Operation log in control systems

Abstract

This Application note describes recording of user reports into operation log in control systems made by the AMiT company and also displaying operation log on local and remote terminals.

Author: Zbyněk Říha
Document: ap0058_en_01.pdf

Appendix

File contents: ap0058_en_01.zip

pd_p1_en_01.dso	Example of work with local operation log.
pd_p2_en_01.dso	Displaying the operation log from remote station on control system.
pd_p3_en_01.dso	Displaying the operation log from remote station on AMREG terminal.

Contents

Contents ... 2
History of revisions ... 3
Related documentation ... 3
1 Definition of used terms ... 4
2 Introduction ... 5
2.1 System Operation Log ... 5
2.2 Application Operation Log ... 5
3 Operation log reports ... 6
3.1 System messages ... 6
3.2 User reports ... 6
3.2.1 Text attachment to the user code ... 7
3.2.2 ErrSig usage ... 8
3.2.3 Report usage ... 13
4 Operation log display ... 16
4.1 Local operation log ... 16
4.2 Operation log from remote station ... 16
5 Technical support ... 18
6 Warning ... 19

History of revisions

Revision	Date	Author of change	Changes
001	1. 7. 2016	Říha Z.	New document.

Related documentation

1. Help file for Psedet part of DetStudio development environment
   file: psedet_cz.chm
2. Help file for screens of DetStudio development environment
   file: Tridet_cz.chm
3. Help file for EsiDet part of DetStudio development environment
   file: esidet_cz.chm

1 Definition of used terms

Operation log depth
Number of records + 1, which can be stored in the operation log.

Loader
A state, when there is only operating system (NOS) running in the control system. Application is either not loaded, or turned off.

WID
Number identifier of variable in database of variables.

Time written in DB-Net format
Time defined in long variable (or in a cell of long matrix) as a number of seconds since 1. 1. 1980.

2 Introduction

Each control system manufactured by AMiT company contains – after loading the operating system (NOS) – an operation log, that serves for evidence of information about errors and operation states of control system.

Operation log is empty after loading the operating system. It is continually filled with reports. After the log is filled completely, the oldest reports are overwritten by newest ones (works as circular (ring) buffer). This mechanism ensures the storage of predefined number of latest reports.

Control system can recognize two types of operation log:
- System Operation Log
- Application Operation Log

The working method with the operation log is the same for both system operation log and application operation log.

2.1 System Operation Log

System operation log is located in the operating system (NOS) and cannot be deactivated in any way. Depth of the system operation log is hard-set for 50 reports.

System operation log can be – **with activated application** – viewed only on control system terminal (by using the LogView item, bound to "$SYS_LOG" value – see chapter "4.1 Local operation log").

It is accessible by communication interface only when there is loader running in the control system – application is deactivated. Variables are available when the loader is running.
- SYS_ERRNDX
- SYS_ERRMTX

Variables are using WID according to conventions stated in chapter "2.2 Application Operation Log".

2.2 Application Operation Log

Application operation log is an image of system operation log and must be user defined. It is created by defining two variables with WIDs special values in application database.

WID	Database type	Description
xx900 *)	I	**Index** Index is the variable with item number in buffer, which will be overwritten by the next report.
xx901 *)	MI	**Buffer** Buffer is the matrix for individual reports. Each report takes up 8 cells in the matrix. While defining the matrix, it is necessary to always define room for one more report. Typically, for log depth equal to 50 (+1 extra report) the matrix dimensions are to be defined as [1, 408].

*) *xx is a control system address, for example in control system no. 4, WID values are 4900 and 4901.*

Application operation log is completely substituting the system log and besides enabling viewing on connected terminal, it also allows the possibility of processing on PC stations connected to the network. Another benefit is the ability of having deeper log (max. 65520 Byte), although on expense of control system's free memory.

In case of application with single isolated control system, system operation log is sufficient in majority of cases.

3 Operation log reports

Each report is stored in the operation log under specified code. Codes can be divided into:
- System (code is hard-set by the operating system).
- User (it is possible to enter any arbitrary code from range 100 to 30000).

Control system works with these codes. It is possible to add any text to each code for clarity – this text can be displayed in example on control system terminal when appropriate code is written into the operation log.

Report texts, which will be shown in operation log can be modified by “Operation Log” tab, which can be displayed by double-clicking the “Operation Log” folder in project’s window.

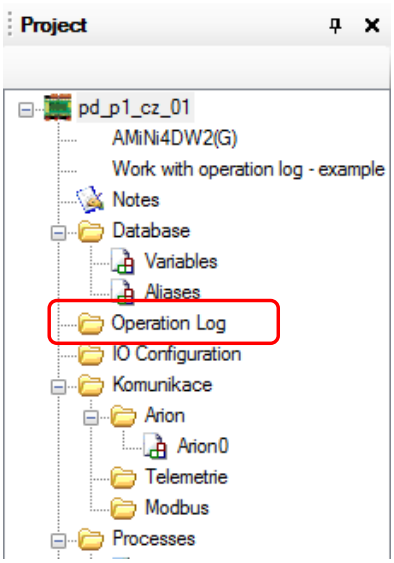


Fig. 1 – “Operation Log” folder in project’s window

Caution
“Operation Log” tab serves only for attaching text to appropriate codes in operation log. Respective code is being written into operation log by the operating system (NOS), eventually by user application of certain SW modules in processes.

3.1 System messages

These are reports independent on application. They are reported by the operating system (NOS), or by function modules libraries and **cannot be removed or filtered in any way.** System reports texts are automatically predefined during creation of the project in DetStudio. List of system errors and reports can be found in Help for Detstudio environment.

If there is application operation log defined in the control system, system reports are written into both application operation log and system operation log.

3.2 User reports

Writing user reports into operation log is handled by these modules:
- `ErrSig` – Module for service and error detection – errors are signalized by horn.
- `ErrSigLA` – Module for service and error detection – errors are signalized by horn.
- `ErrSig32` – Module for service and error detection – errors are signalized by alarm (allows to process upto 32 signals).

- `Report` – Recording of alarm / report / information into the operation log (in integer format).
- `ReportF` – Recording of alarm / report / information into the operation log (in float format).

If there is application operation log defined in the control system, user reports are written only into application operation log.

It is possible to use codes in range from **100** to **30000** for writing user reports.

