The Failover plugin
PRINTED MANUAL

Failover plugin

©2018-2019 AGG Software

Printed: 7/17/2019

Publisher

AGG Software

Production

©2018-2019 AGG Software
http://www.aggsoft.com

Table of Contents

1   Introduction

This data filtering module allows you to create a distributed backup system for receive interfaces while creating a failover system. The system can significantly reduce downtime and data loss in certain emergency situations.

At the configuration stage, you need to install and configure a logger program on the primary and backup computers. During operation, the module monitors the state of received data, and in the case, any problems are detected, switches the receive and handle functions over to the backup program. As soon as the primary interface has been restored, the secondary interface gets automatically disabled.

With regard to each unusual event, the system can generate a notification for the administrator.

2   System requirements

The following requirements must be met for "Failover" to be installed:

Operating system: Windows 2000 SP4 and above, including both x86 and x64 workstations and servers. A latest service pack for the corresponding OS is required.
Free disk space: Not less than 5 MB of free disk space is recommended.
Special access requirements: You should log on as a user with Administrator rights in order to install this module.

The main application (core) must be installed, for example, Advanced Serial Data Logger.

Notes for Microsoft Vista and above:

Since our software saves data to the registry and installs to the Program Files folder, the following requirements must be met:

1. You need Administrator rights to run and install our software
2. The shortcut icon of our software will be located on the desktop;
3. Windows Vista will ask for your confirmation to continue the installation.

NOTE: You can configure the user account only once in order not to see the above dialog box any more. Search Google for the solution of this problem.

3   Installing Failover

1. Close the main application (for example, Advanced Serial Data Logger) if it is running;
2. Copy the program to your hard drive;
3. Run the module installation file with a double click on the file name in Windows Explorer;
4. Follow the instructions of the installation software. Usually, it is enough just to click the "Next" button several times;

5. Start the main application. The name of the module will appear on the "Modules" tab of the "Settings" window if it is successfully installed.

If the module is compatible with the program, its name and version will be displayed in the module list. You can see examples of installed modules on fig.1-2. Some types of modules require additional configuration. To do it, just select a module from the list and click the "Setup" button next to the list. The configuration of the module is described below.

You can see some types of modules on the "Log file" tab. To configure such a module, you should select it from the "File type" list and click the "Advanced" button.
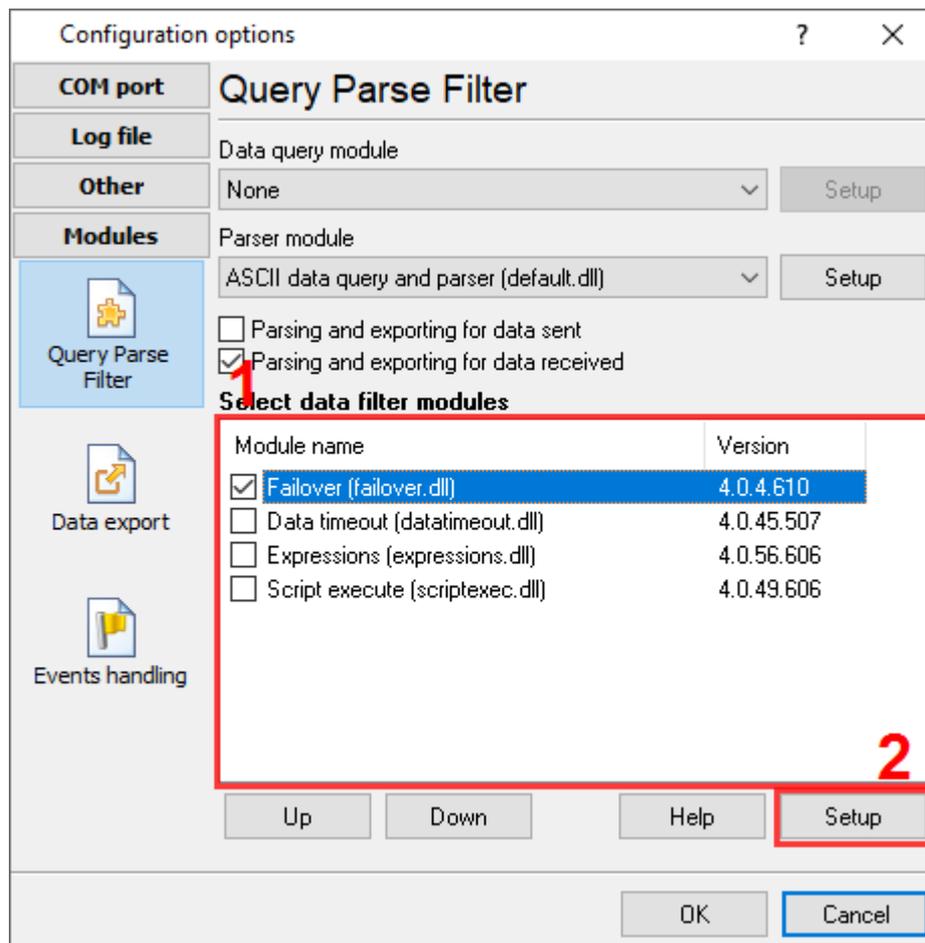
Fig.1. Example of installed module

