



**zenon**  
by COPA-DATA

# zenon manual

## Prediction Interface

v.8.10



**COPADATA**

© 2019 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. Technical data is only used for product description and are not guaranteed qualities in the legal sense. Subject to change, technical or otherwise.

# Contents

<b>1</b>	<b>Welcome to COPA-DATA help.....</b>	<b>5</b>
<b>2</b>	<b>Prediction Interface .....</b>	<b>5</b>
<b>3</b>	<b>Configuration .....</b>	<b>6</b>
3.1	Predictions: Polling transfer .....	6
3.2	Predictions: Spontaneous transfer.....	9
3.3	Metadata Synchronizer .....	11
3.4	Analyzer functionality over REST .....	12
<b>4</b>	<b>Predictive Analytics Driver .....</b>	<b>13</b>
4.1	PA_Drv - Datenblatt.....	14
4.2	Driver history.....	15
4.3	Requirements.....	15
4.3.1	PC 15	
4.3.2	PLC.....	16
4.4	Configuration.....	16
4.4.1	Creating a driver.....	17
4.4.2	Settings in the driver dialog .....	20
4.5	Creating variables.....	26
4.5.1	Creating variables in the Editor .....	26
4.5.2	Addressing.....	30
4.5.3	Driver objects and datatypes .....	32
4.5.4	Creating variables by importing .....	35
4.5.5	Communication details (Driver variables) .....	54
4.6	Driver-specific functions.....	59
4.7	Driver command function .....	61
4.8	Error analysis.....	65
4.8.1	Analysis tool.....	65
4.8.2	Check list .....	66
<b>5</b>	<b>Analyzer Service Node Interface .....</b>	<b>68</b>
5.1	Start and set parameters for service node interface.....	69
5.1.1	Sort, filter and group lists .....	71
5.1.2	Windows service for Analyzer Service Node .....	74
5.2	Connection to the service hub .....	77

5.2.1 Dialog: Service Node Configuration Tool .....	77
<b>5.3 Reports .....</b>	<b>81</b>
5.3.1 Report.....	83
5.3.2 Parameter.....	86
5.3.3 Formats.....	87
5.3.4 Schedules.....	87
<b>5.4 SQL elements .....</b>	<b>94</b>
5.4.1 SQL element.....	96
5.4.2 Parameter.....	98
5.4.3 Columns.....	100
5.4.4 Schedules .....	100
<b>5.5 Prediction .....</b>	<b>107</b>
5.5.1 Prediction model.....	110
5.5.2 Schedules .....	111
<b>6 Example schedule configuration .....</b>	<b>119</b>
<b>7 Service Node Configuration Tool.....</b>	<b>120</b>
<b>8 Metadata Synchronizer.....</b>	<b>122</b>
8.1 Configuration.....	123
8.1.1 Analyzer Server selection dialog.....	124
8.1.2 Database selection dialog .....	125
8.2 Execution .....	125

# 1 Welcome to COPA-DATA help

## ZENON VIDEO-TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel ([https://www.copadata.com/tutorial\\_menu](https://www.copadata.com/tutorial_menu)). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

## GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to [documentation@copadata.com](mailto:documentation@copadata.com).

## PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at [support@copadata.com](mailto:support@copadata.com).

## LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email [sales@copadata.com](mailto:sales@copadata.com).

# 2 Prediction Interface

With **Prediction Interface**, you get predictions from zenon Analyzer as variable values for:

- ▶ Time-based predictions
- ▶ Value-based predictions

You can request these as:

- ▶ Triggered predictions
- ▶ Predictions with schedule

zenon Analyzer calculates the predictions. The **Predictive Analytics** driver provides these variables in zenon Runtime.

## 3 Configuration

In order to be able to use the **Prediction Interface** in the Runtime, you need - depending on your intentions - configurations for:

- ▶ **zenon Editor:**
  - ▶ **Predictive Analytics Drivers** (on page 13)
  - ▶ **Metadata Synchronizer** (on page 122)
- ▶ **zenon Runtime**
- ▶ **zenon Analyzer:**
  - ▶ **Analyzer Service Node Interface** (on page 68)
  - ▶ **Prediction Model Manager**
- ▶ **Service Node Configuration Tool**
- ▶ **Service Grid Configuration Backend**

You can find a transfer schematic diagram and steps for setting parameters in the following sections:

- ▶ **Predictions: Polling transfer** (on page 6)
- ▶ **Predictions: Spontaneous transfer** (on page 9)
- ▶ **Metadata Synchronizer** (on page 11)
- ▶ **Analyzer functionality over REST** (on page 12)

### 3.1 Predictions: Polling transfer

*Polling* predictions means: You use a trigger to trigger predictions if required.

