have to click Yes to remove the folder from the list or No to cancel the operation. The item is not available if the list is empty.

Edit... - the item shows the "**Folder properties**" dialog box (fig. 2.6.3). Change the folder properties and click OK to save them. If you change your mind, click Cancel. The item is not available if the list is empty.

Up - the item moves the folder one position up. The item is not available when the list is empty or when the first folder is selected.

Down - the item moves the folder one position down. The item is not available when the list is empty or when the last folder is selected.

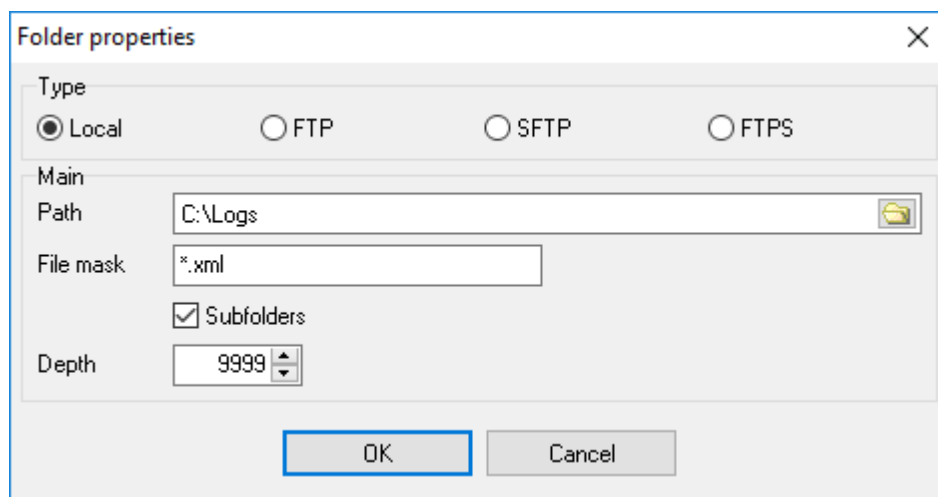


Fig. 2.6.3 Folder properties

Path - the field contains the path to the folder. You can select the folder from the list by pressing the button to the right. The default folder is the current one.

Subfolders - the checkbox enables/disables scanning subfolders. By default, subfolders are scanned.

Depth - the depth of scanning subfolders. 9999 by default.

Note: The scanning depth is counted from the specified path to the folder. For example:

C:\Files\Data\Test - source folder

C:\Files\Data\Test\Folder0 - level 1

C:\Files\Data\Test\Folder0\Folder1 - level 2

etc.

Note: If the specified folder does not exist, you will see an error message on the screen and you will not be able to save the folder properties so it is recommended to select folder from the list by clicking the button to the right from the Path field.

4.6.6.3 Files settings

Select the "**Read options**" tab in the "Configuration options" window and you will see the file and scan settings on the screen (fig. 2.6.4).

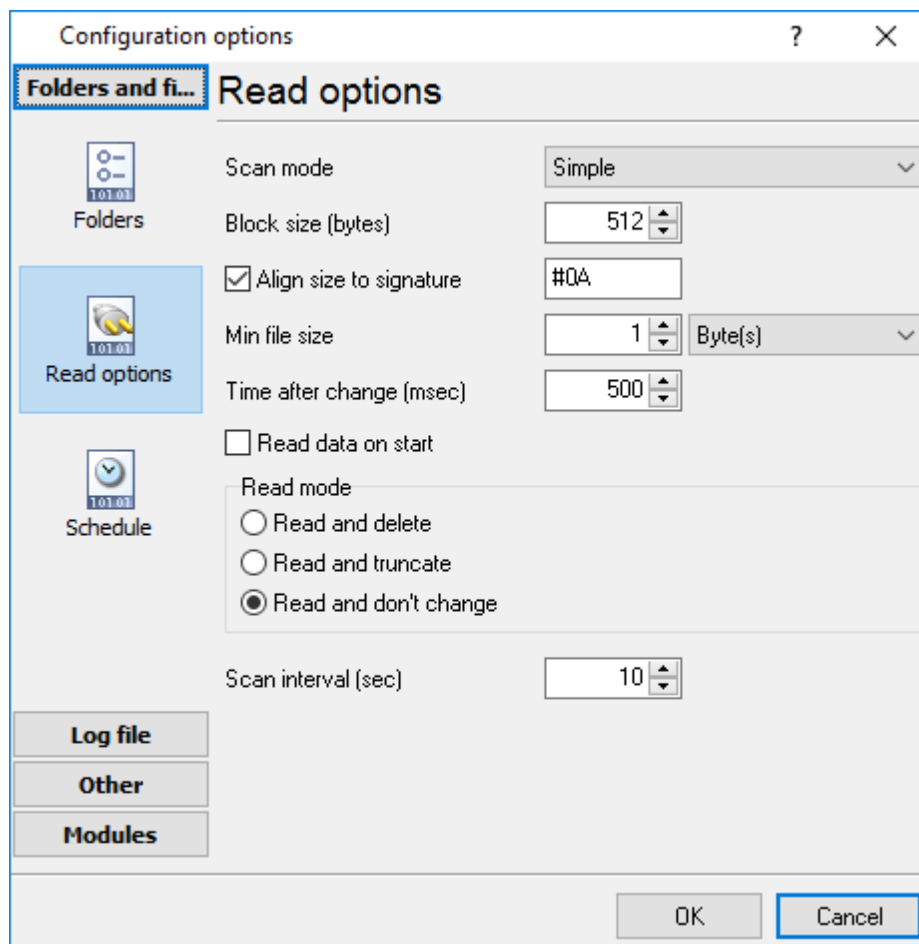


Fig. 2.6.4 Shell mode

Scan mode - the list allows you to select one of the scan modes: **Simple** or **Shell**. In the Simple mode, the program just goes through files in the specified folders and subfolders and checks if the

the file size has changed. The Shell mode uses operating system events to check changes in the file size. The program monitors new files and folders and deleted files and folders in both modes. If the operating system does not support the Shell mode, only the Simple mode will be available in the list. The default mode is Shell.

Shell mode

File mask - the field contains the file mask used to scan files. The program processes only those files that match the mask. For example: *.* - all files are processed, *.txt - only text files are processed, *.exe - only executable files are processed. The default is *.*.

Block size (bytes) - the field contains the size of the block (in pixels) used to read data. If the size is too small, Data will be read multiple times from the file, which will result in a delay. If the size is too large, it may also result in a delay. Choose the optimal data block size. The default is 512 bytes.

Min file size - the field defines the minimum size of a file to be processed. There is a list to the right from this field. It contains measurement units: Byte(s), KByte(s) and MByte(s). If the size of the file is less than that specified in the field with the corresponding measurement unit, it is not processed. The default is 1 Byte(s).

Time after change (msec) - the field contains the delay before the file is read. If the file changes again during the delay, the reading will be delayed again and so on till the file stops changing. The default is 500 msec.

Read data on start - the checkbox allows the program to read data from the file and pass them to the kernel at the program startup. The above parameters are used to read data. It is disabled by default.

Read and delete - if the option is selected, the program will delete the file after the data is read. It is not selected by default.

Read and truncate - if the option is selected, the program will clear the file so that its size equals zero after the data is read. It is not selected by default.

Read and don't change - if this option is selected, the program will leave the file unchanged after the data is read. It is selected by default.

Simple mode

All parameters are the same as for the Shell mode except for the last option.

Scan interval (sec) - the field contains the scan interval value. The default is 30 sec.

Select the "**Schedule**" tab in the Configuration Options window and you will see the log file scan schedule settings on the screen (fig. 2.6.5).

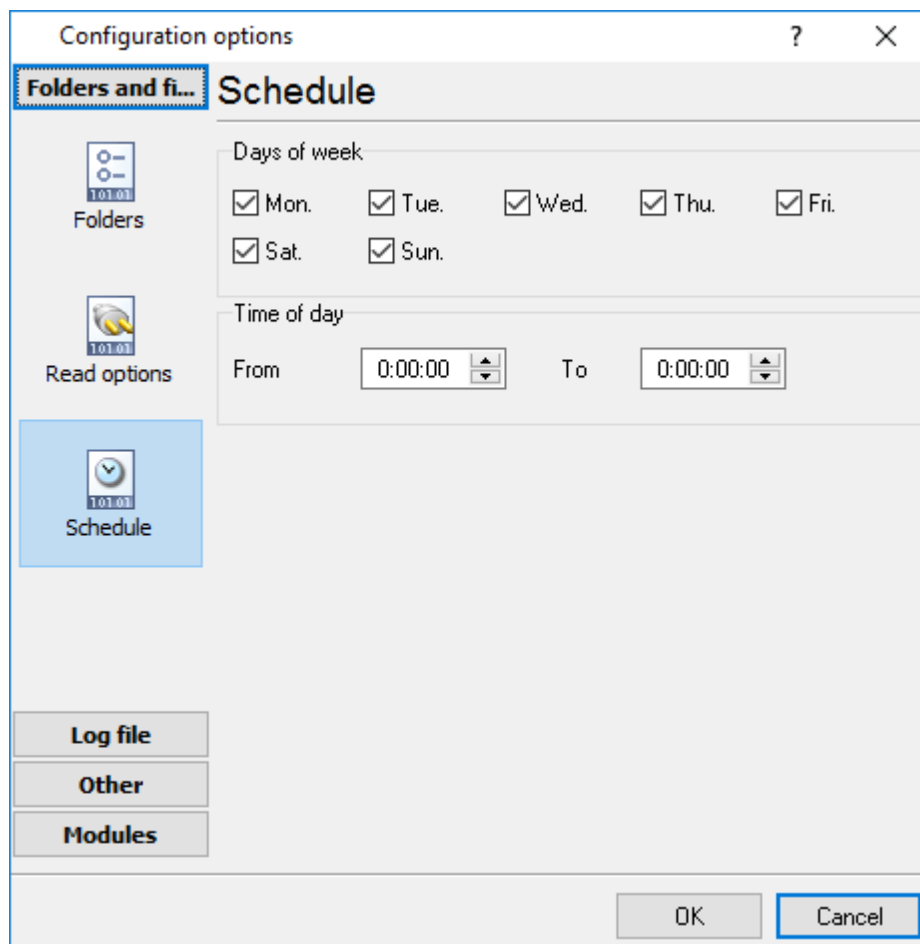


Fig. 2.6.5 Schedule

These options allows you to specify:

1. The days to scan on (the "**Days of week**" group);
2. The time when to scan (the "**Time of day**" group). If you specify the interval as 0:00:00-0:00:00, scanning will be done around the clock.

These options can be useful if, for example, you do not want to scan log files at weekends when the computer is off or when the administrator will not see notifications anyway.

4.6.7 DDE settings

4.6.7.1 Description

DDE

Dynamic Data Exchange (DDE) is a mechanism for interprocess communication in Microsoft Windows operating systems. Although the latest versions of Windows still support this mechanism, it is mainly replaced with more powerful mechanisms — OLE, COM and Microsoft OLE Automation. However, DDE is still used in some things inside Windows, particularly in the mechanism of associating filename extensions with applications. This mechanism can be used to get data from third-party applications.

Purpose

The plug-in is designed to send/receive data between a DDE server and this plug-in. Data exchange is carried out by server groups and their items specified in the configuration (see "Groups, servers and items"). After data is received, it is processed and grouped depending on the specified configuration and is sent for future processing and export.

How the program works

After the program is launched, the configuration tree is analyzed, during which it is checked whether the items exist on the server. If an item does not exist, the corresponding message is added to the log and this item will be highlighted in red when you open the configuration. If the DDE server is sending data at the moment, the program will start receiving the data according to the specified DDE items. Depending on the group settings, items can be exported simultaneously or one at a time, as well as with the date and time stamp. The program can also query DDE servers at the specified time interval and read the values of the DDE items. It is possible to specify a data type for each DDE item. Depending on the specified data type, it is possible to post-process, filter and export data later with the help of additional plug-ins.

4.6.7.2 Groups, servers and items

To start working with the program, you need to add at least one group with one item. Groups can contain logically united items and each item can be located on a different server. Items from one group can be exported simultaneously, which makes it easier to process and export them later. The program allows you to add an unlimited number of groups and items with different properties.

DDE settings

To create a list of groups and items, use the "Groups, servers and DDE items" tree (fig. 2.8.1). You can use the "Action" button or the popup menu to add, remove, edit, view items, assign other names to them and also move items and groups. The contents of the popup menu depend on the currently selected tree item (a group or an item).

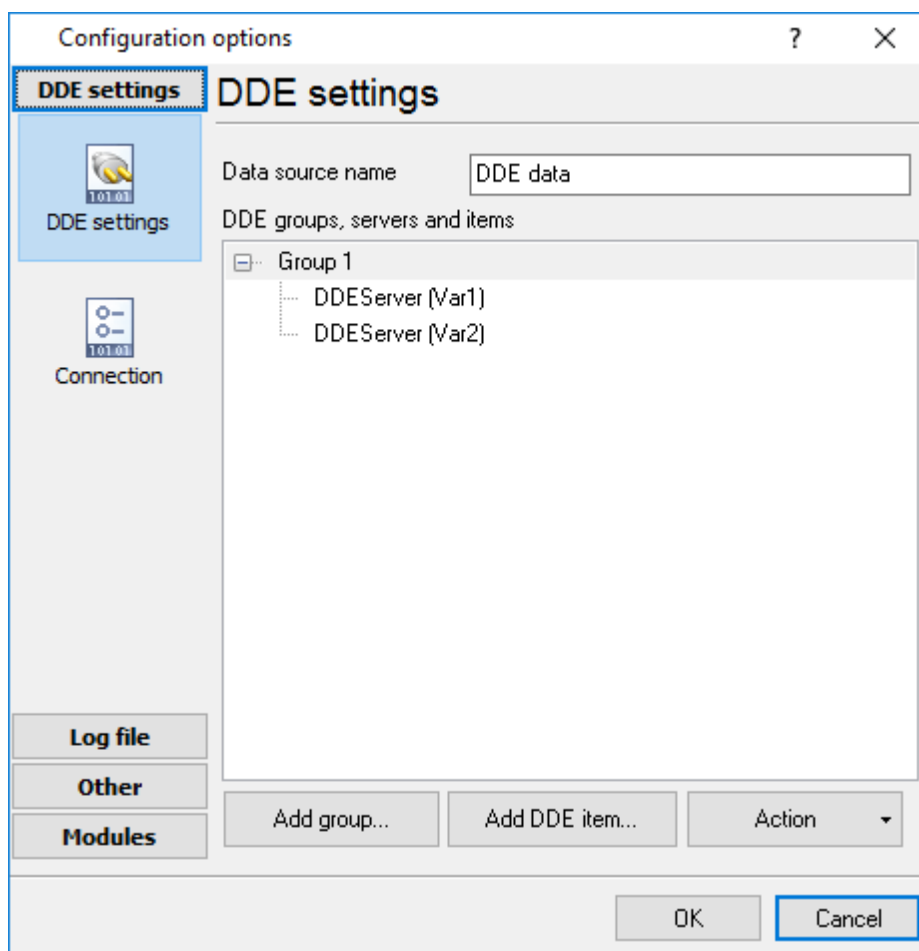


Fig. 2.8.1 DDE settings

Data source name - you can specify any name that will describe the groups of items and will be displayed in the main window of the program.

Groups, servers and DDE items - the tree contains groups of servers and items used to receive/send data.

Action – the button allows you to add groups and DDE items to the configuration, remove groups and items, assign new names to items, view and edit data, move groups and items up/down the tree. A click on the button opens a popup menu with the list of actions (fig. 2.8.2).

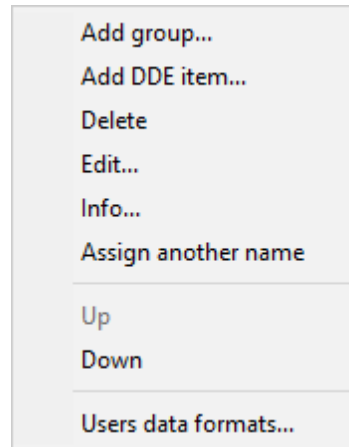


Fig. 2.8.2 Action menu

Adding a group

The "New group" menu item allows you to add a group of items. A click on this item opens the "Group properties" dialog box (fig. 2.8.3).

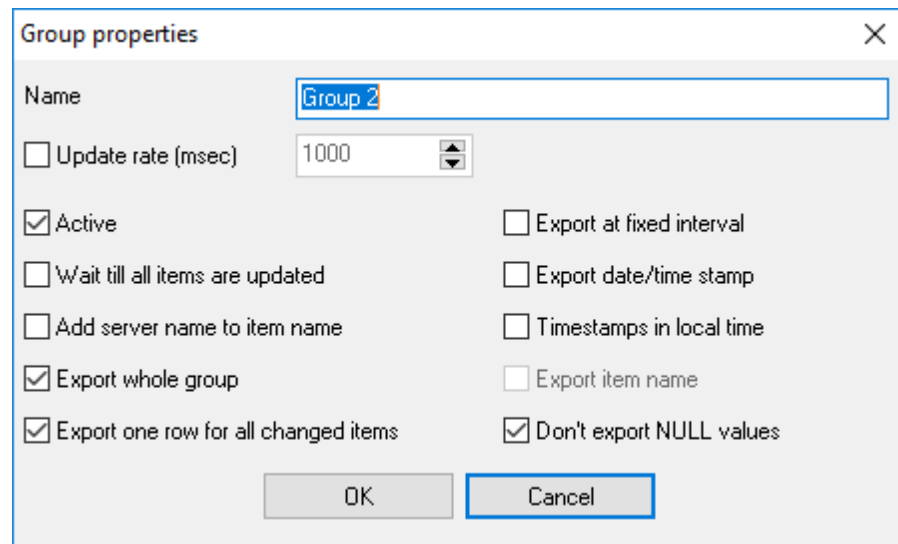


Fig. 2.8.3 Group properties.

Name – the field contains the name of the group. The name may contain any characters. By default, it is Group ###, where ### is the number of the group.

Update rate (msec) - the field contains the update rate value for the items in the group. If this option is enabled, the program will forcibly read the values of the DDE items from the server at the specified interval instead of waiting for the notification that the value of an item has been updated. The value is specified in milliseconds. The default is 1000 milliseconds, i.e. 1 second.

Active – the checkbox determines whether the group is active. If it is selected, the items from the group will be read from the server and exported according to the group properties.

Do not export NULL values - the checkbox makes the program not export the value of an item if

no item value is received from the server it the program fails to set it to the specified data type.

Wait till all items are updated - if this option is enabled, the items of the whole group will be exported simultaneously at the moment when the values of all items in the group are received. After all items in the group are exported, the internal flag in the program is reset and the program again starts to wait till all items in the group are updated.

Export at fixed interval - if this option is enabled, the values of the items will be exported even if they have not been modified.

Add server name to item name - if this option is enabled, the prefix with the name of the DDE server will be added to the name. It is necessary if the values of DDE items with the same name but from different servers are read.

Export whole group - if this option is enabled, the program will export all items in the group no matter whether their values have been modified or not.

Export one row for all changed items - if this option is enabled, the program will export all items in the group in one line, otherwise each item will be exported on a separate line. This option should be enabled if the data will be exported to a database where a separate column is created for every item. This option is available only if the "Export the entire group" option is enabled.

Export date and time stamp - if this option is enabled, the program will automatically add a separate item containing the current date and time during export.

For example: UPDATE_DATE_TIME[7]=2011-09-27 14:09:34

Timestamps in local time - if this option is enabled, the program will use the local time zone to export the date and time stamp, otherwise the time will be in the UTC format.

Export item name - this option is available when items are exported one at a time. If this option is enabled, the program will automatically add a separate item containing the item name during export.

For example: ITEM_NAME[8]="Item2Const";ITEM_VALUE[8]="Time=27.09.2011 0:00:00";
UPDATE_DATE_TIME[7]=2011-09-27 14:09:34

Adding an item

The "New DDE item" menu item allows you to add an item to the group. A click on this item opens the "Item properties" dialog box (fig. 2.8.4).

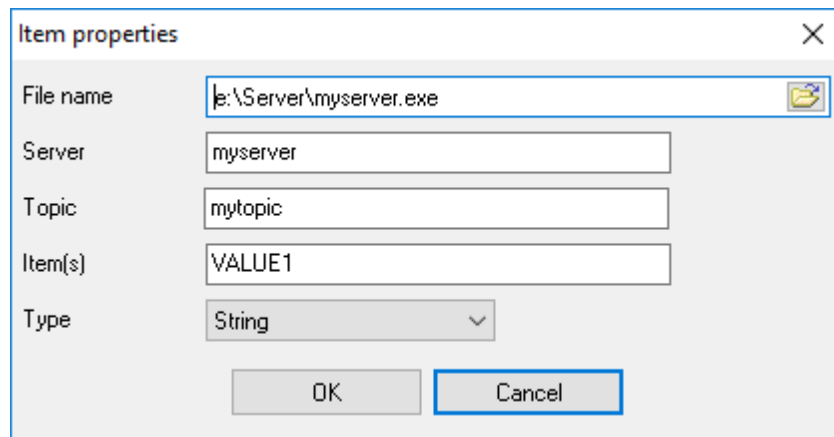


Fig. 2.8.4 Adding an item

File name - the path and name of the DDE server file.

Server - DDE server name.

Topic - describes the group of items on the DDE server.

Item - DDE item name. You can specify several items at once. For example, Item1;Item2;Item3

Type - item data type. The DDE interface works only with string values. But you can specify the data type the program should convert the received values to. Later you can apply the corresponding filter and export plug-ins depending on the data type.

Removing a group or an item

The "Remove" menu item (fig. 2.8.2) allows you to remove the selected group with all its items or the separate item selected in the group.

Editing

The "Edit" menu item (fig. 2.8.2) allows you to edit the properties of a previously added group or item.

Assigning a different item name

The "Assign another name" menu item (fig. 2.8.2) allows you to specify the item name that will be used for exporting. For example, there is an item named "Item1" on the server. You can assign another clearer name to this item or use it later in filter and data export plug-ins.

Moving a group or an item

The "Up" and "Down" menu items allow you to move the selected item one position up or down in the list. If there is only one group or item in the list or if there are no groups and items at all, the menu item will be unavailable.

4.6.7.3 Advanced parameters

This group of advanced parameters allows you to configure the options of interaction with the DDE server.

Automatically reconnect to server - if this option is enabled, the program will constantly keep checking that the connection to the server is not broken. If the program detects the the server has been closed or there is a failure, it will be restarted after the specified time interval.