4   Glossary

Main program – the main executable of the application. For example: Advanced Serial Data Logger and asdlog.exe. It allows creating several configuration. Each configuration may use many plugins.

Plug-in - the additional plugin module for the main program. The plugin module extend the functionality of the main program.

Parser – the plugin module that processes the data flow, singling out data packets from it, and then variables from data packets. These variables are used in data export modules after that.

Core - see "Main program".

5 Settings

Module configuration (Fig.2) shall be applied to two copies of the program (data reception and handling interfaces) installed on one, two or several different computers. You can first configure one instance of the program, export the settings, and import them into another instance. Please note that the program does not synchronize changes in the settings of a running program instance.

Module settings can be applied in one or several program configurations. The "Failover" module will only operate in the configurations where it is enabled (Fig. 1). Failure in any of the monitored configurations will result in switching to the backup interface.

Fig. 1. Enabling the module

Fig. 2. Settings window

Basic settings

This tab allows you to configure the basic parameters of connection between the primary and secondary interfaces. The interface goes to error state whenever a problem is detected. In the event of a problem, the logic of operation is determined by the interface role.

Name - the name of a configuration in which the module is running. This name will be displayed in the message log of the program.

Role - the interface role (primary or secondary). If the Failover module is enabled in more than one configuration, the same role must be selected in each configuration.

The primary interface manages one or more secondary interfaces and controls the switching between them.

The secondary interface starts in standby mode, connects to the primary interface, and waits for turn on and off commands.

Priority - the priority of the secondary interface. Lower value has a higher priority. When switching over to the secondary interface, the interface with a higher priority will be selected.

Primary interface IP - the address of the computer on which the primary interface is configured and running. The secondary interface will continually attempt to connect to the specified address and resume communication with the primary interface.

Port - the TCP/IP port used for connecting interfaces. It must match the primary and secondary interfaces.

Mode - the following interface run modes are possible:

Cold - in this mode, the program does not connect to data sources (servers, devices, files, etc.), and does not receive any data. This is equivalent to pressing the "Pause" button in the main program window. To ensure operation of the secondary interface in this mode, you must disable the option "Start recording automatically" on the tab "Other - Name and security," in the configuration settings of the main program.

Hot - in this mode, the program connects to data sources and receives data from them. Data are then passed to the parser and handled by it. Then data are passed to the data filter modules that are located in the list above the Failover module (Fig. 1). If higher-listed modules skip the data, then the Failover module takes over.

If the interface is currently inactive, the module filters all data. Thus the data escape the data filter modules located lower in the list and data export modules (database, Excel, etc.). If the interface is active (running), the data pass the "Failover" module freely and get processed further on. It should be noted that writing to the log file can be done both before and after filtering. Accordingly, the mode of writing to a log file is also defined in the log file settings.

Switch to secondary interface in case of problems with watchdog items - if this option is enabled, the module will monitor the state of the parser variables in accordance with the parameters on the "Watchdog items" page.

Switch to secondary interface in case of log errors - if this option is enabled, the module will monitor log messages of the program in accordance with the parameters at page "Watch log messages."

Switch to secondary interface if connection with primary is lost - if this option is enabled, in the case of any connection failure, the secondary interface will immediately engage and start receiving and processing data. As soon as the connection is resumed, the running primary interface will be able to suspend the operation of the secondary interface. This option starts only after the secondary interface has successfully connected to the primary interface at least once during the current session of the main program and module.