#### CONFIGURATION

To configure predictions if required:

1. Create the required prediction model in zenon Analyzer with the **Prediction Model Manager**.
2. Prepare the **Service Hub** on zenon Analyzer:
  - ▶ A **Service Hub** is automatically pre-installed and configured on the Analyzer server.
  - ▶ You do not need to do anything more to use this.

- ▶ If you want to use another **Service Hub**:  
Configure zenon Analyzer so that it uses the desired **Service Hub**. Use the **Service Node Configuration Tool** to do this.
- 3. Configure the zenon Editor with the **Service Node Configuration Tool** so that it uses the **Service Hub**.
- 4. Add the **Predictive Analytics** driver in the zenon Editor.
- 5. Select the **Service Hub**.
- 6. Configure the driver.
- 7. Import the variables from the driver and configure them.
- 8. Create the Runtime project and use the prediction variables.

## PREDICTION VARIABLES

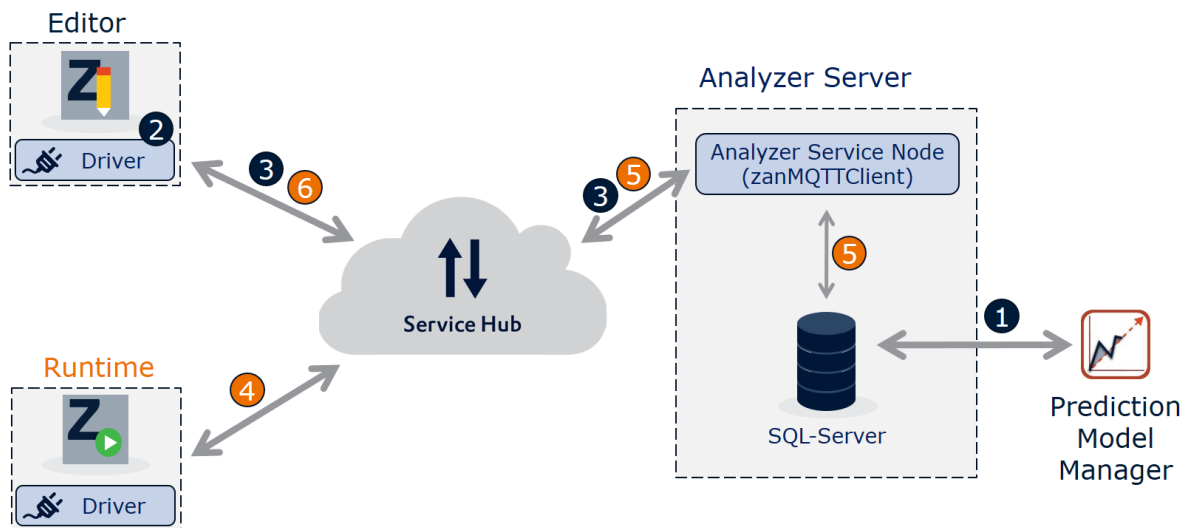
### Created prediction variables:

- ▶ Value-based prediction of requirements with polling communication for 1 prediction:  
3 variables are created:
  - ▶ Input size (value)
  - ▶ Trigger variable to start the prediction
  - ▶ Result variable
- ▶ Value-based prediction of requirements with polling communication for several predictions:  
 $n$  variables are created:
  - ▶ *From* variable: Start of value range
  - ▶ *To* variable: End of value range
  - ▶ Trigger variable to start the prediction
  - ▶  $n$  Result variables as array
- ▶ Time-based prediction of requirements with polling communication for 1 prediction (time stamp):  
3 variables are created:
  - ▶ Input size (time stamp)
  - ▶ Trigger variable to start the prediction
  - ▶ Result variable
- ▶ Time-based prediction of requirements with polling communication for several predictions (time range):  
 $n$  variables are created:
  - ▶ *From* variable: Start of time range

- ▶ *To* variable: End of time range
- ▶ Trigger variable to start the prediction
- ▶ *n* Result variables as array

