

News in OpcDbGateway 5.0x

Application integration software



SAE – Automation, s.r.o. (Ltd.), Nová Dubnica, Sk

Interoperability for your devices and software applications

Objectives

- **Configuration application**

- Expand options and simplify configuration procedures

- **Runtime application**

- Optimization of data exchange and processing algorithms
- Better adherence to the synchronicity processing period
- Expanding Asynchronous Processing Options
- Better collaboration of a powerful application kernel with expansion DLLs

Overview of new configurator properties 1/3

- Configuring multiple OPC groups for one external OPC server in the OPC DA client and more transparent mapping of OPC variables using the configuration assistant.
- OPC variable write option in OPC Client Configurator
- New, a more transparent dialogue for configuring events
- A new, more readable dialogue for configuring triggers and supporting new trigger options
- New, more transparent dialogue for configuring commands

Overview of new configurator properties 2/3

- Assistant for mapping an existing table in a database into a configuration
 - into tables, database operands, and OPC variables of the internal OPC server
 - Assign an indexing MO to all DOs assigned to a table - scroll in the rows of the table
- Create table in database by configuration
- View table contents in database
- Universal procedure for choosing a data provider and creating a connection string

Overview of new configurator properties 3/3

- Easier creation of historical trends, removal of minor mistakes
- Automatically upgrading the configuration database to ver. 5.0
- Interface of "memory operand" (MO) with OPC variable of the internal OPC server via Id - simplifies changes in MO configuration
- Arrange command(s) in order of execution in the tree view

Overview of changes in the runtime application

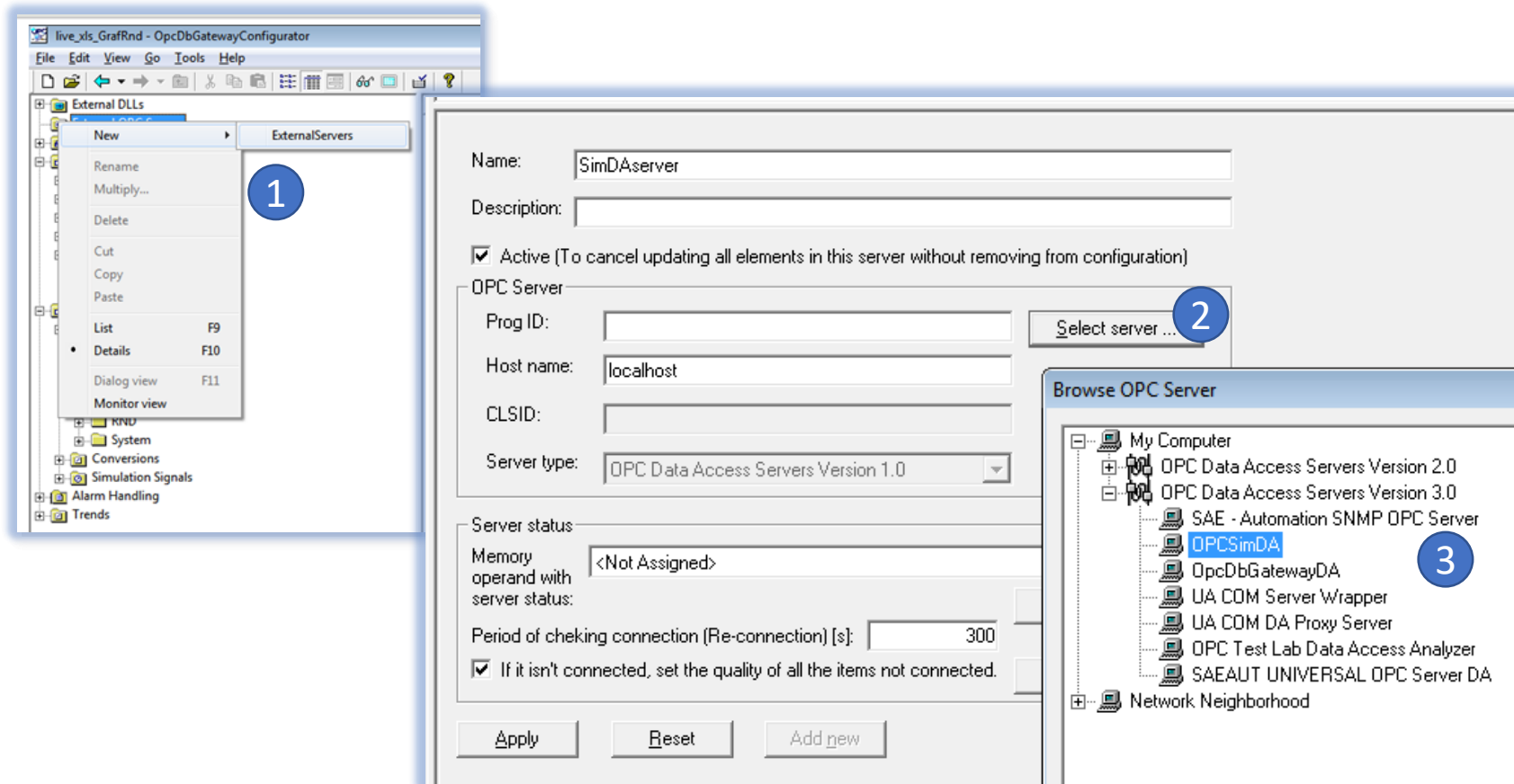
- Using multiple OPC groups on an external OPC server
- Synchronous Cycle Period - Very Accurate Compliance
- Asynchronous Logging - does not adversely affect the Synchronic Cycle Period
- More efficient work with databases - use SQL Server functionality for aggregation functions, accelerate work with large tables, Open tables for synchronous thread in advance and for asynchronous up to event time
- It works as a database client for many different types of databases (to which drivers are installed)

Configuring OPC groups in the OPC DA client

Configuring OPC groups in the OPC DA client 1/6

- **Original Status** - The only OPC group to access one external OPC Server
- **Importance** - structuring access to OPC variables on external OPC servers eg:
 - Read / Write
 - Synchronous / asynchronous
 - Update Rate
- **Testing the connection to the OPC server (Keep alive)**
- **Improvement of user interface and configuration of ext. OPC servers**

Configuring OPC groups in the OPC DA client 2/6



Configuring OPC groups in the OPC DA client 3/6

1. Option to enable/disable ext. OPC server in configuration
2. Period of reconnection
3. MO for OPC server status (connected/not connected)

The screenshot shows the configuration window for an OPC server named 'SimDAserver'. The 'Active' checkbox is checked, indicated by a blue circle with the number '1'. The 'OPC Server' section contains fields for 'Prog ID' (SAEAutomation.OPCSimDA.3), 'Host name' (localhost), 'CLSID' ({33C99B4A-FABF-46A4-8A04-5B4DA37DF2EC}), and 'Server type' (OPC Data Access Servers Version 3.0). The 'Server status' section shows a dropdown menu with 'SimDAserver/IsServerActive' selected, indicated by a blue circle with the number '3'. Below this, the 'Period of checking connection (Re-connection) [s]' is set to 300, indicated by a blue circle with the number '2'. There is also a checkbox for 'If it isn't connected, set the quality of all the items not connected.' and buttons for 'Apply', 'Reset', 'Add new', 'Generate new', and 'Server Status ...'.

Name: SimDAserver

Description:

☒ Active (To cancel updating all elements in this server without removing from configuration)

OPC Server

Prog ID: SAEAutomation.OPCSimDA.3 Select server ...