3.2.1 Text attachment to the user code

Report texts can be added / modified in DetStudio in the "Operation Log" tab, which can be displayed by double-clicking the "Operation Log" folder in project's window.

Operation Log

Add | Tables | Macros for formats

Error	DW table	F table	Prefix	Text	Comment
0			OK	No error	OK
16			nos	Operational system download	Zavedení operačního systému
17			nos	Network loader cold start	Studeny start síťového zavaděče
18			nos	Network loader warm start (Drop-out time: %T)	Teplý start síťového zavaděče
19			nos	Application cold start	Studený start aplikace
20			nos	Application warm start (Drop-out: %T)	Teplý start aplikace
21			nos	Processor error: %0sw	Chyba procesoru
22			nos	Write protect memory access	Zápis do zakázané paměti
23			nos	Interrupt controller internal error, adr.: %d	Interní chyba řadiče přerušení
24	EthState		eth	Interface nr. %0w (card %gh type %sl	Stav rozhraní Ethernet
25			nos	Using a non-existent COM-port %0i	Použití neexistujícího COM-portu
32			Db	WID %0i does not exist.	Neplatný WID proměnné
33			Db	Invalid variable descriptor %0w	Neplatný vnitřní identifikátor proměnné
34	Typy		Db	Invalid variable type %1sw of %0i	Neplatný typ proměnné
35			Db	Invalid row %1w in variable %0w	Neplatný řádek proměnné
36			Db	Invalid column %1w of variable %0w	Neplatný sloupec proměnné
37			Db	Wrong Row Count %1w of %0w variable.	Neplatný počet řádků proměnné
38			Db	Invalid count of columns %1w of %0w variable	Neplatný počet sloupců proměnné
39			nos	Insufficient Memory	Nedostatek paměti
40			aeg	Wrong 1/4h puls, duration: %t	Špatný 1/4h pulz při měření odběru el.energie
50			esi	Application cold start	Studený start aplikace
51			esi	Application warm start (Drop-out: %T)	Teplý start aplikace
52			esi	Error RTOS initializing %w	Chyba inicializace RTOS
53			esi	A wrong user entry in a log %w	Chybný zápis do provozního deníku
54			esi	Invalid matrix index (%w, %l, %w)	Přístup na neplatný index matice

Fig. 2 – Operation log tab

New code (report) can be added to the list by pressing the ***"Insert"*** button.

Operation Log

Add | Tables | Macros for formats

Error	DW table	F table	Prefix	Text	Comment
33			Db	Invalid variable descriptor %0w	Neplatný vnitřní identifikátor proměnné
34	Typy		Db	Invalid variable type %1sw of %0i	Neplatný typ proměnné
35			Db	Invalid row %1w in variable %0w	Neplatný řádek proměnné
36			Db	Invalid column %1w of variable %0w	Neplatný sloupec proměnné
37			Db	Wrong Row Count %1w of %0w variable.	Neplatný počet řádků proměnné
38			Db	Invalid count of columns %1w of %0w variable	Neplatný počet sloupců proměnné
39			nos	Insufficient Memory	Nedostatek paměti
40			aeg	Wrong 1/4h puls, duration: %t	Špatný 1/4h pulz při měření odběru el.energie
50			esi	Application cold start	Studený start aplikace
51			esi	Application warm start (Drop-out: %T)	Teplý start aplikace
52			esi	Error RTOS initializing %w	Chyba inicializace RTOS
53			esi	A wrong user entry in a log %w	Chybný zápis do provozního deníku
54			esi	Invalid matrix index (%w, %l, %w)	Přístup na neplatný index matice
1000				en-US text	en-US comment
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku
32774			val	Valve %0w - %1w did not reach the OPEN end switch	Ventil nedojel v časovém limitu na koncový spí...
32775			val	Valve %0w - %1w did not reach the CLOSE end switch	Ventil nedojel v časovém limitu na koncový spí...
32776			val	Valve %0w - %1w reached the OPEN end switch	Ventil dojel na koncový spínač OTEVŘENO
32777			val	Valve %0w - %1w reached the CLOSE end switch	Ventil dojel na koncový spínač ZAVŘENO
32800	Bufer		lcd	Read request %sl rejected	Nelze vložit čtecí požadavek
32801	Bufer		lcd	Write request %sl rejected	Nelze vložit zápisový požadavek
32802			lcd	User Login %0w	Přihlášení uživatele
32803			lcd	User Logout %0w	Odhlášení uživatele

Fig. 3 – New code in the operation log

Each item of newly inserted report can be edited by the *"**F2**"* button.

Code shown in the "Error" column must match the code used in module, which writes reports into the operation log. Text, which is suppose to show on the control systems' display must be entered to the "Text" column.