## TRANSFER PROCESS

### Runtime Analytics (dataflow polling)



#### In the Editor

1. The prediction models are created in the **Prediction Model Manager** and saved in the zenon Analyzer metadata database.
2. The configured **Service Hubs** are read during driver configuration. The selected **Service Hub** is saved in the driver settings.
3. When browsing the variables:
  - a) The driver queries the list of available zenon Analyzer from the service hub.
  - b) After selection of an Analyzer server: A request for the call of the list of the metadata databases is sent to the **Service Hub** and forwarded to the Analyzer server. The response is sent from the Analyzer server to the **Service Hub**. This forwards it to the driver.
  - c) After selection of a metadata database: Query of the projects from the metadata database that provides the prediction models from the **Service Hub**. The **Service Hub** reads the list of available projects that are used for prediction by the Analyzer server - including variables, prediction models, schedules. It forwards it to the driver.

#### In the Runtime:

4. The variable for **Input** or **From-To** must be configured.  
If the trigger variable has been activated: The driver subscribes to the **Service Hub** and sends the variable information to the **Service Hub**. The driver then waits for the response.



5. The **Service-Hub** forwards the request to the **Analyzer Service Node**. This calls up the SQL server to carry out the prediction calculation. After it has been completed, the **Analyzer Service Node** sends the information back to the **Service Hub**.
6. The **Service Hub** forwards the result to the driver. The driver signs off from the **Service Hub**.

## 3.2 Predictions: Spontaneous transfer

*spontaneous* predictions means: Predictions are automatically created on the basis of a schedule. zenon Runtime subscribes to such a schedule and receives the predicted value automatically as soon as it is calculated. Only the time-based prediction is supported in spontaneous communication.

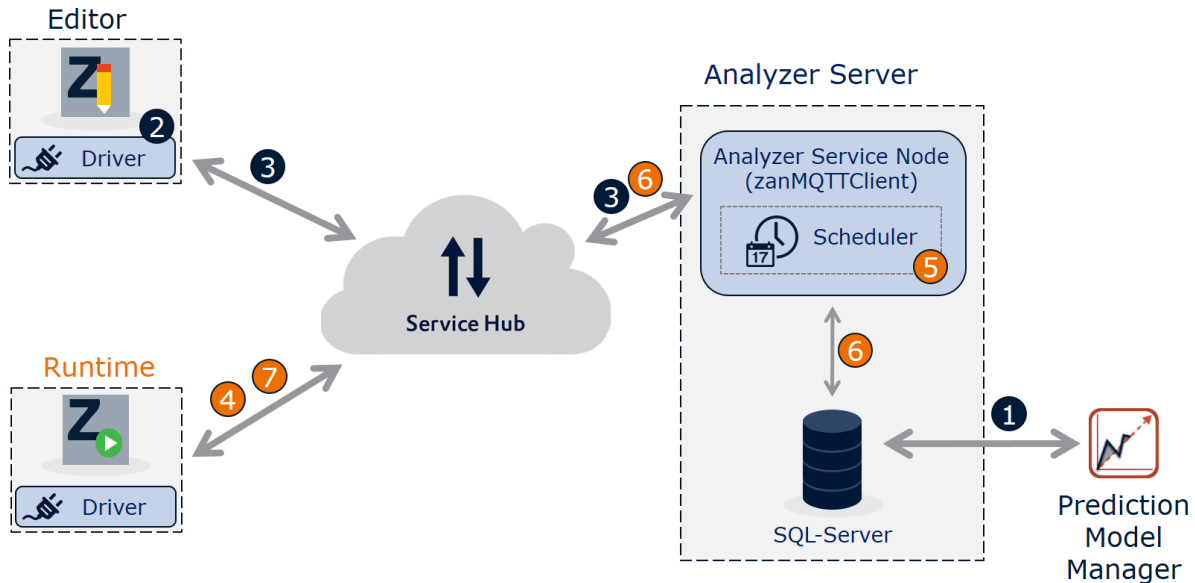
### CONFIGURATION

To configure spontaneous predictions:

1. Create the required prediction model in zenon Analyzer with the **Prediction Model Manager**.
2. Prepare the **Service Hub** on zenon Analyzer:
  - ▶ A **Service Hub** is automatically pre-installed and configured on the Analyzer server.
  - ▶ You do not need to do anything more to use this.
  - ▶ If you want to use another **Service Hub**:  
Configure zenon Analyzer so that it uses the desired **Service Hub**. Use the **Service Node Configuration Tool** to do this.
3. Open the **Service Node Interface** in ZAMS and create the schedules for the required prediction models.
4. Configure the zenon Editor with the **Service Node Configuration Tool** so that it uses the **Service Hub**.
5. Add the **Predictive Analytics** driver in zenon Editor.
6. Select the **Service Hub**.
7. Configure the driver.
8. Import the variables from the driver and configure them.
9. Create the Runtime project and use the prediction variables.  
These are created automatically if a new prediction is available.

## TRANSFER PROCESS

### Runtime Analytics (dataflow spontaneous)



#### In the Editor:

1. The prediction models are created in the **Prediction Model Manager** and saved in the zenon Analyzer metadata database.
2. The configured **Service Hubs** are read during driver configuration. The selected **Service Hub** is saved in the driver settings.
3. When browsing the variables:
  - a) The driver queries the list of available zenon Analyzer from the service hub.
  - b) After selection of an Analyzer server: A request for the call of the list of the metadata databases is sent to the **Service Hub** and forwarded to the Analyzer server. The response is sent from the Analyzer server to the **Service Hub**. This forwards it to the driver.
  - c) After selection of a metadata database: Query of the projects from the metadata database that provides the prediction models from the **Service Hub**. The **Service Hub** reads the list of available projects that are used for prediction by the Analyzer server - including variables, prediction models, schedules. It forwards it to the driver.

#### In the Runtime:

4. When starting the Runtime, the **Prediction Analytics** driver subscribes to the required schedule on the **Service-Hub**.
5. When starting the zenon Analyzer with the **Analyzer Service Node**, the integrated scheduler is also started. Predictions are triggered according to the schedule.
6. If the **Analyzer Service Node** received the predicted values, they are published on the **Service-Hub**.

7. When new predicted values appear on the **Service Hub**, they are always sent to the **Prediction Analytics** driver automatically and automatically displayed in the Runtime

### 3.3 Metadata Synchronizer

The **Metadata Synchronizer** (on page 122) sends metadata from zenon to the zenon Analyzer.

An overview of what it does:

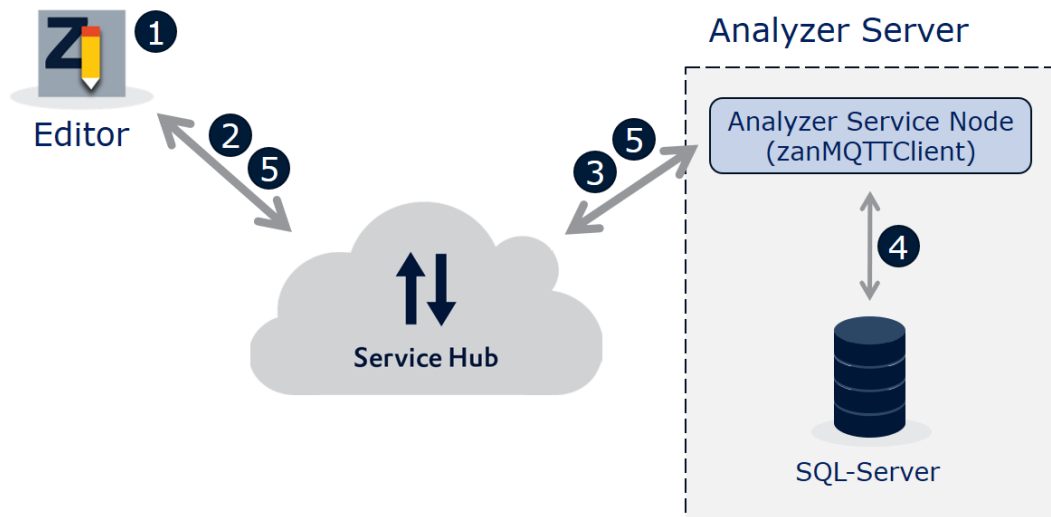
- ▶ Integrated into the zenon Editor directly (no wizard).
- ▶ Ensures fast data transfer.
- ▶ Synchronizes metadata at the press of a button.
- ▶ Only transfers data that is needed by zenon Analyzer.
- ▶ Works without a global project