Generate program event in case of problem - if this option is enabled, the program will generate an event with the specified identifier when a problem is detected and before switching over to another interface. This event can be handled by event handling modules and, for example, send a message to the administrator using the "Events notification" module. As the context of this event, the following values are passed:

    NAME - the name of a data interface.
    PRIORITY - the priority of a secondary interface.
    ERROR - the error type. One of the following values: ITEMS, LOG, SQL

Example of using "Events notification" module to handle an event: https://www.aggsoft.com/asdl-data-timeout.htm
Documentation for the "Events notification" module": https://www.aggsoft.com/serial-data-logger/plugins/events.pdf

Watchdog items

The tab "Watchdog items" (Fig. 3) allows you to specify a list of parser variables whose state you want to monitor. It is assumed that the program receives and handles data with a certain frequency.
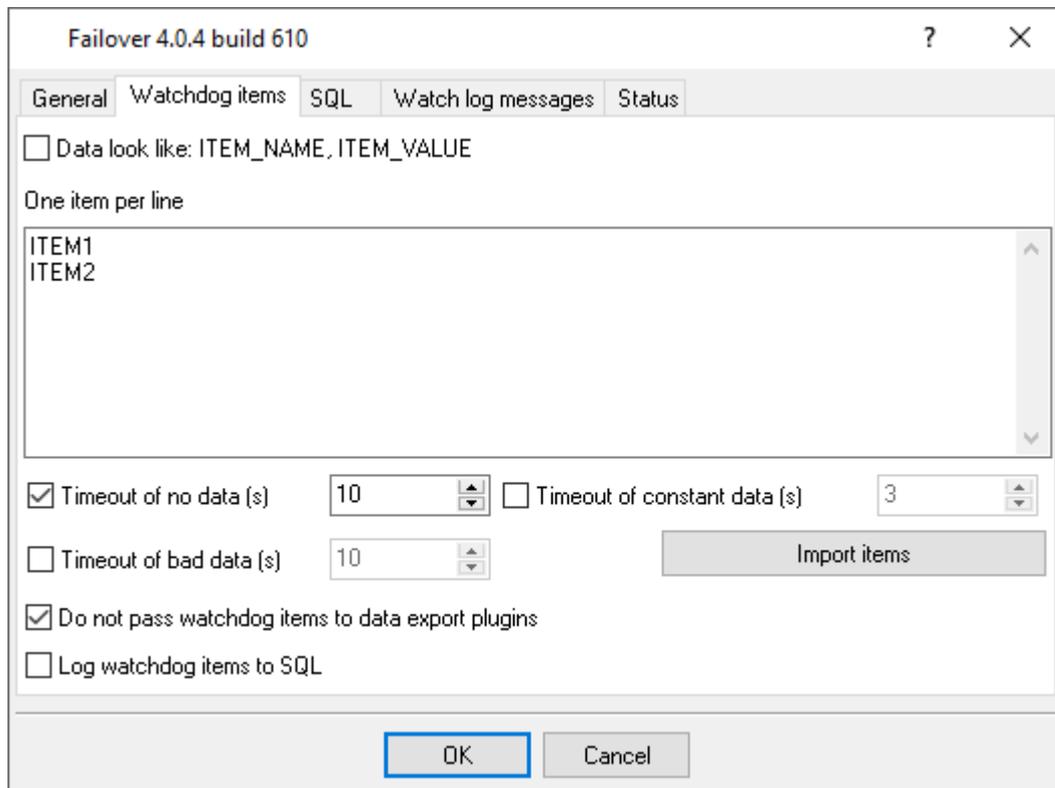
Fig. 3. Watchdog items parameters

Data look like: ITEM_NAME, ITEM_VALUE - some of our loggers can output data in this format when there are a lot of values, and it is inconvenient to send them as a single string. In this case, the ITEM_NAME variable contains the name of the value, and the ITEM_VALUE variable contains the value of the variable.

For example,

ITEM_NAME[3]="ITEM1";ITEM_VALUE[3]="123".

If this option is enabled, the module will extract the name (ITEM1) and value (123) and monitor their status. If this option is enabled, then ITEM_NAME (ITEM1) should be specified in the variable list, rather than "ITEM_NAME."

For example, this format is used in Advanced OPC Data Logger when you enable these options in the OPC tag group settings:

Fig. 4. Example of OPC Data Logger group settings

Timeout of no data - if enabled, the module will set an error flag; in the case, no data were specified during the set time period (in seconds). For the primary interface time count begins immediately after the program starts, and for the secondary interface - after its activation.