If item does not exist, try to read it after - if this option is enabled, the program will not exclude items that do not exist on the server from its queries, but it will check their availability periodically. It may be necessary if the DDE server generates the list of DDE items dynamically during its operation.

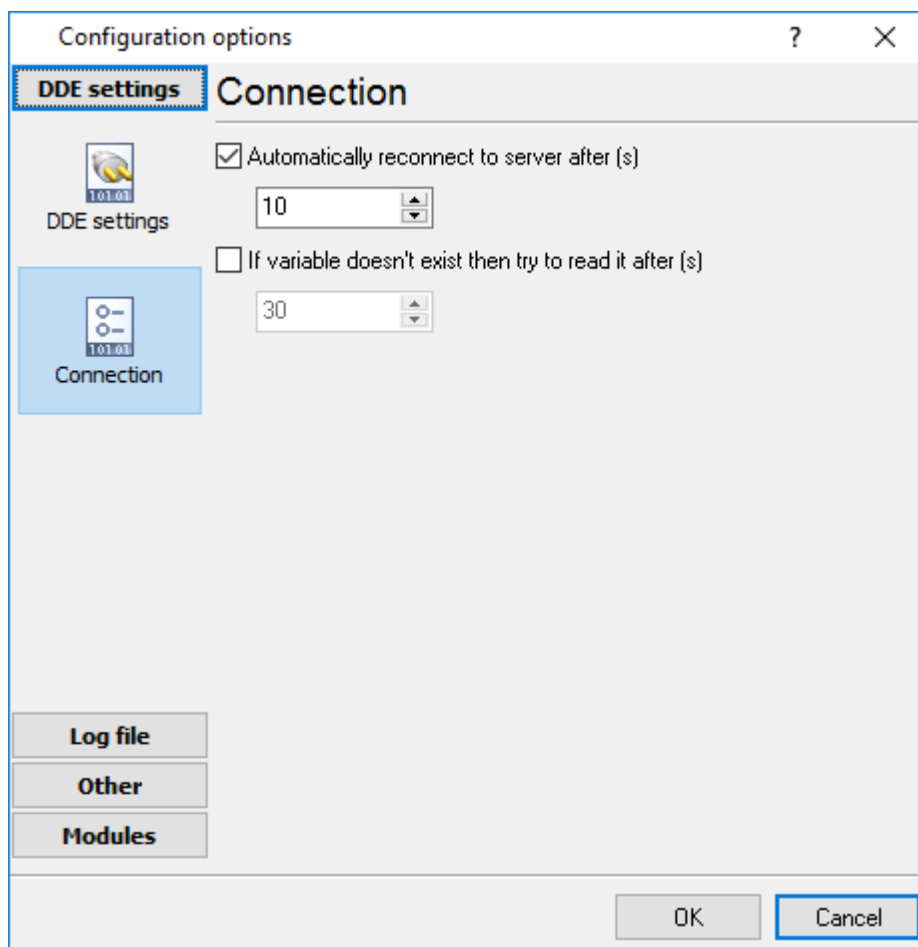


Fig. 2.8.5 Advanced parameters

4.6.7.4 Step-by-step instructions

4.6.7.4.1 Writing DDE to MSSQL

4.6.7.4.1.1 Writing a lot of DDE items

If you need to write a lot of DDE items to an MSSQL database, it is not effective to create a table with a separate column for each item. Because the values of items can be modified non at the same time, which will result in entries with a lot of empty columns generated in the database.

It will be more effective to create a table consisting of three columns

```
CREATE TABLE [dbo].[dde_data](  
    [REC_ID] [int] IDENTITY(1,1) NOT NULL,  
    [ITEM_NAME] [nchar](25) NOT NULL,  
    [ITEM_VALUE] [nchar](255) NULL,  
    [TIMESTAMP] [datetime] NULL  
) ON [PRIMARY]
```

ITEM_NAME - will be the name of the item;

ITEM_VALUE - will contain the string value of the item;

TIMESTAMP - will contain the date and time when the value was modified.

1. Create a new user in the database or give the permissions to write and read data from the created table to an existing user.
2. Create a group of DDE items in the program (fig. 1) with the properties shown in the picture.

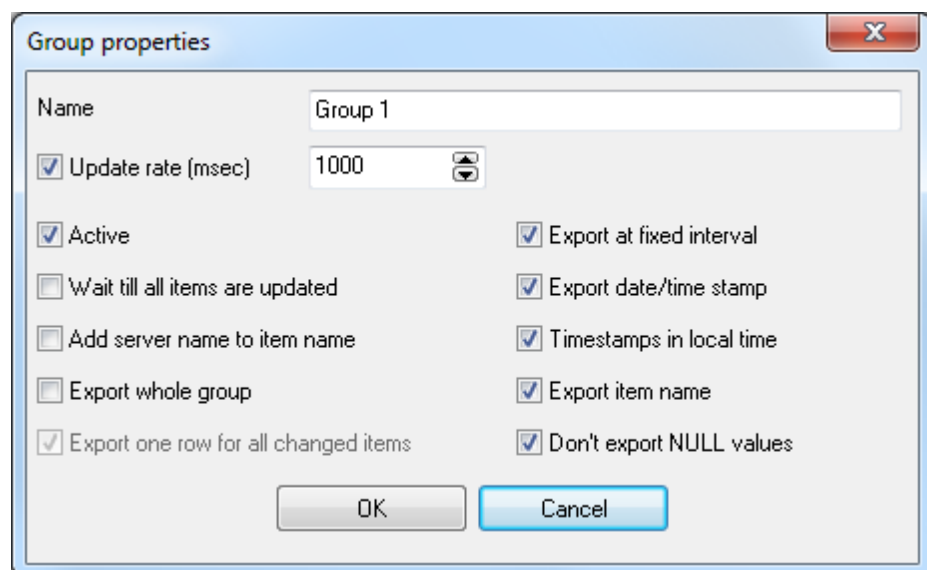


Fig. 1 Group properties

3. Add the necessary items to the group.

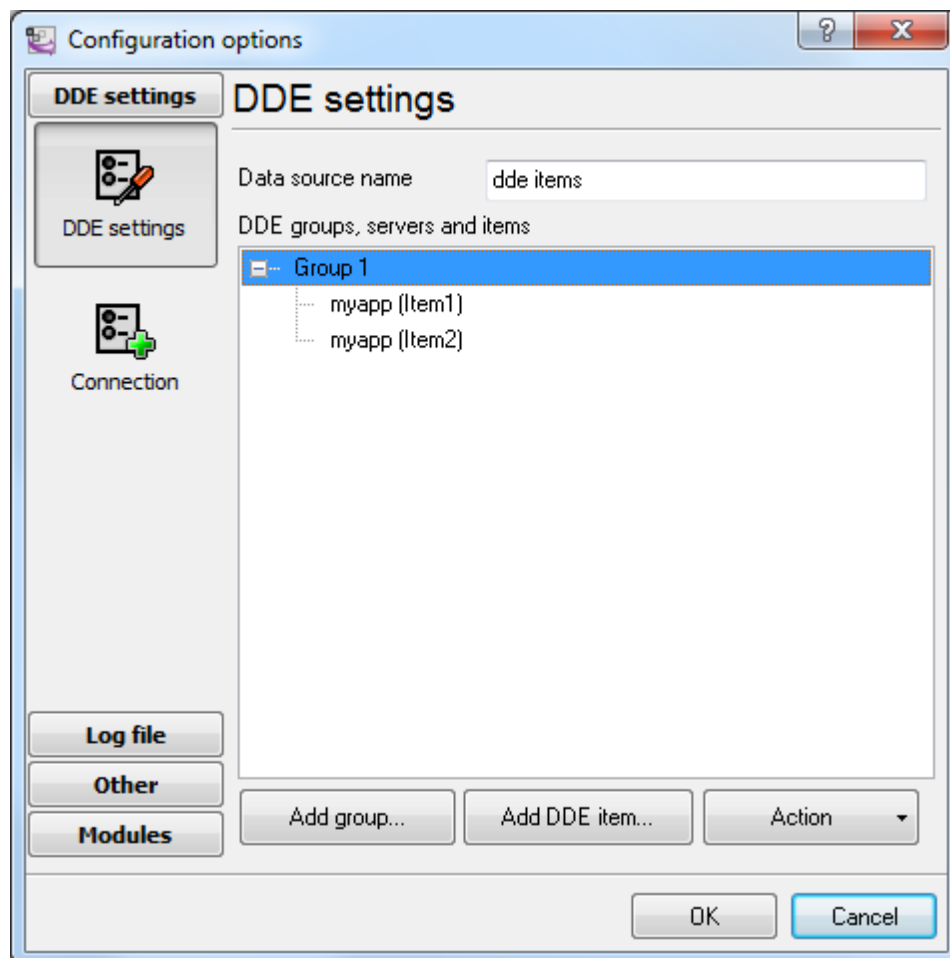


Fig. 2 Item list

4. Click OK. Data like that should appear in the main window of the program:

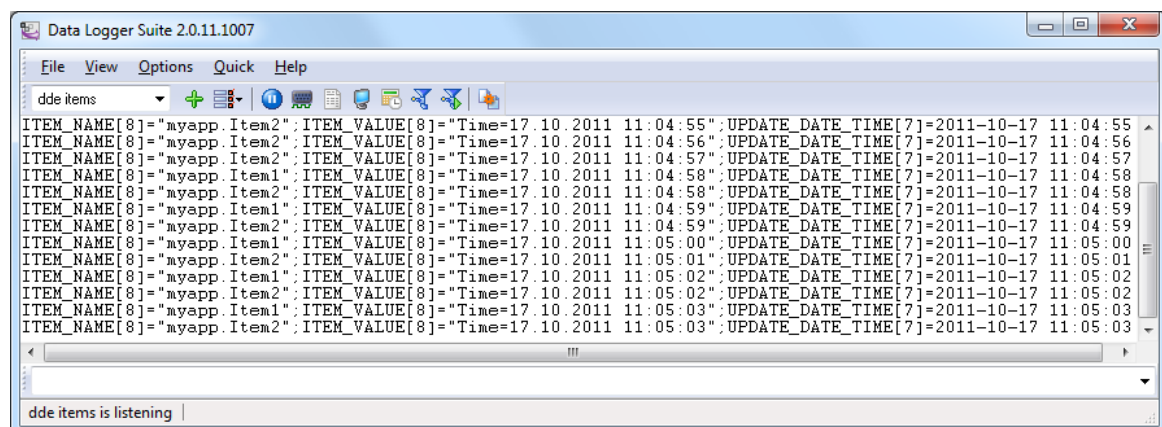


Fig. 3 Data

Every new value of DDE items appears on a new line here. Every line contains the additional "ITEM_NAME" and "UPDATE_DATE_TIME" items with the item name and timestamp, respectively.

5. Enable the parser for the data (fig. 4)

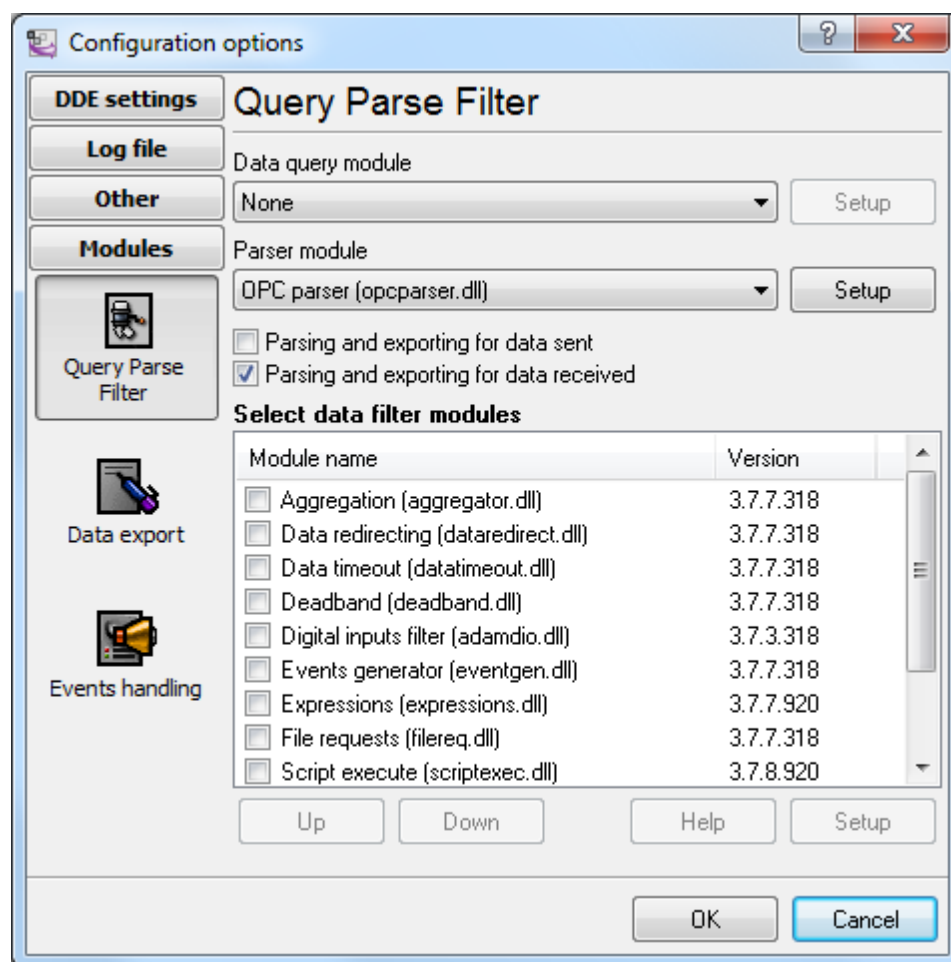


Fig. 4 Enabling the parser

6. Select the data export plug-in (fig. 5)

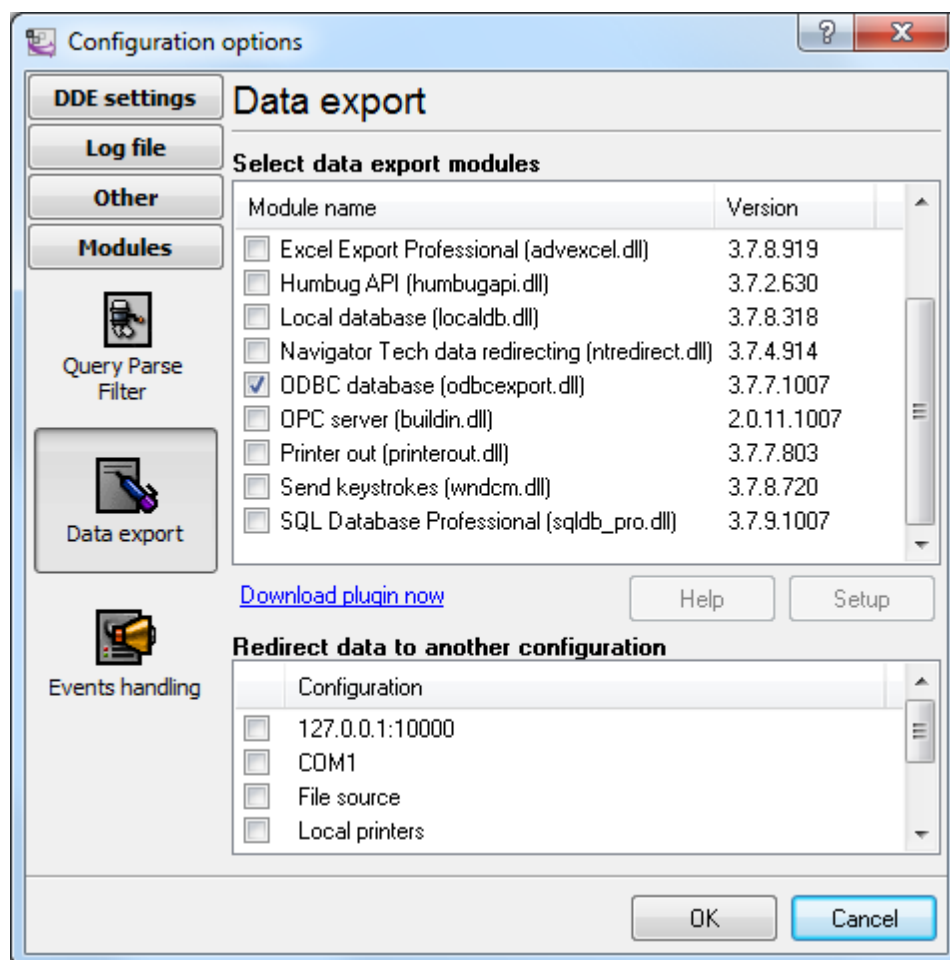


Fig. 5 Selecting the data export plug-in

7. Configure the data export plug-in (fig. 6-7)

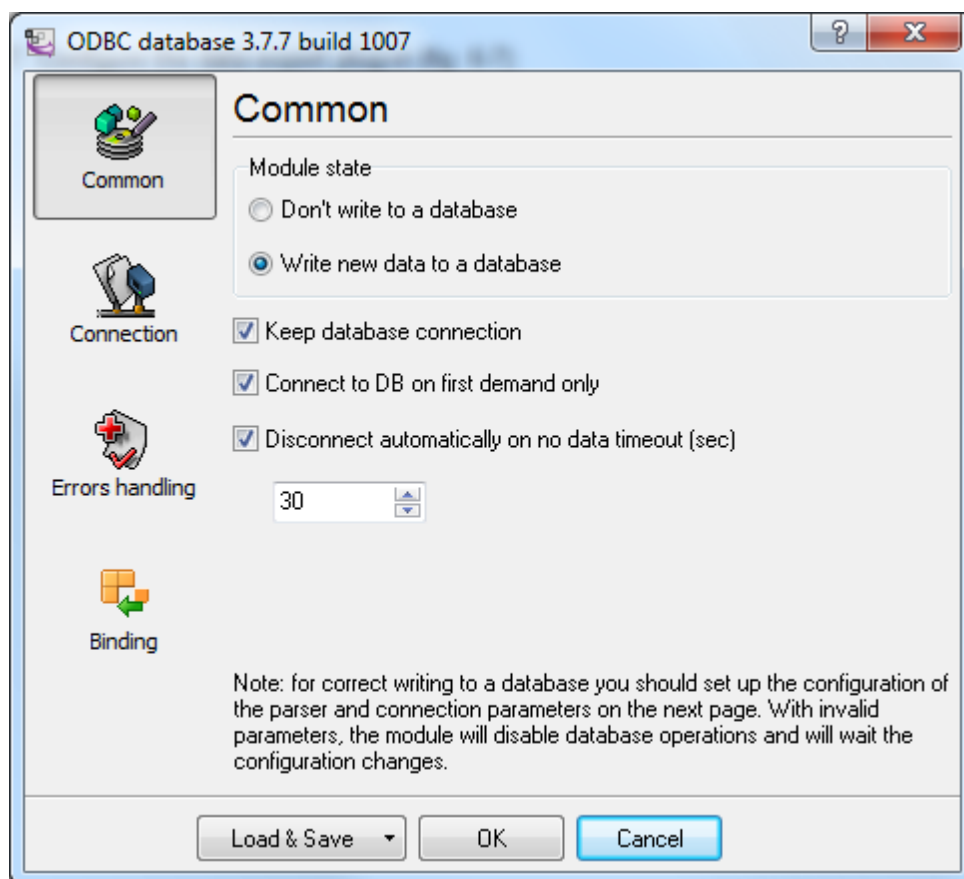


Fig. 6 Configuring the data export plug-in. General.

To set up a connection, you should create and configure the ODBC data source for connecting to your MSSQL database. Click the "Configure" button to do it. After you create the data source, click "Update" and select the data source from the list.

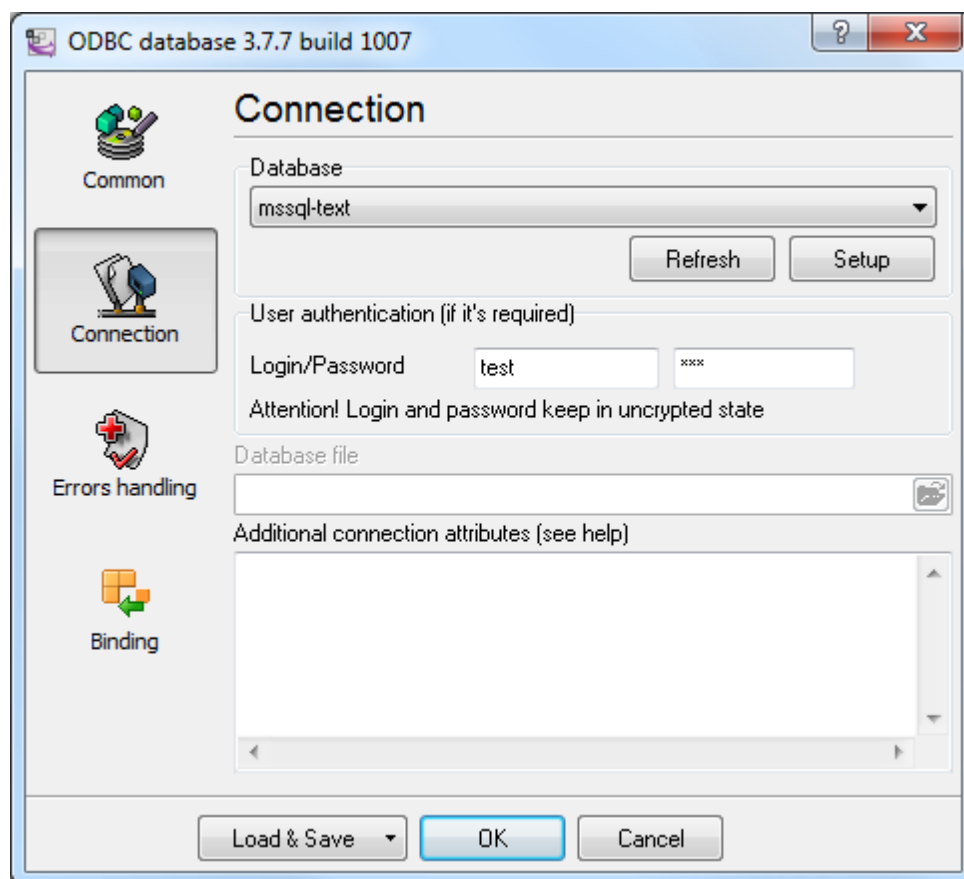


Fig. 7 Configuring the data export plug-in. Connection.