3.2.2 ErrSig usage

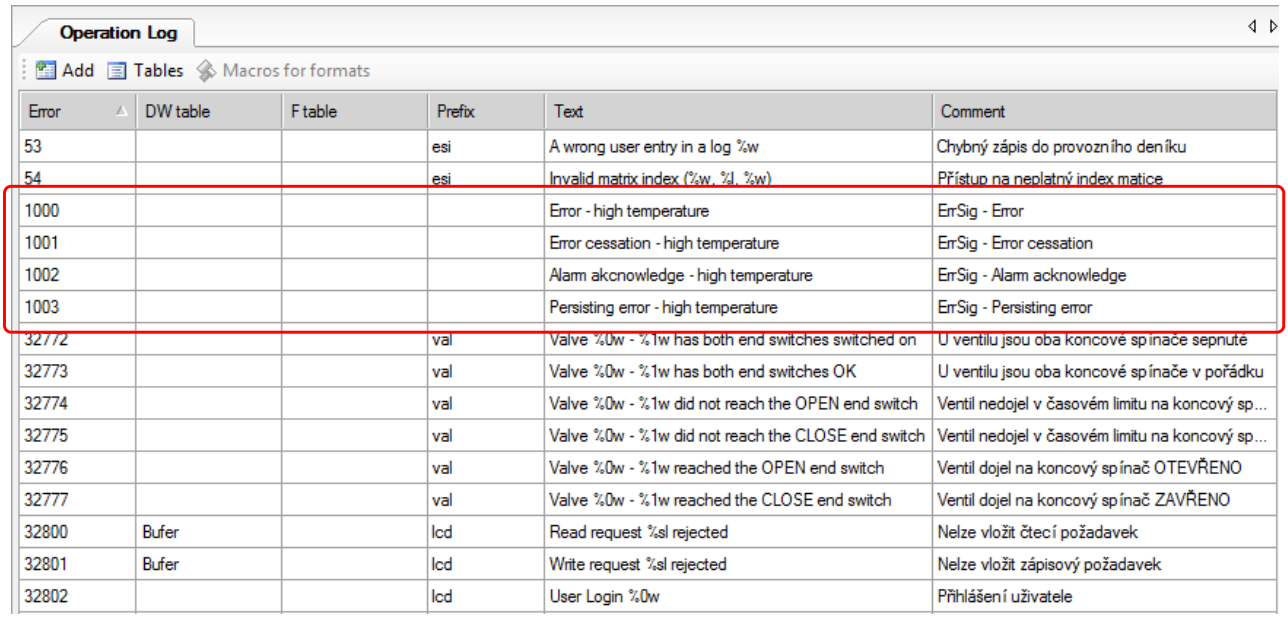
`ErrSig` module is recommended to use for alarm service, which can happen during usage. Detailed description of its available parameters can be found in Help for DetStudio environment. From the usage of operation log point of view, following parameters are important: `Code`, `Data1`, `Data2`, eventually `Data3`. In case of `ErrSig` module usage, it looks like this:

```
                                                                  Data3┐
ErrSig Error, 0x0001, Acknowledge, 0x0001, Alarm.0, Ignore.0, 10, 20, 1000, 0, 0, 0
                                                                  │    │  └Data2
                                                                  │    └Data1
                                                                  └Code
```

4 reports will be sent to the operation log with following codes:

- 1000 – Information about error detection,
- 1001 – Information about error cessation,
- 1002 – Information about error acknowledgement,
- 1003 – Information about persisting error even after acknowledgement.

Therefore, it is required to put four codes into the tab with texts for operation log (see Help for `ErrSig` module in the DetStudio environment.



Operation Log

Add | Tables | Macros for formats

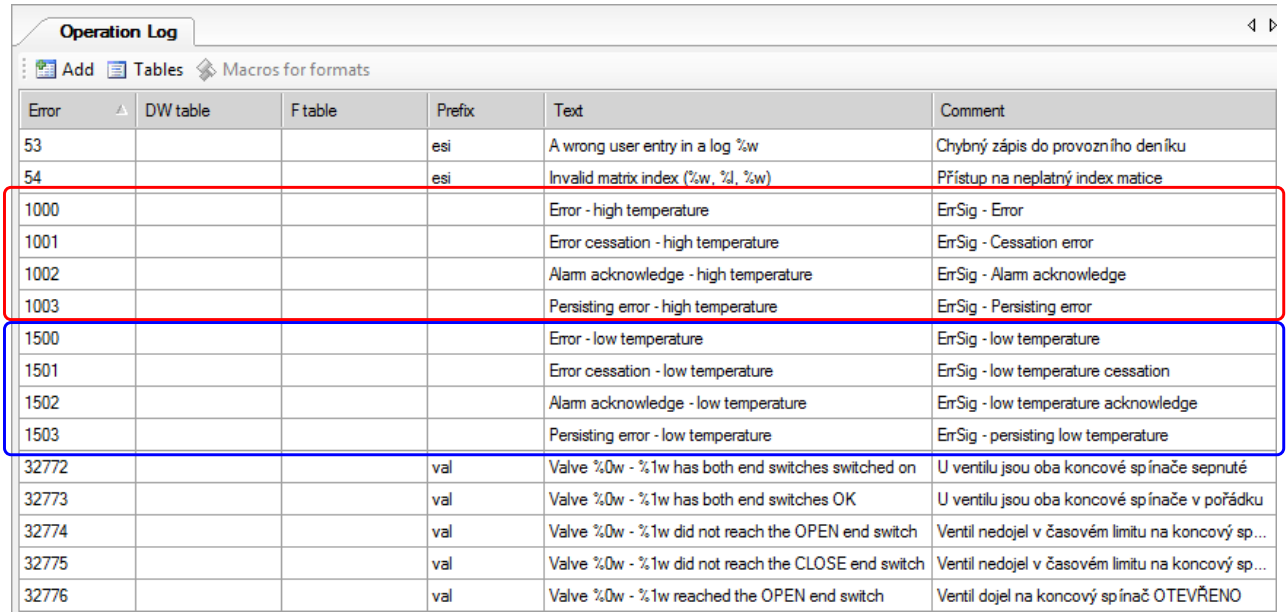
Error	DW table	F table	Prefix	Text	Comment
53			esi	A wrong user entry in a log %w	Chybný zápis do provozního deníku
54			esi	Invalid matrix index (%w, %l, %w)	Přístup na neplatný index matice
1000				Error - high temperature	ErrSig - Error
1001				Error cessation - high temperature	ErrSig - Error cessation
1002				Alarm akcnowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku
32774			val	Valve %0w - %1w did not reach the OPEN end switch	Ventil nedojel v časovém limitu na koncový sp...
32775			val	Valve %0w - %1w did not reach the CLOSE end switch	Ventil nedojel v časovém limitu na koncový sp...
32776			val	Valve %0w - %1w reached the OPEN end switch	Ventil dojel na koncový spínač OTEVŘENO
32777			val	Valve %0w - %1w reached the CLOSE end switch	Ventil dojel na koncový spínač ZAVŘENO
32800	Bufer		lcd	Read request %sl rejected	Nelze vložit čtecí požadavek
32801	Bufer		lcd	Write request %sl rejected	Nelze vložit zápisový požadavek
32802			lcd	User Login %0w	Přihlášení uživatele

Fig. 4 – Four reports for one `ErrSig` module

If there is a second `ErrSig` module used which is suppose to generate different reports, it is possible to use additional four codes. `ErrSig` module should therefore have set different value in the `Code` parameter than 1000 to 1003. For example:

```
ErrSig Error, 0x0002, Acknowledge, 0x0002, Alarm.1, Ignore.1, 10, 20, 1500, 0, 0, 0
```

Reports in tab with texts for operation log for both `Errsig` modules should look according to following table:



Operation Log

Add | Tables | Macros for formats

Error	DW table	F table	Prefix	Text	Comment
53			esi	A wrong user entry in a log %w	Chybný zápis do provozního deníku
54			esi	Invalid matrix index (%w, %l, %w)	Přístup na neplatný index matice
1000				Error - high temperature	ErrSig - Error
1001				Error cessation - high temperature	ErrSig - Cessation error
1002				Alarm acknowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
1500				Error - low temperature	ErrSig - low temperature
1501				Error cessation - low temperature	ErrSig - low temperature cessation
1502				Alarm acknowledge - low temperature	ErrSig - low temperature acknowledge
1503				Persisting error - low temperature	ErrSig - persisting low temperature
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku
32774			val	Valve %0w - %1w did not reach the OPEN end switch	Ventil nedojel v časovém limitu na koncový sp...
32775			val	Valve %0w - %1w did not reach the CLOSE end switch	Ventil nedojel v časovém limitu na koncový sp...
32776			val	Valve %0w - %1w reached the OPEN end switch	Ventil dojel na koncový spínač OTEVŘENO

Fig. 5 – Reports for two `Errsig` modules

Reports from first `ErrSig` module are framed in red colour, reports from second `ErrSig` module are framed in blue colour.

From abovementioned should be apparent that codes in operation log are not necessary defined as a line without spaces.

In case of multiple `ErrSig` modules usage, the amount of reports defined in the "Operation Log" tab can be lowered considerably by using the `DataX` parameters from `ErrSig` module.

Reports defined in abovementioned steps can be defined in the control system code by using same `Code` parameter. Separate modules can be differentiated by `Data1` or `Data2` parameters followingly:

```
ErrSig Error, 0x0001, Acknowledge, 0x0001, Alarm.0, Ignore.0, 10, 20, 3000, 0, 0, 0
ErrSig Error, 0x0002, Acknowledge, 0x0002, Alarm.1, Ignore.1, 10, 20, 3000, 1, 0, 0
```

Abovementioned code uses `Data1` parameter for recognition between each report. In such case it is sufficient to define only 4 reports in the "Operation Log" tab. Although they must be complemented by information processing from `Data1` parameter. `Data1` (and `Data2`) parameter can be in "Operation Log" tab processed by so called DW table, which allows (on basis of `Data1` and optionally `Data2` parameter value) to display the data from operation log dynamically. Table can be accessed from toolbar in tab by the ***"Tables"*** button.

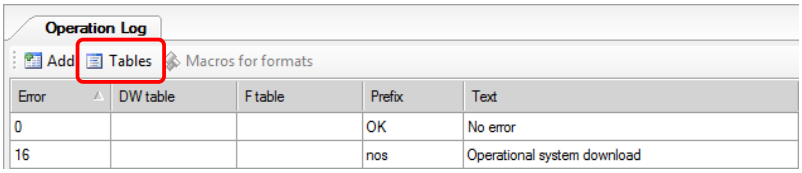


Fig. 6 – Button for tables definition

After pressing the button, table editor for operation log will open – it is possible to define new table here by the ***"Add DW"*** button (with name "Errors").

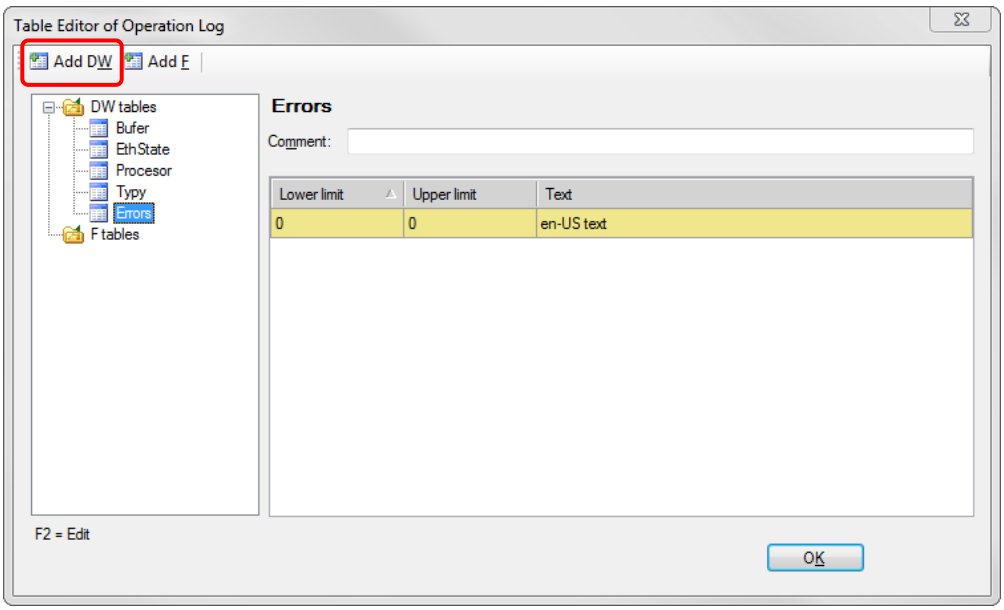


Fig. 7 – Table editor with "Errors" table

It is possible to define one value or value range in each table, to which specific text in operation log will be assigned. There is a request currently, that each value added to the `Data1` parameter of the `ErrSig` module must have one text report. In such case, it is necessary to input into top margin and bottom margin same number with text, which should be displayed in the operation log. Resulting look of the table:

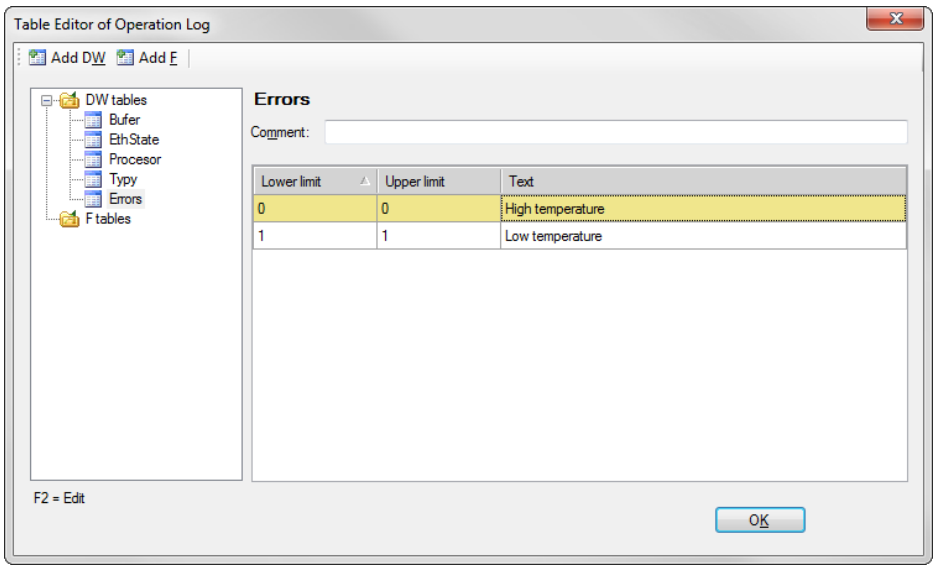


Fig. 8 – Defined table with errors

Such defined table must be assigned in the "Operation Log" tab by the "DW table" column to corresponding user reports. It is also necessary to input so called Macro (formatting parameter) into text. This can be inputted into own report when there is report text editing for operation log in progress – by the *"**Macros for formats**"* button (if text editing in the "Text" cell is finished, the button is unavailable).

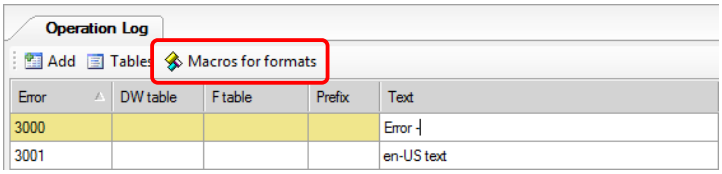


Fig. 9 – Available "Macros for formats" button

After clicking the button, the table with each macro will be displayed (formatting parameters), where you can select the way of processing the value from `DataX` module of `ErrSig`. If DW table is used, it is necessary to select the macro, which will substitute the value with text from the table (see following figure).

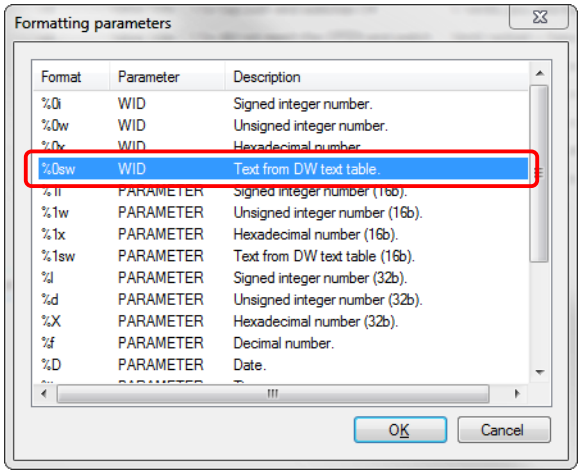
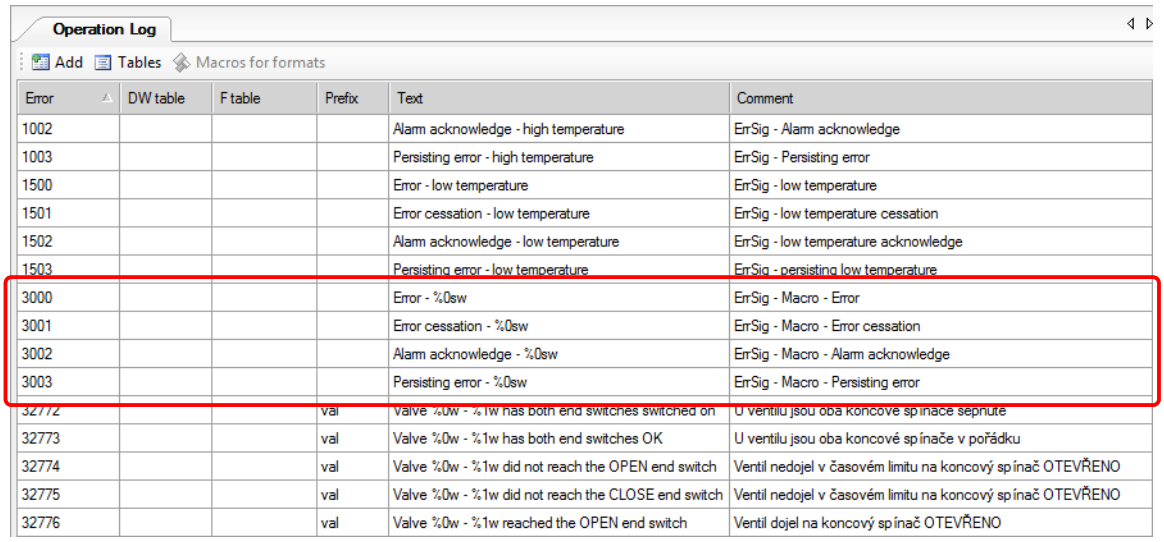


Fig. 10 – Selection of requested macro

By clicking the *"**OK**"* button, requested macro will be inserted into text of own report. Chosen macro must then be placed (e.g. by copying) to each report that is supposed to be working with the "Errors" table. Rows of appropriate reports will look as follows:



Error	DW table	F table	Prefix	Text	Comment
1002				Alarm acknowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
1500				Error - low temperature	ErrSig - low temperature
1501				Error cessation - low temperature	ErrSig - low temperature cessation
1502				Alarm acknowledge - low temperature	ErrSig - low temperature acknowledge
1503				Persisting error - low temperature	ErrSig - persisting low temperature
3000				Error - %0sw	ErrSig - Macro - Error
3001				Error cessation - %0sw	ErrSig - Macro - Error cessation
3002				Alarm acknowledge - %0sw	ErrSig - Macro - Alarm acknowledge
3003				Persisting error - %0sw	ErrSig - Macro - Persisting error
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku
32774			val	Valve %0w - %1w did not reach the OPEN end switch	Ventil nedojel v časovém limitu na koncový spínač OTEVŘENO
32775			val	Valve %0w - %1w did not reach the CLOSE end switch	Ventil nedojel v časovém limitu na koncový spínač OTEVŘENO
32776			val	Valve %0w - %1w reached the OPEN end switch	Ventil dojel na koncový spínač OTEVŘENO

Fig. 11 – Reports using DW table

This method will always display static texts from rows with codes 3000 to 3003. Texts in DW table "Errors" will be then added dynamically. Texts are dynamically changed according to value sent in `Data1` parameter from `ErrSig` module.

