



DataTag Link

Connecting
Unconnectable

USER'S MANUAL

Datatag Link

© 2021-2022 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.

Publisher

AGG Software

Production

© 2021-2022 AGG Software

<http://www.aggsoft.com>

Table of Contents

Part 1 Introduction	1
1 About the program	1
Part 2 License, Registration and technical support	1
1 License	1
2 How to acquire a license	3
3 Support	3
Part 3 Installation	3
1 System requirements	3
2 Limitations of the trial version	4
3 Installation	4
Part 4 Settings	5
1 Introduction	5
2 File menu	7
3 Edit menu	7
4 View menu	8
5 Program settings	8
6 Data sources	14
OPC DA	14
OPC UA	15
MQTT	17
SQL	19
MODBUS	23
7 Command like parameters	27
8 FAQ and known issues	27
9 Cron time format	28

1 Introduction

1.1 About the program

Program Overview

Datatag Link is a utility that enables you to link one or more values (tags) from different data sources and ensure two-way real-time communication between these sources. Data transmission between sources may be accompanied by automatic conversion of data types. At the same time, each transaction is logged to a separate log file.

Our program enables you to save money on purchasing conversion hardware. Using more advanced and high-performance computers makes it possible to ensure a higher transmission rate, a higher number of synchronized values or to link sources that are not reachable with hardware solutions.

Features:

- Connecting to OPC DA, OPC UA, MQTT, SQL, or MODBUS servers in any combination.
- Adding up to 65,000 tags to enable synchronization in one application.
- OPC UA. Supporting secure connections, as well as operation via the Internet.
- MQTT: Supporting simple values and those valid in JSON.
- SQL: supporting all widely used databases.
- MODBUS: Supporting MODBUS TCP and MODBUS RTU connections.
- Configuring all parameters in the dialog mode.
- Supporting automatic startup and running as a background service, establishing communication between sources even before the user logs into the system.

Typical usage:

- Bridging multiple interfaces i.g. MQTT-OPC, OPC-SQL, etc.
- Simple logging of data sources in SQL.
- Linking databases and industrial equipment (SQL-MODBUS).

Company website: <https://www.aggsoft.com/>

2 License, Registration and technical support

2.1 License

Copyright © 1999-2022 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 Datatag Link 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 Datatag Link 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 customers 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 Datatag Link 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 update 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 How to acquire a license

The unlicensed program works in the trial mode. The license key removes all limitations from the trial version and allows you to use our technical support for one year.

If you want to buy a program through the Internet visit the [order 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 have the form of order registration. Enter your personal information and choose the most convenient payment method for you. Further, you will get a notification and follow the notes in it.

You may find more information about our policies, payment terms and payment methods, frequently asked questions on our [web site](#).

2.3 Support

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

3 Installation

3.1 System requirements

Processor: Any CPU compatible with Intel® x86 or x64

Operating system: Windows XP SP4 and higher (all versions) x86 and x64. The program also runs under server operating systems.

Note: the program will not run under Windows 9x, Me and NT.

CPU operating speed

Minimum: 1 GHz.

Recommended: 1.5 GHz or higher.

Our software successfully runs on some slower CPUs with 4 or more cores.

Memory

Minimum: 512 MB.

Recommended: 1024 MB or more.

Free disk space. The program requires approximately 5 MB. However, additional disk space is required for system log files.

Special access requirements. To be able to use the software, you must log on as Administrator. This is necessary to access certain hardware features and install license keys.

3.2 Limitations of the trial version

Limited version: The total trial period is 21 days. The program allows you to link up to 10 tags. The continuous runtime limit is 2 hours. After exceeding the runtime, you can restart the program and continue testing it.

Full version: unlimited functionality.

3.3 Installation

Basic Installation Requirements

If Datatag Link is already running, then shut down the program. If Datatag Link is running as a service, then stop the service in the Windows control panel.

If a beta version of Datatag Link is installed on your computer, delete it before installation.

Run the installation file.

Follow the instructions of the installation wizard during the installation process.

During the installation process, you will be asked to select the types of data sources between which you are going to establish communication.

By default, Datatag Link will be installed in the folder "c:\Programs Files\Datatag Link" on your system disk. You may as well change the destination folder.

Once installation is done, configure the program as described [below](#).

4 Settings

4.1 Introduction

When you launch the program, the main window will open (Fig. 1). This window consists of the following parts:

1. **Main menu** - contains the menu commands described in the Menu section of this help file. To access the main menu, you can use the hotkeys (Alt + first letter). Some nested menu items can be accessed with special shortcuts.
2. **Toolbar** - contains buttons to quickly access the most frequently used functions.
3. **Tag List** - contains a list of created links between tags in different data sources and their status.
4. **Status Panel** - displays running time of the program.

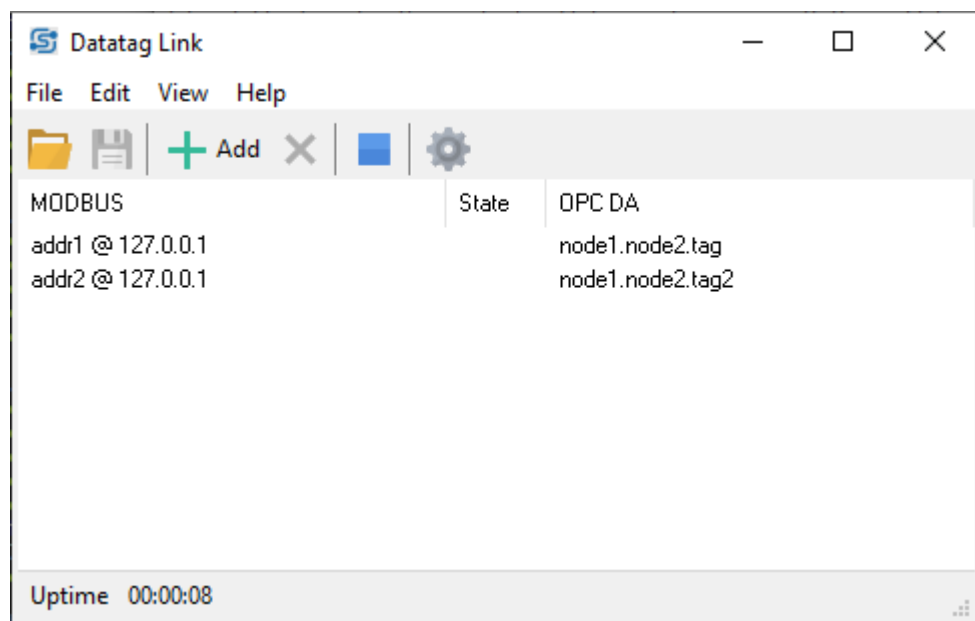





Fig. 1. The main window.

When you click the Add button , a window will be displayed with the connection settings for each data source (if required) and tag parameters (see Fig. 2). The parameters of each connection type are described in more detail in the sections below.