#### To transfer data with the Metadata Synchronizer:

1. Prepare the metadata database in zenon Analyzer.
2. Prepare the **Service Hub** on zenon Analyzer:
  - ▶ A **Service Hub** is automatically pre-installed and configured on the Analyzer server.
  - ▶ You do not need to do anything more to use this.
  - ▶ If you want to use another **Service Hub**:  
Configure zenon Analyzer so that it uses the desired **Service Hub**. Use the **Service Node Configuration Tool** to do this.
3. Configure the zenon Editor with the **Service Node Configuration Tool** so that it uses the **Service Hub**.
4. Select, in the zenon Editor, for each project that is to be included in the synchronization process, from:
  - ▶ the desired **Service Hub**
  - ▶ the desired **Analyzer Server**
  - ▶ The desired metadata database
5. Transfer the data with a mouse click.

## TRANSFER PROCESS

### Metadata Synchronizer (dataflow)



1. The **Metadata Synchronizer** collects, validates and packs the data from the projects in the workspace.
2. The **Metadata Synchronizer** creates a subscription on the **Service Hub**, writes the packed data to the **Service Hub** and waits for the results of the zenon Analyzer.
3. The **Service Hub** forwards the packaged data to the Analyzer server.
4. The data is unpackaged and validated there. The metadata in the metadata database is then saved, updated or deleted.
5. Status messages and errors are sent back to the **Service Hub**, forwarded to the **Metadata Synchronizer** and displayed in the output window of the zenon Editor.

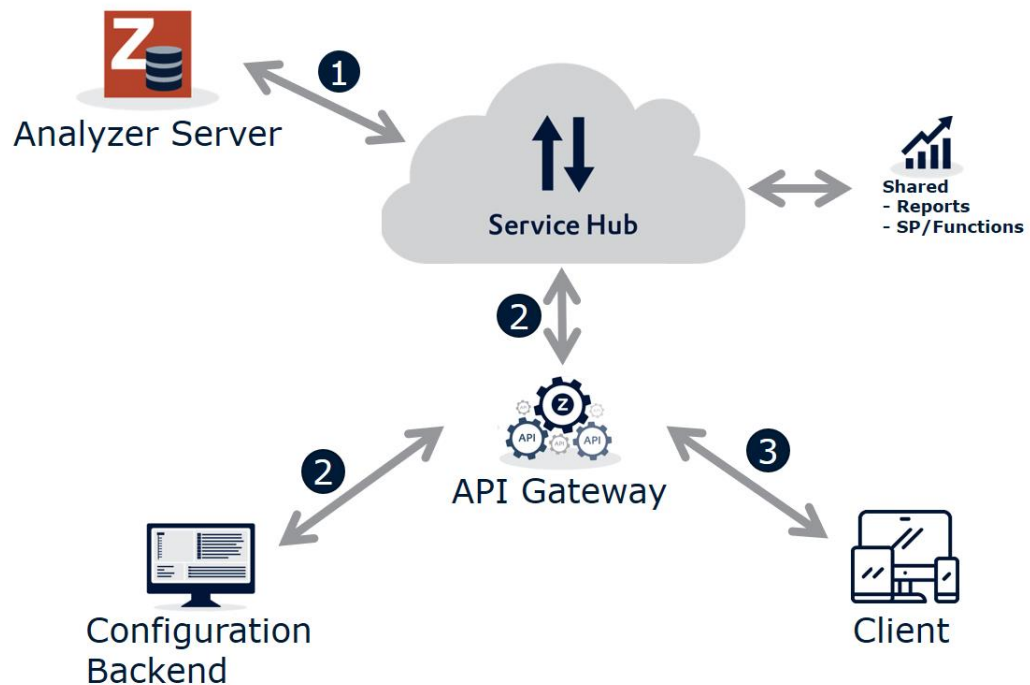
### 3.4 Analyzer functionality over REST

zenon Analyzer now has a **REST Interface** for secure data transfer.

It serves as an interface to third-party systems for local and global use. Furthermore, reports and SQL functionality such as stored procedures or functions can be requested.

## REST FUNCTIONS

### Analyzer functionality over REST



1. Create reports, stored procedures and functions in zenon Analyzer.
  - a) Configure zenon Analyzer so that it uses the **Service Hub**.
  - a) Open the configuration of the **Service Nodes** and configure reports and SQL functionality for access via the REST interface.
2. Open the configuration backend and approve the available reports for clients there.
3. Call up the reports and SQL functionality via REST.