Benefit of this solution is lesser demand for the text amount added into table with reports for operation log. For each new error that must be programmed, one `ErrSig` module with same value appointed to `Code` parameter will suffice; with unused value appointed for `Data1` parameter.

```
ErrSig Error, 0x0004, Acknowledge, 0x0004, Alarm.2, Ignore.2, 10, 20, 3000, 2, 0, 0
```

And add one text into defined DW table with name "Errors".

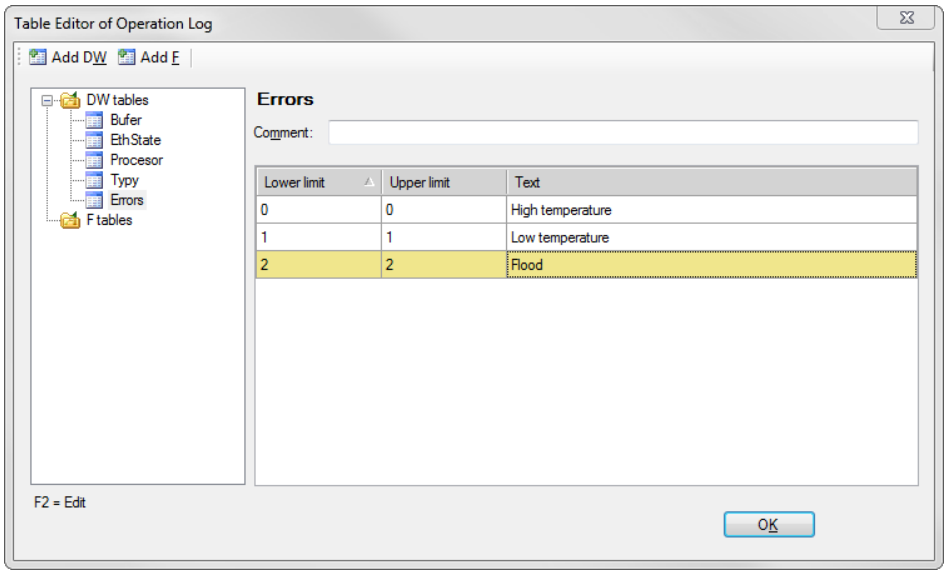


Fig. 12 – DW table with added reports

ErrSig module is not exclusive only for technology alarms usage. It is also suitable for giving enough information for application author or service technician. Typical example of usage might be heat exchanger station, where ErrSig module can be used for sensor of input temperature to the heat exchanger station. If the temperature of supplied medium is insufficient, ErrSig module will write a report into operation log. Application author or service technician will be immediately after new report notified, that the issue is not on side of the heat exchanger station itself, but on the side of the heat supplier.

3.2.3 Report usage

Report (ReportF) module can be used to record any report. E.g. information about technology setting changes by user, or about switching between automatic / manual mode etc. From the usage of operation log point of view, following parameters are important: Code, Data1, Data2, eventually Data3.

If you need to report switch to the manual mode to the operation log, it is necessary to define alias "@Manual", which sets the value to True everytime, when user switches into manual mode. Report module can be used for recording the information about manual mode as follows:

```
Report @Manual, 2000, 0, 0, 0
                  |     |  |  └ Data3
                  |     |  └ Data2
                  |     └ Data1
                  └ Code
```

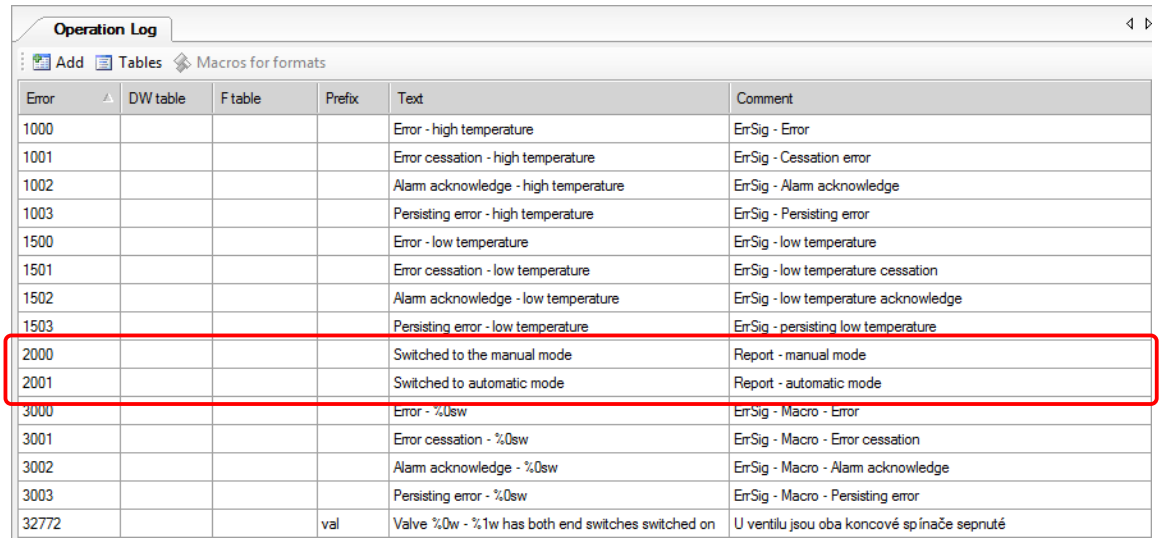
Because the Report module reacts only on rising edge of selected alias / bit, only information about switching to manual mode will be recorded to the operation log. In case of requirement to record information about return to automatic mode, it is necessary to create different alias, that will signalize such return.

```
Let @Automat = not(@Manual)
```

The record itself must be tied with another Report module.

```
Report @Automat, 2001, 0, 0, 0
```

Into tab with texts for operation log then fill corresponding codes.