8. Binding (fig. 8) allows you to specify which data to which columns the program should add. You should specify the column name and "bind" the item to it from the main window of the program (the parser item). You should also specify the data type of the column.

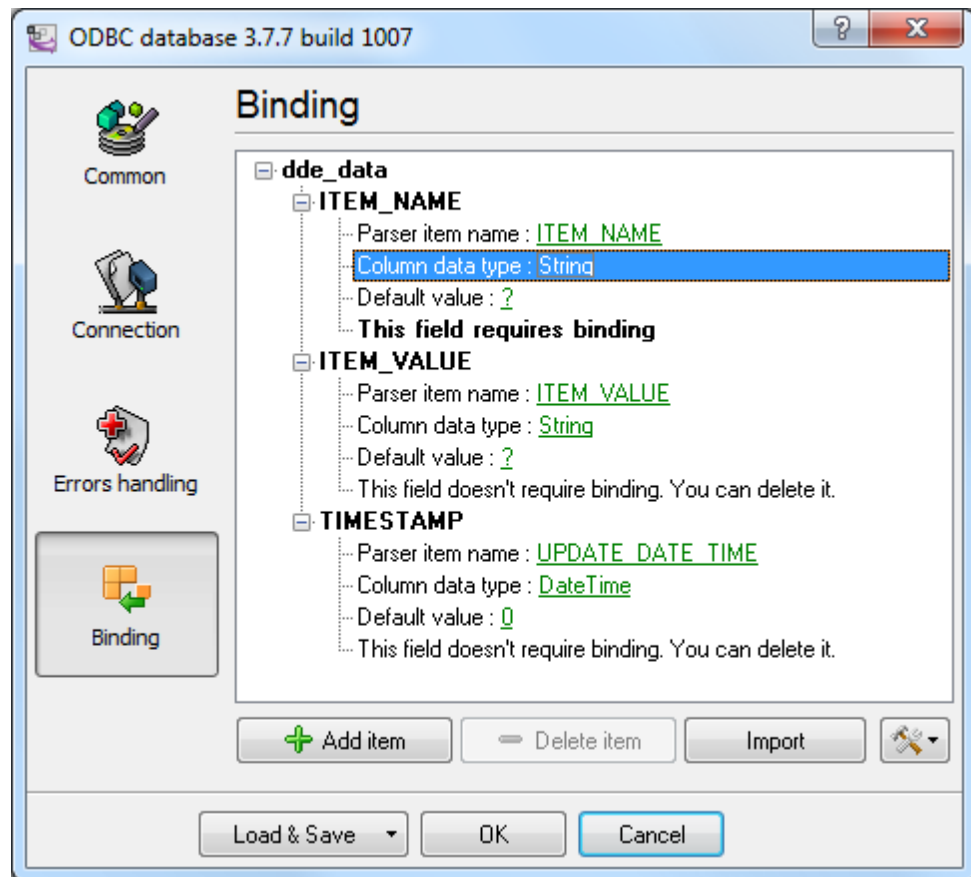


Fig. 8 Configuring the data export plug-in. Binding.

9. Click "OK" to save the changes.
10. Check the status bar to make sure the data is being successfully processed (fig. 9).

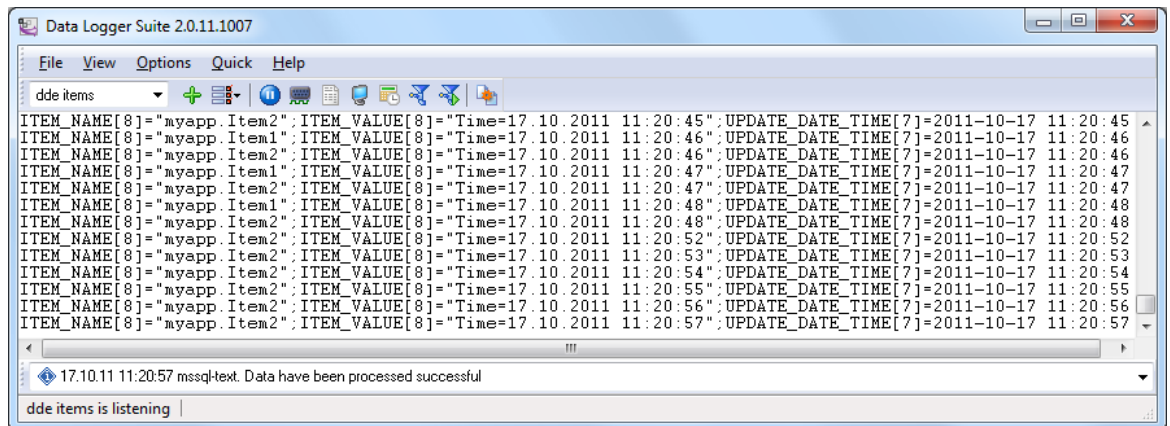


Fig. 9 A message about data being successfully written

	REC_ID	ITEM_NAME	ITEM_VALUE	TIMESTAMP
1	1	Item1	Time=27.09.2011 16:24:14	2011-09-27 16:24:14.000
2	2	Item2	Time=27.09.2011 16:24:16	2011-09-27 16:24:16.000
3	3	Item1	Time=27.09.2011 16:24:19	2011-09-27 16:24:19.000
4	4	Item2	Time=27.09.2011 16:24:22	2011-09-27 16:24:22.000
5	5	Item1	Time=27.09.2011 16:24:24	2011-09-27 16:24:24.000
6	6	Item2	Time=27.09.2011 16:24:27	2011-09-27 16:24:27.000
7	7	Item1	Time=27.09.2011 16:24:29	2011-09-27 16:24:29.000
8	8	Item2	Time=27.09.2011 16:24:32	2011-09-27 16:24:32.000
9	9	Item1	Time=27.09.2011 16:24:34	2011-09-27 16:24:34.000
10	10	Item2	Time=27.09.2011 16:24:37	2011-09-27 16:24:37.000
11	11	Item1	Time=27.09.2011 16:24:39	2011-09-27 16:24:39.000
12	12	Item2	Time=27.09.2011 16:24:42	2011-09-27 16:24:42.000
13	13	Item1	Time=27.09.2011 16:24:44	2011-09-27 16:24:44.000
14	14	Item2	Time=27.09.2011 16:24:47	2011-09-27 16:24:47.000
15	15	Item1	Time=27.09.2011 16:24:50	2011-09-27 16:24:50.000
16	16	Item2	Time=27.09.2011 16:24:52	2011-09-27 16:24:52.000
17	17	Item1	Time=27.09.2011 16:24:55	2011-09-27 16:24:55.000
18	18	Item2	Time=27.09.2011 16:24:57	2011-09-27 16:24:57.000

Fig. 10 Data in an MS SQL 2008 database

4.6.7.4.1.2 Writing several DDE items to separate columns

If you need to not only log the changes of the DDE item values in the MSSQL database, but also analyze and process them after that, it will be more convenient to create a table where there will be a separate column for every item. This method can be applied if there are not many DDE items.

```
CREATE TABLE [dbo].[dde_data_2](
  [REC_ID] [int] IDENTITY(1,1) NOT NULL,
  [TIMESTAMP] [datetime] NULL,
  [ITEM1] [nchar](30) NULL,
  [ITEM2] [nchar](30) NULL
) ON [PRIMARY]
```

ITEM1 - will store the value of item 1;

ITEM2 - will store the value of item 2;

TIMESTAMP - will contain the date and time when the value was modified.

1. Create a new user in the database or give the permissions to write and read data from the created table to an existing user.
2. Create a group of DDE items in the program (fig. 1) with the properties shown in the picture.

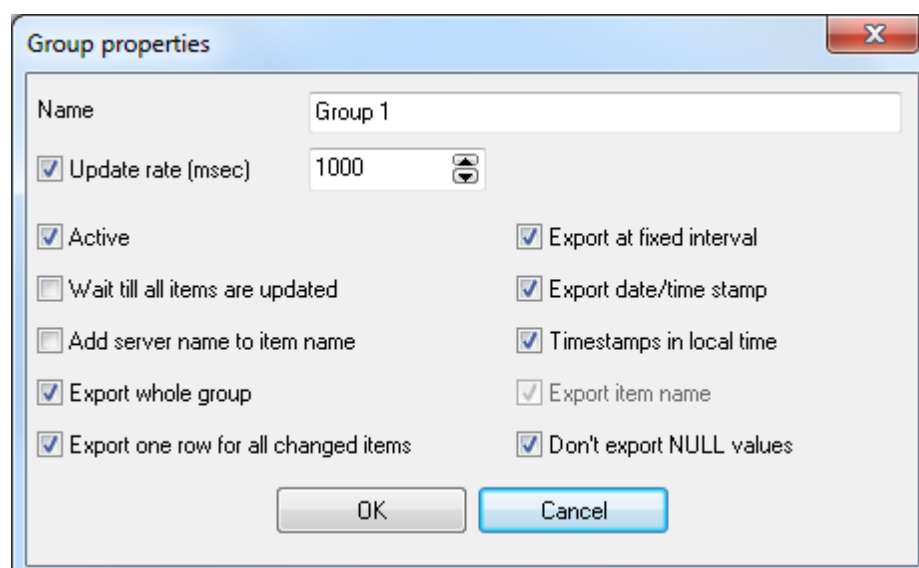


Fig. 1 Group properties

3. Add the necessary items to the group.

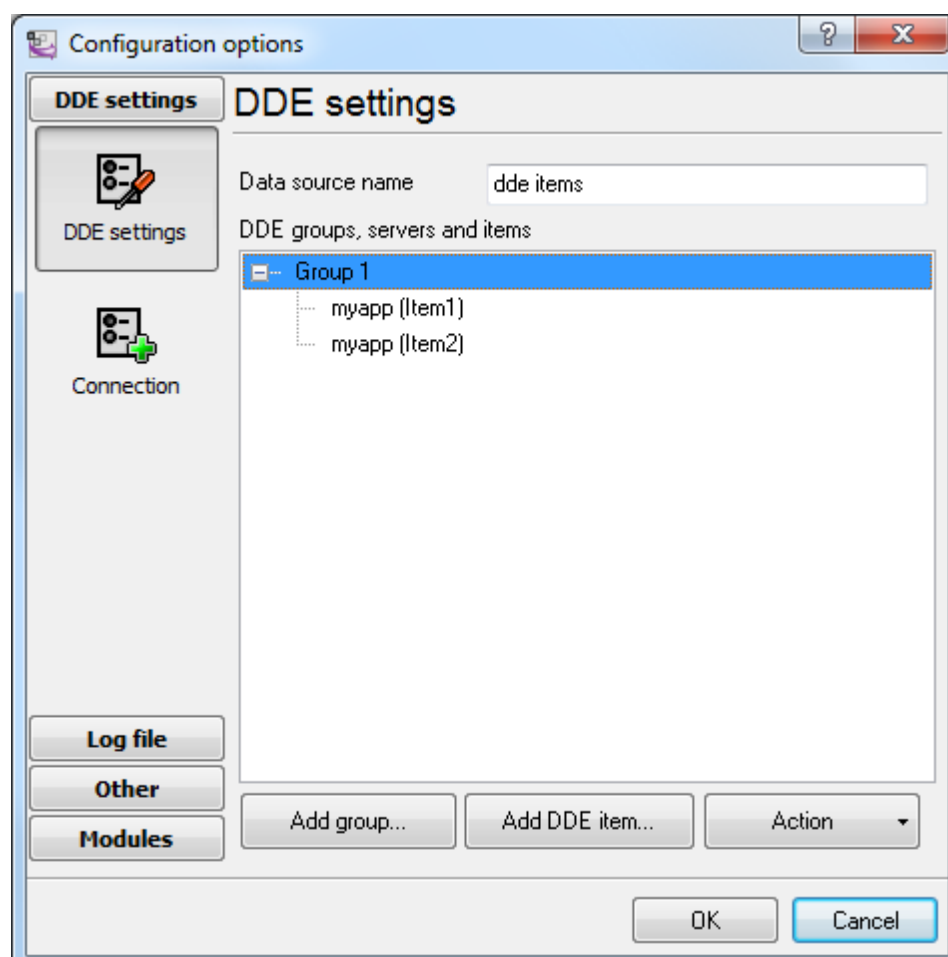


Fig. 2 Item list

4. Click OK. Data like that should appear in the main window of the program:

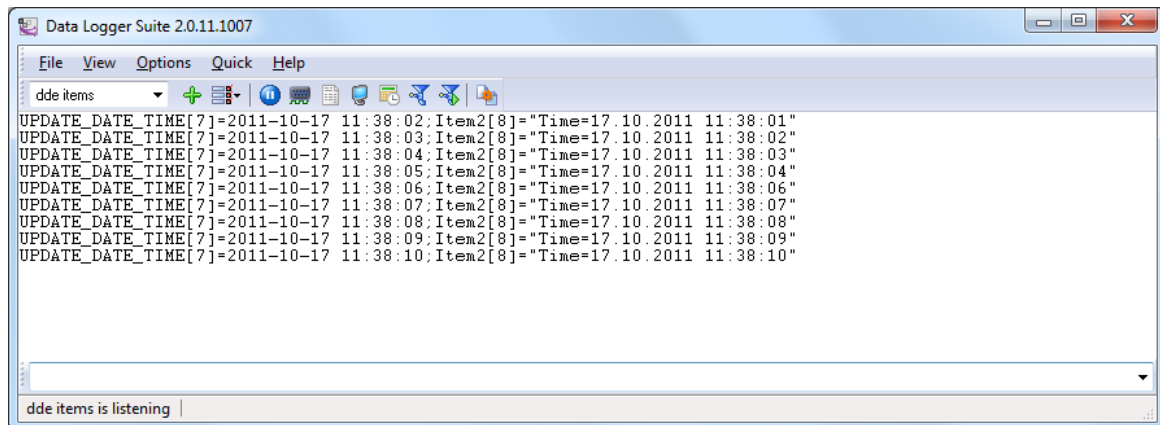


Fig. 3 Data

Every line contains the value of all items and the additional "UPDATE_DATE_TIME" item with the timestamp.

Steps 5-7 are similar to those from the previous example "Writing a lot of DDE items".

8. Binding (fig. 4) is slightly different from that in the previous example because the table contains a different set of columns.

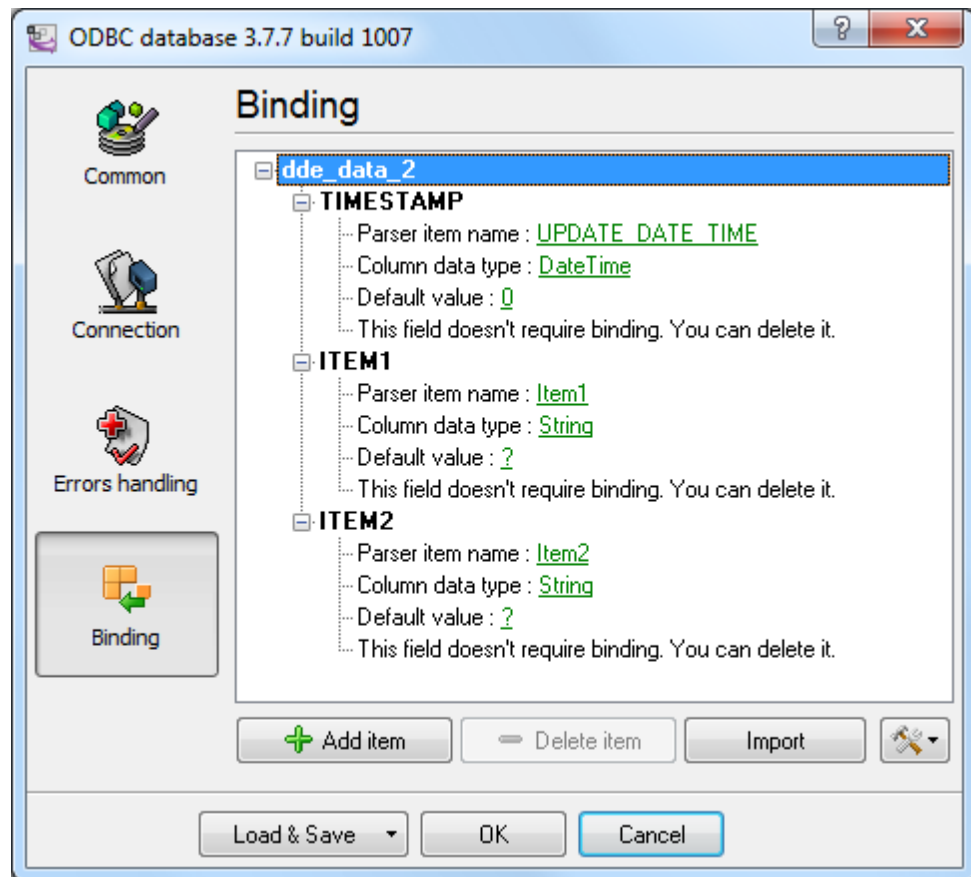


Fig. 4 Configuring the data export plug-in. Binding.

Click "OK" to save the changes.

9. Check the status bar to make sure the data is being successfully processed (fig. 5).

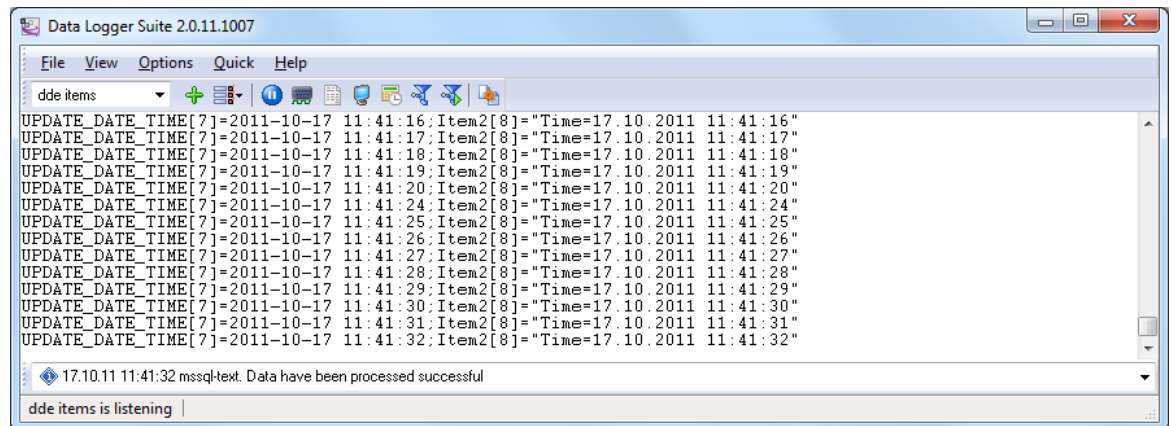
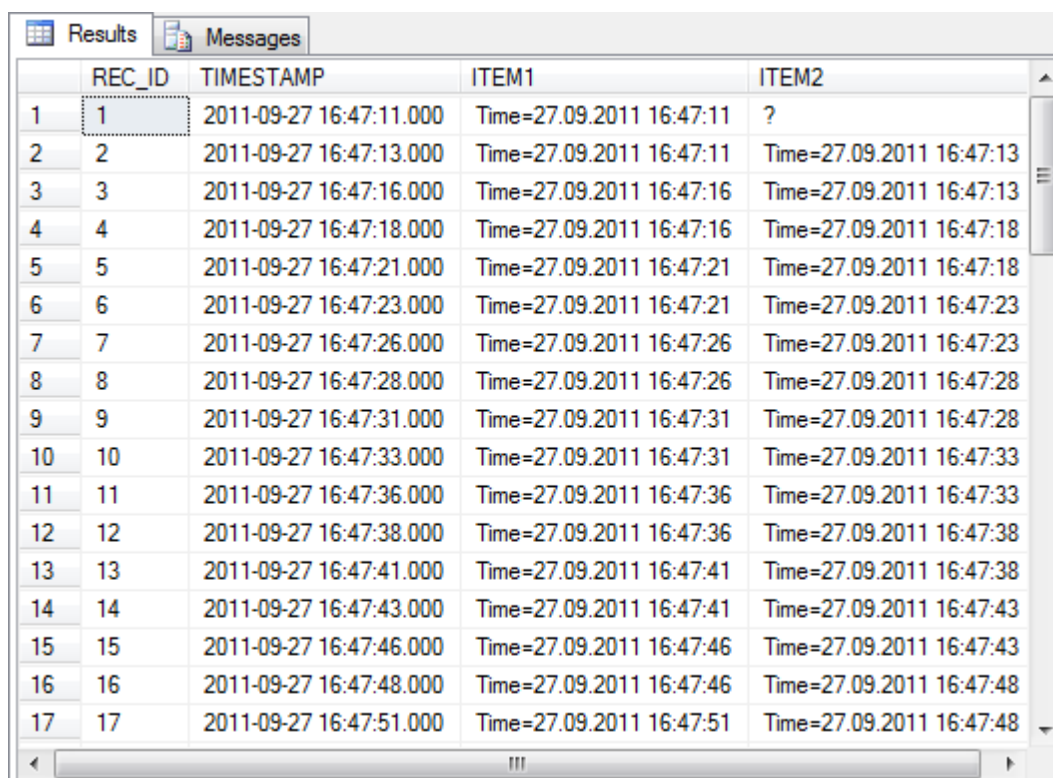


Fig. 5 A message about data being successfully written



	REC_ID	TIMESTAMP	ITEM1	ITEM2
1	1	2011-09-27 16:47:11.000	Time=27.09.2011 16:47:11	?
2	2	2011-09-27 16:47:13.000	Time=27.09.2011 16:47:11	Time=27.09.2011 16:47:13
3	3	2011-09-27 16:47:16.000	Time=27.09.2011 16:47:16	Time=27.09.2011 16:47:13
4	4	2011-09-27 16:47:18.000	Time=27.09.2011 16:47:16	Time=27.09.2011 16:47:18
5	5	2011-09-27 16:47:21.000	Time=27.09.2011 16:47:21	Time=27.09.2011 16:47:18
6	6	2011-09-27 16:47:23.000	Time=27.09.2011 16:47:21	Time=27.09.2011 16:47:23
7	7	2011-09-27 16:47:26.000	Time=27.09.2011 16:47:26	Time=27.09.2011 16:47:23
8	8	2011-09-27 16:47:28.000	Time=27.09.2011 16:47:26	Time=27.09.2011 16:47:28
9	9	2011-09-27 16:47:31.000	Time=27.09.2011 16:47:31	Time=27.09.2011 16:47:28
10	10	2011-09-27 16:47:33.000	Time=27.09.2011 16:47:31	Time=27.09.2011 16:47:33
11	11	2011-09-27 16:47:36.000	Time=27.09.2011 16:47:36	Time=27.09.2011 16:47:33
12	12	2011-09-27 16:47:38.000	Time=27.09.2011 16:47:36	Time=27.09.2011 16:47:38
13	13	2011-09-27 16:47:41.000	Time=27.09.2011 16:47:41	Time=27.09.2011 16:47:38
14	14	2011-09-27 16:47:43.000	Time=27.09.2011 16:47:41	Time=27.09.2011 16:47:43
15	15	2011-09-27 16:47:46.000	Time=27.09.2011 16:47:46	Time=27.09.2011 16:47:43
16	16	2011-09-27 16:47:48.000	Time=27.09.2011 16:47:46	Time=27.09.2011 16:47:48
17	17	2011-09-27 16:47:51.000	Time=27.09.2011 16:47:51	Time=27.09.2011 16:47:48

Fig. 6 Data in an MS SQL 2008 database

The first line contains a question mark in the "ITEM2" column. It is the default value specified in "Binding". It was used because the value of ITEM1 was received at the moment when a connection to the server was established, while the value of ITEM2 was not received yet. That is why an empty value was exported and it was replaced with the default.