Timeout of constant data - if enabled, the module sets an error flag, if data are coming in without changing. This timeout must be longer than the no data timeout. Interval count starts as soon as any value has been received.

Timeout of bad data - if enabled, the module sets an error flag if data come in, but the variable value is NULL. Interval count starts as soon as any bad value has been received.

Import variables - by clicking this button, you copy all the variables known to the parser to the list, which can then be shortened or supplemented with variables. Before you fill in the list, you must wait for the values from the data source to be retrieved and processed.

Do not pass watchdog items to data export plugins - if enabled, the module will filter tracked variables from the data packet, and they will not enter the data export modules. In combination with the following option, you can provide for different modes of writing tracked values into the database.

Log watchdog items to SQL - if enabled, the module will generate an event with the identifier specified in the "SQL" tab (Fig. 4). All variables of the data packet containing one or more tracked variables will be passed as the context of this event. This event can be handled by both an event handling module and the SQL Database Pro module. The settings of either module must contain a query executed on some event (Fig. 5). Transfer of data to SQL is performed only in the currently running interface.

Switch to secondary interface in case of problems with SQL - if enabled (Fig. 5), the module will monitor errors in the program message log when executing SQL query associated with this event. Switching to a different interface will happen when any problem with SQL occurs.
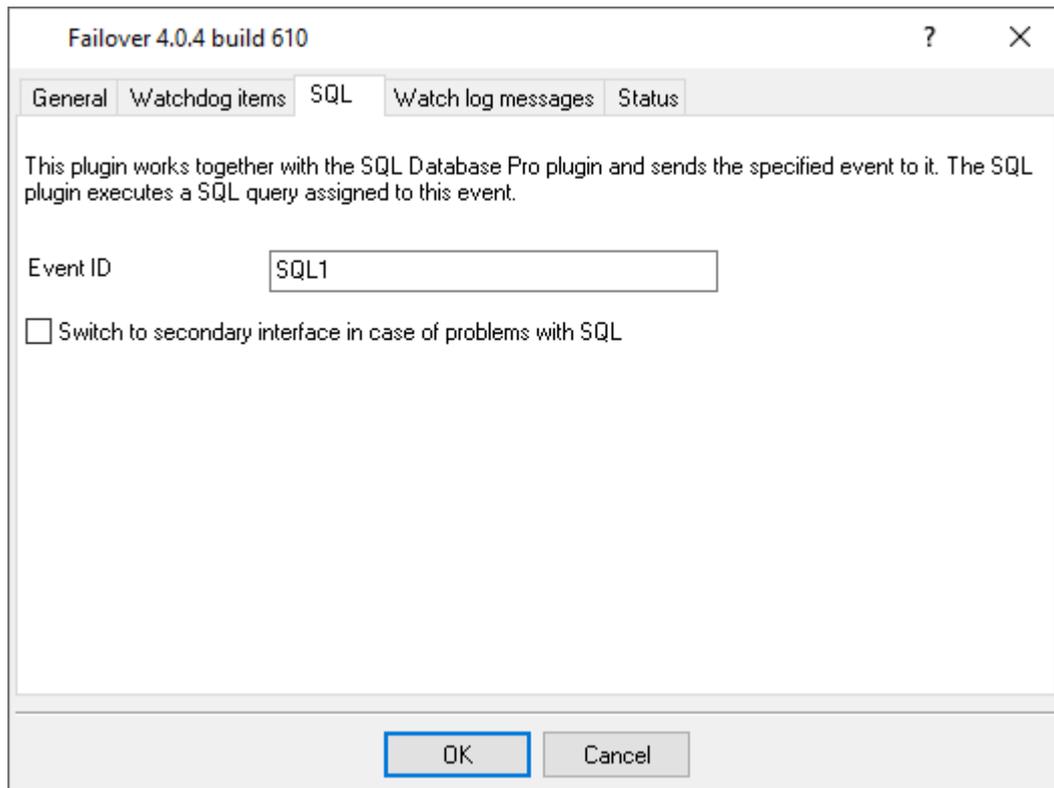
Fig. 5. Event parameters for SQL module
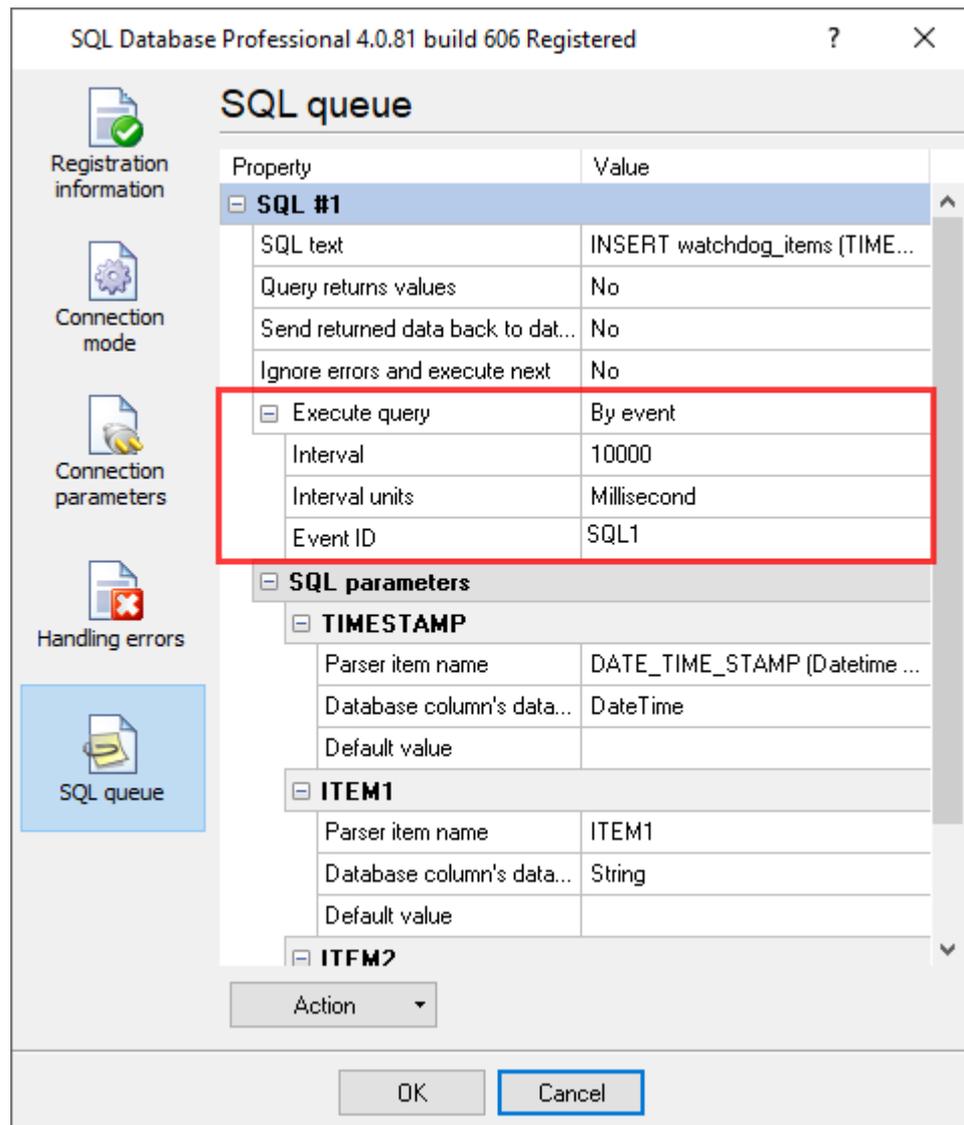
Fig. 6. Example of SQL query

Tracking messages in a log

In the "Watch log messages" tab (Fig. 7), you can specify a list of text strings or regular expressions, the appearance of which in the log indicates problems in the interface.

Simple text strings are case-sensitive. The specified string can be in any position in the message.