Host name: localhost

CLSID: {33C99B4A-FABF-46A4-8A04-5B4DA37DF2EC}

Server type: OPC Data Access Servers Version 3.0

Server status

Memory operand with server status: SimDAserver/IsServerActive

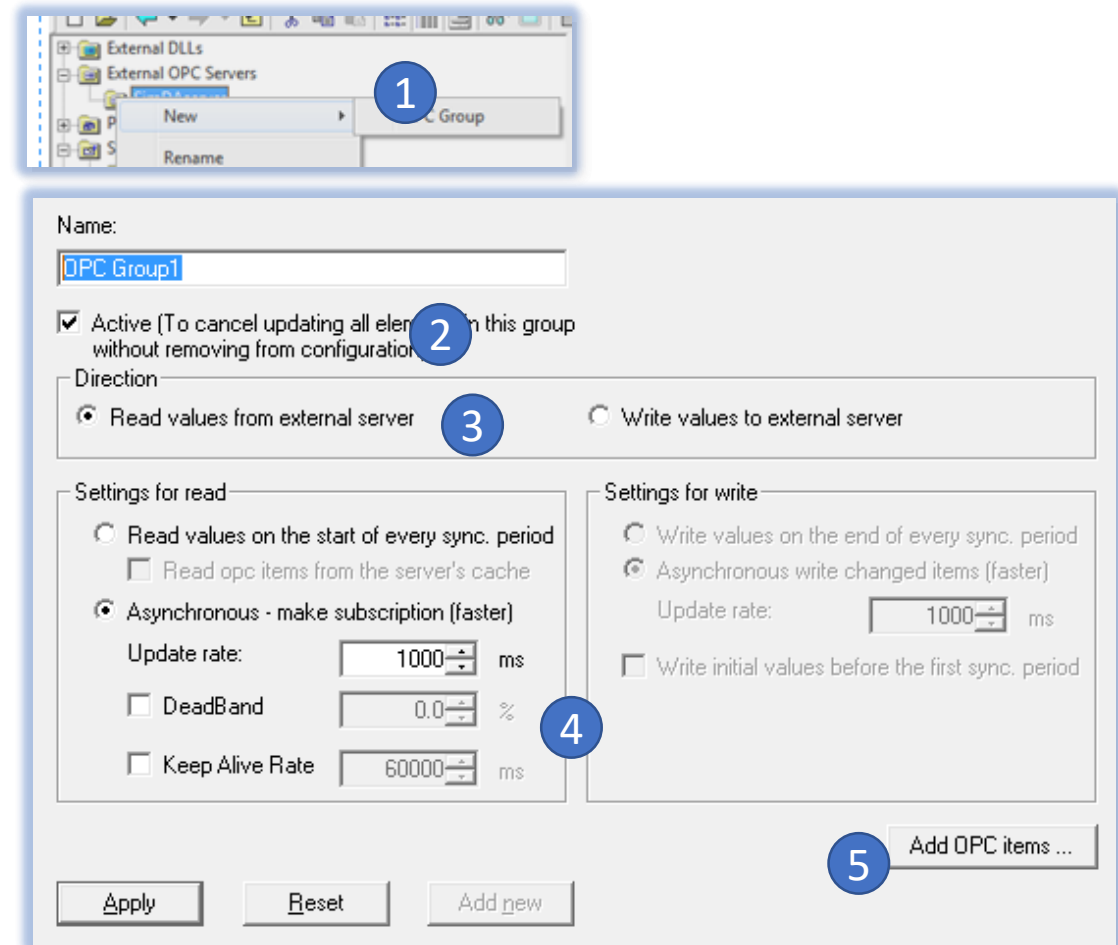
Period of checking connection (Re-connection) [s]: 300 Generate new

☒ If it isn't connected, set the quality of all the items not connected. Server Status ...

Apply Reset Add new

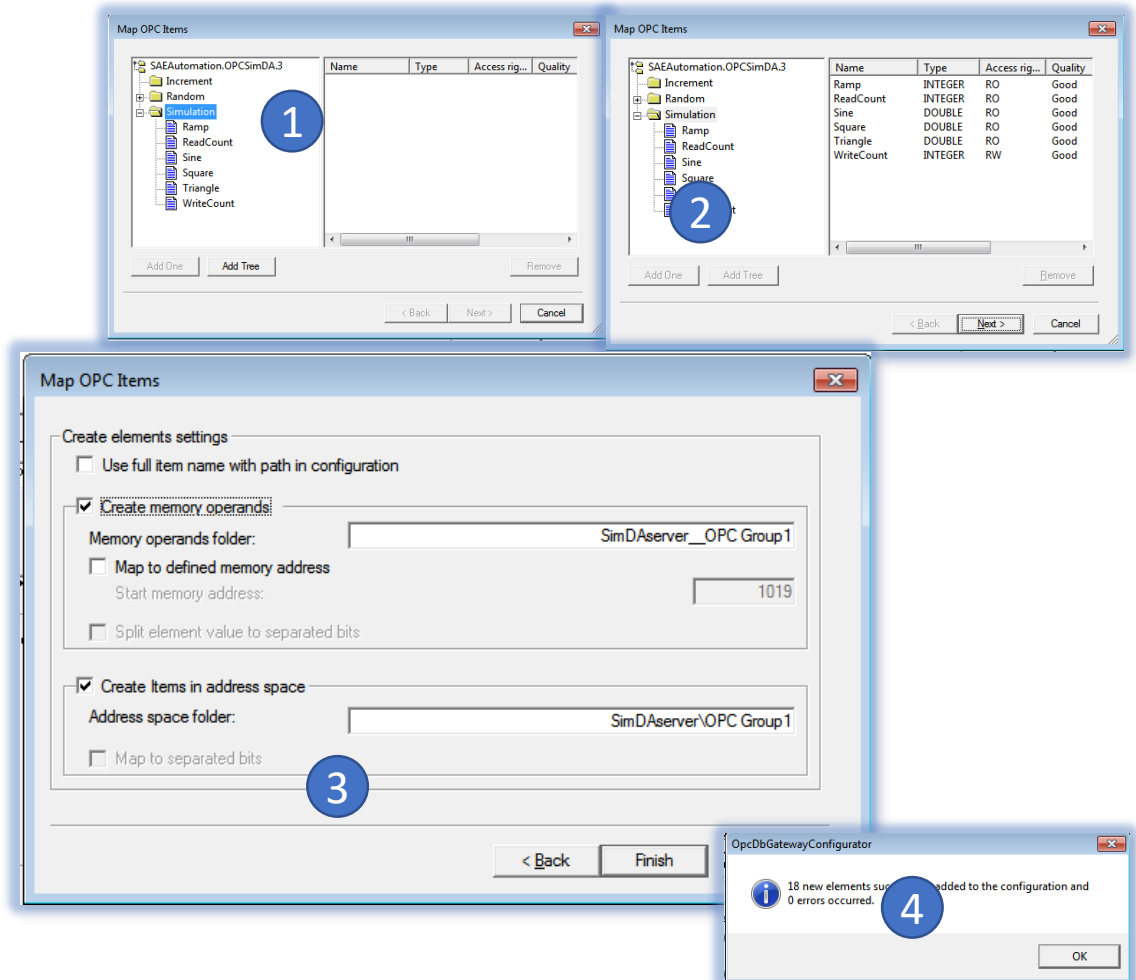
Configuring OPC groups in the OPC DA client 4/6

1. Create a new group
2. Disable Group Configuration
3. Read / Write
4. Read / write properties
5. Adding OPC variables to a group



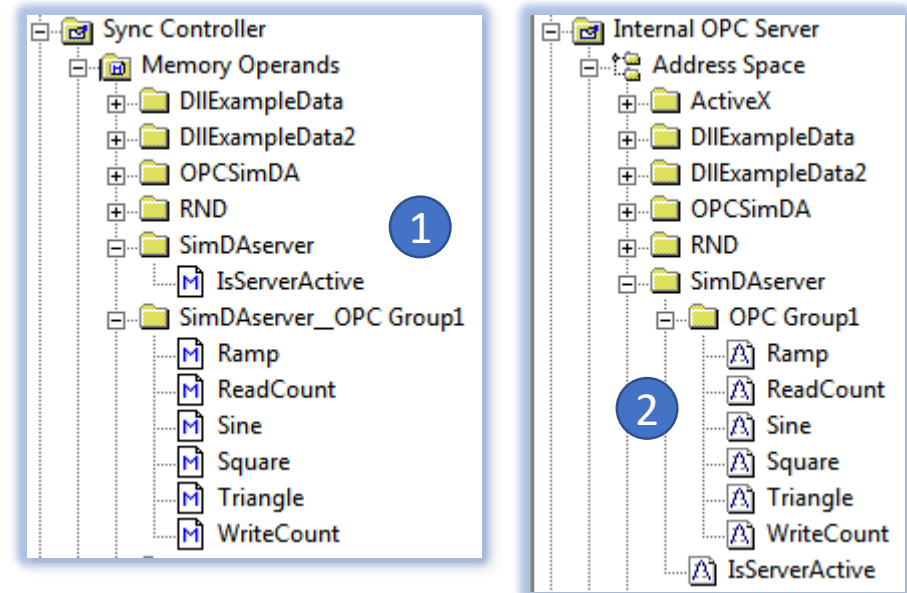
Configuring OPC groups in the OPC DA client 5/6

1. Select directory
2. Select OPC variables from the directory
3. Mapping properties on MO and OPC variables of int. OPC server
4. Mapping information