4.6.7.4.2 Writing DDE to MySQL

Configuring how to write DDE data to a MySQL database is very much similar to configuring [how to write data to MSSQL](#)^[68]. Only main differences are described in this manual.

1. Download and install the 32-bit version of [MySQL ODBC Connector](#) (the ODBC driver) for MySQL from [mysql.org](#).

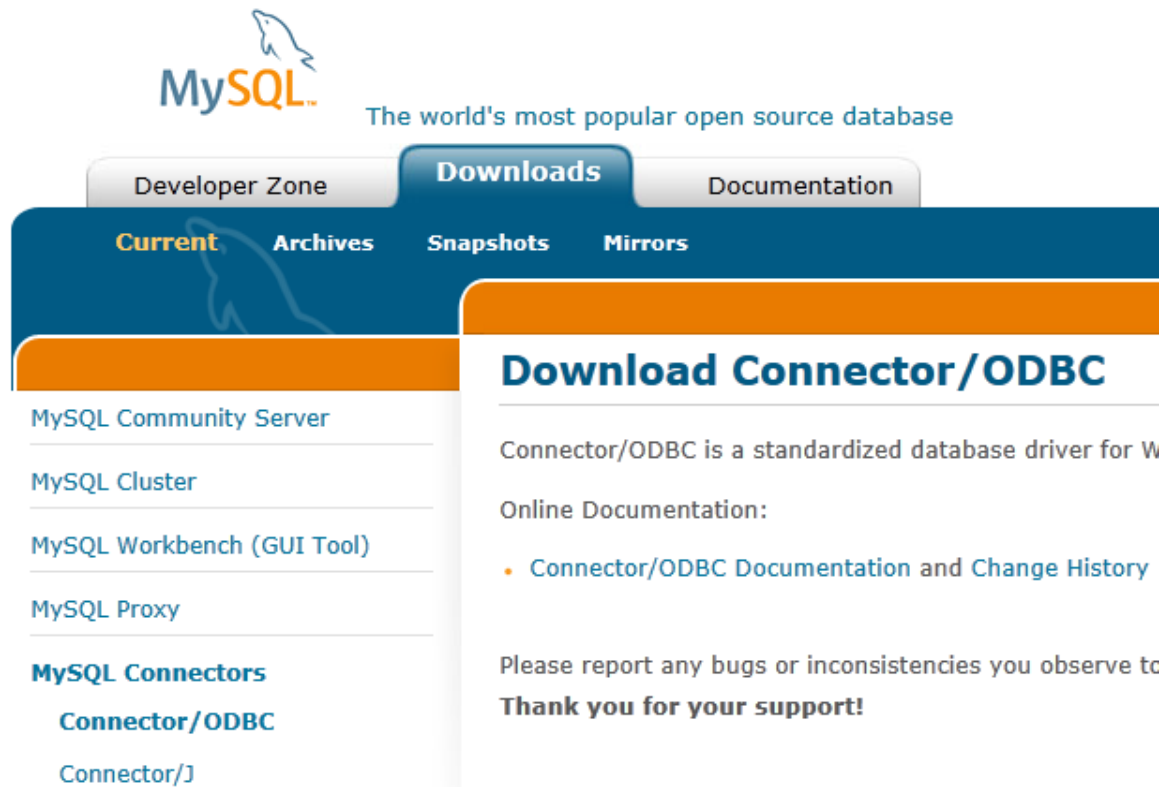


Fig. 1 Driver section

2. The structure of a table in the database will look like this:

2.1 In case of [writing a lot of DDE items](#)^[68]

```
CREATE TABLE `dde_data` (
  `REC_ID` int(11) NOT NULL AUTO_INCREMENT,
  `TIMESTAMP` datetime DEFAULT NULL,
  `ITEM_NAME` varchar(25) NOT NULL,
  `ITEM_VALUE` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`REC_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2.2 In case of [writing several DDE items](#)^[75]

```
CREATE TABLE `dde_data_2` (
  `REC_ID` int(11) NOT NULL AUTO_INCREMENT,
  `TIMESTAMP` datetime NOT NULL,
  `ITEM1` varchar(30) DEFAULT NULL,
  `ITEM2` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`REC_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

If you specify any other data type than String for ITEM1 or ITEM2 in the DDE settings, you should create the columns of the corresponding types while creating a table.

3. Create an ODBC data source for your MySQL database.

4. During "Binding", you should take into account that the names of the table and its columns are case-sensitive in MySQL.

4.6.7.4.3 Writing DDE items to a database

If you want to write data to a database other than MSSQL or MySQL described in the "[Writing DDE to MSSQL](#)^[68]" or "[Writing DDE to MySQL](#)^[79]" sections, you have the following options:

1. If there is an ODBC driver for this database, the configuration process is similar to that described in the "[Writing DDE to MSSQL](#)^[68]" or "[Writing DDE to MySQL](#)^[79]" sections. You will only have to create a table complying with the syntax of your database. For example, the Microsoft Access ODBC drivers come together with the operating system starting from Windows XP and you can use the Microsoft Access visual development tools to create a database;
2. If it is an SQLBase, Oracle, SQLServer, Sybase, DB2, Informix, Interbase, Firebird, MySQL or PostgreSQL database, you can use the "SQL Database Pro" data export plug-in (fig.1) instead of the "ODBC database" plug-in. You should download and install this plug-in separately after you install the program. In this case, you also [add DDE items](#)^[68] and select the parser, but you configure the "[SQL Database Pro](#)" plug-in instead of the "ODBC database" plug-in the way it is described in the [manual](#) on our website.

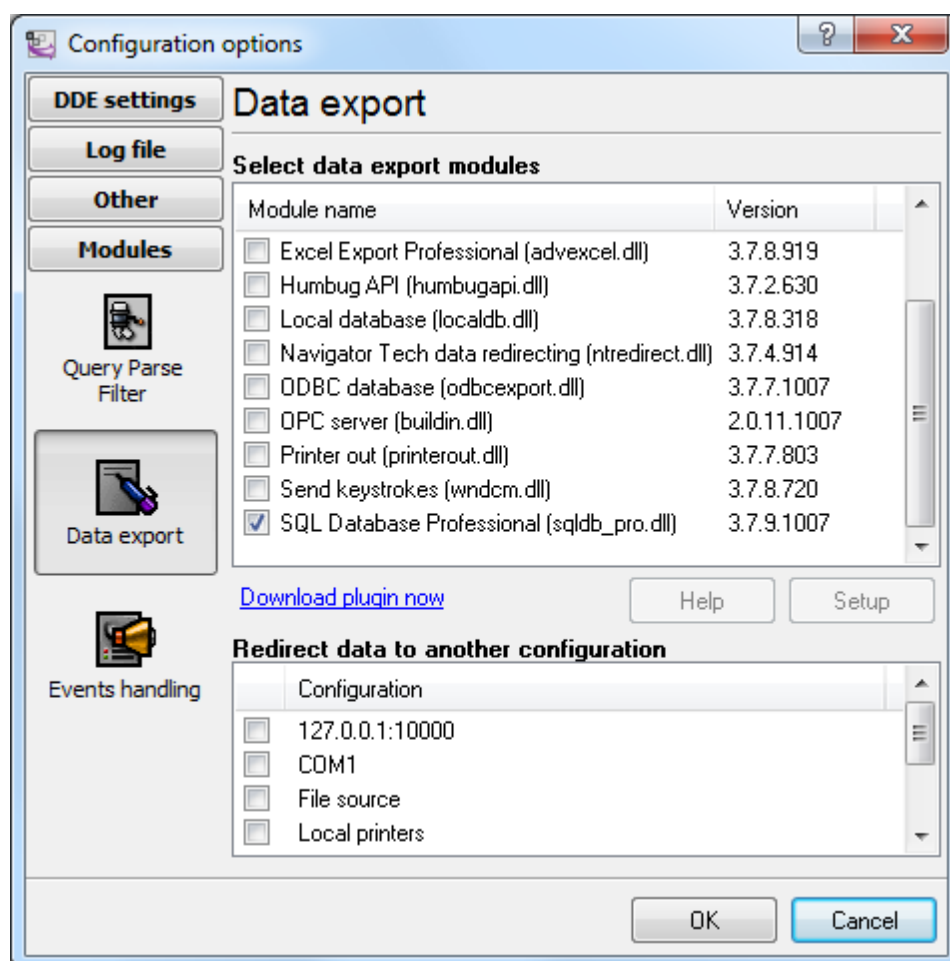


Fig. 1 Selecting the data export plug-in

4.6.7.4.4 Writing DDE data to Excel

Excel has built-in tools for getting DDE data. All you have to do is specify the formula like this for a cell:

```
=myserver|mytopic!myDDEitem
```

where:

myserver - DDE server name;
mytopic - topic name;
myDDEitem - DDE item name.

Once the item is updated on the server, it will be automatically updated in Excel.

This method has a few drawbacks

1. It is rather difficult to create the history of changes for the value of a DDE item. You will have to write VBA code for that.
2. Excel will work very slowly in case of a large number of items.

Our program offers an alternative way to generate files that can be opened in Excel and that will contain the history of value changes.

1. You should add the necessary DDE items the way it is described in the "[Writing DDE to MSSQL](#)" section (fig. 1-3)
2. Enable writing the received data to a log file of the CSV format (fig. 1). By default, the program generates files in the Excel format. You can customize the format of the CSV file if you click the "Advanced" button.

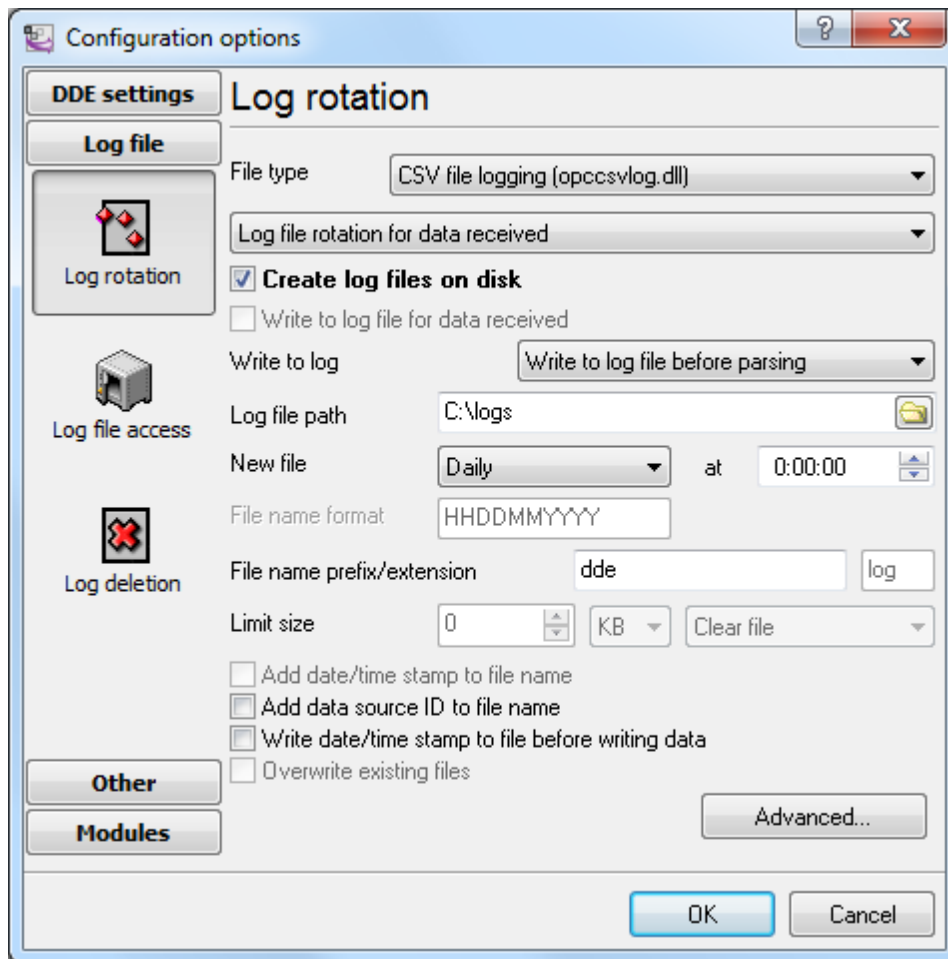


Fig. 1 Configuring the log file

Note 1. You should take into account that the "C:\logs" folder must be available for writing. Do not try to write log files to the root folder of the C:\ drive, the "Program Files" folder of the "Windows" folder. Starting from Windows Vista, only the Administrator can write to them.

Note 2. If you use the program as a Windows service, by default the service will be run by the system account that has restricted permissions to create and write files.

4.6.8 SQL data

4.6.8.1 Introduction

The SQL data interface allows to poll a database table in SQL-compatible databases (such as SQLBase, Oracle, Microsoft SQLServer, Sybase, DB2, Informix, Interbase, Firebird, MySQL, PostgreSQL) for new data and send the new data to other targets using data export plugins. For example, you can build SQL-to-DDE or SQL-to-OPC bridges, export SQL data to a text file and upload it to a server.

To access databases, the direct driver access methods provided by the developer of the corresponding database are used. It makes it possible to reduce system requirements, lower the traffic between database clients and servers, and, what is more important, use features unique for

every database (for example, stored procedures in Microsoft SQLServer or Oracle).

If our module does not support direct access to some database, it is possible to work through the ODBC driver that is also supported by our module. Unlike our other module for ODBC Databases, this module provides broader opportunities for controlling the database connection and the process of executing databases operations and implements quite a new approach to data publishing.

In this module the user can to create SQL queries himself.

4.6.8.2 Connection parameters

The connection parameters described below configure the connection on the software level. You can set these parameters on the "**Connection parameters**" tab (fig. 2.10.1).

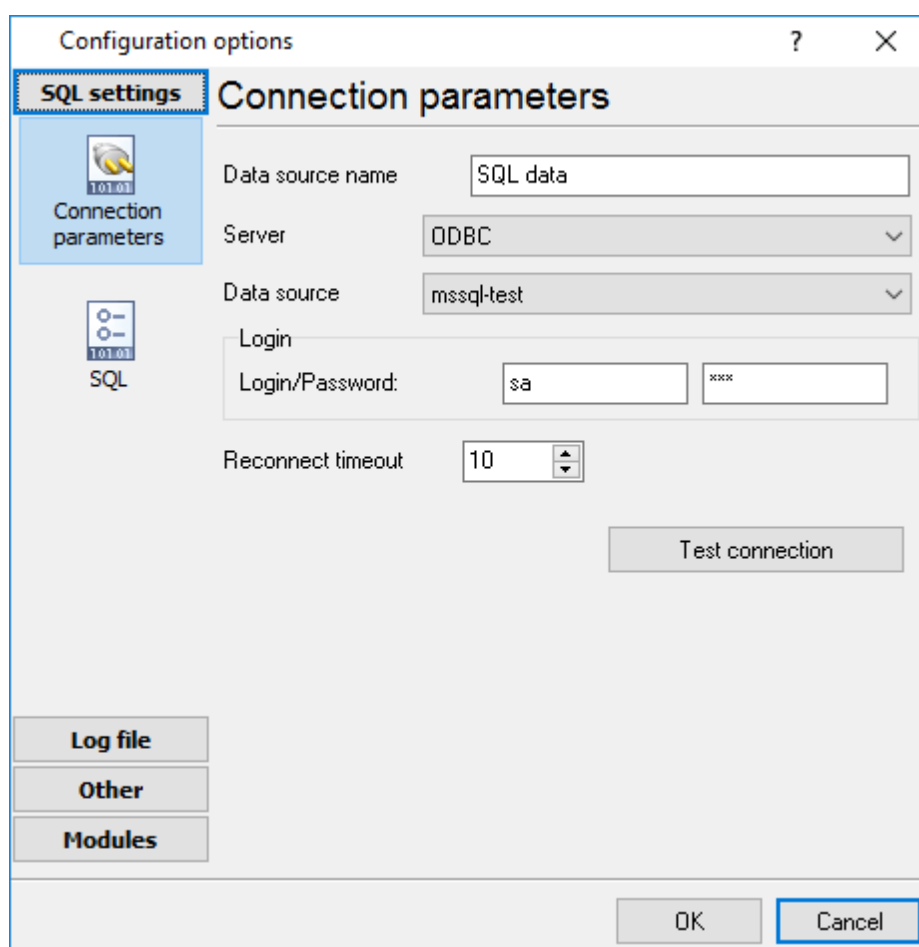


Fig. 2.10.1 Connection parameters

Using the "**Stay connected**" and "**Idle timeout**" options, you can specify the connection type. The "Stay connected" option makes the module connect to the database once needed and maintain the connection until the program is closed. The "Idle timeout" option makes the module disconnect if no data is exported for the number of seconds.

Server - database type.

Depending on the selected database type, use the "**Database**" field to specify the following:

- **DB2, Informix and ODBC** – specify DSN set in the "ODBC Administrator" or the complete description of the data source with the parameters supported by the selected server. You can see an example of this field for the Informix server below:

```
SERVICE=ids_srv;HOST=yourhost;PROTOCOL=OLSOCTCP;SERVER=ids_srv;
DATABASE=sysmaster;UID=informix;PWD=informix.
```

- **Interbase** – specify the path to the necessary database and the network protocol (see the examples below).

Value	Protocol
<server_name>:<filename>	TCP
\\<server_name>\<filename>	NetBEUI
<server_name>@<filename>	SPX

- **Oracle** – specify host name/service name.
- **MySQL, MS SQL Server or Sybase SQL Server** – if you establish a connection with a remote server, specify its name and the database name separated with a colon. For example, remsrv.dbname points to the DBNAME database situated on the REMSRV sever. Specify **(local)** for a local database. For example, 192.168.1.1:server_name

Then specify username and password to access the database in the "Login" and "Password" fields respectively.

For MS SQL Server you may not specify the login and password. In this case the program will use Windows Authentication.

For ODBC you may not specify the login and password. In this case the program will use login and password from the ODBC alias settings.

Use the "Additional connection attributes" field to specify connection parameters unique for each server.

Value	Description	Note
AUTOCOMMIT	Use autocommit	
APPLICATION NAME	The name of the application that will be sent to the server	Only for MSSQL and Sybase
HOST NAME	The name of the workstation that will be send to the server	Only for MSSQL and Sybase
COMMAND TIMEOUT	The number of seconds to wait until any operation is finished	Only for MSSQL, ODBC, SQLBase, Sybase
COMPRESSED PROTOCOL	Use compression while exchanging data between the client and the server. By default, the value is TRUE	Only for MySQL
ENABLE BCD	Change the NUMERIC data type into the BCD data type before sending data to the server	Only for Oracle, Interbase

ENABLE INTEGERS	Change the NUMERIC data type into the INTEGER data type before sending data to the server	Only for Oracle, Interbase
ENABLE MONEY	Change the NUMERIC data type into the CURRENCY data type with the precision (1-4) before sending data to the server	Only for MySQL
ENCRYPTION	Use encrypted passwords when accessing the database. By default, this value is FALSE	Only for Sybase
FIELD REQUIRED	Display an error message if any field has the NULL value when a query is executed	
FORCE OCI7	Use OCI7 (SQL*Net 2.x - Oracle7 interface) to access the Oracle server	Only for Oracle
LOCAL CHARSET	Set the encoding character set	Only for Interbase
LOGIN TIMEOUT	The number of seconds to wait for user authorisation	Only for DB2, Informix, ODBC, MSSQL, MySQL, Sybase
MAX CURSORS	The maximum number of simultaneously opened cursors	Only for MSSQL and Sybase
MAXCHARPARAMLEN	The maximum line length. By default, it is 255	
MAXFIELDNAMELEN	The maximum length of a field name. By default, it is 50	Only for Oracle
MAX STRING SIZE	Limit the size of strings to this value. Longer strings will be considered a blob	Only for Firebird, Interbase, ODBC
NEW PASSWORD	Use this value when the server returns the 'Password expired' message	Only for Oracle8
QUOTED IDENTIFIER	Use identifiers in quotes	Only for MSSQL and Sybase
PREFETCH ROWS	The number of rows to be prefetched in order to minimize network traffic (Oracle8: this option does not work if SELECT contains fields of the LONG type)	Only for DB2, Informix, ODBC, Oracle8
ROLE NAME	Specifies the role the server should assign to the client when it is connected	Only for Interbase and Oracle (SYSDBA/SYSOPER roles)
SERVER PORT	Specifies the server port for connecting via TCP/IP	Only for MySQL, PostgreSQL
SINGLE CONNECTION	Specified whether to use a single process/connection. By default, it is FALSE	Only for MSSQL and Sybase
SQL DIALECT	Installs SQL Dialect (1,2,3) for the client	Only for Interbase
TDS PACKET SIZE	Specifies the size for a TDS packet. If the server does not support this size, a "Login failed" error will occur in the process of connecting	Only for Sybase
TRANSACTION LOGGING	If it is FALSE, transaction logging will be disabled, so rollback will be unavailable	Only for SQLBase
RTRIM CHAR OUTPUT	Delete spaces on the right for fields of the CHAR type. By default, it is TRUE	Only for DB2, Informix, Interbase, Oracle, ODBC and Sybase
XA CONNECTION	Indicates that it is necessary to connect to the TM service with the name specified in the "Database name" field. By default, it is	Only for Oracle8i

	FALSE	
XXX API LIBRARY	Specifies the interface library type to use for connecting, where XXX stands for server type. For example, Oracle, SQLServer, Interbase, etc.	