For each connection, it is possible to configure certain transmission directions  . For example, you can enable only one-way data transmission or temporarily disable communication.

Link configuration

MODBUS

Connection

127.0.0.1 + Add - X

Name	Device address	Memory	Memory offset
addr1	1	Coils	0

Number of registers: 1

Data type: Boolean (Bool)

Byte order in register/value: BA DC A - lowest, F - highest

Polling

Interval: 5000 milliseconds

Time using Unix Cron schedule

WD: * M: * D: * ?

H: * MIN: * SEC: */10

MODBUS → OPC DA (checked) MODBUS ← OPC DA (checked)

OPC DA

Tag name: node1.node2.tag


Data type: Float 32-bit (F32)

You can specify the path in a tree hierarchy like: node1.node2.value

Server name

OK Cancel

Fig. 2. Tag linking parameters.

After setting up a new link of tags and clicking OK in the settings window, a new connection will appear in the main window (Fig. 2). Once you have configured the list of all connections, click File - Save  to apply all the changes. The next time you run the program, the last saved configuration will be loaded.

By pressing the Pause button  on the toolbar, you will temporarily suspend the program.

In the Status column the symbol "<" or ">" will appear indicating transmission from one source to the other.

Furthermore, you can configure the program parameters in a [separate window](#) (message log, service mode, data display type, etc.).

4.2 File menu

Open

Hotkey: Ctrl+O

The function loads a program configuration from a file. The loaded configuration will completely supersede the current settings.

Save

Hotkey: Ctrl+S

The function saves changes in the current configuration. When you click Save, the changes will be applied.

Save As

The function saves the changes to a separate file. It enables the user to save a backup copy of the configuration or transfer it to another computer.

Moreover, this menu item allows you to export a list of tags in CSV format. You can edit it in a text editor or create a spreadsheet using Excel or other application. After editing the file, you can import it using the Open tab.

Display Message Log

The function opens a message log file that lists warning messages and errors that may occur while the program is running. It is recommended to use this file during troubleshooting.

Exit

The function is used to terminate program execution.

4.3 Edit menu

Add

Hotkey: Insert

The function is used to add and configure a new link between tags.

Edit

The function is used to edit a connection from the list. The edit window will open when you double-click on a connection in the list.

Copy

The function is used to copy a connection from the list. By using the function, you copy complete settings which you can edit anytime. The function is useful when adding multiple connections of the same type.

Delete

Hotkey: Delete

The function deletes a previously created connection. This menu item will only become available when you select one or more connections from the list.

Options

Hotkey: Ctrl+P

The function opens a new dialog box with [program settings](#).

4.4 View menu

AutoFit Column Width

The function is used to change a column width to automatically fit the text length.

AutoFit Column Size

Thanks to this function, the program will automatically adjust your column size to fit the size of the main window.

Transactions File

Hotkey: Ctrl+L

The function opens a new program window that lists the last 10,000 transactions. The content of this window is identical to that of the [transactions file](#).

4.5 Program settings

The program settings do not depend on the type of data source. The settings feature a number of options to customize the appearance and behavior of the program. To access the settings, go to Edit menu in the main window.

View Options

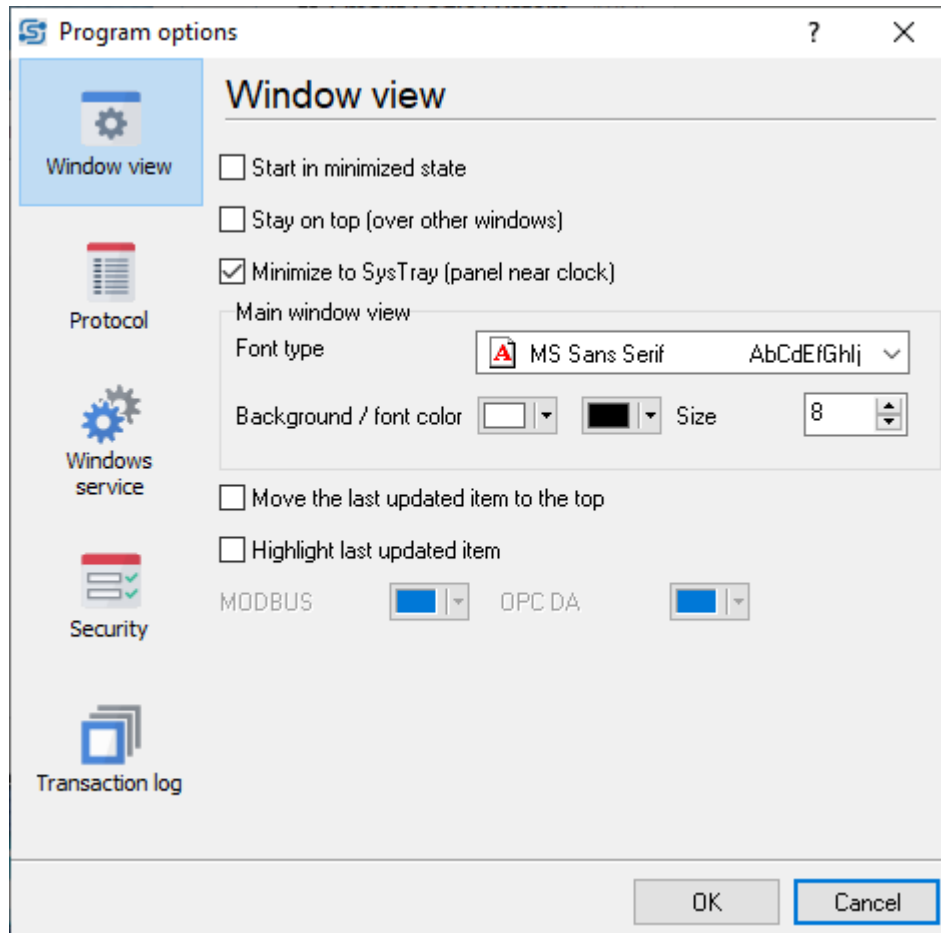


Fig. 3. View Options.

Run Minimized - when you launch Datatag Link, the main window will stay minimized in the taskbar or will be minimized as an icon in the system tray.

Minimized in System Tray - the main window will be automatically minimized, and the icon will be placed next to the clock.

Always On Top - the main window will always be on top of other programs on the desktop.

Main Window View - allows you to customize the data display window (font type, background, color).

Move Last Updated Item on Top - when this option is enabled, with each data transaction the corresponding link of tags will move to the top of the list for three seconds. This makes it easy to monitor progress when there are many tags in the list.

Highlight Last Updated Item - when this option is enabled, with each transaction, a line in the list will be highlighted by a custom color for three seconds. The color may be different depending on the data sending source.

Protocol File

A log file records user-defined types of program messages that may occur while the program is running.