Configuring OPC groups in the OPC DA client 6/6

1. The result of mapping to the memory operands
2. Result of OPC mapping to the items in internal OPC server

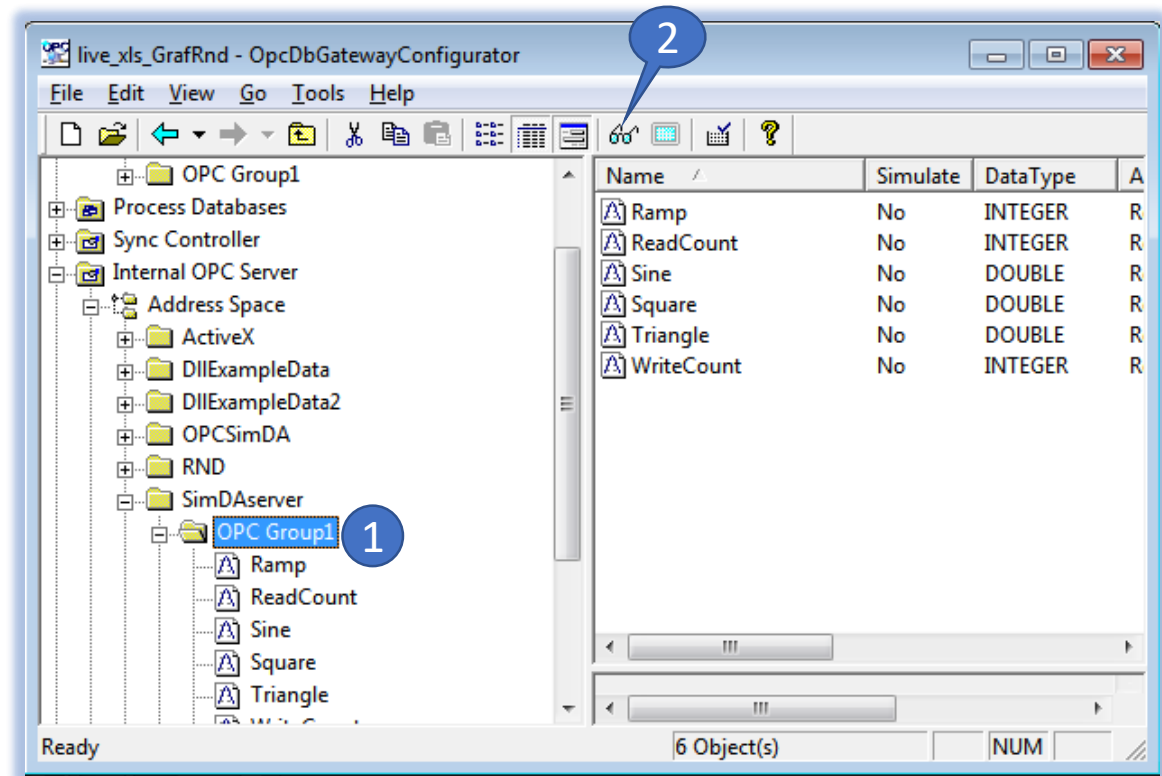


OPC variable write option in OPC Client of the Configurator

Importance: To test writing to the OPC variable on the OpcDbGateway runtime application, it was necessary to use an external OPC client application. Currently, OPC client in the OpcDbGateway configuration application can be used for that.

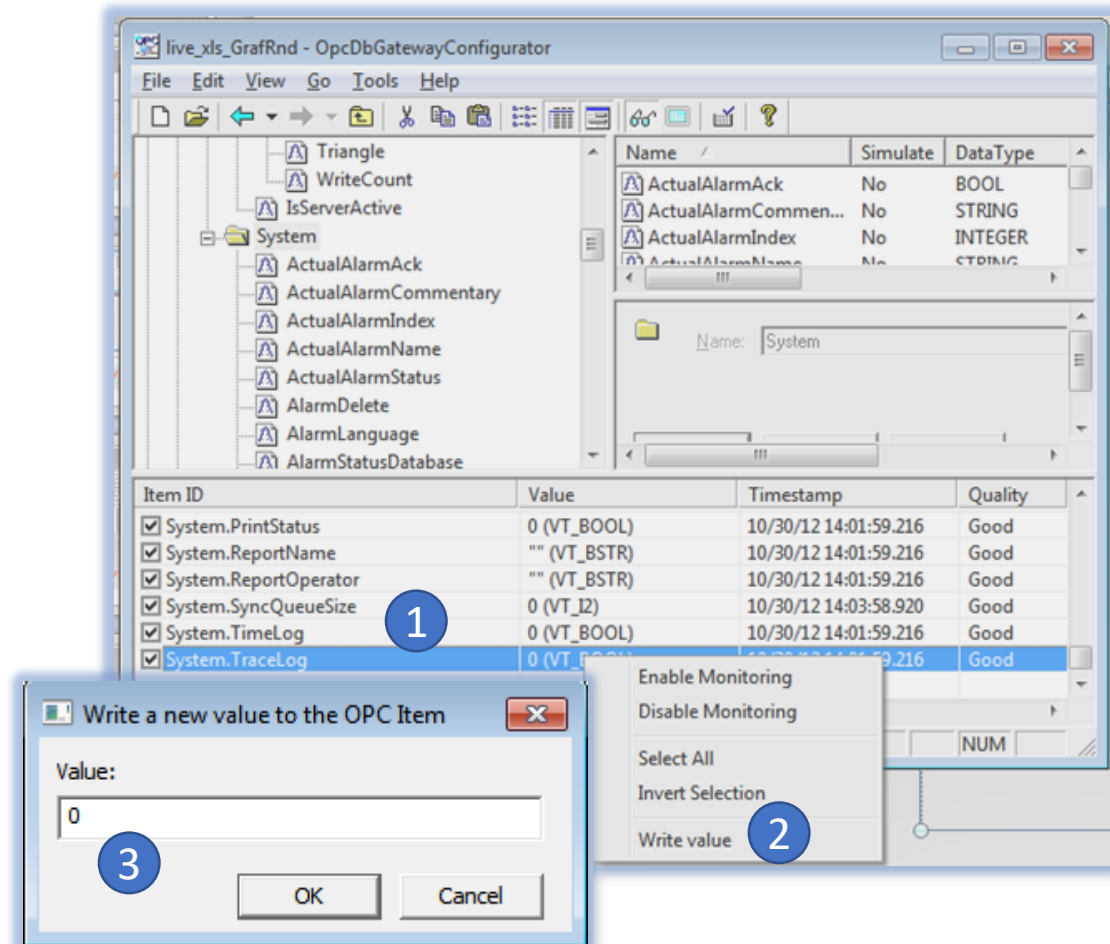
Option to write the OPC variable in the monitor view 1/2

1. Select a monitored directory
2. Start OPC client in the configurator and thus also the internal OPC server of the runtime application



Option to write the OPC variable in the monitor view 2/2

1. Selection of the OPC variable in the OPC client monitor
2. Selecting a listing
3. Write new value




Dialogbox for Event Configuration

Dialog box for Event Configuration 1/2

Name:

Description:

Trigger: 

Priority: (Event with higher priority will be performed earlier.)

☐ Synchronous event (call in synchronous thread)

Action type

Call

☐ Functional block ☒ External program

Log

☐ Write message to log ☐ Make new log file

☐ Check used space ☐ Make new alarm log file

Report/Database


☐ Create report ☐ Compare report files and list table

☐ Do query


Test


☐ Beep

Settings related to action

Functional block: 

Program path with arguments:

User message: 

Query: 

Report type:

Report template:

Dialog box for Event Configuration 2/2

- Changes:
 - It is highlighted which events are triggered as synchronic (currently only function block calls) and which are asynchronous.
- The events are divided by function type into:
 - Call - function block, external program/script - choice of MO through which parameters are passed
 - Log
 - Work with databases
 - Testing - currently only single - beep
- Only those parameters that are defined in the given event type are activated

Dialog box for trigger configuration

Dialog box for trigger configuration 1/4

Name: Triggger 1

Description:

Triggering condition

Memory operand is true

☒ SimDAServer__OPC Group1/Square

☐ One time valid, when is set (Unchecked: valid in every synchronous period)

Execution time

☒ ☒ 30. 10. 2012 01:00:00.000 (h:m:s.ms)

☐ (Unchecked), Computer time (Unchecked)

Periodically

☐ Stri

☒ 0 00:00:00.000 (d h:m:s.ms)

0

Apply

Reset


Add new

Name:

Description:

Triggering condition

Memory operand is true

☒ 

☐ One time valid, when is set (Unchecked, valid in every synchronous period)

Execution time

☒ (h:m:s.ms)

☐ Time related to start OPC server (Checked), Computer time (Unchecked)

Periodically

☒ (d h:m:s.ms)

Dialog box for trigger configuration 2/4

- Trigger always has a trigger condition:
 - Time
 - The value of the memory operand (MO) - TRUE
 - Once (Trigger activates, the value of MO is automatically inverted without bumping)
 - Repeatedly (for synchronic events trigger is triggered repeatedly while the MO is TRUE)
 - The MO evaluation is always performed at the end of the synchronous cycle
 - Time & MO (MO value is evaluated at a defined time)
- Trigger may have a repeat:
 - At least 1x - then the trigger is activated twice
 - Unlimited - until OpcDbGateway is running
 - Repeat has a defined repeat period always defined