A string with a regular expression must have the "RE:" title. Syntax and capabilities correspond to Perl 5.10. By default search is case-sensitive. To change search parameters, you can use the constructs (?is). You can find a more detailed description of regular expressions in Google.
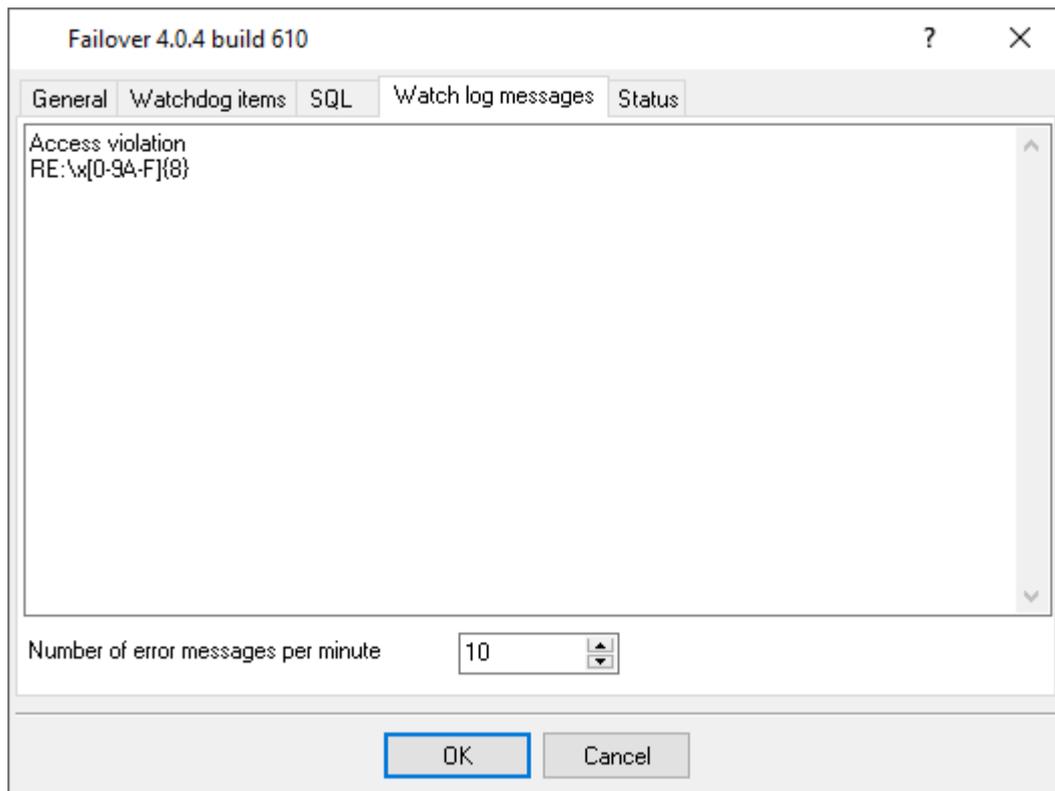
Fig. 7. Message tracking parameters

Status

Using the "Status" tab (Fig. 8), you can monitor the status of the current interface.

Problems - a list of problems detected in this interface.

Active interface - the name of a currently active (running) interface. As for the secondary interface, its IP address is specified.

Self-recovery errors - when running in cold mode, it is assumed that the interface can recover when reconnecting to the data source. The module attempts to do this three times. If it fails to restore operation after three consecutive attempts, the interface is considered faulty and gets completely disconnected before human intervention. To reset the self-recovery error counter, go to the settings of "Failover" module and shut it down by clicking "OK" button. This action also resets all error states (such as no data available). Thus the status of the idle one is "error-free" again, and it can resume operation.
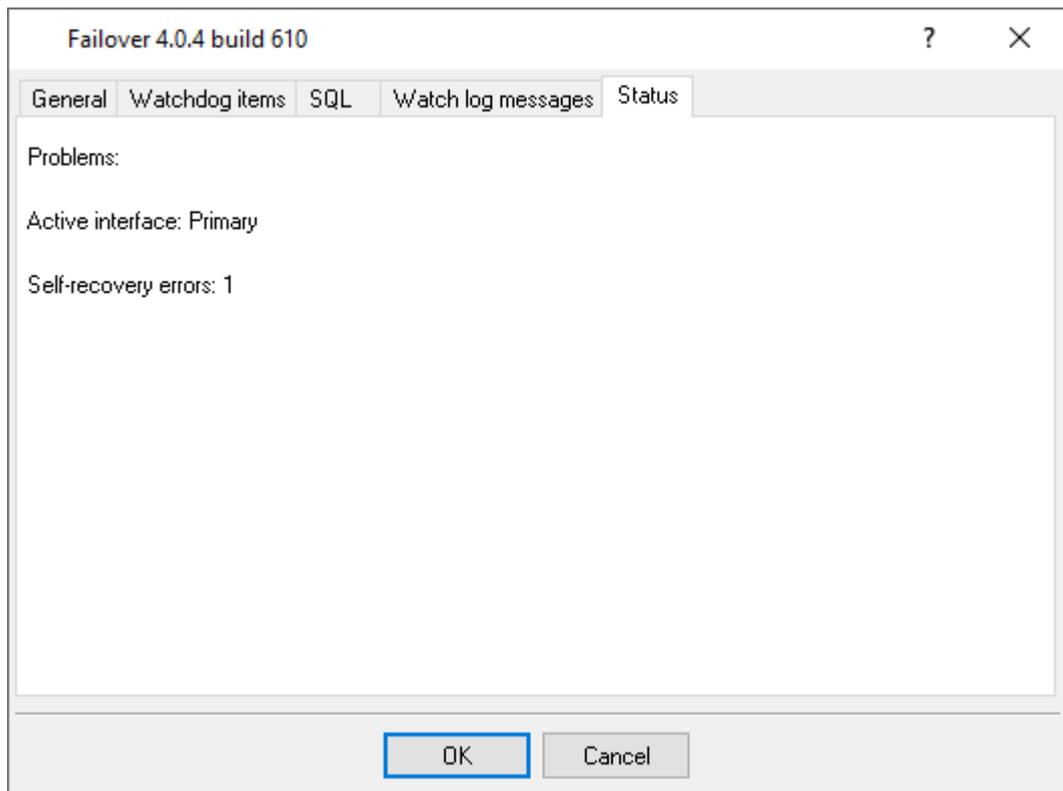
Fig. 8 Status