## 4 Predictive Analytics Driver

The **Predictive Analytics Driver** gets prediction data from the Analyzer server.

Predictions can be:

- ▶ spontaneous sent via schedule from the Analyzer Server and received in the driver or
- ▶ Polling from the driver with incoming values queried on the Analyzer server

In doing so, a package can contain:

- ▶ An individual value prediction: Prediction value is written to a simple variable or
- ▶ A value list: Prediction values are written to an array of simple variables

## 4.1 PA\_Drv - Datenblatt

General:	
Driver file name	PA_Drv.exe
Driver name	Predictive Analytics Engine driver
PLC types	Predictive Analytics Engine
PLC manufacturer	COPA-DATA; zenon system driver

Driver supports:	
Protocol	MQTT
Addressing: Address-based	Name based
Addressing: Name-based	Symbolic Address
Spontaneous communication	X
Polling communication	X
Online browsing	X
Offline browsing	--
Real-time capable	X
Blockwrite	--
Modem capable	--
RDA numerical	--
RDA String	--
Hysteresis	X
extended API	X
Supports status bit <b>WR-SUC</b>	X

Driver supports:	
alternative IP address	--

Requirements:	
Hardware PC	Ethernet Port
Software PC	--
Hardware PLC	--
Software PLC	--
Requires v-dll	--

Platforms:	
Operating systems	Windows 10; Windows 7; Windows 8; Windows 8.1; Windows Server 2008 R2; Windows Server 2012; Windows Server 2012 R2; Windows Server 2016

## 4.2 Driver history

Date	Build number	Change
13.12.18	53630	Created driver documentation

## 4.3 Requirements

This chapter contains information on the requirements that are necessary for use of this driver.

### 4.3.1 PC

**Certificate Bundles** must be present for the connection to the **Service Hub**.

### 4.3.2 PLC

At least 1 prediction model must be available in a database on the Analyzer server. If spontaneous communication is required, there must be at least 1 schedule for at least 1 prediction model. Furthermore, the Analyzer server must be connected to a service hub.

Prediction models can be configured in the **Prediction Model Manager** of **zenon Analyzer**. The schedules can be configured in the zenon Analyzer Management Studio, whereby this must be started on the same computer as the Analyzer server.

## 4.4 Configuration

In this chapter you will learn how to use the driver in a project and which settings you can change.