Error	DW table	F table	Prefix	Text	Comment
1000				Error - high temperature	ErrSig - Error
1001				Error cessation - high temperature	ErrSig - Cessation error
1002				Alarm acknowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
1500				Error - low temperature	ErrSig - low temperature
1501				Error cessation - low temperature	ErrSig - low temperature cessation
1502				Alarm acknowledge - low temperature	ErrSig - low temperature acknowledge
1503				Persisting error - low temperature	ErrSig - persisting low temperature
2000				Switched to the manual mode	Report - manual mode
2001				Switched to automatic mode	Report - automatic mode
3000				Error - %0sw	ErrSig - Macro - Error
3001				Error cessation - %0sw	ErrSig - Macro - Error cessation
3002				Alarm acknowledge - %0sw	ErrSig - Macro - Alarm acknowledge
3003				Persisting error - %0sw	ErrSig - Macro - Persisting error
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté

Fig. 13 – Reports for Report module

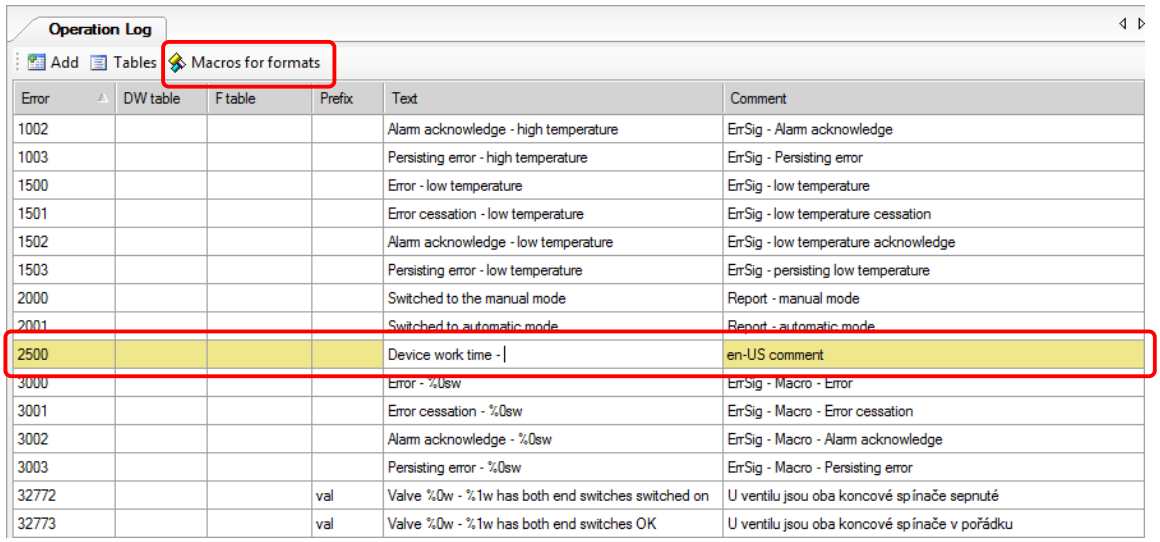
Abovementioned ensures that code 2000 will be recorded into operation log each time, when "@Manual" alias is set to True value and code 2001 will be recorded each time, when "@Automat" alias is set to True value.

`Report` module is also able to write any value into operation log. Depending on the type and value size, it is possible to use its `DataX` parameters. If the value must be written in float type, it is necessary to use `ReportF` module.

If there is a request to write time value (in DB-Net format) saved in long type variable named "TimeDBNet", `Report` module will look like this:

```
Report @Operation, 2500, 0, TimeDBNet, 0
```

It is necessary to input code defined in `Report` module and part of the report into tab with texts for operation log.



Error	DW table	F table	Prefix	Text	Comment
1002				Alarm acknowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
1500				Error - low temperature	ErrSig - low temperature
1501				Error cessation - low temperature	ErrSig - low temperature cessation
1502				Alarm acknowledge - low temperature	ErrSig - low temperature acknowledge
1503				Persisting error - low temperature	ErrSig - persisting low temperature
2000				Switched to the manual mode	Report - manual mode
2001				Switched to automatic mode	Report - automatic mode
2500				Device work time - |	en-US comment
3000				Error - %0sw	ErrSig - Macro - Error
3001				Error cessation - %0sw	ErrSig - Macro - Error cessation
3002				Alarm acknowledge - %0sw	ErrSig - Macro - Alarm acknowledge
3003				Persisting error - %0sw	ErrSig - Macro - Persisting error
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku

Fig. 14 – Report editing in progress in the `Report` module

If editing report for operation log is in progress, button ***"Macros fo formats"*** is available in toolbar in tab "Operation Log". After clicking the button, the table with each macro will be displayed (formatting parameters), where you can select the way of displaying the variable, used in `Report (ReportF)` module with `DataX` parameters.

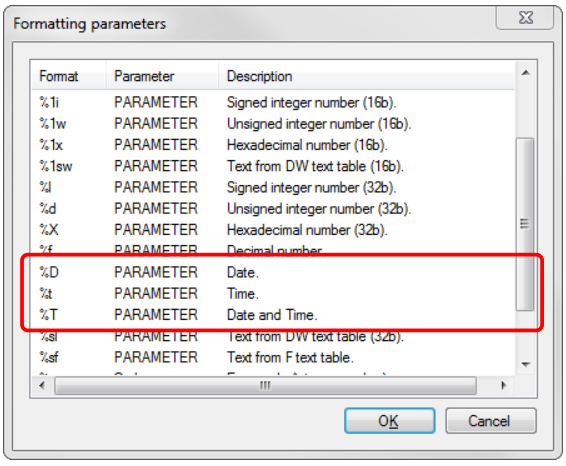
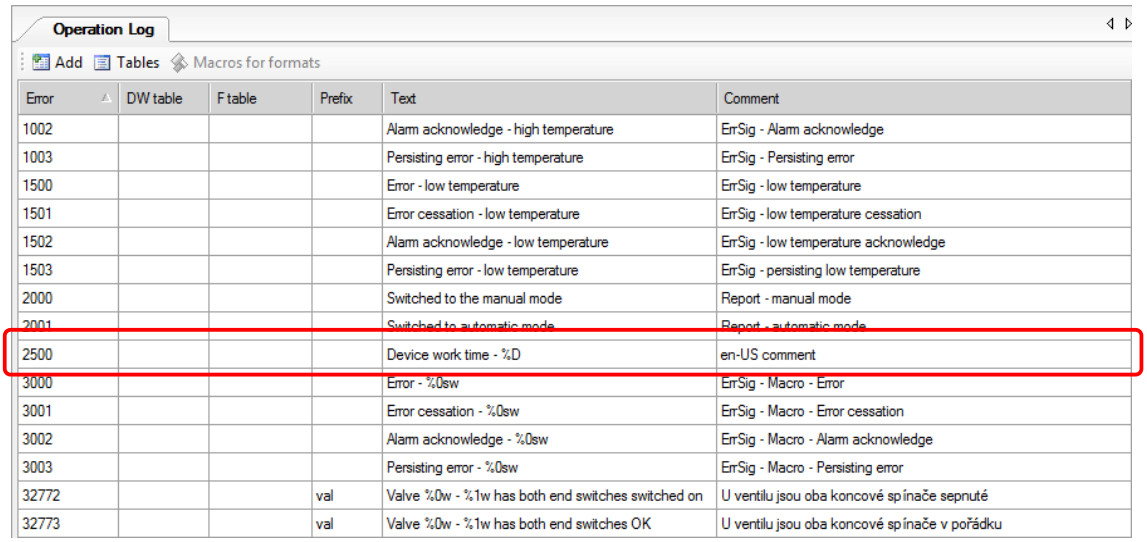