6   How to run two instances of a program on one computer

To run two instances of the program in the desktop mode, you need to complete the following actions:

1. Create a copy of the main program file in the program folder. For example, by creating a copy of asdlog.exe in asdlog1.exe.
2. Run each copy of the program and select "Main menu - Options - Save settings to INI."
3. Configure each instance of the program.
4. Optionally, you can create a shortcut for each copy on the desktop or in autorun.

Please note that after the program is updated, the copy of the file will not be updated. You will have to do it manually after each update.

To run two instances of the program in service mode, you must complete the following additional actions:

5. Create a copy of the file *srv.exe. For example, by creating a copy of asdlog.exe in asdlog1.exe.
6. Register a copy of the service in the system. To do this, run cmd.exe, go to the folder with the program and run the command "asdlogsrv1.exe /AI" (without quotes) for each copy.

7   Troubles?

7.1   Possible problems

No data for publication/exporting – no data is passed for exporting. Solution: configure the parser, make sure that one or more variables are declared in the parser.

Error on binding variable with name %s [%s] – the error usually occurs if data does not correspond to the specified format. For example, the date and time format does not correspond to the data.

Unable to disconnect from the database [%s] and **Unable to connect to a database [%s]** – it is impossible to connect/disconnect to/form the database. You should check the parameters of the database connection. The analysis of the additional information will help you locate the error.

Database access error [%s]. Stop operations with the database? – the message appears if an error occurs during an attempt to execute an SQL query if the second variant of reacting to errors is selected. The message implies a "Yes" or "No" answer. The analysis of the additional information will help you locate the error.

Unable to verify your SQL script [%s] – the message appears when an attempt to analyze your SQL query fails. Check if the syntax of your SQL query is correct.

Tested successfully – the message appears if your database connection is successfully tested. It requires no additional actions.

Database isn't used – the message appears if the module is temporarily disabled (the "Temporarily disabled" check box is selected) or the database name field is empty. Check the connection parameters.

Database isn't selected - the message appears if the database type is not selected. Check the connection parameters.

Database: %s – %s contains the database name. The message appears if the database connection is successful. Usually, you see it when you call the module for the first time. It requires no additional actions.

Invalid data block length (columns=%d,length=%d) – an internal application error. It means that the data sent by the parser is in an invalid format. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

The time of connection is not due yet (%d,%d) – the message appears during an attempt to connect to the database after the connection to it has been lost and the "Reconnect after" option is enabled. No additional actions are required.

Invalid procedure call. Bad arguments –an attempt to call the module using invalid parameters. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

Writing to the database is complete - the message appears if your queue of SQL queries is successfully executed. It requires no additional actions.

Writing to the database is complete with errors – the message appears if the executing your queue of SQL queries was interrupted by an error. It requires no additional actions.

Your SQL is empty. Please, specify some SQL text first – the message appears if you do not enter the text for your SQL query. Check if the options on the "SQL queue" tab are configured correctly.

Invalid temporary path – the path to the temporary file specified by you does not exist. Enter a new path in the "Temporary folder" field on the "Errors handling" tab.

%s, %d – will be replaced by additional information.