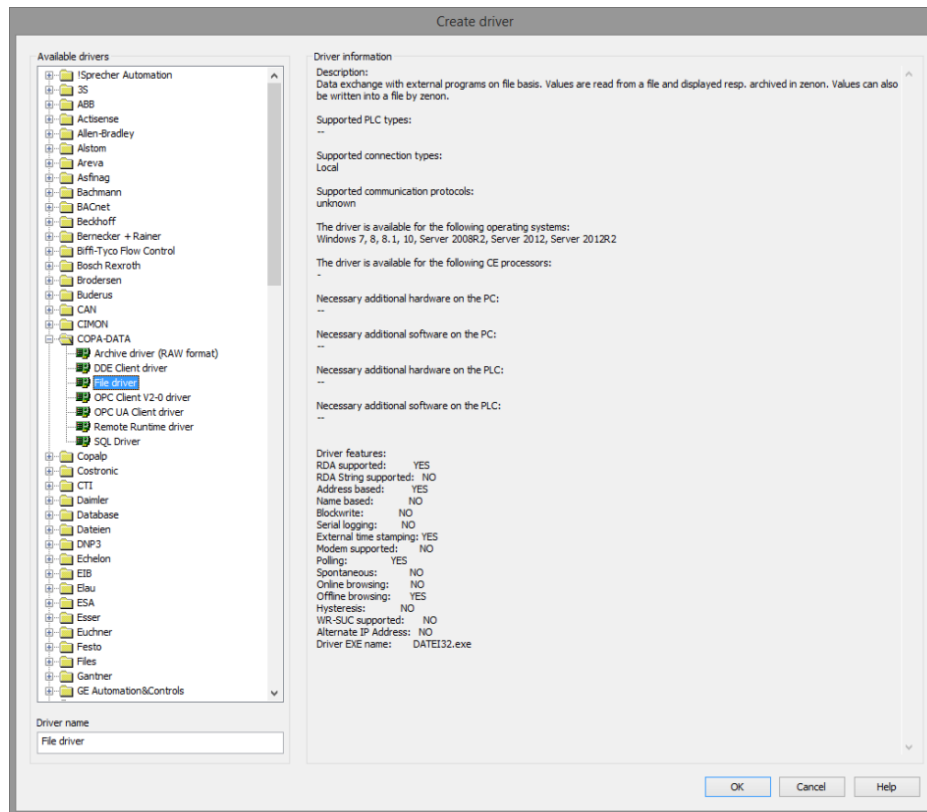
### Information

Find out more about further settings for zenon variables in the chapter Variables (main.chm::/15247.htm) of the online manual.



## 4.4.1 Creating a driver

In the **Create driver** dialog, you create a list of the new drivers that you want to create.



Parameter	Description
<b>Available drivers</b>	<p>List of all available drivers.</p> <p>The display is in a tree structure:  <code>[+]</code> expands the folder structure and shows the drivers contained therein.  <code>[-]</code> reduces the folder structure</p> <p>Default: <i>no selection</i></p>
<b>Driver name</b>	<p>Unique <b>Identification</b> of the driver.</p> <p>Default: <i>Empty</i></p> <p>The input field is pre-filled with the pre-defined <b>Identification</b> after selecting a driver from the list of available drivers.</p>
<b>Driver information</b>	<p>Further information on the selected driver.</p> <p>Default: <i>Empty</i></p> <p>The information on the selected driver is shown in</p>

Parameter	Description
	this area after selecting a driver.

## CLOSE DIALOG

Option	Description
OK	Accepts all settings and opens the driver configuration dialog of the selected driver.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.



### Information

The content of this dialog is saved in the file called Treiber\_[Language].xml. You can find this file in the following folder:

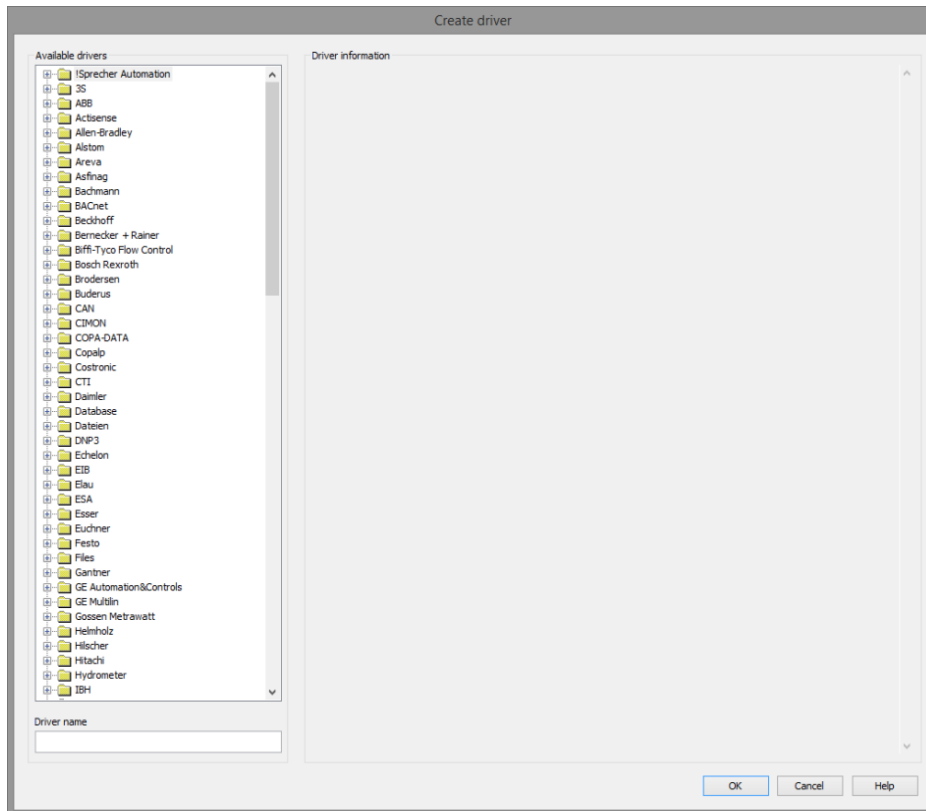
*C:\ProgramData\COPA-DATA\zenon[version number].*

## CREATE NEW DRIVER

In order to create a new driver:

1. Right-click on **Driver** in the Project Manager and select **New driver** in the context menu.  
Optional: Select the **New driver** button from the toolbar of the detail view of the **Variables**.  
The **Create driver** dialog is opened.