Dialog box for trigger configuration 3/4

- Trigger activation date and time:
 - it can be entered from milliseconds to years.
 - It can be entered in **real** or **absolute** terms such as:
 - The time interval from the start of the OpcDbGateway ranges from ms to one day.
 - Date and time on the local computer.
 - The time is stored in UTC shape and displayed according to the local time setting.
- **Trigger Date:**
 - If not specified, the OpcDbGateway launch date is taken and only time is taken into account. If the server is run later than the specified time, it is activated (in the case of a periodic or repeated trigram) the next day. (Exactly on the current day the trigger is not activated)
 - If it is specified - if the activation time is old, the next time is triggered according to the period. If no period is specified, the trigger will never be activated.
- For synchronic and asynchronous events **the same trigger can be used.** (It was not possible before)

Dialog box for trigger configuration 4/4

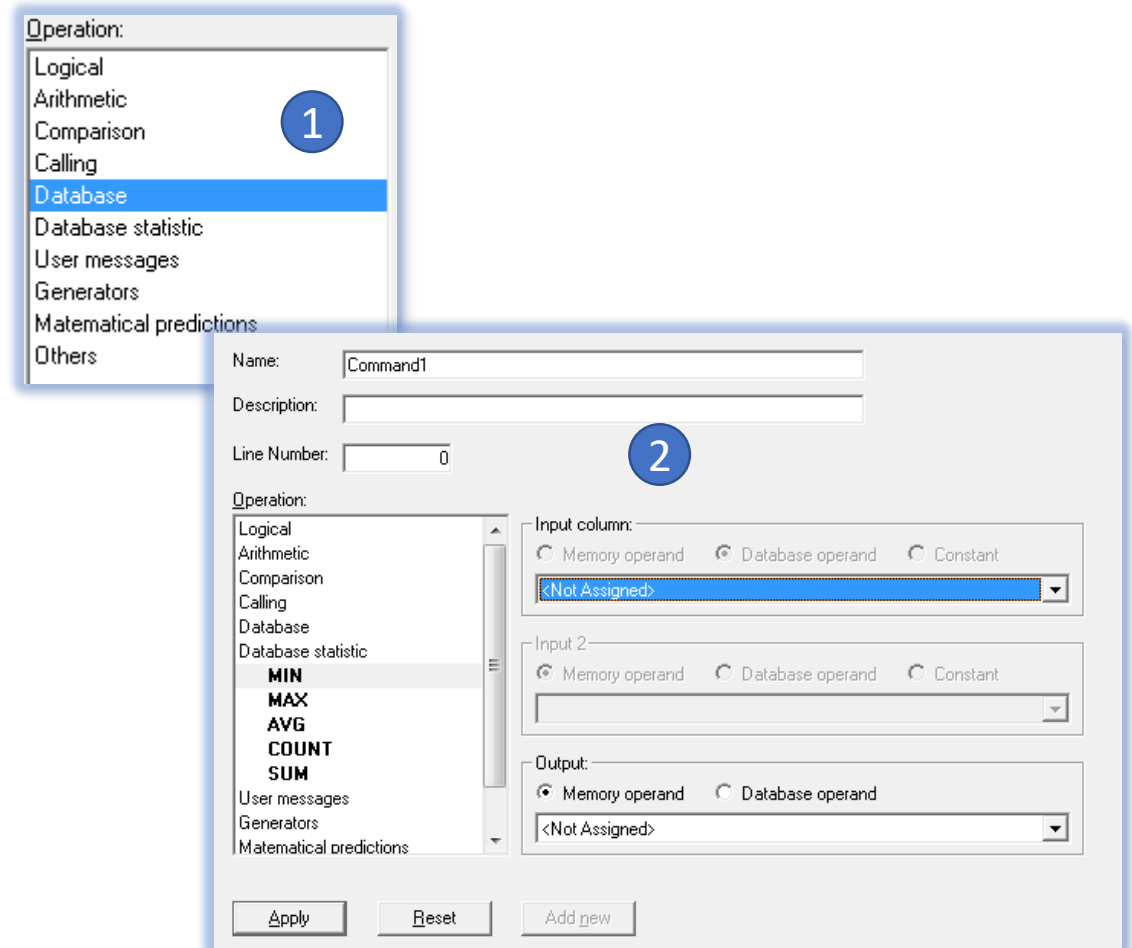
Periodic trigger

- Period - time between activations of the trigger.
- The repeat period can be entered from milliseconds to years.
- Two types of periods:
 - Exact - always the same interval from ms to days
 - Calendar - annual, monthly, weekly, daily. The length of the interval (annual, monthly) may change.
- The first trigger activation of the periodic trigger:
 - If the start time is set as well as the period - wait for the start time and then take place according to the given period.
 - If the start time is not specified – the first activation is at the start time of the synchronous thread (after FB START / RESTART).
- If the trigger is defined with a limited number of repeats and with an activation memory operand, the number with the operand memory value - FALSE is also counted to the number of repeats .

A more transparent dialog for
working with commands

A more transparent dialog for working with commands

1. Choosing a command type
2. Selection of Input / Output operands



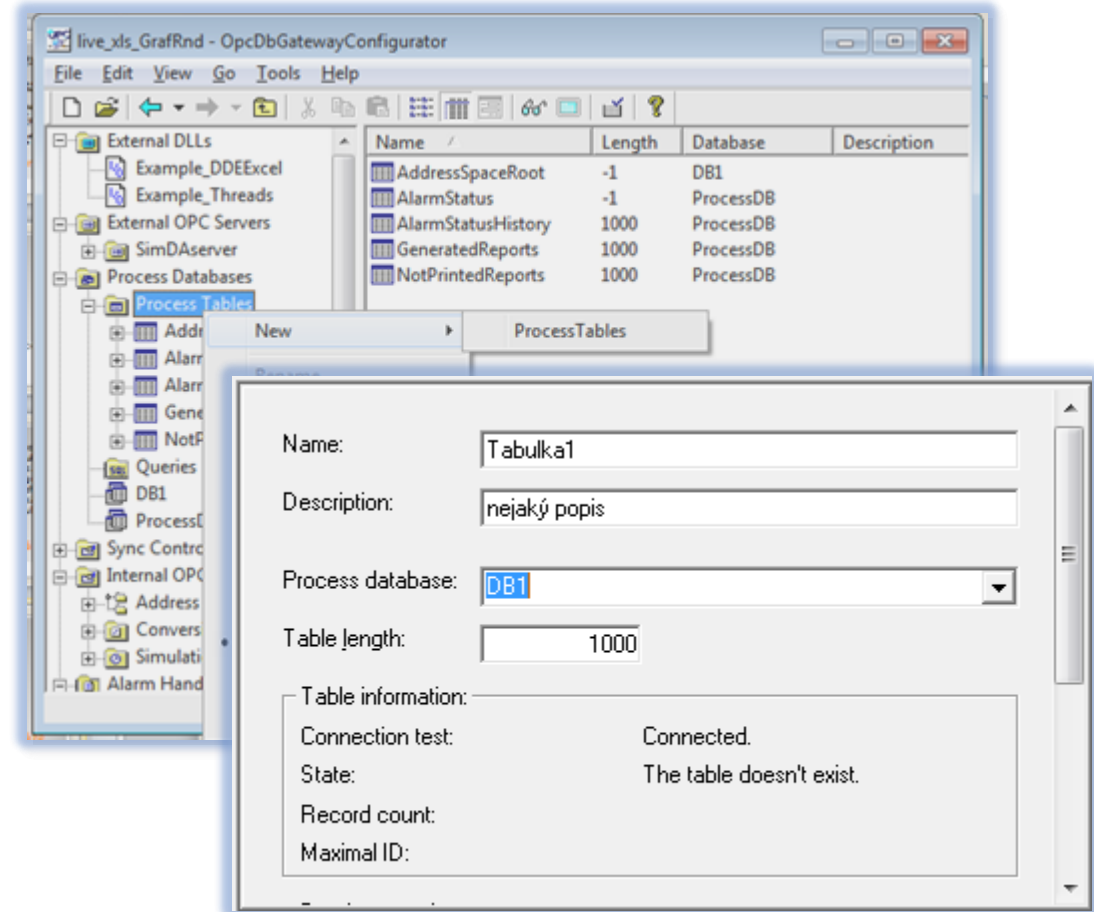
Create table in database by
configuration