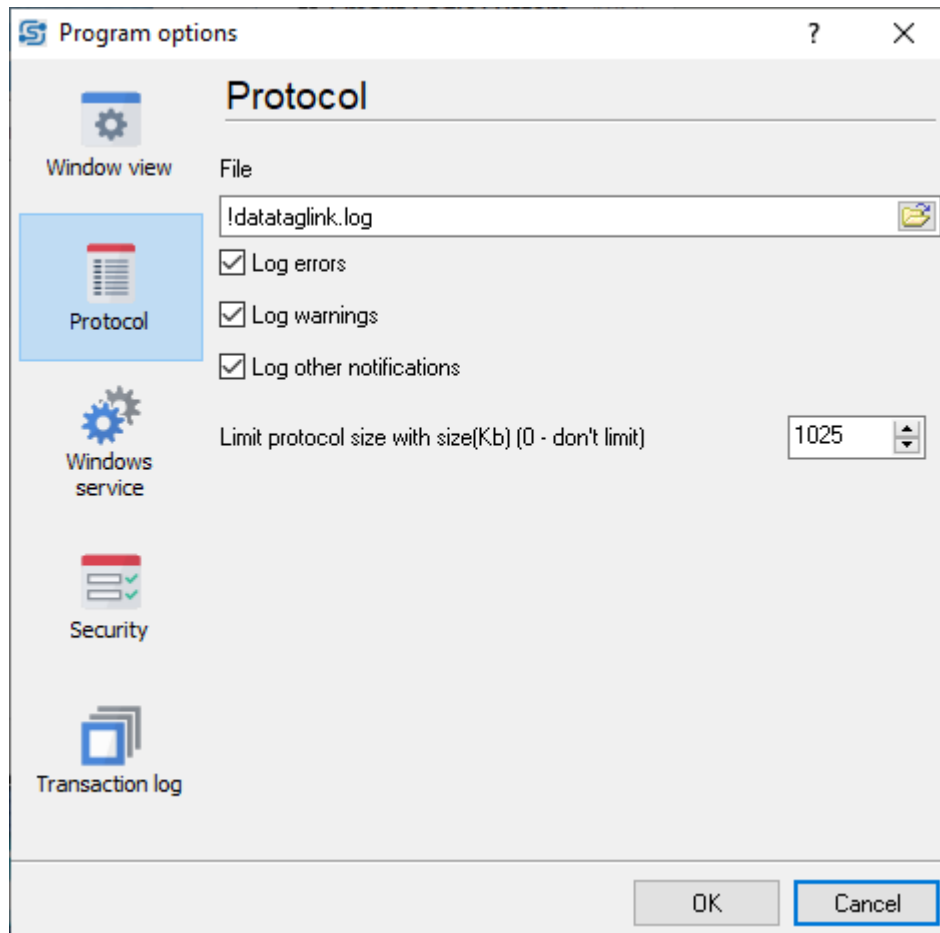


Fig. 4. Log file.

File - use this field to specify a log file path.

Log Errors, Warnings, and Other Messages - here you can specify the types of messages that will be recorded in the log file.

Limit Protocol Size (Kb) - defines the maximum size of the log file. Once the set limit is exceeded, the program will start writing to a new file.

Windows Service

Using a Windows Service enables you to:

- Automatically launch the program when Windows starts, even before the user logs in.
- Run the program in the background. In this mode, a general user will not be able to close the program.
- When run as a service, it is possible to configure restart settings in the case of errors.

Attention: To be able to change the configuration or any settings of the service (start, stop, pause, resume), you must launch the program as an administrator. When using Windows Vista, you must run the program with elevated permissions (right-click on the icon - Run as administrator).

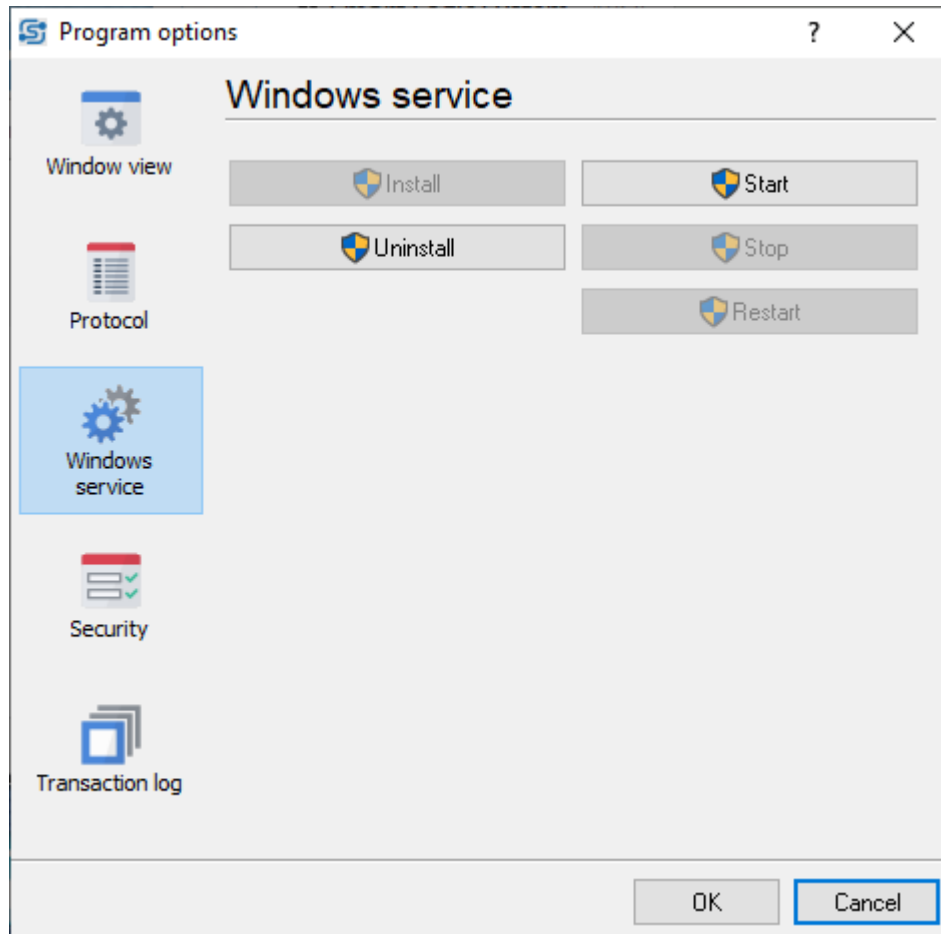


Fig. 5. Windows service.

Install - install a service into the operating system.

Delete - stop and delete a service.

Start, Stop, Restart - functions that are used to manage a service status

It is also possible to manage a service using the [command line](#).