After you set up your database connection, you can immediately test it by clicking the "**Test connection**" button. The program will try to connect to the database. The process can take rather long (up to three minutes) depending on the database type. A message will be displayed as the test result. If an error occurs, the message will contain the server response that will help you find out what has caused the error.

4.6.8.3 SQL

The program can retrieve data using two methods:

1. **Table name.** In this case, then program will read all columns from the specified table. The program will automatically build the SELECT SQL query.
2. **SQL statement.** You can specify your own SQL statement with necessary columns. Also you can query data from several tables. For example:

```
SELECT COL1, COL2 FROM DATA
```

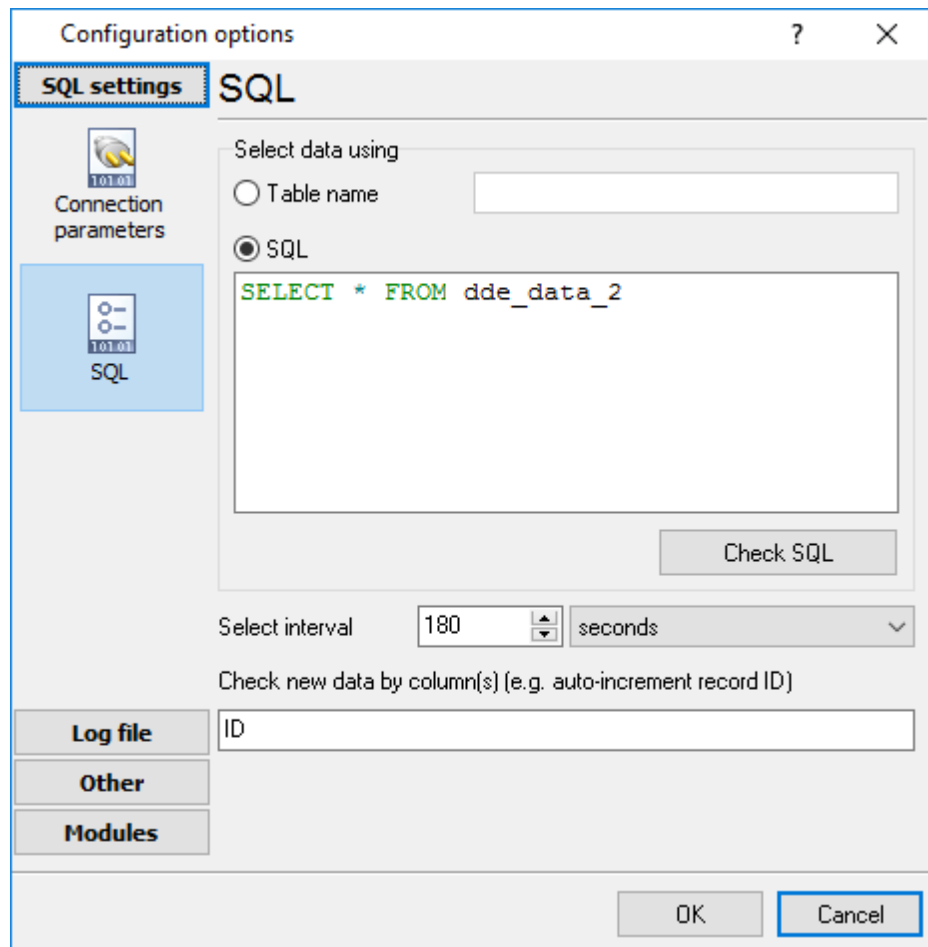


Fig. 2.10.2 SQL

Select interval - the program will execute the selected query the every specified interval.

Check new data by column - you can specify a column name that the program will use to select new data only. After each query the program will store a maximum value and next time will select new data only. If you'll not specify a column name then the program will read all data from a database with every query. It can be useful, if your table contains just few records that are being updated periodically.

4.6.9 USB HID

4.6.9.1 Selecting USB devices

This tab contains the list of HID's. To enable monitoring a device, select it in the list. Note that a USB device can be compound, i.e. one physical device may represent several logical devices. The HID list contains logical devices.

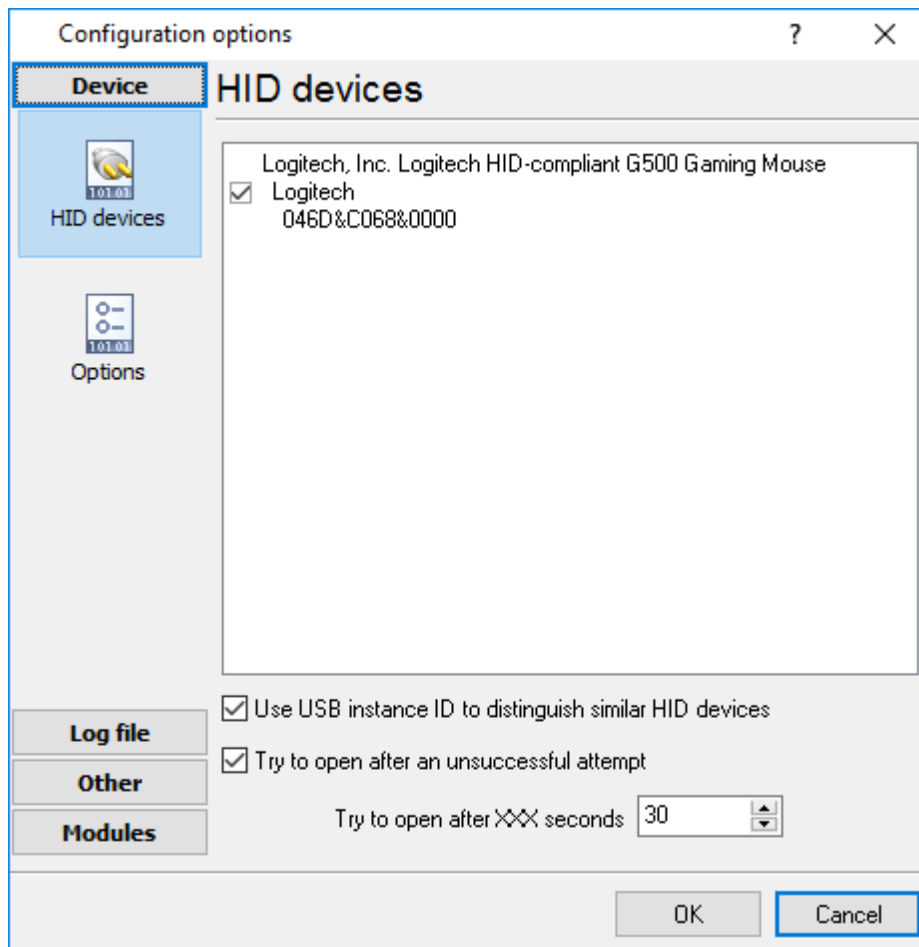


Fig.1 HID list

Use the instance number to differentiate between similar devices – enable this option if you have several similar devices connected to the computer. In this case, the program will use the number of the physical USB port to differentiate between these devices. However, the devices need to be always connected to the same ports. If you have only one HID, you can disable this option. In this case, the program will control this HID model connected to any port.

Try to open again after a failed attempt – if this option is enabled, the program will wait till the device appears and automatically start logging data from it in case the HID is connected and disconnected periodically.

4.6.9.2 HID data log options

Data between the computer and a USB device is transmitted over the USB as binary data packets called HID reports. Our program can log both these binary data packets and already decoded values.

Log decoded data – the program transform data packets into a readable form

Log VID and PID – if this option is enabled, the program will add manufacturer and HID model IDs to every decoded data packet.

Log the USB device name – if this option is enabled, the program will add the name of the USB device to the decoded data packet.

One value per line – this option changes the format used to log decoded values and makes it possible to write values received from several different devices to the database. If you are going to log data to a plain text file, it is recommended to disable this option.

Log raw data – if this option is enabled, the program will log unprocessed (RAW) data.

Log mouse moves – mouse moves generate a lot of USB packets. If you want to log data about every mouse gesture, enable this option.

4.6.9.3 Samples

[Configuring writing data from a USB HID to the database;](#)
[USB HID to MS SQL \(writing a large number of variables\);](#)
[USB HID to MS SQL \(writing several variables\);](#)
[Configuring writing data from a USB HID to MySQL;](#)
[Exporting data from a USB HID to Excel in real time.](#)

4.6.10 Additional parameters

4.6.10.1 Data view change

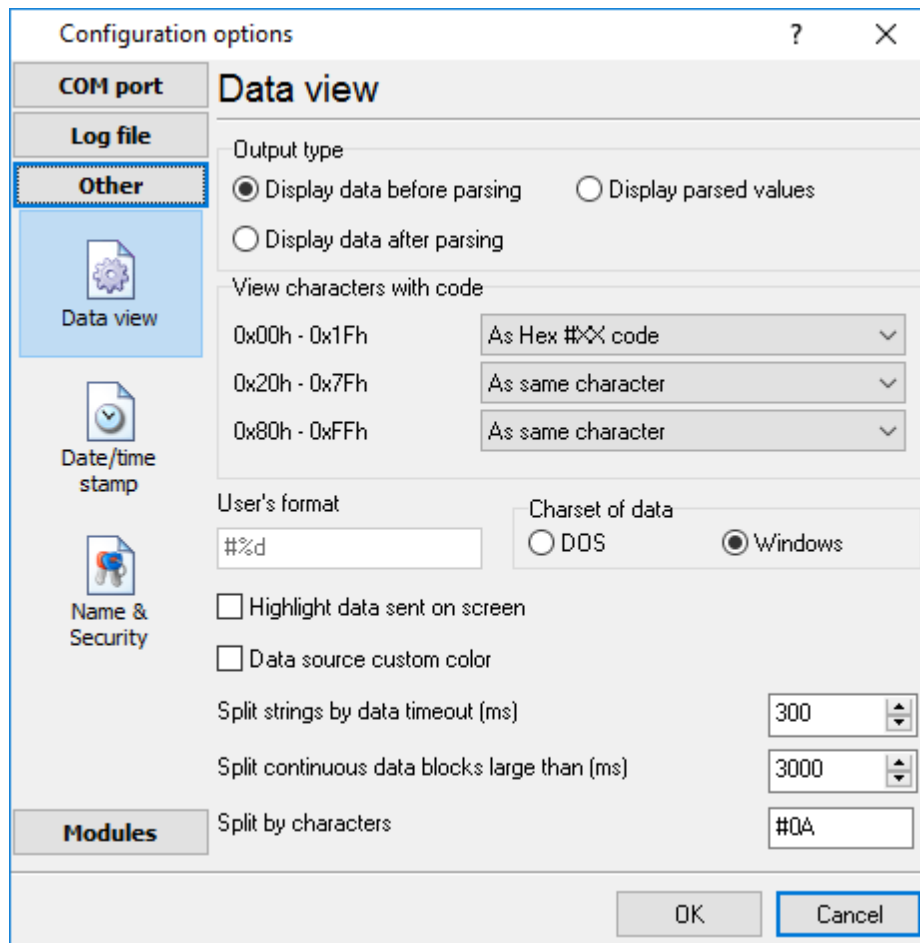


Fig. 3.1.1. Data view

Data view settings, that can be configured on the "Data view" tab:

1. **View characters with code** - the program can interpret and decode bytes as characters. You can select decoding mode for each characters range. If the range doesn't have the corresponding character, that's why these data can be displayed only in hexadecimal and decimal code.
2. You can set up data byte display **users format**. The directive %d shows to display an decimal code, the directive %x - hex code. You can set any framing characters before/after the user format.
3. **Highlight data sent on screen** - string with sent data will be highlighted by the set color.
4. **Character set** - allows to define the character set of incoming data. Windows - Windows ANSI character set, DOS - OEM character set.
5. **Data source custom color** - if you've created several configurations then you can define a custom color for each data source that allows to distinguish data flows on the "All data" page in the main window.
6. **Split strings by data timeout** - this option allows to visually split data packets in the program

window. A data packets that will be received after the specified interval will be showed on a new line. If this value is set to 0 then data packets will not split.

7. **Split continuous data blocks large than** - this option allows to visually split continuous data flow in the program window. The program will show data from a new line every specified interval.
8. **Split by characters** - this option allows to visually split continuous data flow in the program window using the specified symbols. For example (fig. 3.1.1), the program will use a character with the 0Ah hexadecimal code that is equal to the "LF" ASCII code.

4.6.10.2 Date/time configuration

This group of options (fig. 3.2.1) allows you to configure how data and time stamps appear in the log file and on the screen. You can configure the stamp format in the [program options](#)^[109].

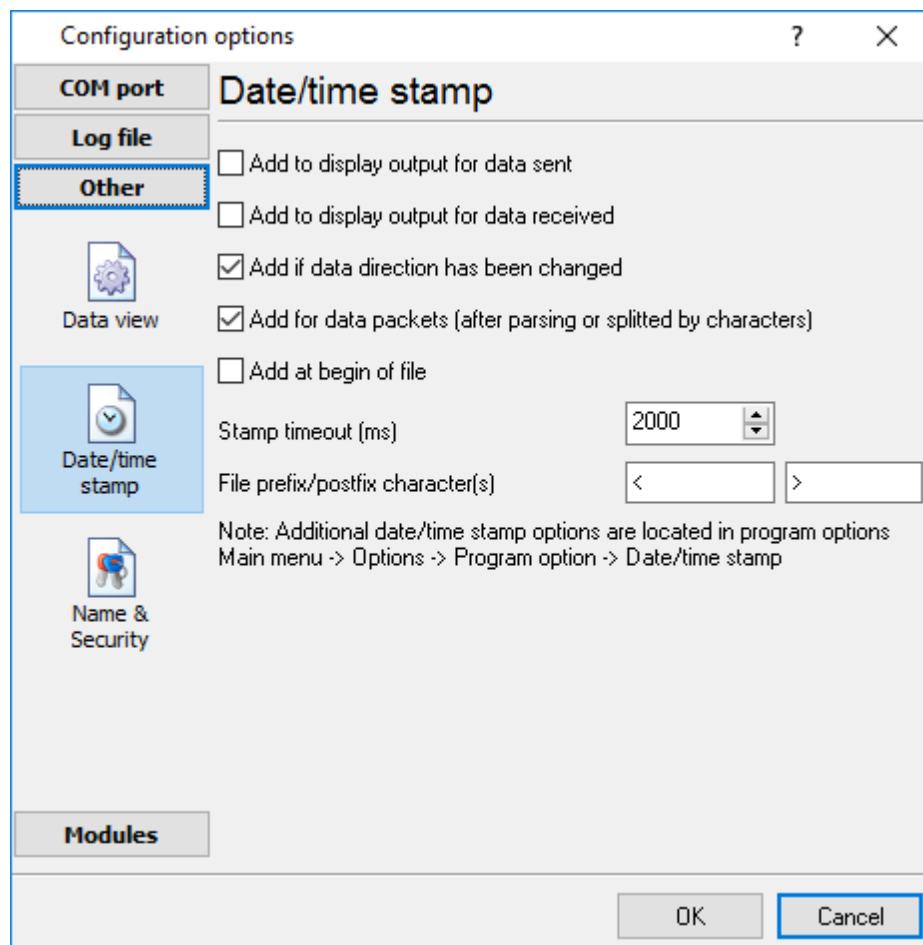


Fig. 3.2.1 Time stamp configuration

Add to display output for data sent - the time stamp will be added for the sent data displayed on the screen. The stamp will be added according to the timeout (if the data flow is uninterrupted) or when a data packet is sent.

Add to display output for data received - the same but for the received data.

Add if data direction has been changed - if the program is sending and receiving data, the time

stamp will be also added when the data transfer direction changes (sending/receiving).

Add for data packets - if the data is displayed after it is processed, the stamp will be added to each processed data packet.

Add at begin of file - the stamp will be added at the beginning of every new log file.

Stamp timeout - if the data flow is uninterrupted, the stamp will be added regularly at the interval specified in milliseconds.

File prefix/postfix character(s) - the program will use these characters instead of those specified in the [program options](#)^[109] while writing data to a file. For example, it allows you to add the new line character or another sequence of characters before or after the stamp. Example: >#0D#0A

4.6.10.3 Name and security

This group of options (fig. 3.3.1) allows you to configure the following parameters:

Friendly name - this name will be added before the port number or the data source in the drop-down list in the main window of the program. It allows you to describe the data source.

Start logging automatically - if this option is enabled, the program will start receiving and logging data automatically when it is launched.

The "Security" option group allows you to protect user operations in this particular configuration with a password. You can specify advanced security options applied to the entire program in the [program options](#)^[110].

Ask password before start and stop - the password will be required when the user clicks the "Start/Pause" button in the main window of the program.

Ask password before configuration edit - the password will be required when the user tries to open the Configuration options dialog box.



Fig. 3.3.1. Name and security

4.6.11 Log files

4.6.11.1 Log rotation

The main function of Data Logger Suite is logging data to a file (so-called, log file). The "Log rotation" tab has a rich set of options for it. (fig. 4.1.1).

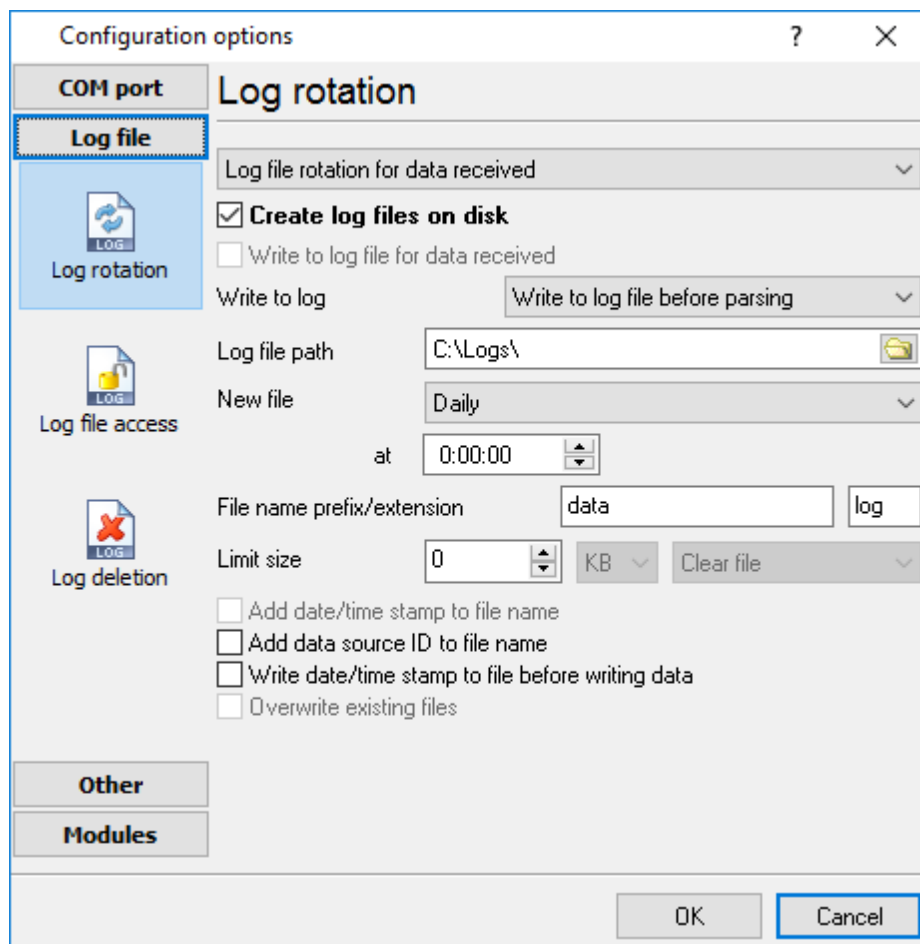


Fig. 4.1.1 Log-file forming modes

First of all, select log file what you can configure:

- **Log file for data received** - all data received will be saved here.
- **Log file for data sent** - all data sent will be saved here. If you want to save data to the same file, as data received, then select the "Log file rotation for data sent" from the list and enable two options: "Create log files on disk" and "Write to log for data received". Yes, of course, you should configure a log rotation for data received before.

Set the "Create log file on disk" option to checked state. Then you can set path to folder, where files will be created with the help of dialog window, which will be showed up after clicking a button with picture of opening folder. You should select a necessary folder in the dialog window and click the "OK" button.

Note: The program can work with network paths too, but in this mode, the program will increase

data flow over a network and can be failed with exceptional errors.

A log file name can be stamped with date and time. In this case a new log file is created periodically. The time stamp format depends on the selected period. For instance, if the "**File name prefix**" field is set to "sample", the "**File extension**" field to "log" and the "**File name format**" option is "Daily", then each log file created will have the format "sampleYYYYMMDD.log". On March 21st, 2003, the log file will be "sample20030321.log". Please, note, that the final extension (after the final period), remains at the end of the file name.

Log rotation mode is defined by the following key parameters:

- **File name prefix** - text string, which will be added at file name beginning;
- **File name extension** - text string, which will be a file extension (characters after dot);

Limit size - the "Limit size" field specify the maximum size in kilobytes of any log file. If you'll specify zero size, then the file size will not limited. You may select from the following modes:

1. **Clear file** - if the log file size will exceed the limit specified, then the log file content will be deleted and file filling will start from beginning.
2. **Rename old** - if the log file size will exceed the limit specified then the existing log file will be renamed.
3. **Shift (no threshold)** - the older data over the limit specified will removed from the log file.
4. **Shift (with threshold)**. In this mode the program will wait when the file size will exceed the limit specified + the threshold value. After this, the older data over the limit specified will removed from the log file.