Create table in database by configuration 1/4

- OpcDbGateway can work with existing tables on the selected database or create new tables according to the needs of a specific application.
- If the tables are defined in the configurator, then, to prevent possible errors, it is convenient for the configuration application to create defined tables on the database automatically

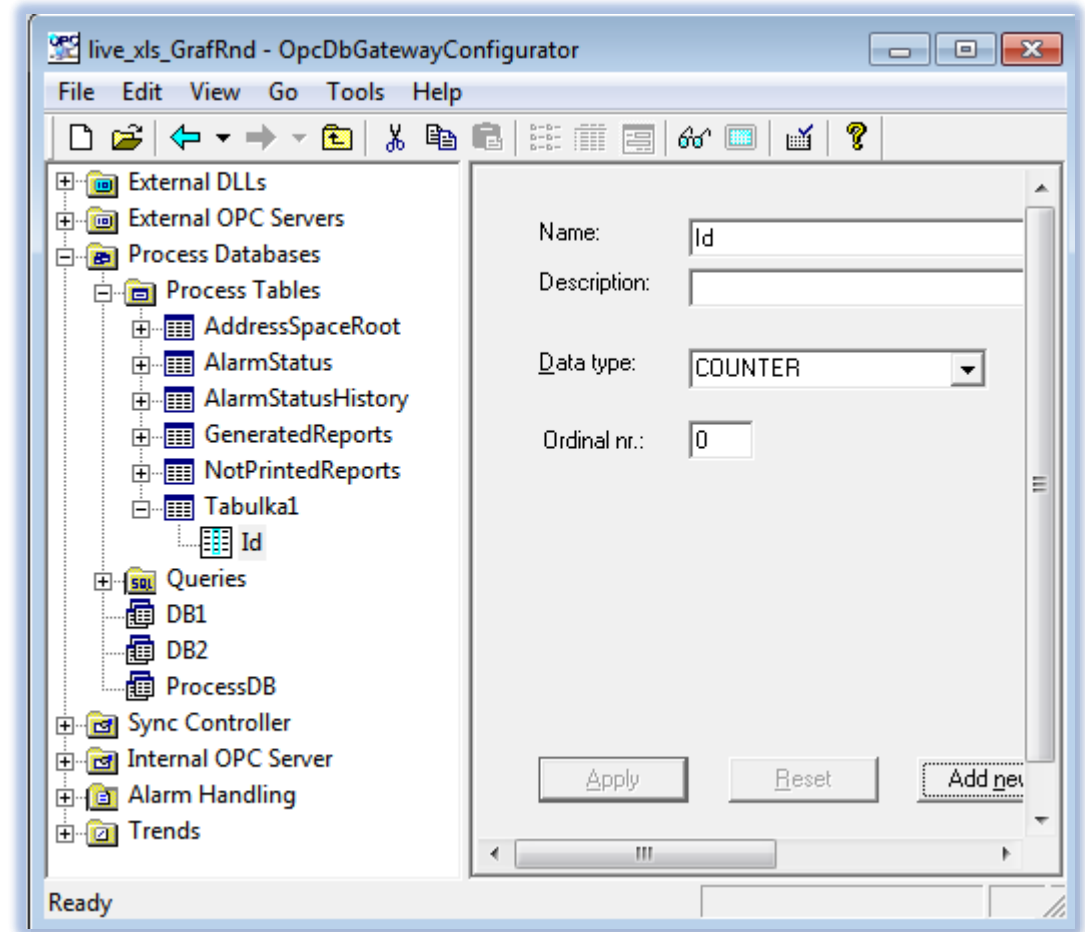
Create table in database by configuration 2/4

- Define the table in the configuration.
- One parameter is the database where the table should be created (must be added in advance in the Process databasses in the configuration)



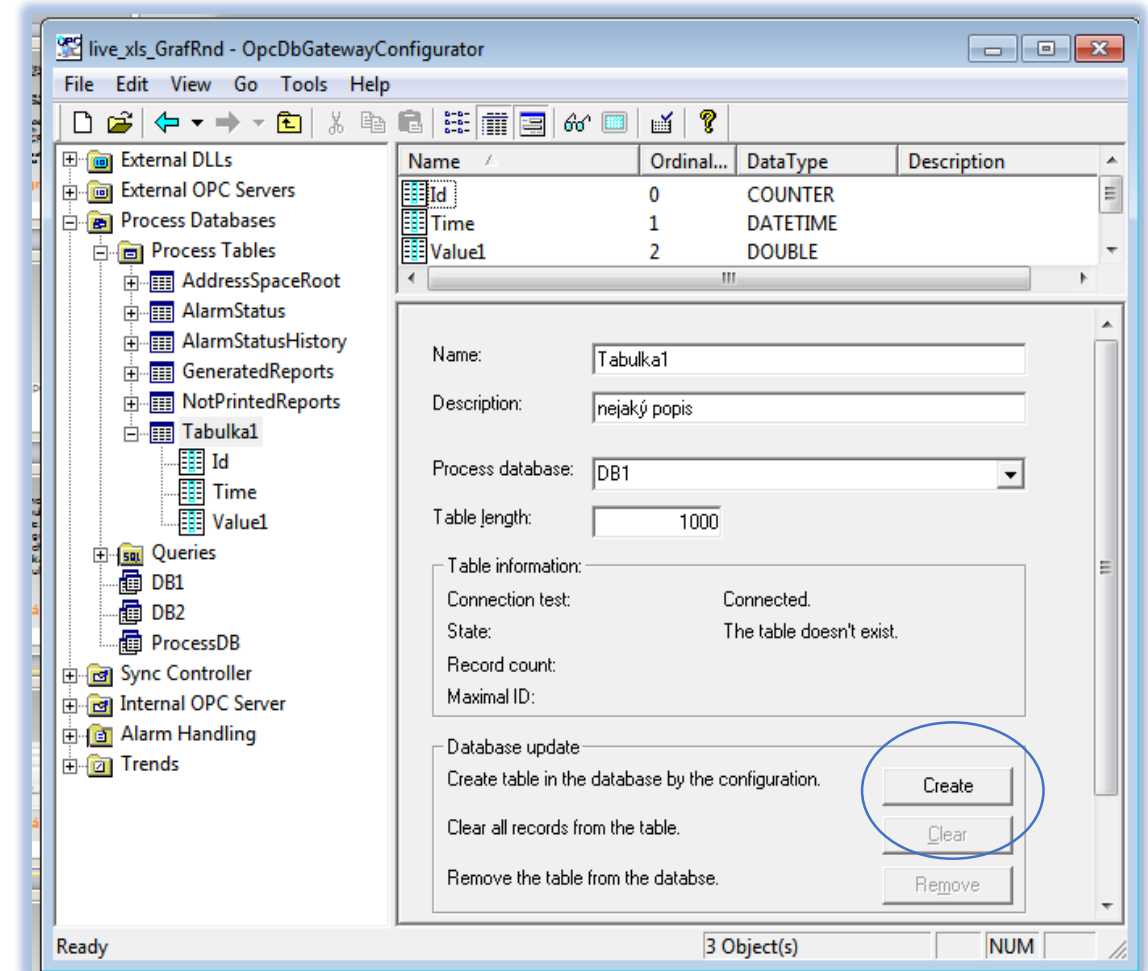
Create table in database by configuration 3/4

Define table columns.