Security

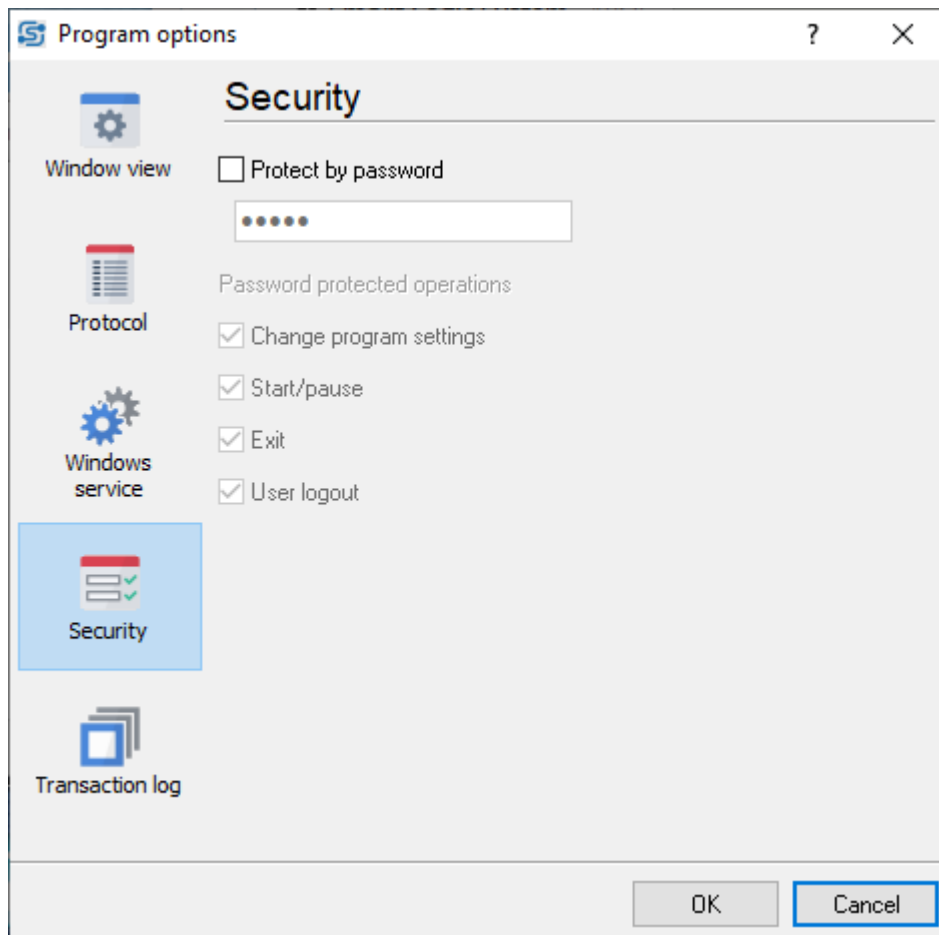


Fig. 6. Security.

This tab enables you to selectively password protect certain actions with the program. Enable the Password Protect option, set a password, and select the restricted actions from the list.

Change Program Settings - refers to any changes in the program configuration.

Start/Pause - the function enables you to start / stop data transmission using the button in the main window of the program.

Exit - shuts down the program. The function does not affect the program operation when it is running as a background service.

User logout - if the program is running on the desktop, it will block attempts to terminate the user session, which may cause the program to stop. In this case, you will have to shut down the program manually and then end the user session.

Transactions Log

These options are used to configure saving of the transaction history to a CSV file.

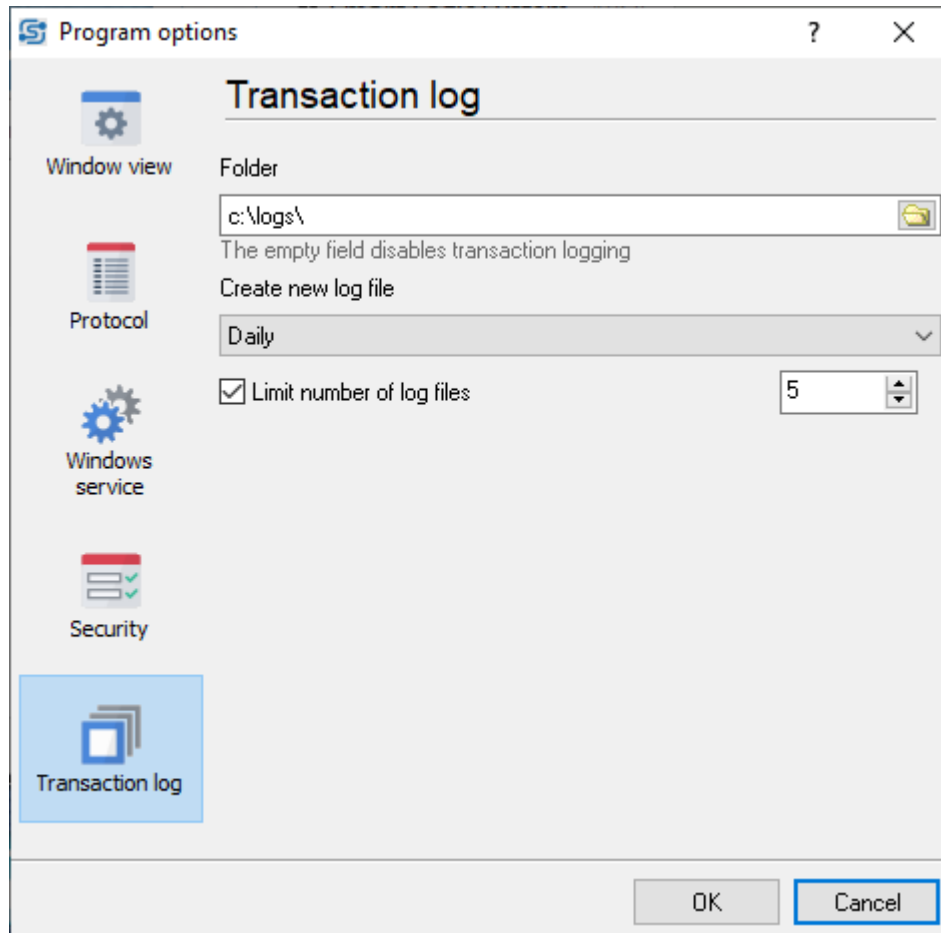


Fig. 7. Transaction log.

Folder - used to specify a folder path where CSV files will be created. If the path is not specified, saving transactions to a file will be disabled.

Create a New Log File - the function allows you to select a mode of creating new log files, depending on the amount of data transmitted.

Limit Number of Log Files - when the option is enabled, the program will automatically delete old log files at the time of creating a new log file.

Transaction File Format

A transaction file has a fixed format that corresponds to the CSV format. See an example of records below:

```
"Timestamp";"Sender";"Server";"Item";"DataType";"Value"
"2022-04-15 15:14:16.382";"MODBUS";"127.0.0.1";"node1.node2.tag2";11;TRUE
```

Timestamp - timestamp of data transmission in the computer's time zone.

Sender - a source of the data sender.

Server – an ID of the recipient's server, if it is available in the settings.

Item – a tag ID on the recipient server.

Data Type – a numeric ID of the data type value.