If the program works continuous for a long time, it is possible that the log file will have large size and this file will be inconvenient for looking and analysing. For this there is the possibility to create files in dependence with the time on PC. You can select one variant predefined or set up new one:

- **Daily** - file will be created with name containing prefix, and date in format DDMMYYYY, where DD is two-digit day sign, MM is two-digit month sign and YYYY is four digits of the current year. The file name extension will be added at the end of file;
- **Monthly** - file will be created with name containing prefix, and date in MMYYYY format. The file name extension will be added at the end of file;
- **Each data packet in different file** - in this mode the program splits data flow to a different file. In this mode you should configure the parser or the program will split a data by timeout about 300 ms.
- **Don't create new file** - in this mode the program will write all data to one file. It is recommended for a small data flow. Otherwise your log file will be too big and a performance of the program will fall down;
- **User's format** - file will be created with name containing prefix and date in showed by you format (for example, DDMMYYYY). The file name extension will be added at the end of file. The file may not contain format signs, then file name will be constant. You should not use characters, that the OS doesn't allows in file name, such as "/", "\", "*", "?" and some others;
- **Weekly** - create a new file every week. The file name will contains a week number;
- **After data timeout** - the program will create a new file if the program didn't receive any data at the specified interval.
- **Hourly** - file will be created with name containing prefix, and date in format YYYYMMDDHH, where HH is two-digit hour sign, DD is two-digit day sign, MM is two-digit month sign and YYYY is four digits of the current year. The file name extension will be added at the end of file;
- **Constantly named file** - the current log file will have a constant name. When creating a new file the existing log file will be saved using the new file name that will contain a data and time

stamp.

Date and time formatting codes:

d - day, not adding null(1-31).
 dd - day ,adding null(01-31).
 ddd - day of the week in text form(Mon-Sat) according to standard, set on this computer.
 dddd - day of the week in full text form(Monday-Saturday) according to standard, set on this computer.
 m - month, not adding null(1-12).
 mm - month, adding null (01-12).
 mmm - month in text form(Jan-Dec) according to standard, set on this computer.
 mmmm - month in full text form (January- December) according to standard, set on this computer.
 yy - year in the form of two last digits(00-99)
 yyyy - year in the form of four last digits (0000-9999).
 h - hours, not adding null (0-23).
 hh - hours, adding null (00-23).
 n - minutes, not adding null (0-59)
 nn - minutes, adding null (00-59).
 s - seconds, not adding null (0-59).
 ss - seconds, adding null (00-59).

Example: you want to create log file every hour. It is desired that file name starts from "sample_log" and the file extension "txt".

Answer: set file prefix = sample_log_, file extension= txt (without dot!). In file name format show HHDDMMYYYY. Now file will be created every hour. Naturally, you can set any formatting characters combination, described higher.

If you want to access to a log file while the program work, then you should configure [access mode settings](#)^[97] for the log file in the next chapter.

Add date/time stamp to file name - this option is available for modes #4 and #7 and allows to add date and time to the file name.

Add data source ID to file name - if this options is activated then then the program will append the data source name at the beginning of the file name. For example, COM1-sample20030321.log.

Write data/time stamp to file before writing data - if this options is activated then then the program will write a date/time stamp to a file before each data portion.

Overwrite existing files - this option is available for modes #4 and #7 and allows to delete an existing log file before creating a new log file.

4.6.11.2 Log file access

During work can be such situations, when it is necessary to get access to a file with *current* data (current log file) from other applications (for example, for data processing). But while you are accessing the current log file Data Logger Suite can't write data to a log file and all data at this moment will be lost. We recommend to use a temporary file for data storage. It is most safe way. (fig. 4.2.1).

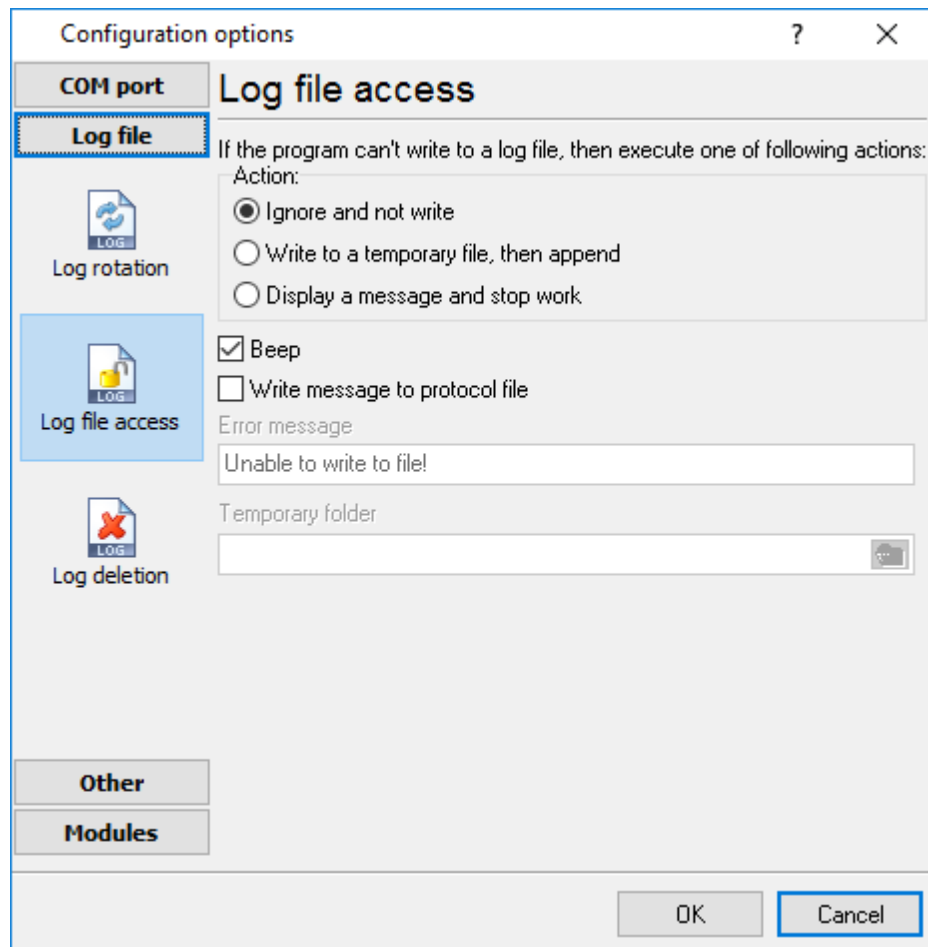


Fig. 4.2.1 File access mode.

You can select one from following variants:

- **Ignore and not write** - with this mode, the data will be **lost**;
- **Write to a temporary file, then append** - a temporary file will be created, to which writing will be done. After access to current file will be got, temporary file content will be added to the end of main file. But mind that if file is created in dependence of time, there can be a situation when at temporary file forming name of the main file will be changed. Then temporary file will be added to the end of newly created file.
- **Display a message and stop work** - data will be lost until dialog window is closed.

You can set up your message text, which will be displayed at writing error to data file. The sound signal can be on for an additional indication. You can also enable writing a message to a protocol file.

4.6.11.3 Log deletion

The deletion of files (fig. 4.3.1) will help you to avoid stuffing your hard disk with needless information. Log files can be deleted either depending on the time of storing or when the maximal number of files is exceeded.

When deleting files by the time of their storage, the files that were modified last time before the specified period are deleted.

When controlling the number of files, the files with the oldest modification dates are deleted first.

You can select both variants of file deletion. In that case files will be deleted when either of the conditions is true.

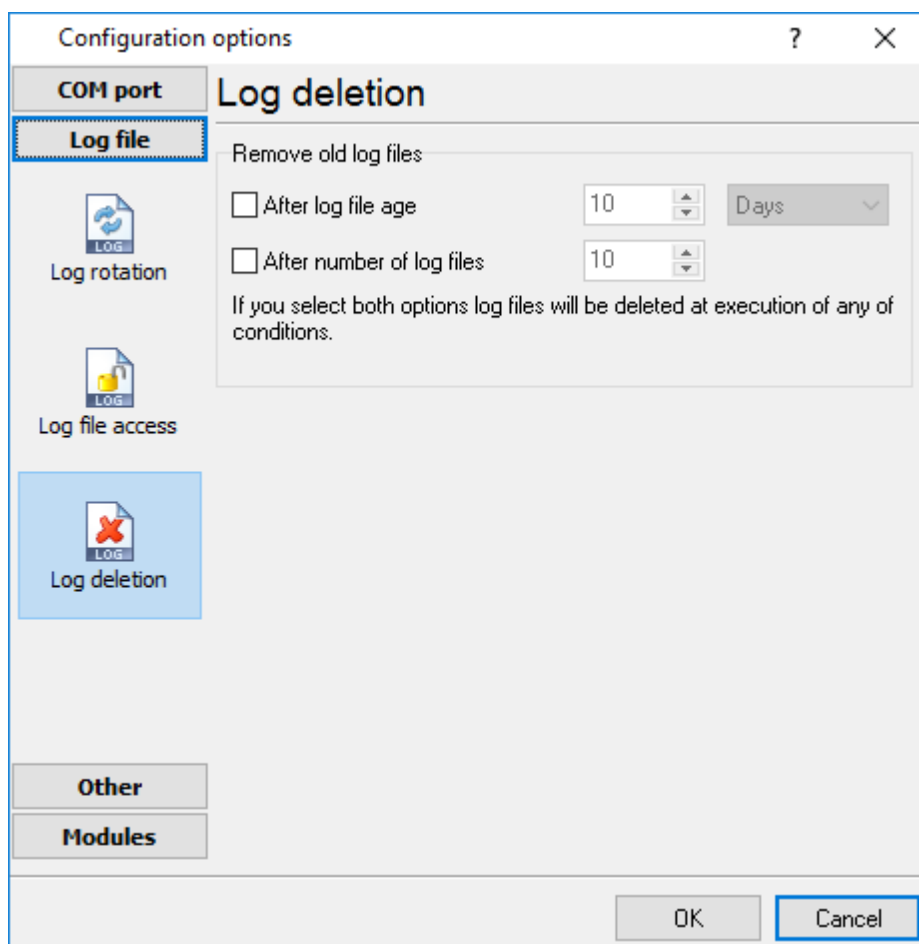


Fig. 4.3.1 Log deletion

By default the program is writing data to a text file that isn't compatible with the CSV format. But you can create Excel compatible CSV files. Simply activate logging to a CSV file..

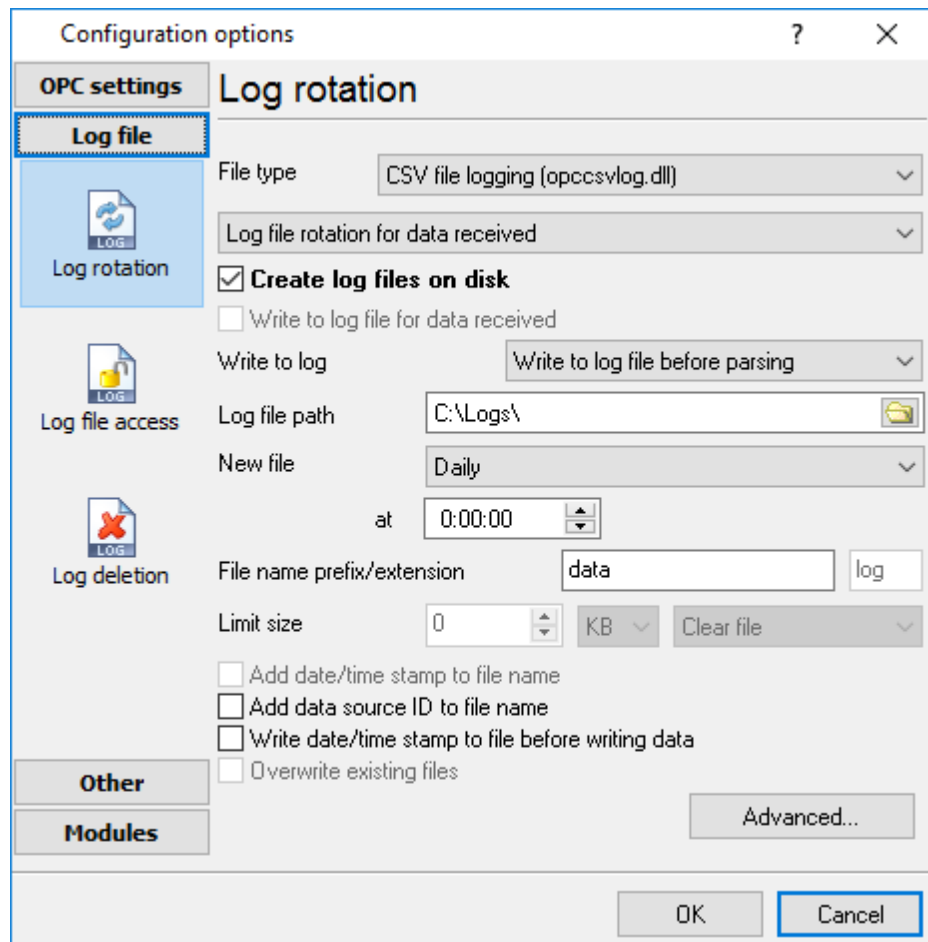


Fig. 4.4.1 CSV logging

4.6.12 Modules

4.6.12.1 Introduction & setup

To extend program functionality we implemented plug-ins modules. Module structure lets reduce your program size and purchase costs (you pay only functionality, which you need), to low down program distributive download time, your computer processor load and reduce disk space.

Data Logger Suite supports few types of modules (fig. 5.1.1 - 5.1.3):

- **Data query** - transmits queries or commands out the data source to control or query your instruments directly;
- **Data parser** - data parser that allows you to parse, filter and format more complex data from more sophisticated devices. Some of the advanced features of the parser are the ability to

work with raw binary or hex data;

- **Data filter** - data filters allow you to filter your data and modify values of parser variables;
- **Data export** (fig.11) - Data Logger Suite has many modules for passing serial data directly to other applications, such as as keystrokes where incoming data is passed to other programs as a sequence of keys, as DDE Server that passes data to other programs using Dynamic Data Exchange, ODBC for exporting data to a database and many others;
- **Events handling** (fig.12) - an external plug-ins used to handle events generated by the Data Logger Suite software. Once an event occurs (for example: "Data source is opened" or "Configuration changed"), the plug-in creates a text message using the specified template and sends a notification, do some actions, such as execute programs, scripts and etc. The form of the notification or actions depends on the plug-in settings.

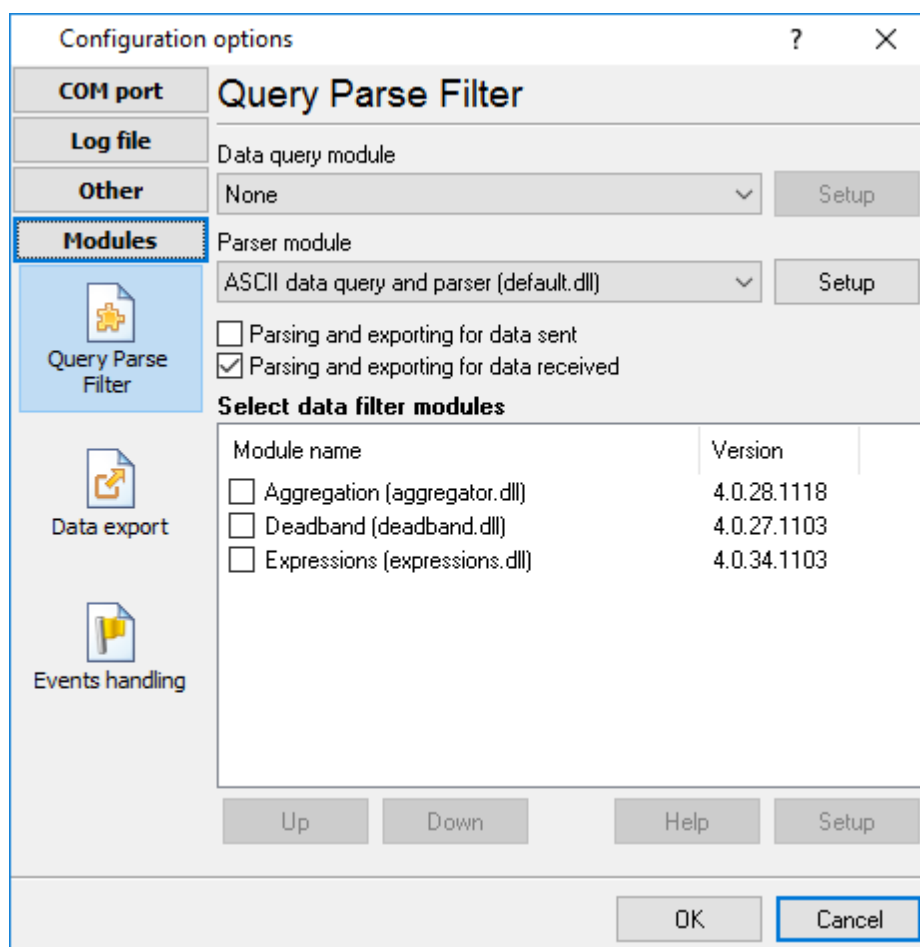


Fig. 5.1.1. Activating plug-ins

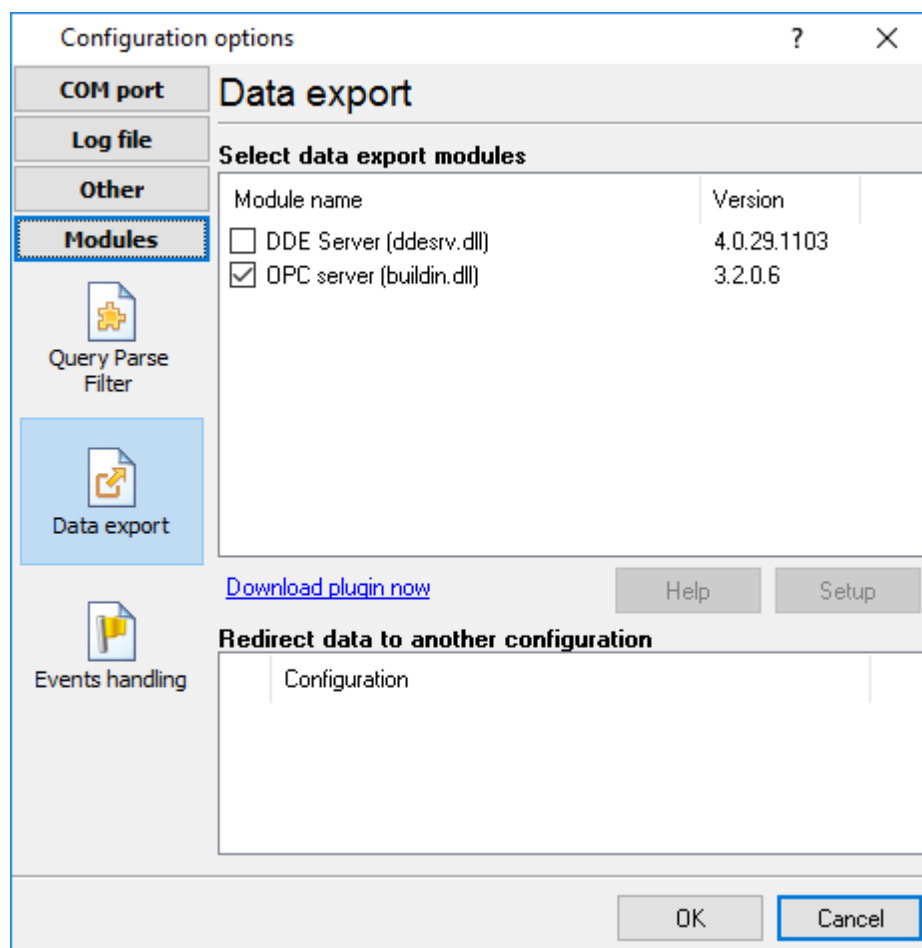


Fig. 5.1.2 Activating data export plug-ins

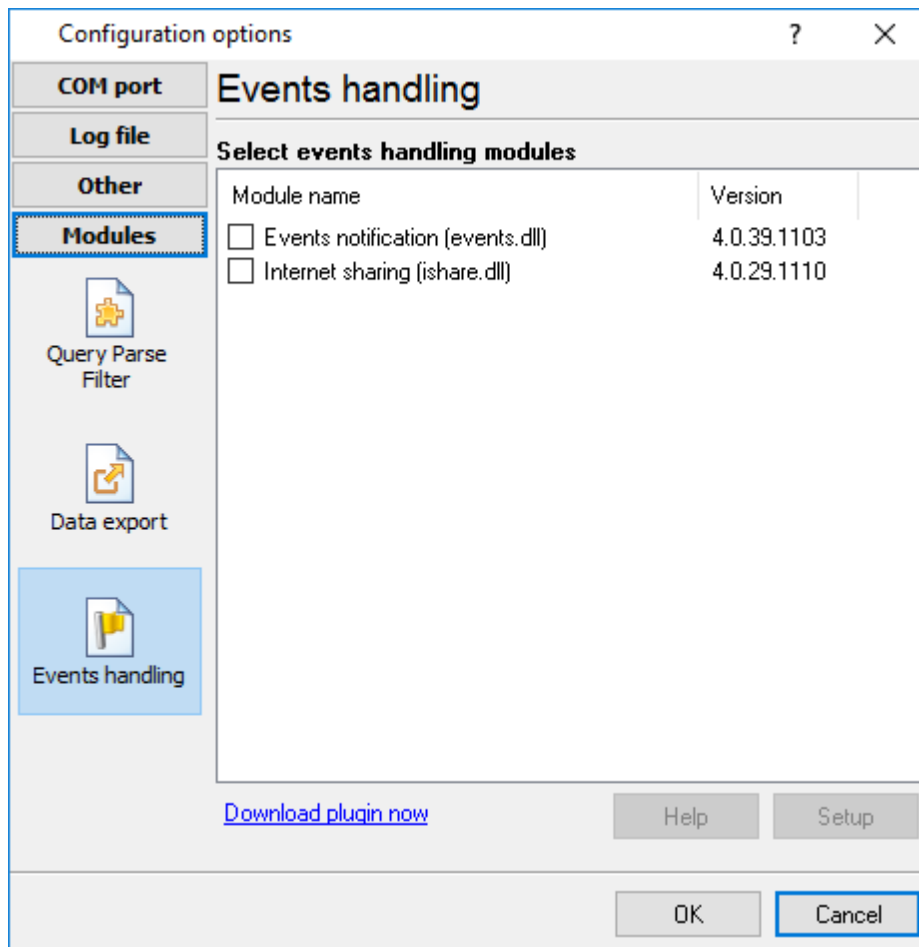


Fig. 5.1.3 Activating events handling plug-ins

You can parse and export data sent and received (fig. 5.1.1). By default, only data received will be parsed.

Installation

You can easily install a new module. usually, you should start the installation file and click the "Next" button for few times. The installation wizard will detect a place of your Data Logger Suite software and place a plug-in module and all distributive files to the "Plugins" folder, which is in the program folder (by default X:\Program Files\Data Logger Suite\Plugins).