Create table in database by configuration 4/4

- Create a defined table on the database

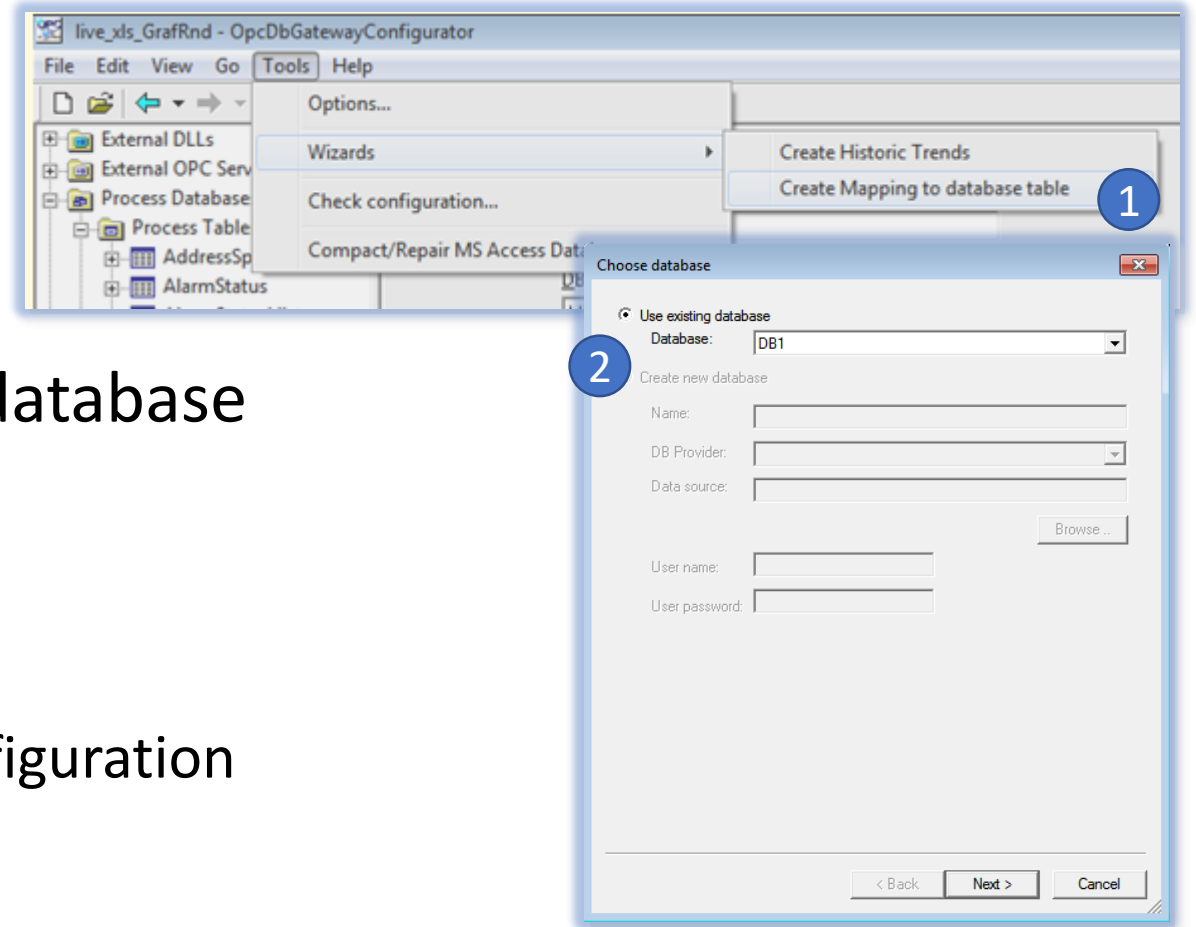


Map tables in database to
configuration

Map tables in database to configuration 1/8

- Transposing existing tables to the selected database into configuration and then mapping cells and columns to database operands quickly and significantly reduces the possibility of configuration errors.
- If we already have a table within defined tables, it is a quicker procedure than creating tables in the configuration and then mapping to the database - you do not have to configure table columns
- Not all columns from the database table must be configured in a table in the configuration (we can only map the columns for which we want to define DO)
- Also included is mapping of individual table cells to database operands. The database column can also be the entire column of the table.
- If we work with the database only through SQL commands, then the tables do not need to be included in the configuration. Just have the database itself and create a query for it.

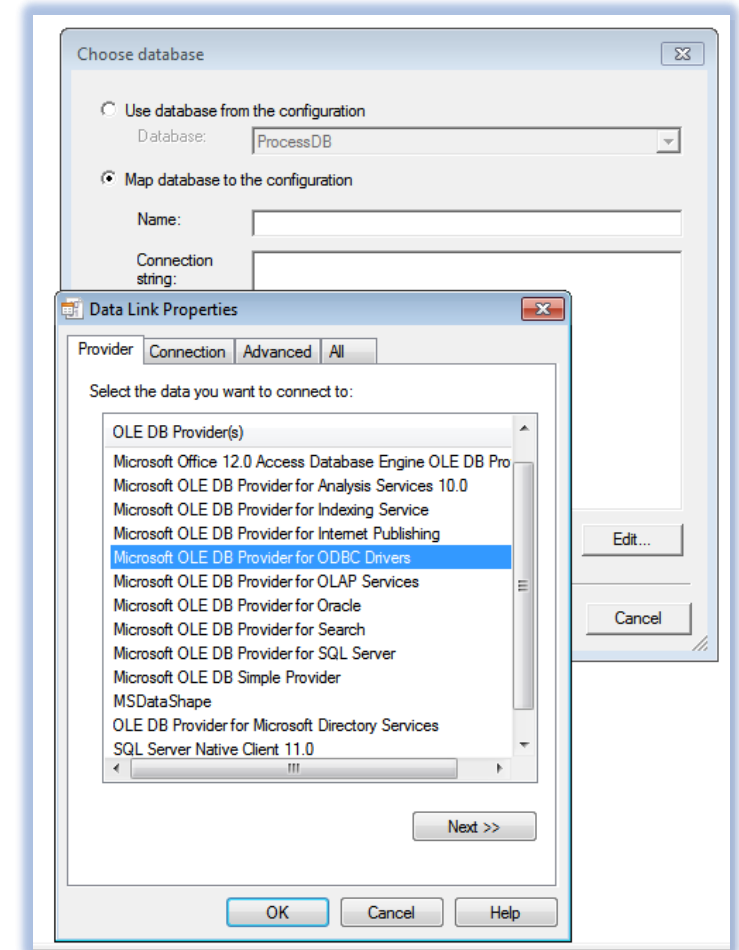
Map tables in database to configuration 2/8



1. Start-up Assistant for mapping database tables
2. Select a database
 - Already defined in configuration
 - Existing but not included in the configuration

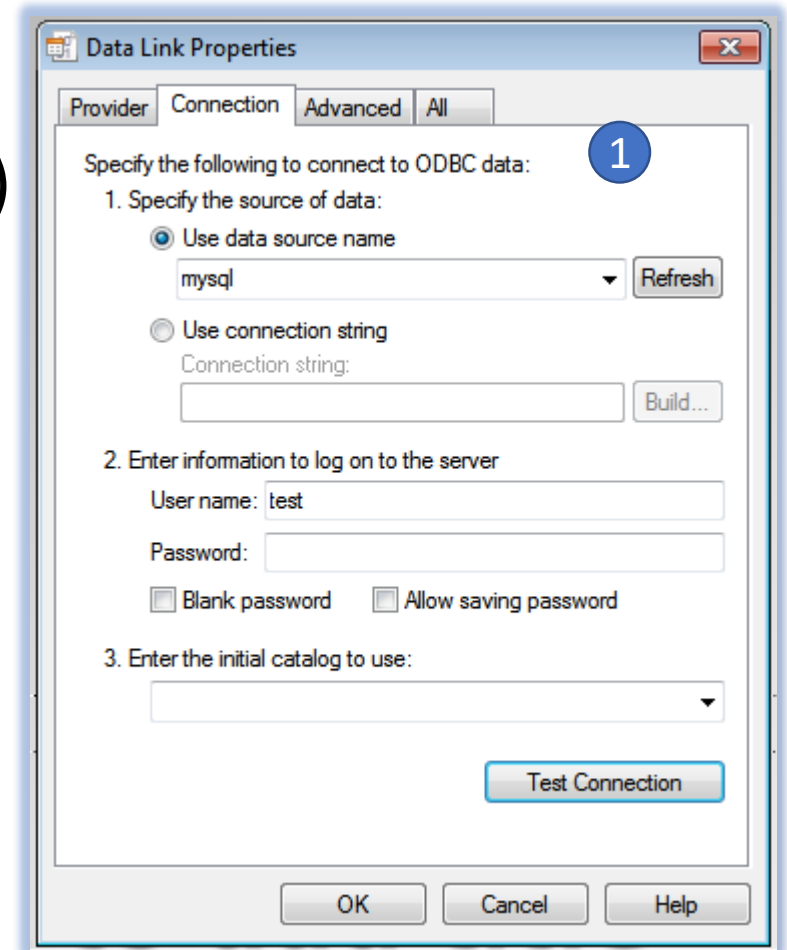
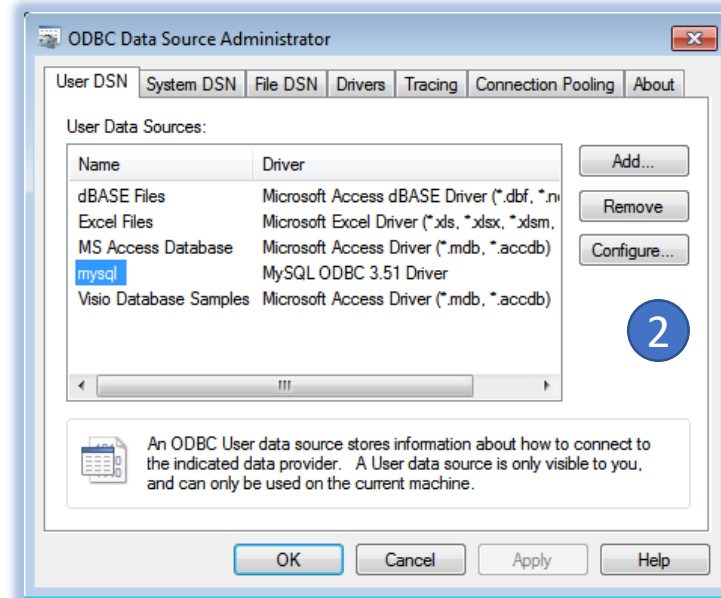
Map tables in database to configuration 3/8