- 1 - Null.
- 2 - Smallint (I16).
- 3 - Integer (I32)
- 4 - Single (F32)
- 5 - Double (F64)
- 6 - Currency (F64)
- 7 - Date
- 8 - String
- 11 - Boolean (TRUE or FALSE)
- 16 - ShortInt (I8)
- 17 - Byte (U8)
- 18 - Word (U16)
- 19 - LongWord (U32)
- 20 - Int64 (I64)
- 21 - UInt64 (U64)

Value - transmitted value.

4.6 Data sources

4.6.1 OPC DA

The data source is an OPC DA server. That means that any client software compatible with the OPC standard can receive data from Datatag Link in real time. In order to connect to the server, you need to know its attributes, which can be viewed by clicking on the Server Name button (see Fig. 8). Before using an OPC server, you need to install OPC Core Components Redistributable on the computer where the server and your client software will be running. You can get the package on the [official website](#) of the OPC Association (registration on the website is required).

OPC DA

Tag name: node1.node2.tag

Data type: Float 32-bit (F32)

You can specify the path in a tree hierarchy like: node1.node2.value

Server name

Fig. 8. OPC DA.

Tag Name - you have to specify an ID of the OPC tag that you want to link to another data source. An OPC server supports hierarchical representation of data, and you can specify a path to the tag in the OPC tag tree as node1.node2.tag (dot is used as a separator). When creating a tag name, it is recommended to use only Latin letters and numbers.

Data Type - this field is used to specify the data type of OPC tag. When necessary, in the case the data types of different sources do not match, the program will try to convert such data types. When Auto is specified in the Data Type field, the program will use the data type of paired source.

4.6.2 OPC UA

The data source is an OPC UA DA server. This type of the OPC server is a later version of the [OPC DA](#) standard. It allows you to easily receive data from remote computers without installing additional software or using a complex configuration procedure.

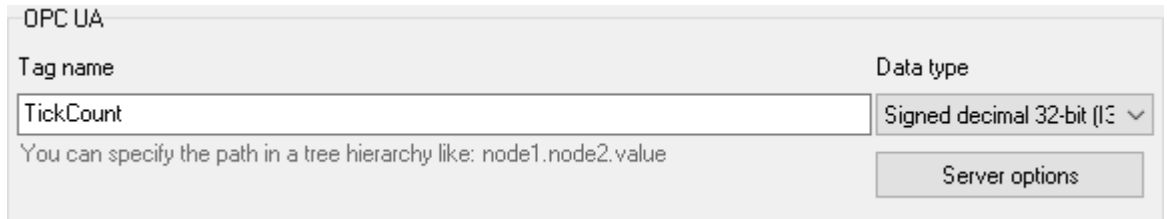


Fig. 9. OPC UA.

Tag Name - you need to specify the ID of an OPC tag that you want to link to another data source. An OPC server supports hierarchical representation of data, and you can specify a path to the tag in the OPC tag tree as node1.node2.tag (dot is used as a separator). When creating a tag name, it is recommended to use only Latin letters and numbers.

Data Type - this field is used to specify the data type of OPC tag. When necessary, in the case the data types of different sources do not match, the program will try to convert such data types. When Auto is specified in the Data Type field, the program will use the data type of paired source.

Server Options - this button opens an additional window to configure an OPC server (see Fig. 10). Kindly note that you can only configure one OPC UA server, and it will have the same parameters for all connections.

The screenshot shows a 'Server configuration' dialog box with the following fields and values:

- Name:** opc ua server
- IP address:Port:** 0.0.0.0 (IP address) and 10001 (Port)
- Login/Password:** test (Login) and **** (Password)
- Enable secure (SSL/TLS) connection:**
- Certificate storage location:** CertificateStore_\
- Local discovery server address:** (Empty)

Buttons: OK, Cancel

Fig. 10. OPC UA server parameters.

Name - the name of an OPC server displayed when connecting to the server from an OPC client.

IP Address and Port - network parameters used to connect to the server. IP address 0.0.0.0 is set by default and allows you to connect to the server via any network interface available on the computer. You can use any server port. However, it is recommended to use a port number greater than 1000.

Login and Password - when login and password are set, a server will verify these credentials while connecting OPC clients. You may leave these fields empty.

Certificate storage location - if a path to the certificate's store is specified in this field, an OPC client will use a secure connection to connect to the server, and its public certificate will be placed in that folder. There should be three subfolders in this folder:

- **private** – a folder with the certificate of the OPC server itself (server_cert.der) and its private key (server_key.der). Without these certificates, the program cannot encode traffic.
- **rejected** - this folder contains certificates of the clients that are denied access to the server.
- **trusted** - this folder contains certificates of the clients that are allowed access to the server. Until the client's certificate appears in this folder, the client will not be able to access the server.

Local discovery server address is the address of a special server on the network or on a local computer that stores information about all active OPC UA servers and facilitates connection to them. If you do not have an LDS server, leave this field blank.

4.6.3 MQTT

This data source is an MQTT client that connects to another MQTT broker, then subscribes to the specified values and transmits data between the sources.

Fig. 11. MQTT.

Broker - the program allows you to set up a connection to one or more brokers at the same time. Choose a connection from the list of previously configured brokers, or use the Add button to set up a new connection to the broker (Fig. 12). The other buttons in the window are used to change connection settings and delete connections.

Tag Name - the function enables you to specify the path and name of a value on the MQTT broker. The "/" character is used as a separator in the path.

Data Type - since all data on the MQTT broker are presented in the text form, the data type is not specified, but the program expects that the format of the values should be matching those specified in the connection settings.

Retain - when sending data to an MQTT broker, the program will set a Retain flag, which indicates to the broker that it is necessary to save the last received value.

QoS - in this field you can specify priority of a value from 1 (regular) to 4 (maximum).

JSON Value - oftentimes, a single tag in an MQTT broker stores a text representation of an object in JSON format, which may contain multiple values. JSON format allows you to unify the format for storing values, thus reducing the number of individual tags on an MQTT broker, which may be appropriate in complex applications. The value specified in this field defines the name of the value in the JSON string. When receiving data from a broker by subscription, the program will automatically locate the specified value and convert the data type. When sending data, the program will only change this value.

Connecting to an MQTT broker

The screenshot shows a 'Broker configuration' dialog box with two tabs: 'Connection' and 'Data format'. The 'Connection' tab is active. It contains the following fields and controls:

- Name:** A text box containing 'test.mosquitto.org'.
- IP address:Port:** A text box containing 'test.mosquitto.org:8883'.
- Login/Password:** Two empty text boxes.
- Enable secure (SSL/TLS) connection:** A checked checkbox.
- CA certificate:** A text box containing 'mqtt\ca_cert_mosquitto.org.crt' with a file selection icon on the right.
- Client certificate:** A text box containing 'mqtt\cert\client.crt' with a file selection icon on the right.
- Client private key:** A text box containing 'mqtt\cert\client.key' with a file selection icon on the right.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom. The 'Cancel' button is highlighted with a blue border.