After program restart a module will be loaded and initialized. If module is supported with our software, its short description you will see in modules list (Fig. 5.1.1-5.1.3). Most modules require additional settings. If you want to configure the plug-in module, simply click the "Setup" button near it. If you selected the module and the "Setup" button is not active, then module doesn't have additional settings and can work without additional settings. Please, read users manual of a corresponding plug-in for additional information.

Configuration steps

1. Select and configure a query module. You may use a module of this type if you need to send

- some data to your device (for example, initialization strings or request strings).
2. Select and configure a parser module. This step is necessary, because filter and export modules can use parsed data only. If you didn't select the parser module, then you can't configure the data filter and data export modules.
 3. Activate and configure data export modules. You can select one or more modules simultaneously. The program will use selected modules simultaneously. Please, note, the program can't use the data export module, if you didn't configure the parser module.
 4. Activate and configure event modules. You can select one or more modules simultaneously.

4.6.12.2 OPC server

Since the version 2.1.1 Data Logger Suite has an internal OPC server. It means, that any OPC compatible client application can get data from Data Logger Suite without any additional software. For connecting to the OPC server our OPC server has an unique attributes (Fig. 5.2.1). Before using the OPC server on your PC you should download and install the OPC Core Components Redistributable from www.opcfoundation.org (registration required).

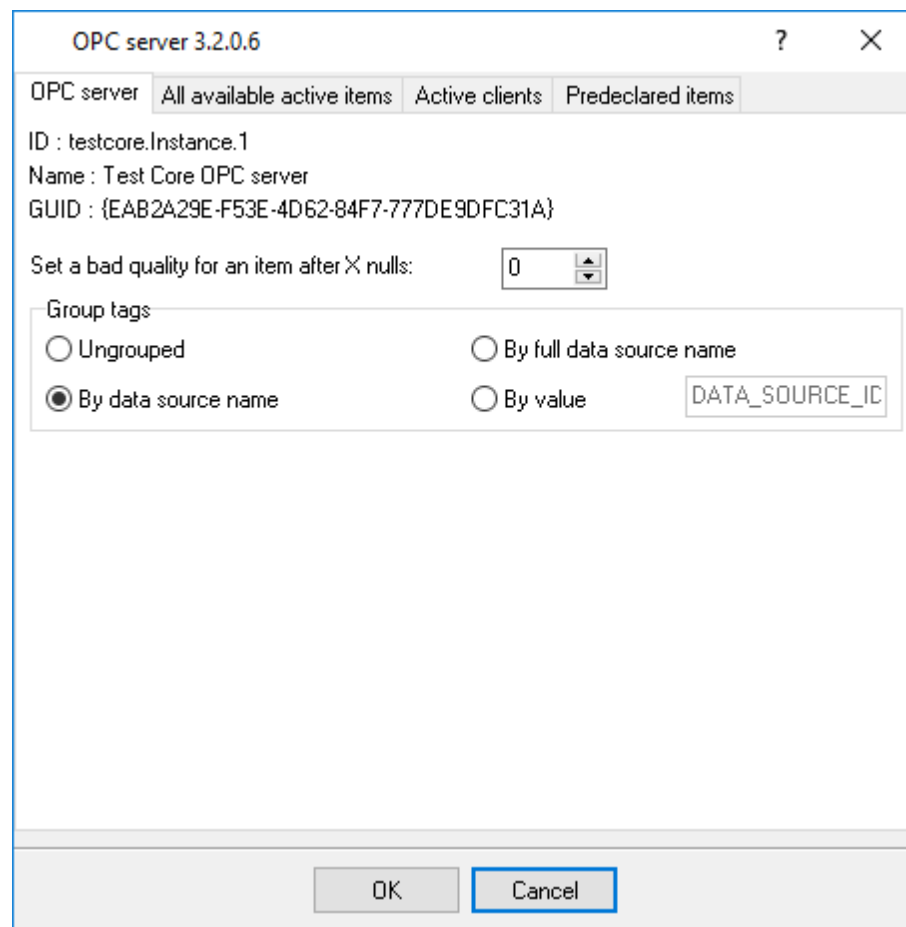


Fig. 5.2.1 OPC server parameters

Data Logger Suite are parsing all incoming data to one or more variables and OPC client are getting it (fig. 5.2.2). After connecting to the OPC server you will get list of all variables.

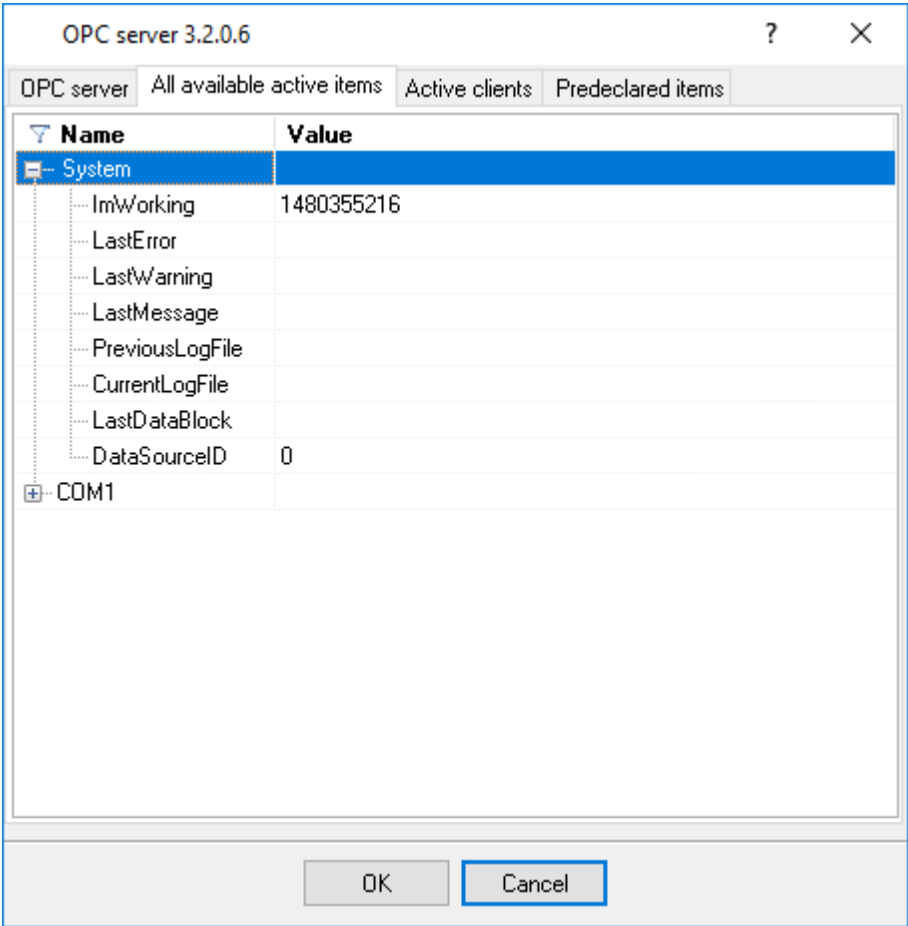


Fig. 5.2.2 OPC server active items

Clients activity is showed on the "Active clients" tab. The top node is client, below is group of items and connected items. By double-clicking, you can get a detailed information about each node.

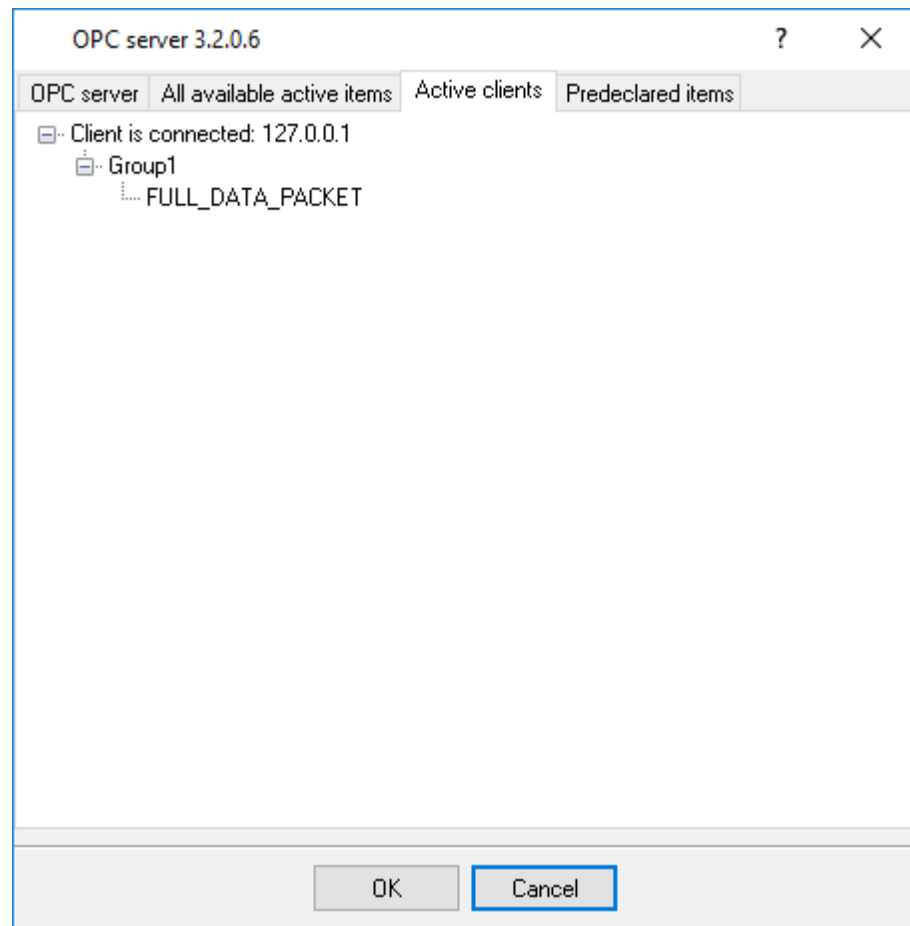


Fig. 5.2.3 OPC server clients

Data Logger Suite creates new variables at "on-the-fly" mode. The Data Logger Suite starts without any variables and get it only after first data had been received. If your client OPC will connect to the OPC server before, than data had been processed, then it will get empty list of variables and your OPC client should poll the OPC server for updating list of variables. If your OPC client doesn't allow it, then your can pre-define all variables (fig.5.2.4). In this case the OPC server will create these variables with empty values, immediate after starting and your OPC client will get these names while connecting.

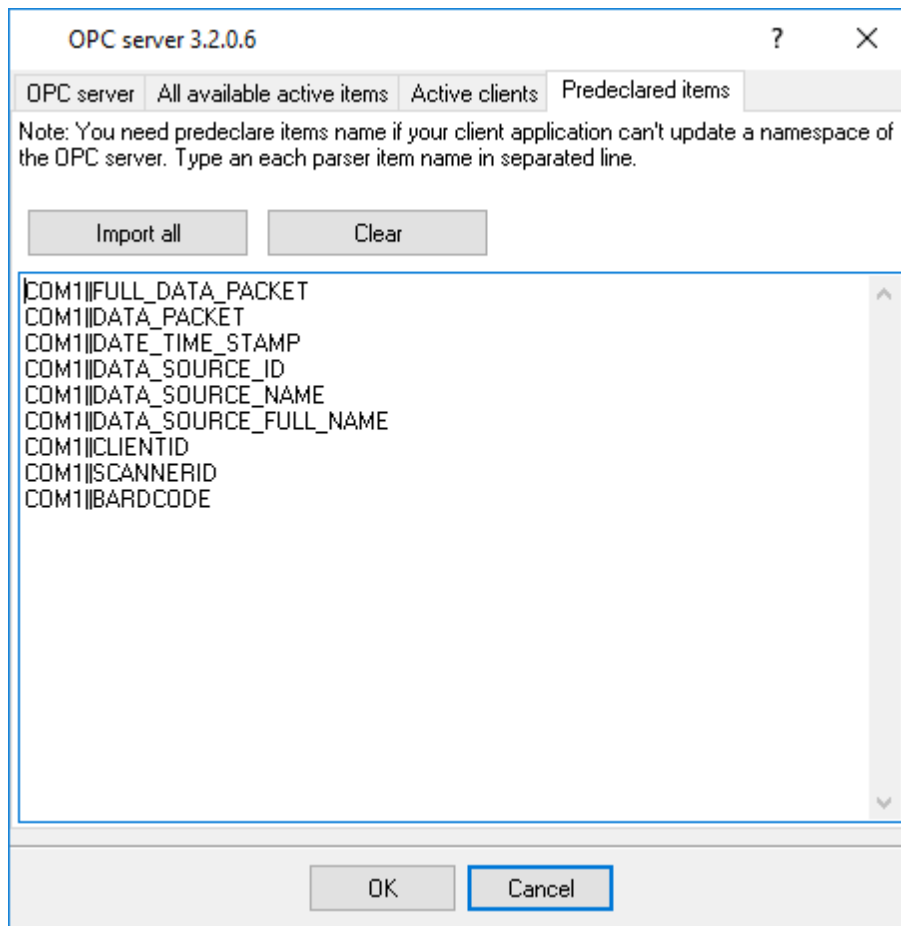


Fig. 5.2.4 OPC server pre-declaration

4.7 Program options

4.7.1 Window view

This tab in program options (fig. 6.1.1) allows you to customize appearance of the main window of the program (fig. 1.1.1⁷). You can access this tab through the "Options -> Program options" menu item in the main window.

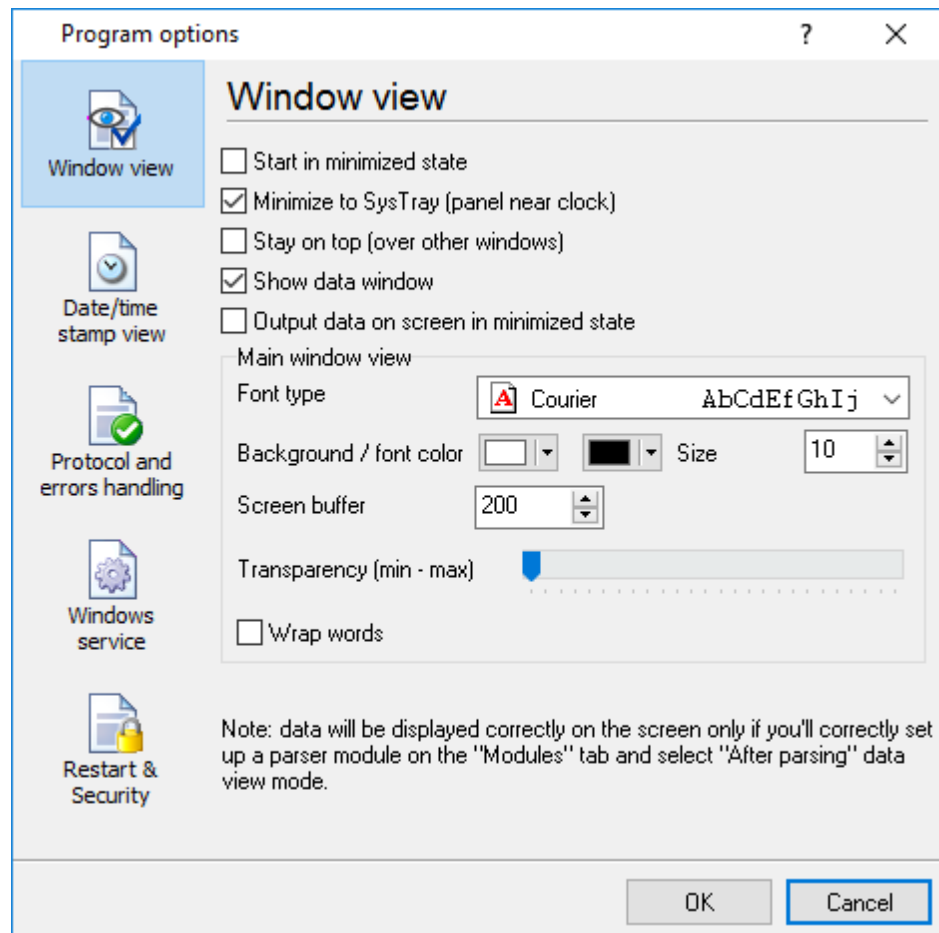


Fig. 6.1.1 Window view setting

You can set following parameters:

- **Start in minimized state** - at start Data Logger Suite will automatically put its icon on panel near clock (SysTray fig. 6.1.2);
- **Minimize to SysTray** - while the main window of Data Logger Suite minimizes, the program will automatically put its icon to the panel near clock;
- **Show data window** - if you specify this option, then the program will display all data in the main window. If you are logging many ports on slow PC, then you can computer CPU load rate with disabling of this option;
- **Output data on screen in minimized state** - if you'll enable this option, then the program will display processed data in minimized state. If you are logging many data sources on slow PC, then you can decrease computer central processor load rate with disabling of this option;
- **Font type** - the data will be displayed with this font type in the main window only. We recommend to use mono-spaced fonts in this field, such as: Terminal, Courier etc.;
- **Screen buffer** - at exceeding of value specified the data window will be cleaned;
- **Window view** - will let you setup data window view (font color, font type, background color).
- **Transparency** - in Windows 2000 and later will let you set transparency of the main window. The most left position is normal window view and most right position is maximum transparency.
- **Wrap words** - if you didn't configure a parser module or your data flow doesn't contain a blocks separator, then your data without this option enabled will be displayed as one long string in the data window.

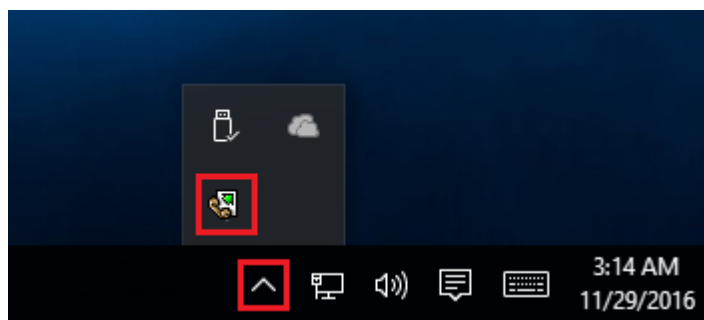


Fig. 6.1.2 Systray - panel near clock

4.7.2 Date/time stamp view

This group of options (fig. 6.2.1) allows to configure the format of date/time stamps that will be used in the main program window and log files.

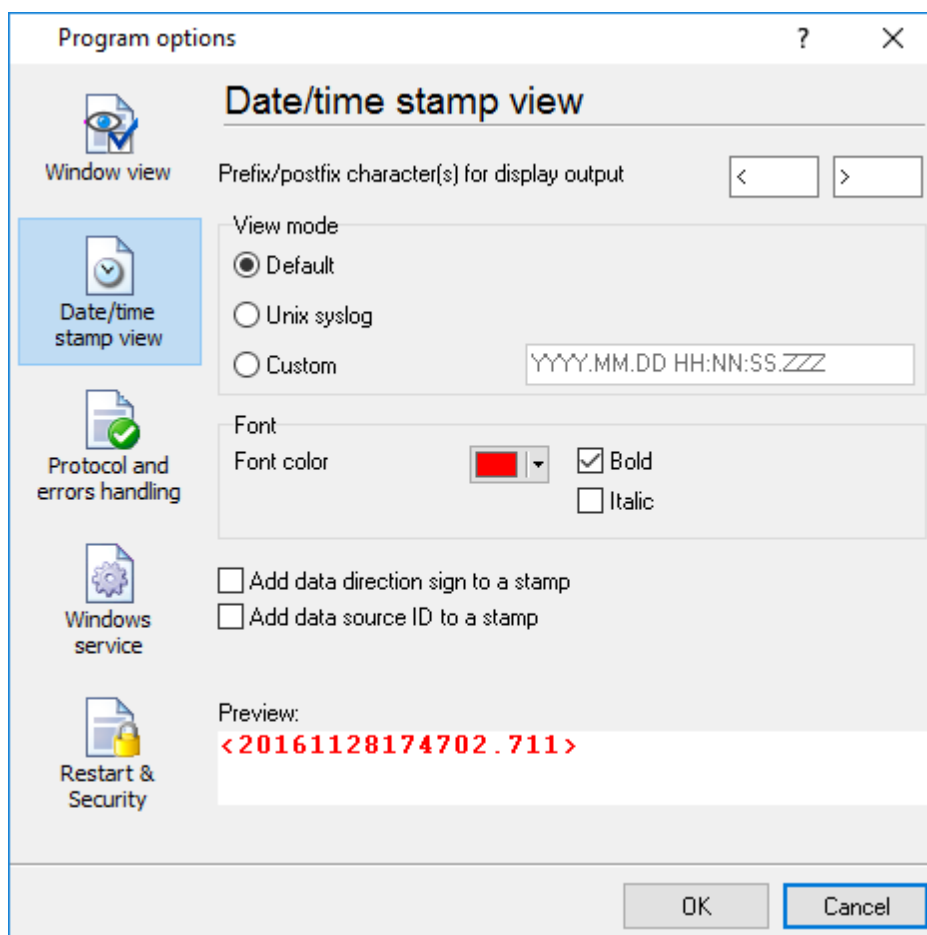


Fig. 6.2.1 Configuring data/stamp view

Prefix/Postfix characters for display output - these options allow to define beginning and ending characters of a date/time stamp that will be showed in the program window. When outputting data to

a log file the program uses [individual characters](#)^[92] for each configuration.

View mode - allows to select the standard or define the custom format of the date/time stamp.

Font - this group allows to define the color and font of date/time stamp.

Add data direction sign to a stamp - if this option is activated then the program will append TX or RX to the end of the stamp.

Add data source ID to a stamp - if this option is activated then the program will data append data source ID at the beginning of the stamp. For example, COM1.

4.7.3 Protocol and errors handling

While the program execute, she generates many messages about errors and events. All these messages are being registered in a protocol file. This can be start or stop of the program, some messages from plug-in modules etc. On this tab you can define the kind of messages, which you want to put a protocol file (fig. 6.3.1). Here you can set maximum protocol file size and a formatting mode. Usually, the protocol file is in a program folder and has the name of the program with the 'log' extension.