Configure a database not yet included in the configuration using a universal dialog to connect to a defined data source



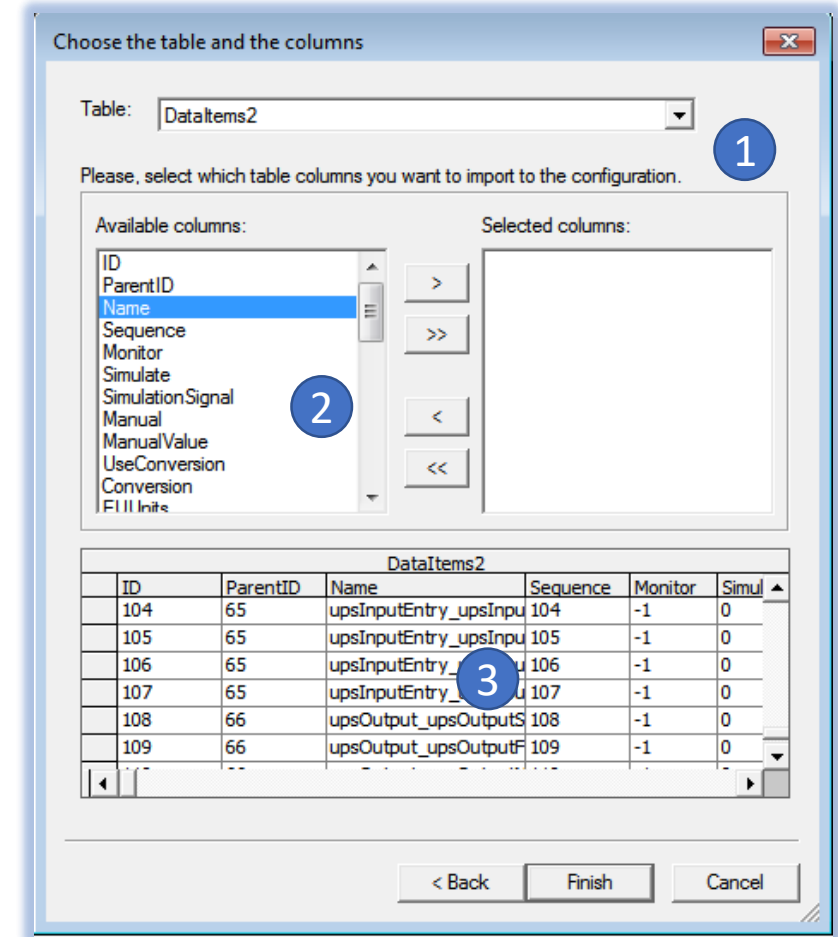
Map tables in database to configuration 4/8

1. Select "data connection"
2. "Mysql" was defined as an ODBC data source)



Map tables in database to configuration 5/8

1. Select a table from the database
2. Choosing the columns in a table in the configuration (just select those to which we want to create DB operands)
3. View data from a database table



Map tables in database to configuration 6/8

1. Choosing the prefix of the DB operand name
2. Selection of a cell from a table that is mapped to the DB operand
3. A summary of cells that will be mapped to DB operands

Map DB operands

☒ Map the cells of the table to DB operands.

Prefix name of DB operand: DataItems2_2

When you click on a cell, then add it.

ID	ParentID	Name	Sequence	Monitor	Simul
111	68	upsOutputEntry_upsOu	111	-1	0
112	68	upsOutputEntry_upsOu	112	-1	0
113	68	upsOutputEntry_upsOu	113	-1	0
114	68	upsOutputEntry_upsOu	114	-1	0
115	68	upsOutputEntry_upsOu	115	-1	0
116	69	upsBypass_upsBypassF	116	-1	0

Column	Row	Value
2 Name	2	upsOutputEntry_...
2 Name	3	upsOutputEntry_...