Fig. 12. Broker connection settings.

Name is a text description of a broker that is displayed in the list, as in fig. 11.

IP Address and Port are the IP address or URL of the broker on a local network or the Internet. A broker port name uses a colon as a separator.

Login and Password - the function enables authorization in the case a broker does not allow anonymous connections.

Enable Secure Connection Mode – the function encrypts all traffic between the program and the broker. This encryption uses a set of certificates and keys.

Certificate of Certification Authority (CA) - a file that must be provided by a broker's administrator.

Client Certificate and **Private Key** of the Client are generated by the user. To achieve that, you can use an OpenSSL utility (not included in the scope of supply). Also, to better protect the connection, it is possible to specify the client's public key on the broker.

Data Format Settings (Fig. 13) are used when converting values into a string in order to transmit them to an MQTT broker. These settings do not apply to values in JSON format.

The screenshot shows a dialog box titled "Broker configuration" with a close button (X) in the top right corner. It has two tabs: "Connection" and "Data format". The "Data format" tab is active. It contains three rows of configuration options:

Option	Format
Boolean True	Date
true	YYYY-MM-DD
Boolean False	Time
false	HH:NN:SS
Decimal point	Date and time
	YYYY-MM-DD HH:NN:SS

Below the last row, there is a note: "Leave fields empty to use the default format from your OS". At the bottom of the dialog, there are two buttons: "OK" and "Cancel". The "Cancel" button is highlighted with a blue border.

Fig. 13. Data format.

4.6.4 SQL

This data source allows reading and writing data in a supported SQL database. The program will periodically check a database for new data, and if a value has changed, it will transmit it to the linked data source.

The screenshot shows the SQL configuration window with the following details:

- Database:** A dropdown menu showing 'test db', with '+ Add', a minus sign, and a red 'X' button.
- Parameters:**
 - Data type:** Unsigned decimal 32-bit (U32)
 - Mode:** Table and column
 - Polling interval:** 30 seconds
- Table/column:** Two text input fields containing 'modbus_data' and 'value1'.
- Row identifier:** A table with the following structure:

Column	Type	Value	Data type
ID	Exact match	1	Unsigned decimal 32-bit

Below the table, a note states: 'The table name may include the schema name'.

Fig. 14. SQL.

Database - the program allows setting up connections to several different databases at the same time. Use the list of previously configured connections to select a connection, or click the Add button to configure a new connection (Fig. 16). The other buttons in the window are used to change connection settings and delete connections.

Data Type - defines the data type of the value to be read from the database. It must match the data type of the column where the data is stored.

Mode – the function is used to select a method of receiving and recording data.

Interval - the function sets an interval for checking availability of new data in the database.

Table and Column Mode

It is required to specify the name of the table and column, and the program will automatically generate SQL queries for reading, writing and updating data.

The row ID field is used to indicate the column name with a unique value for each row (Primary key). The program uses that name to receive a value from the corresponding row in the database. If there is no such row available when data are written into the database, a new row will be added.

If Exact Match is specified in the Type field, the program will find that single value by the identifier specified in the Value field. In this case, a table is a matrix of values in which a new value always overwrites the old one.

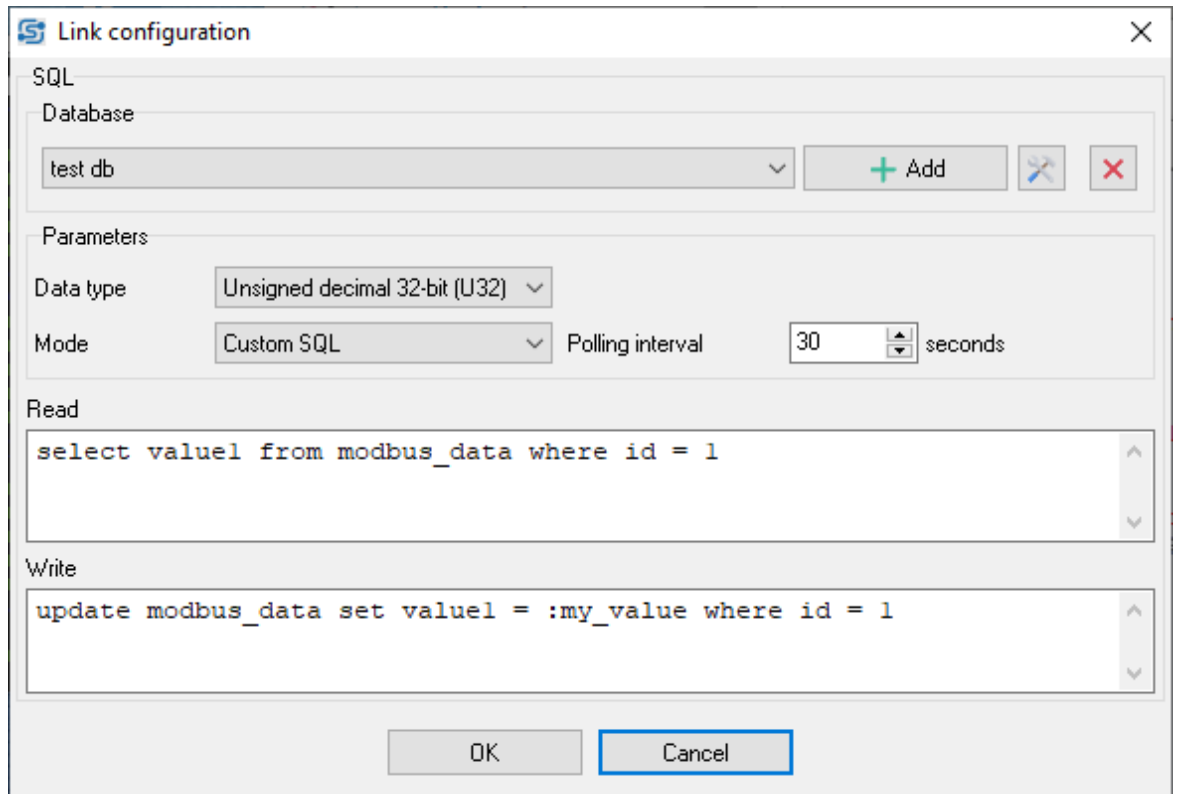
When the Addition value is selected in the Type field, it is assumed that the unique identifier will increase when a new value is added. While reading data, the program selects from the database all new values whose identifier is greater than the last received identifier value. When writing a value to the database, the program will always add a new line. In this case, the table will contain all modifications of a value. And it is up to the user to remove old or unnecessary values from the table.

Custom SQL Mode

In this mode, it is the user who creates an SQL query to read and write data (Fig. 15). It is convenient to use this function to receive and write data through a stored procedure on an SQL server using a complex data processing algorithm.

In this mode, when writing data, the place of a value in the SQL query is marked with a special parameter ":my_value". When inserting values, the data type of value is regarded.