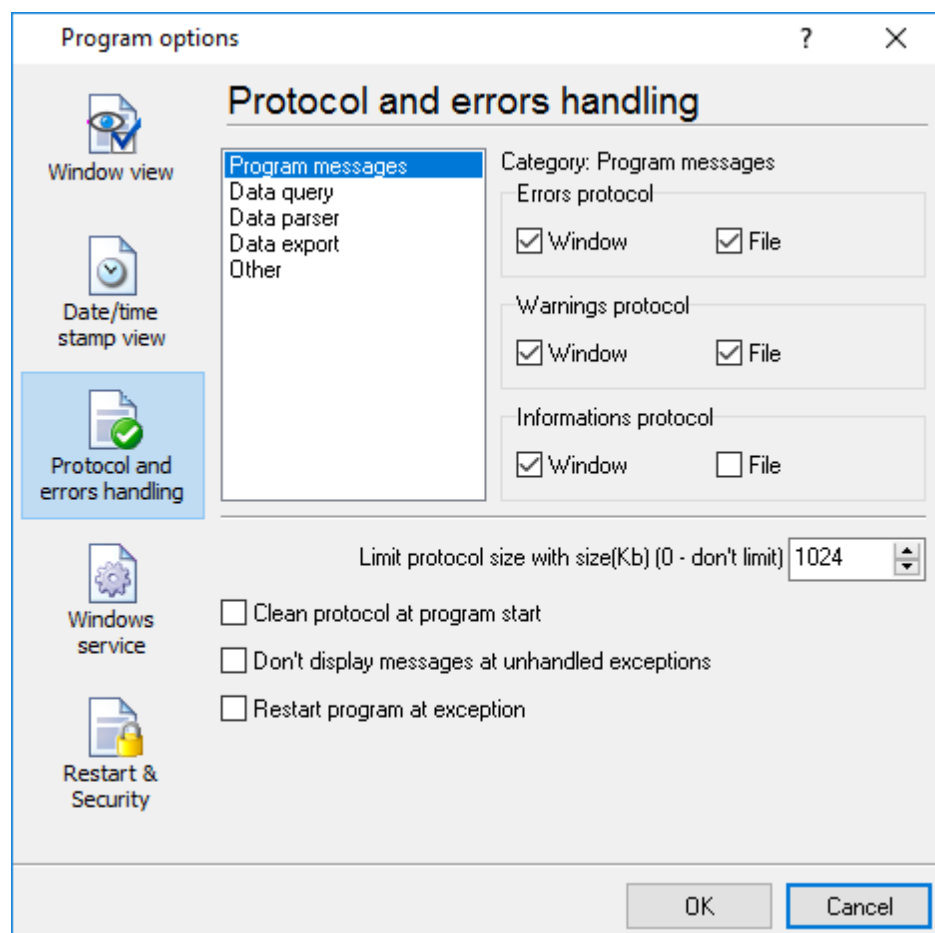


Fig. 6.3.1 Protocol settings

Data Logger Suite works with three types of messages:

- **Information messages** - this type of messages informs you on operations which are fulfilled in the current time;
- **Warnings** - warns you of possible failures or possible errors. Interference of the user is not required, but check is required;
- **Errors** - the program has detected an error which elimination needs involvement of the user.

There is the possibility to log following events:

- **Program messages** - messages about start or stop of the program, etc.;
- **Data query** - messages which are generated in a data query module;
- **Data parser** - messages which are generated in a data parser module;
- **Data export** - messages which are generated in a data export module;
- **Other** - messages that can not be associated with types above.

You can write each type of messages to a protocol file or/and to the list in the main window. Please, specify necessary options for each message type at "Window" and "File" fields.

If you don't want to allow to grow a protocol file size to an unlimited size, then you can enable the "Clean protocol at program start" or limit protocol file size in the "Size" field.

Some exceptional messages can occur while the program execute. In most cases these messages crash the program and the most safe way is to restart the program. Please, specify the "Restart program at exception" option and the program will be restarted automatically.

If you want look all program messages, then you can disable the "Don't display messages at unhandled exceptions" check box and the program will open the exception message window with detailed information.

4.7.4 Service mode on Windows 2000+

4.7.4.1 Configuration

Windows NT+ services use will let you:

- control service on local and remote computers, including remote computers with Windows NT 4.0 system;
- setup actions on emergency service restore in case of failure, for example auto service or computer restart (only on computers with Windows 2000 or later);
- create for services other names and descriptions, to find them easier (only on computers with system Windows 2000 or later);
- run service before user login (password input);
- service can be setup on automatic start after operation system load.

Note 1: you must be logged in as an administrator in order to change the configuration or control the service in any way (start, stop, pause, continue).

Note 2: On Windows Vista and later you should start the program with elevated administrator

privileges.

If you want to use the program as a service application, then, please, go to the "Options -> Program options -> Windows service" tab (fig. 6.4.1), then enable the "Use program as a service" check box. Later, please, specify the startup type of the service. There can be following variants:

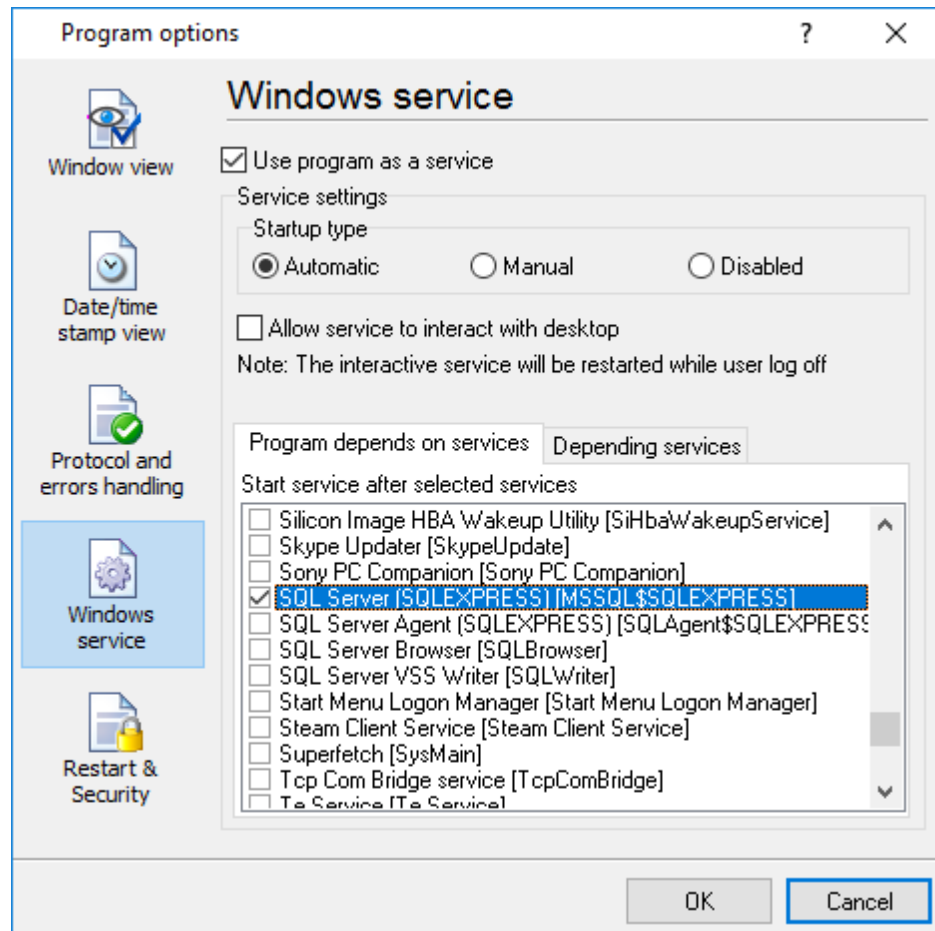


Fig. 6.4.1 Service settings

1. **Automatic** - service will be started automatically at every Windows start, before user login;
2. **Manual** - you can start the service application in the "Services" window in the Control panel (fig. 6.4.2);
3. **Disabled** - service can't be started.

If you want to change program settings while service mode, then enable the "Allow service to interact with desktop" option. In this case, the program will put the icon to the SysTray (fig. 6.4.2). But then your interactive service will be restarted while user log off.

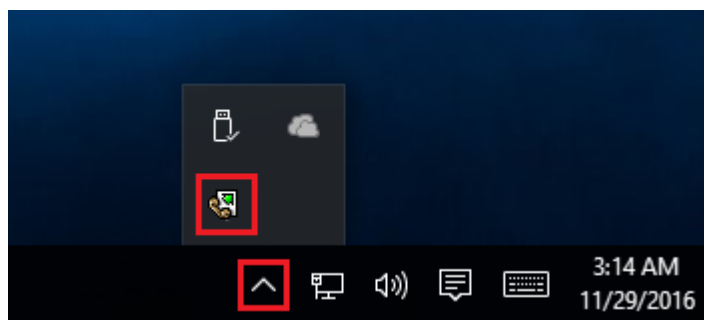


Fig. 6.4.2 Service icon in Systray

If you use data bases or special drivers, before service run, these applications must be run. You can specify what services should be started before Data Logger Suite in the "Program depends on services" window (fig. 6.4.3).

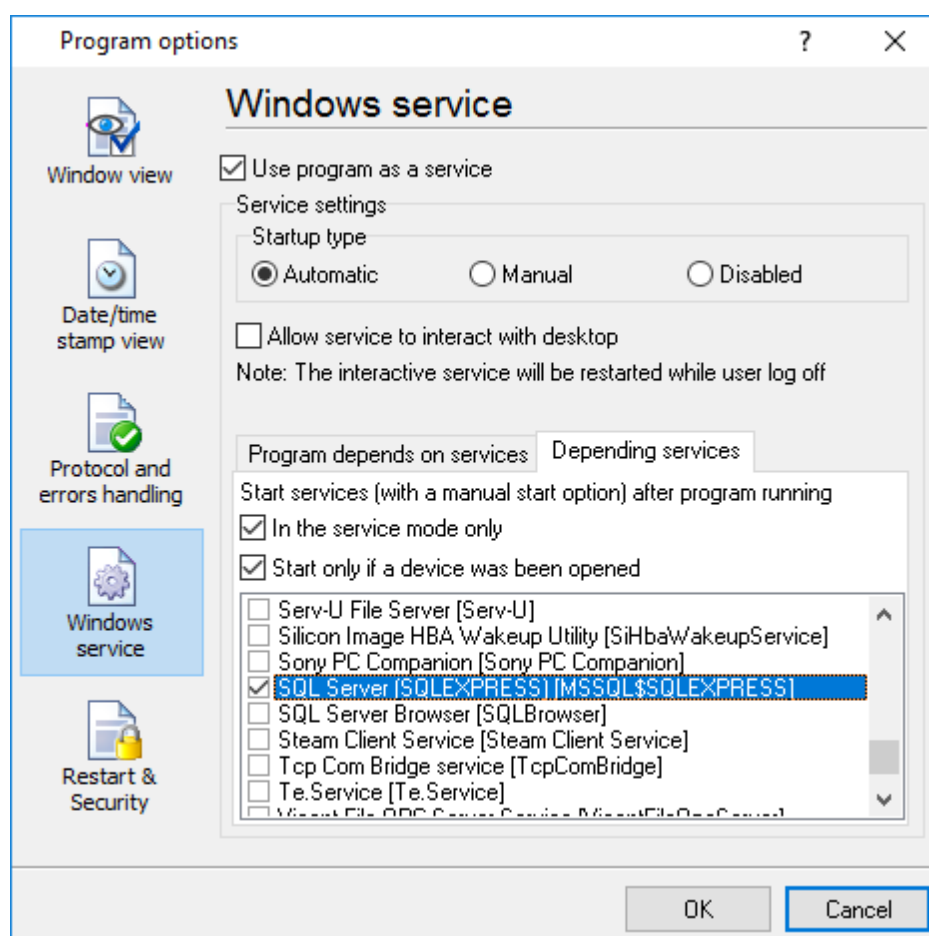


Fig. 6.4.3 Service settings #2

In some case you need to start Data Logger Suite before starting other services. In this case you should:

- Switch a start mode of a target service to "Manual";
- Start Data Logger Suite;

- Set on the target service at the "Services depend on ASDL" tab;
- Select mode when you need start these service;
- Restart Data Logger Suite.

After you installed the service mode of the program, restart a computer or run the service manually from the "Services" window in the Control panel (fig. 6.4.4)

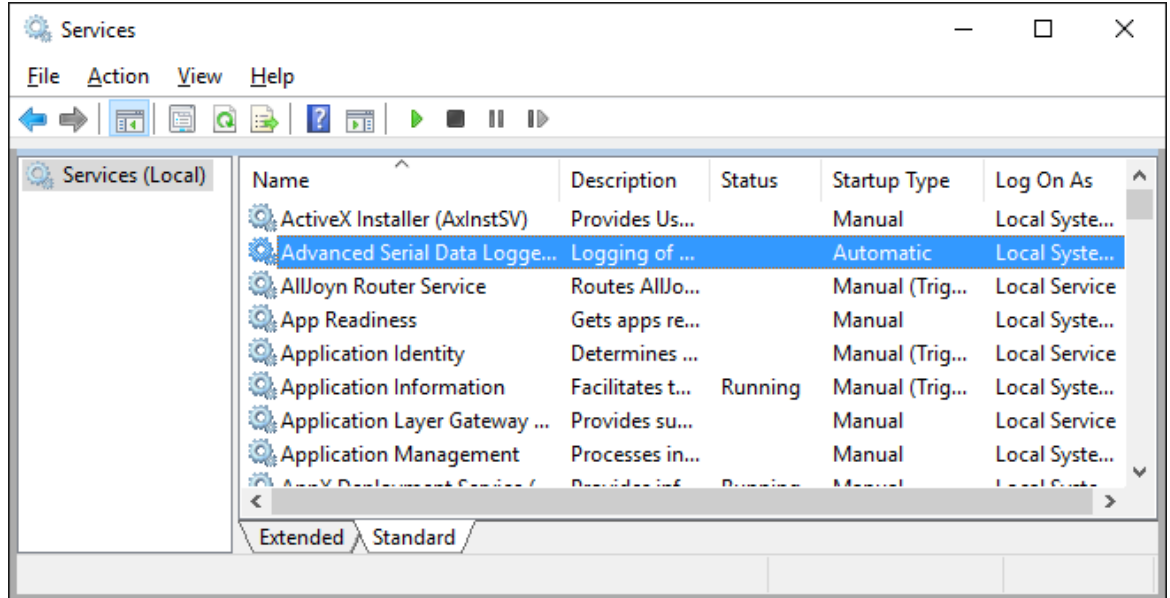


Fig. 6.4.4 Manual service run (in Windows 2000)

After start of the service, the service processes names will be displayed in a process list: dataloggersuitesrv.exe and dataloggersuite.exe (fig. 6.4.5). The 'dataloggersuitesrv.exe' application makes interface between the service manager and the Data Logger Suite software. Unlike srany.exe utility, our service stops safely.

The screenshot shows the Windows Task Manager window with the 'Performance' tab selected. At the top, it displays system resource usage: CPU at 37%, Memory at 47%, Disk at 94%, and Network at 0%. Below this, the 'Processes' tab is active, showing a list of running applications and background processes. The 'Task Manager' process is highlighted in yellow, showing it is using 12.2% CPU, 9.0 MB of memory, and 0 MB/s of disk and network. Other processes listed include 'Advanced Serial Data Logger - L...', 'Application Frame Host', 'COM Surrogate', 'Cortana', 'Device Census', 'Host Process for Windows Tasks', and 'Host Process for Windows Tasks'.

Name	CPU	Memory	Disk	Network
Apps (1)				
Task Manager	12.2%	9.0 MB	0 MB/s	0 Mbps
Background processes (39)				
Advanced Serial Data Logger - L...	0%	8.4 MB	0 MB/s	0 Mbps
Advanced Serial Data Logger - L...	0%	0.6 MB	0 MB/s	0 Mbps
Application Frame Host	0%	2.0 MB	0 MB/s	0 Mbps
COM Surrogate	0%	2.5 MB	0 MB/s	0 Mbps
Cortana	0%	70.6 MB	0 MB/s	0 Mbps
Device Census	0%	1.2 MB	0.1 MB/s	0 Mbps
Host Process for Windows Tasks	0%	0.7 MB	0 MB/s	0 Mbps
Host Process for Windows Tasks	0%	4.8 MB	0 MB/s	0 Mbps

Fig. 6.4.5 Process list

If you want to configure the program as a service, then you must be logged with administrator rights. The service application can be controlled, stopped or removed with help of a command line string. Run `dataloggersuitesrv.exe` with following parameters:

- `/?` - short help;
- `/I` - install service for start in manual mode;
- `/A` - install service for start in automatic mode;
- `/D` - install service in off state;
- `/R` - remove service from computer.

4.7.4.2 Windows Vista+ notes

One of the ways Vista's security was improved was by separating system services and user applications into separate 'sessions'. Keeping the system services isolated helps to better secure them, but also makes any interactive interface unavailable to the user. That's where the Interactive Services Detection service comes in. When a service needs to interact with the user, Interactive Services Detection presents a dialog that will switch the user to the session where the service is running so they can interact with the service. For an excellent, detailed description of this, see next

paragraph.

Many sites recommend disabling this service, but doing so will result in you not being able to interact with any services that require your attention. This service is run manually by default, so there is little point to disabling it unless you don't want to be bothered by important information from the software you may be trying to run.

- **Display Name:** Interactive Services Detection
- **Service Name:** UI0Detect
- **Process Name:** UI0Detect.exe
- **Description:** Enables user notification of user input for interactive services, which enables access to dialogs created by interactive services when they appear. If this service is stopped, notifications of new interactive service dialogs will no longer function and there may no longer be access to interactive service dialogs. If this service is disabled, both notifications of and access to new interactive service dialogs will no longer function.
- **Path to Executable:** %windir%\system32\UI0Detect.exe
- **Default Startup:**
 - * Home Basic: Manual
 - * Home Premium: Manual
 - * Business: Manual
 - * Enterprise: Manual
 - * Ultimate: Manual

4.7.5 Restart & Security

Sometimes the program should be restarted. For example, if you've changed the program settings remotely and want to reload program automatically with the new settings. To do that, specify the time for restarting the program on the "Restart & Security" tab in program options "Options->Program options". Just specify the time of day, when the program should be being restarted.

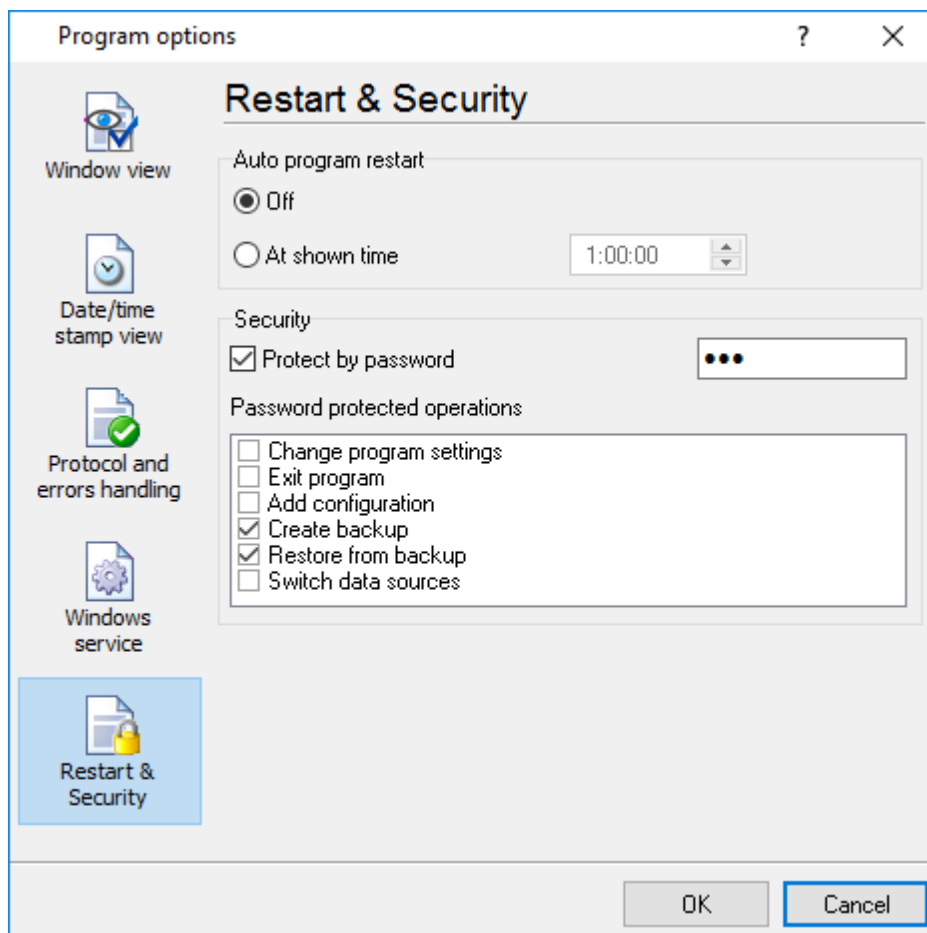


Fig. 6.5.1 Program restart settings

On this tab you can also protect some actions with the program by a password. To do that, activate the "Protect by password" option, define a password and select protectable actions.

6 Having problems?

6.1 Program doesn't run or work

It is necessary to make sure in proper time installation on your computer, so as if you put clock after program installation, protection from use after trial period works.

Also program won't work, if you started SoftIce application debug environment or some other. In any other case, please, inform our developers about your problems, our address support@aggsoft.com.

6.2 FAQ

Question: Why COM port doesn't open?

Answer: Probably, some other program already uses it (COM port). This can be DOS application, for example.

Question: What to do?

Answer: Close application, using this communicative port (for DOS application close also DOS session window). Or use other communicative port. Probably, at start or some program stop was made some fatal error and COM port wasn't properly closed.

Question: Is it possible to set variable data transmit rate or transmit 9 data bits?

Answer: No, Windows operation system doesn't let such liberties.

Question: What cutoff point type to use: DB25 or DB 9?

Answer: It makes no difference, select only in program corresponding COM-port. USUALLY DB25 - COM2, DB9 - COM1

Question: Is cable connection direct or null-modem?

Answer: Everything depends on your device cutoff point. Usually it is necessary to use null-modem cable, in which signals are moved apart in this way:

Device		Computer
RXD	<- ->	TXD
TXD	<- ->	RXD
GND	<- ->	GND

if device uses special signals DTR and others and you don't want to use hardware data transmit control, at device side connect 7 and 8 contacts of DB9 cutoff point or analogous DB25 cutoff point signals.

More hardware tips and articles you can found on our site <http://www.aggsoft.com>.

Question: Is data receive becomes at once after program start or data receive must be started?

Answer: Everything depends on your device type, usually in device settings is selected to echo view data to COM port. If your device doesn't support this mode, write what initialization string must be sent to read out data, and we add this possibility to the program.