Fig. 15 – Table with macro list (formatting parameters)

When requesting to display long variable in date and time format, it is possible to select one of the three parameters marked in previous figure.

Resulting report for operation log definition will serve for cases, when there is a need to display only date and it will look as follows:



Operation Log

Add | Tables | Macros for formats

Error	DW table	F table	Prefix	Text	Comment
1002				Alarm acknowledge - high temperature	ErrSig - Alarm acknowledge
1003				Persisting error - high temperature	ErrSig - Persisting error
1500				Error - low temperature	ErrSig - low temperature
1501				Error cessation - low temperature	ErrSig - low temperature cessation
1502				Alarm acknowledge - low temperature	ErrSig - low temperature acknowledge
1503				Persisting error - low temperature	ErrSig - persisting low temperature
2000				Switched to the manual mode	Report - manual mode
2001				Switched to automatic mode	Report - automatic mode
2500				Device work time - %D	en-US comment
3000				Error - %0sw	ErrSig - Macro - Error
3001				Error cessation - %0sw	ErrSig - Macro - Error cessation
3002				Alarm acknowledge - %0sw	ErrSig - Macro - Alarm acknowledge
3003				Persisting error - %0sw	ErrSig - Macro - Persisting error
32772			val	Valve %0w - %1w has both end switches switched on	U ventilu jsou oba koncové spínače sepnuté
32773			val	Valve %0w - %1w has both end switches OK	U ventilu jsou oba koncové spínače v pořádku

Fig. 16 – Report from `Report` module with macro (formatting parameter)

4 Operation log display

4.1 Local operation log

`LogView` serves for operation log display – can be found in toolbox in “General” section. Its parameterization is independent on system / application operation log usage. It is necessary to tie it with the “$SYS_LOG” value (“Variable” parameter link in properties window) in both cases.

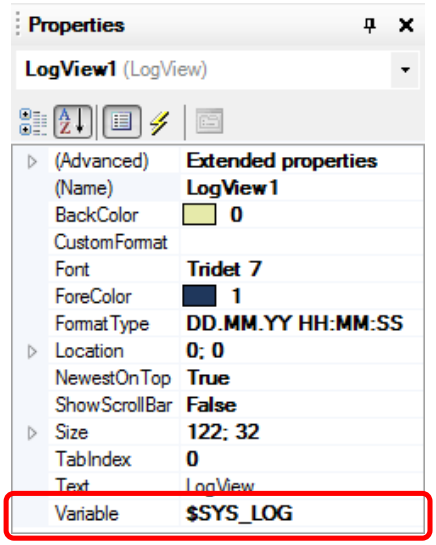


Fig. 17 – `LogView` component link to the “$SYS_LOG” value

If there is an application operation log defined in the control system, its reports will be displayed. If not, reports from system operation log will be displayed.

4.2 Operation log from remote station

`LogView` component is used for displaying the operation log from remote station (same as for local operation log display).

If there is a request to display operation log from remote station (from different control system), it is necessary to:

- Define operation log on the side, where the operation log is suppose to display.
- Create link by the “Project/Remote stations” menu (see Help for DetStudio environment) on the control system (or AMREG terminal) side, where the operation log is suppose to display.
- Define same reports in local operation log on the side of control system (or AMREG terminal) side, where the operation log is suppose to display.

Note

For definition of same reports, export of all reports from operation log from remote station in csv format and subsequent import into control system project (or AMREG terminal) can be used. Hence it is not necessary to rewrite reports manually.

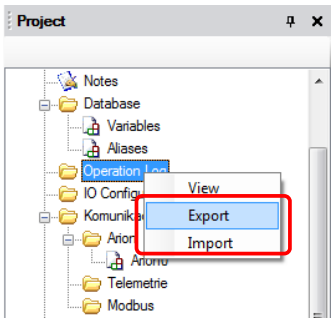


Fig. 18 – Contextual menu for import / export of reports from operation log into *.csv

Caution
If the operation log will be displayed on the AMREG terminal, it is necessary to use `DbNet` *object in the project, to which the "$SYS_LOG" value from remote station must be entered.*

If the operation log from remote station is displayed on control system's terminal, it is necessary to use `LogView` component for displaying. If it is displayed on the AMREG terminal, it is necessary to use `LogViewDb` component for displaying. **It is necessary to link the component to the "$SYS_LOG" value from remote station, not to the local "$SYS_LOG" value.**

5 Technical support

All information about working with the operation log is available in technical support department of AMiT company. Technical support can be reached by e-mail on address: **support@amit.cz**.

6 Warning

AMiT, spol. s r. o. does not provide any warranty concerning the contents of this publication and reserves the right to change the documentation without obligation to inform anyone or any authority about it.

This document can be copied and redistributed under following conditions:

1. The whole text (all pages) must be copied without any changes.
2. All redistributed copies must retain the AMiT, spol. s r. o. copyright notice and any other notices contained in the documentation.
3. This document must not be distributed for purpose of making profit.

The names of products and companies used herein can be trademarks or registered trademarks of their respective owners.