When reading data from the database, the program takes the value from the first column that the SQL query returns. An SQL query can return multiple rows.



The screenshot shows a 'Link configuration' dialog box with the following sections:

- SQL**
 - Database:** A dropdown menu showing 'test db' with '+ Add', a pencil icon, and a red 'X' icon.
 - Parameters:**
 - Data type:** A dropdown menu showing 'Unsigned decimal 32-bit (U32)'.
 - Mode:** A dropdown menu showing 'Custom SQL'.
 - Polling interval:** A numeric input field showing '30' with up/down arrows, followed by the text 'seconds'.
 - Read:** A text area containing the SQL query: `select valuel from modbus_data where id = 1`.
 - Write:** A text area containing the SQL query: `update modbus_data set valuel = :my_value where id = 1`.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

Fig. 15. SQL queries.

Database Connection Parameters

Fig. 16. Database Connection Parameters.

Name is a text description of a database that is displayed in the list, as in Fig. 14.

Type – the type of database.

Server is the IP address or name of the computer on which a database is running.

Database - the name of a database on your database server.

Login and Password - this function enables user authorization.

Test Connection - the function checks if it is possible to connect to the database.

Additional parameters - you can specify database-specific parameters here. For example, a non-standard port like

SERVER PORT=2200

Parameter	Description	Database
SERVER PORT	The non-standard port for IP connections	All

4.6.5 MODBUS

This data source can function as the master using MODBUS RTU or MODBUS TCP protocols and receive values from the slave.

Data exchange can be carried out via a COM port or over an Ethernet network.

When exchanging data over a single connection, the program will line up all requests in one queue and execute them sequentially. Write requests always get to the front of the queue, ahead of read requests. A request at the beginning of the queue will be executed first.

When a connection over several ports is set up, data exchange will be carried out in parallel, each connection having its own queue.

It is not allowed to create different connections using the same port number.

Fig. 17. MODBUS.

Connection - the program allows you to configure a connection to several slave devices at the same time. Use the list of previously configured connections to select a connection, or click the Add button to configure a new connection (Fig. 18). The other buttons in the window are used to change connection settings and delete connections.

Name - specifies a value name for your data.

Device Address - the field is used to specify the address of your device in the network of MODBUS devices. According to the MODBUS protocol specification, an address is a number from 0 to 255. If you specify 0 as the device address, then all devices on the network will respond to this request.

Memory - defines the memory area of the slave device from which values will be read.

Memory Offset - Each memory area can have up to 65536 registers with an offset from 0 to 65535 (depends on the slave device). Sometimes in the device documentation, an offset is not indicated, rather an address of the type 40100. In this case, the first digit indicates the memory area and the

other digits indicate the address from the range 1 to 65535. That means you need to deduct 1 from the address and specify the resulting value as an offset. Example:

Address: 40100
Memory area code: 4
Memory address: 100
Offset: 99

Memory area codes:

0 - Coils.
1 - Discrete inputs.
3 - Input registers.
4 - Holding registers.

Number of Registers - specifies the number of serial registers in the device memory in which a value is stored. A value can occupy several serial registers in the device memory, depending on the type of the value. One register stores two bytes of data. For example, the data type "I32" will occupy two registers in the memory (32 bits → 4 bytes → 2 registers of 2 bytes each).

Data Type - defines the data type of value. As a rule, memory areas 0 and 1 store logic values, while 3 and 4 store numeric values.

Byte order in the register/value depends on the data storage method in internal memory. Although there is no specific standard for data storage, the most common storage method is when the oldest register of a value and byte in the register are transmitted in a data packet first, which corresponds to BA and DC order.

Polling

Interval - you can set the polling interval for a MODBUS device to check for a new value in a MODBUS register. A lower value provides a faster reaction to changes, but requires more CPU resources and lead to higher network traffic.

Cron schedule - these parameters allows you to set a [flexible schedule](#) to poll a MODBUS device. You can use it for rare reads.

Connection

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

- Name:** 127.0.0.1
- Type:** COM
- Mode:** MODBUS/TCP
- Request timeout (ms):** 1001
- Parameters:**
 - COM port:** COM11
 - Baud rate:** 57600
 - Data bits:** 8
 - Parity:** None
 - Stop bits:** 1
- RS485 interface mode
- Buttons:** OK, Cancel

Fig. 18. Connection via COM port.

Connection options have a few differences depending on the type of connection.

The common options are:

Name - the name of a connection that is displayed in the list of connections (Fig. 17).

Type - connection type (COM port or network).

Mode - specifies the type of MODBUS protocol. Usually, MODBUS RTU is used when transmitting data via a COM port, and MODBUS TCP is used when transmitting data over a network. However, in some cases, MODBUS RTU may be used with a network connection.

MODBUS RTU (Enron) - in this mode, a single register contains a 32-bit value in the controller's memory, apart from the standard register size of 16 bit.

MODBUS RTU (Slave) or MODBUS TCP (Slave) - in this mode, the program acts as a MODBUS slave device. It does not send any requests and only responds to requests of another MODBUS master. When a linked data interface sends data to the MODBUS interface, the program saves data in the corresponding register in the program's memory, and the MODBUS master device may read this value later. All uninitialized registers contain zeros. The MODBUS request should contain the device address defined on one of tag links (Fig. 17).

Request Timeout is the interval from sending a request to receiving a response from a MODBUS device. If the timeout is exceeded, the program will automatically cancel the current request and proceed to the next request in the queue. If the response comes after this interval, it will be ignored.

The timeout value depends on several factors: connection speed, device performance, the amount of data transmitted in a request or response.