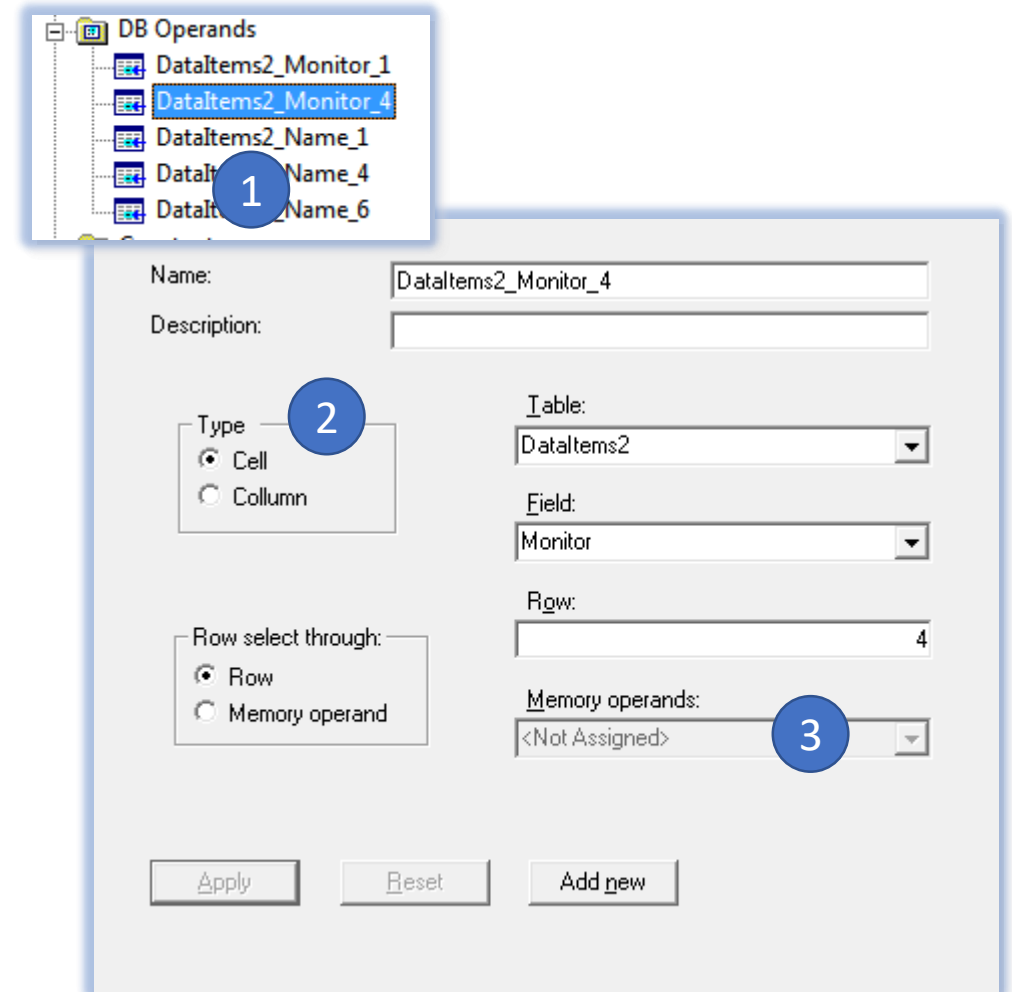
Remove

< Back Finish Cancel

Map tables in database to configuration 7/8

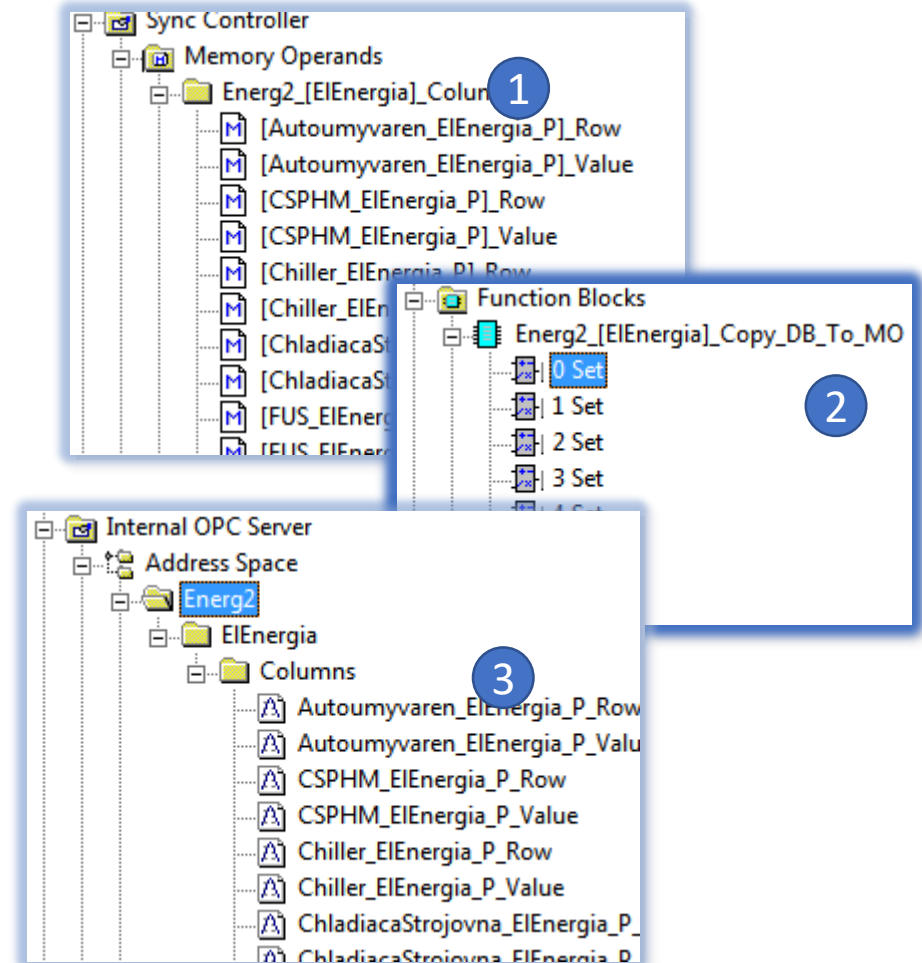
1. Pohľad na vytvorené DO
2. DO priradené bunkám je možné zmeniť pomocu Type z „Cell“ na „Column“.
3. Konkrétna bunka je potom adresovaná prostredníctvom MO (kurzor)

Viaceré stĺpcové DO môžu využívať rovnaký MO pre adresáciu – takto sa dá adresovať celý riadok tabuľky



Map tables in database to configuration 8/8

1. Automatically create MO to DO so that other data sources (eg internal OPC server) can be linked to a database table
2. Automatically create a function block for copying DO to MO (you can also define copy in the opposite direction)
3. Automatic creation of OPC variables to MO



View Order of Commands in Tree View

View Order of Commands in Tree View

The image displays a software interface for managing commands. On the left, a tree view shows a hierarchy starting with 'Main', which contains a list of commands: '10 10 Result=0', '20 20 CALL DLL DIExample', '30 30 CALL Inputs=Outputs', '40 40 Rnd', '50 50 Rnd', '60 60 Rnd', and '70 70 Rnd'. The command '10 10 Result=0' is selected. On the right, a detailed view for this command is shown. It includes fields for 'Name' (10 Result=0), 'Description' (empty), 'Line Number' (10), and 'Operation' (SET). Below these are sections for 'Output type' (Mem selected), 'Output' (DIExampleData/Result), 'Input1 type' (Const selected), 'Input 1' (FALSE(0)), and 'Input 2' (<Not Assigned>). At the bottom, there are 'set' and 'Add new' buttons. In the foreground, a table lists the commands in order, and a separate box shows the details for the 'Main' function block.

LineNr	Name	Description
10	10 Result=0	
20	20 CALL DLL DIExample	
30	30 CALL Inputs= Outputs	
40	40 Rnd	
50	50 Rnd	
60	60 Rnd	
70	70 Rnd	

Name: Main
Description: The Main function block is called every PlcPeriod

View the contents of the database
tables in the configurator

View the contents of the database tables in the configurator

1. Command to view table content
2. View table content

The image shows a software interface for managing a database. The main window has several sections:

- Name:** EIEnergia
- Description:** (empty field)
- Table length:** -1
- Table information:**
 - Connection test: Connected.
 - State: The table exists. (marked with a blue circle '1')
 - Record count: 76478
 - Maximal ID: 76214
- Database update:**
 - Create table in the database by the configuration. (Create button)
 - Clear all records from the table. (Clear button)
 - Remove the table from the database. (Remove button)
- Buttons at the bottom: Apply, Reset, Add new

A button labeled "Show table..." is circled in blue, indicating the command to view the table content.

Below the configurator, a window titled "Records in the table" displays a list of records. A blue circle '2' is placed over the first record, indicating the view of the table content.

Record ID	Time	Value	Record ID	Time	Value	Record ID	Time	Value	Record ID	Time	Value
1877	20. 1. 2009 13:01:00	0	1878	20. 1. 2009 13:16:00	8	1879	20. 1. 2009 13:31:00	8	1880	20. 1. 2009 13:46:00	6
1881	20. 1. 2009 14:01:00	1	1882	20. 1. 2009 14:16:00	5	1883	20. 1. 2009 14:31:00	6	1884	20. 1. 2009 14:46:00	5

Changes in the runtime
application

Changes related to trigger 1/3

If the trigger is activated before the first period (based on initialized values), synchronous and asynchronous events that occur at the same time as the synchronous thread are included in the queue.

All triggers (which do not have a defined start time) and synchronous thread have the same start time. This time is rounded off for seconds to the current time.

Against the previous version, several types of trigger have been canceled, with ambiguous behavior. One trigger type was triggered, the trigger condition of which is a combination of memory operand value and time.

If trigger is dependent on memory operand and time when it is triggered, the value of the operand memory is checked only at the time of trigger activation.

Changes related to trigger 2/3

- Streamline and refine triggering of individual triggers:
 - The trigger repeat period can be for several milliseconds, while preserving high repeat accuracy.
 - The number of triggered times is not limited. In the previous version, the maximum was 62.
- Windows is not considered to be a real-time OS, resulting in the following limitations:
 - The current time increment is 15 milliseconds.
 - Switching between threads and processes takes some time.
 - The accuracy of time is not guaranteed and depends on the particular computer.
 - The local computer time can be corrected from the domain server or from time.windows.com based on the "w32time" service that detects the current time (if it's important to keep the ms-period accurate, it's best to turn it off)
- The process priority in Windows where the trigger triggered event occurs in OpcDbGateway has increased by default to ABOVE_NORMAL_PRIORITY_CLASS. (The user may change it with "PriorityClass" in the registers.)

Changes related to trigger 3/3

- If both the start time and the period are specified, first wait for the start time and then take place according to the given period. If the start time is not specified, trigger is first activated at the OpcDbGateway start time, which is the same as the start of the synchronous thread.
- Due to optimization, the current time is not obtained for every command you execute in the synchronous thread. The time is obtained at the beginning of the period, and the time is used to write to all memory operands that change in the given period. However, the log file (s) logs things with the current time (not with the start of the period).

Better collaboration of the runtime application kernel with extension DLLs

When the OpcDbGateway starts, the "Start" or "Restart" function blocks in the synchronous thread are started.

Functions "OnStart" and "OnStop" in enhancing DLLs are called from the synchronous thread. The "OnStart" function is called immediately after calling the "Start" or "Restart" function block before the synchronous thread starts. The "OnStop" function is called immediately after calling the "Stop" function block.

This allows initiation related to the DLL activity before FB MAIN's 1st performance

Asynchronous Logging - does not adversely affect the Synchronic Thread Period

Performing logging operations is relatively time-consuming

Je využíváné len na diagnostické účely

From the perspective of the quasi-parallel processing of the function blocks, it is of no practical importance that the execution of the FB in the synchronous thread is blocked by completing the write to the log file

More efficient work with databases

- Internal implementation of aggregated database functions over individual columns (MIN, MAX, SUM, AVG, COUNT) using SQL commands above the database - great acceleration of execution.
- Opening and closing tables should not adversely affect the synchronic cycle period.
- In the synchronous cycle, more work is done with DB operands - so the tables are open already at the beginning.
- In the asynchronous cycle, more SQL statements work less frequently, but usually with more data - so the tables are opened before use.

Names of OPC variables

- Restricted limitation that the names of OPC Data items can contain only 'a-Z', '_', '0-9' and has to begin with 'a-Z'
- OPC variable names can be up to 255 characters (previously only 50)

Results

Results

- Significant improvement of functionality on the client side (data aggregation)
 - Groups on ext. OPC DA servers
 - It works as a client for many databases (for those with which the database drivers are installed)
- Better data processing functionality
 - Event functionality - Extension of options and triggers
 - Cyclical processing (more accurately observing the cycle period)
- Simplify and Expand Configuration Options:
 - collaboration with databases
 - trigger
- There is no need for a separate test client for writing to the OPC Server OPC variable
- Several errors and disputed functionality have been removed