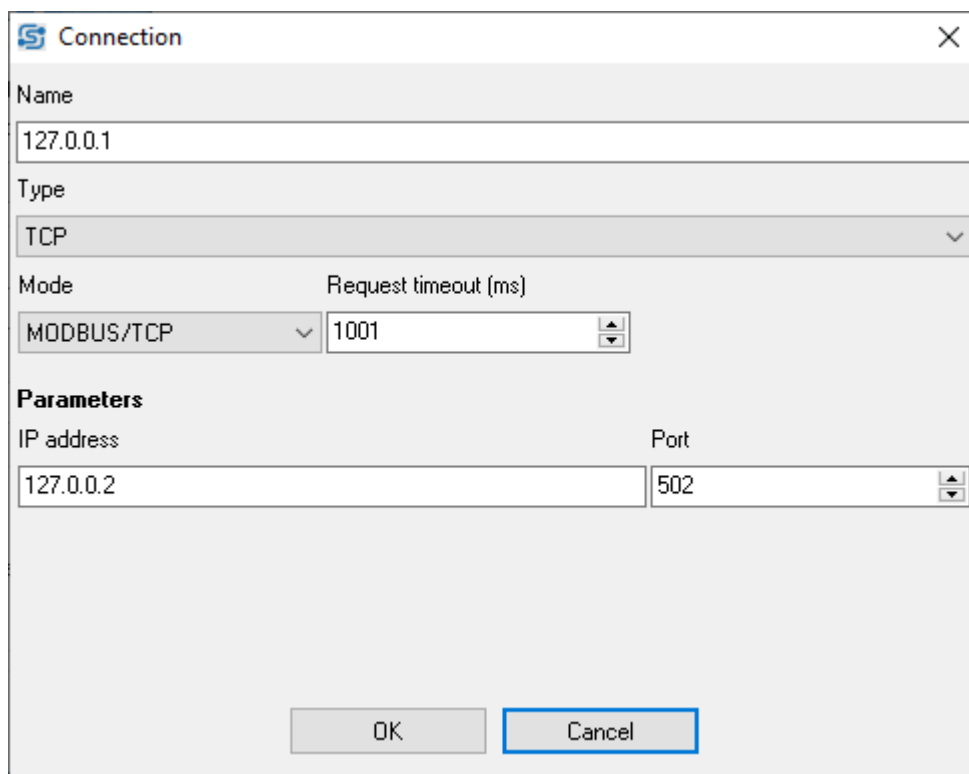
Connection Options via COM port

COM port - allows selecting a port from the list. The program automatically detects all available ports in the operating system.

Speed, Data bits, Parity, Stop bits - these options define settings of communication with a device and depend on the device settings.

RS 485 Interface Mode - the option enables automatic control of the program by the signal of RTS COM port. During data transmission, the program will always keep this port enabled. This may be required when using external RS232-RS485 hardware converters.

Network Connection Parameters



The screenshot shows a 'Connection' dialog box with the following fields and settings:

- Name:** 127.0.0.1
- Type:** TCP
- Mode:** MODBUS/TCP
- Request timeout (ms):** 1001
- Parameters:**
 - IP address:** 127.0.0.2
 - Port:** 502

Buttons: OK, Cancel

Fig. 19. Network connection.

IP Address - IP address of the device.

Port – the field is used to specify a TCP port to connect to. The standard value is 502. Kindly note that some MODBUS devices limit the number of simultaneous connections, and sending requests to one device from different computers may not be possible.

4.7 Command like parameters

/I - install the service with manual startup.

/A - install the service with automatic startup.

/D - install the service with disabled status.

/S - start the service.

/P - stop the service.

/SS - start the service if it is pending. Or restart it if it is already running.

/R - remove the service from the system.

/C=Y - changes the service startup settings, where Y can take the values of I, A, D (see above).

/FILE=XXXXX - when the program starts, a configuration will be loaded from the file XXXXX.

Note: You can specify several parameters in the command line in the order in which they are executed.

Examples:

```
datataglink /FILE=c:\DataLink\Config.xml
```

```
datataglink /A
```

```
datataglink /R
```

4.8 FAQ and known issues

Q: I would like to manage configuration, create and delete connections of the program using my own software. Is that possible?

A: All settings are stored in an XML configuration file, which is easy to edit with any text editor or create in your own application. Once you have changed the configuration, use the [command line](#) to restart the service running in the background.

Q: I can't connect to an OPC DA server from a remote computer.

A: A remote connection to such a server is possible in theory, however is difficult in practice due to variety of Windows versions, security settings and [DCOM](#) software components. We recommend using [OPC UA](#) if possible.

4.9 Cron time format

The CRON format is a simple yet powerful way to describe time and operation periodicity. The traditional (inherited from the Unix world) CRON format consists of five fields separated with spaces:

<Second> <Minutes> <Hours> <Month days> <Months> <Weekdays>

Any of the five fields can contain the * (asterisk) character as its value. It stands for the entire range of possible values. For example, every minute, every hour and so on. In the first four fields, you can also use the proprietary "?" (w/o quotes) character. See its description below.

Any field can contain a list of comma-separated values (for example, 1,3,7) or an interval (subrange) of values defined by a hyphen (for example, 1-5).

You can use the / character after the asterisk (*) or after an interval to specify the value increment. For example, you can use 0-23/2 in the "Hours" field to specify that the operation should be carried out every two hours (old version analog: 0,2,4,6,8,10,12,14,16,18,20,22). The value */4 in the "Minutes" field means that the operations must be carried out every four minutes. 1-30/3 is the same as 1,4,7,10,13,16,19,22,25,28.

You can use three-word abbreviations in the "Months" (Jan, Feb, ..., Dec) and "Weekdays" (Mon, Tue, ..., Sun) fields instead of numbers.

Examples

Note: the <Second> field equal 0 in all examples

Format	Description
* * * * *	every minute
59 23 31 12 5	one minute before the end of the year if the last day in the year is Friday
59 23 31 Dec Fri	one minute before the end of the year if the last day in the year is Friday (one more variant)
45 17 7 6 *	every year on the 7th of June at 17:45
0,15,30,45 0,6,12,18 1,15,31 * 1-5 *	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays
*/15 */6 1,15,31 * 1-5	00:00, 00:15, 00:30, 00:45, 06:00, 06:15, 06:30, 06:45, 12:00, 12:15, 12:30, 12:45, 18:00, 18:15, 18:30, 18:45, if it is the 1st, 15th or 31st of any month and only on workdays (one more variant)
0 12 * * 1-5 (0 12 * * Mon-Fri)	at noon on workdays
* * * 1,3,5,7,9,11 *	every minute in January, March, May, July, September, and November
1,2,3,5,20-25,30-35,59 23 31 12 *	on the last day in the year at 23:01, 23:02, 23:03, 23:05, 23:20, 23:21, 23:22, 23:23, 23:24, 23:25, 23:30, 23:31, 23:32, 23:33, 23:34, 23:35, 23:59

0 9 1-7 * 1	on the first Monday of every month at 9 in the morning
0 0 1 * *	at midnight on the 1st of every month
* 0-11 * *	every minute till noon
* * * 1,2,3 *	every minute in January, February, and March
* * * Jan, Feb, Mar *	every minute in January, February, and March
0 0 * * *	every day at midnight
0 0 * * 3	every Wednesday at midnight

You can use the proprietary "?" character in the first four fields of the CRON format. It stands for the start time, i.e., the question mark will be replaced with the start time during the field processing: minute for the minute field, hour for the "Hours" field, month day for the month day field, and month for the month field.

For example, if you specify:

? ? * * *

The task will be run at the moment of startup and will continue being run simultaneously (if the user does not restart the program again, of course) – the question marks are replaced with the time the program was started at. For example, if you start the program at 8:25, the questions marks will be replaced like this:

25 8 * * * *

Here are some more examples:

- ? ? ? ? * - run `_only_` at startup;
- ? * * * * - run at startup (for example, at 10:15) and continue being run in exactly one hour: at 11:15, 12:15, 13:15 and so on;
- * ? * * * - run every minute during the startup hour;
- */5 ? * * * - run on the next day (if CRON is not restarted) at the same hour every minute and so on every day, once in five minutes, during the startup hour.