2. The dialog offers a list of all available drivers.

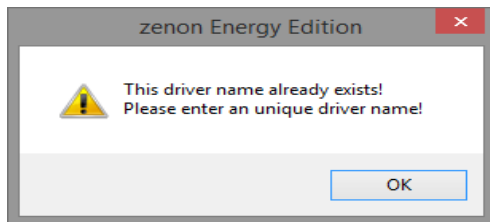


3. Select the desired driver and name it in the **Driver name** input field.  
This input field corresponds to the **Identification** property. The name of the selected driver is automatically inserted into this input field by default.  
The following is applicable for the **Driver name**:
  - ▶ The **Driver name** must be unique.  
If a driver is used more than once in a project, a new name has to be given each time.  
This is evaluated by clicking on the **OK** button. If the driver is already present in the project, this is shown with a warning dialog.
  - ▶ The **Driver name** is part of the file name.  
Therefore it may only contain characters which are supported by the operating system.  
Invalid characters are replaced by an underscore (\_).
  - ▶ **Attention:** This name cannot be changed later on.
4. Confirm the dialog by clicking on the **OK** button.  
The configuration dialog for the selected driver is opened.

**Note:** The language of driver names cannot be switched. They are always shown in the language in which they have been created, regardless of the language of the Editor. This also applies to driver object types.

## DRIVER NAME DIALOG ALREADY EXISTS

If there is already a driver in the project, this is shown in a dialog. The warning dialog is closed by clicking on the **OK** button. The driver can be named correctly.



## ZENON PROJECT

The following drivers are created automatically for newly-created projects:

- ▶ **Intern**
- ▶ **MathDr32**
- ▶ **SysDrv**



### Information

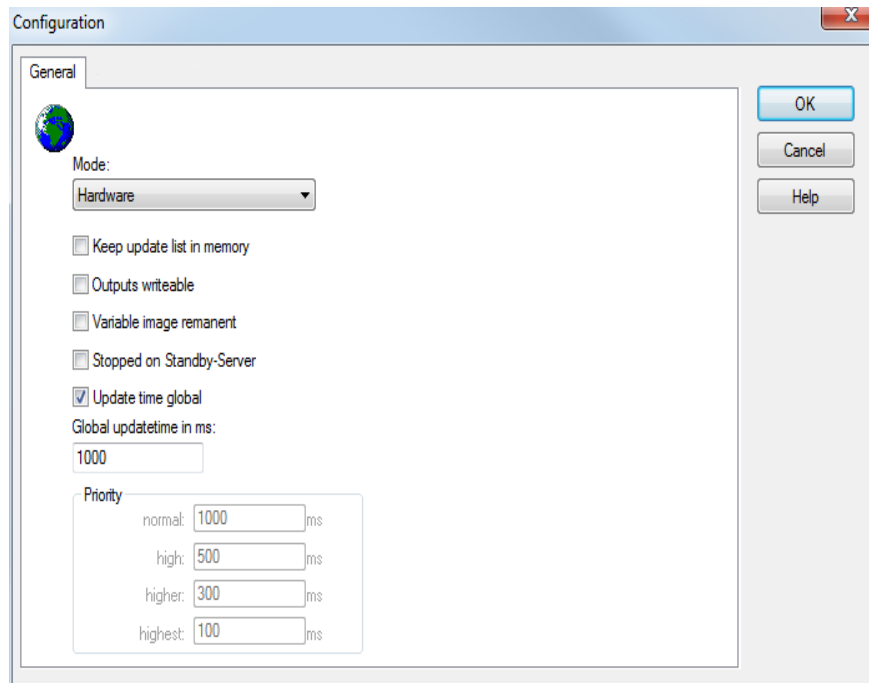
Only the required drivers need to be present in a zenon project. Drivers can be added at a later time if required.

## 4.4.2 Settings in the driver dialog

You can change the following settings of the driver:

### 4.4.2.1 General

The configuration dialog is opened when a driver is created. In order to be able to open the dialog later for editing, double click on the driver in the list or click on the **Configuration** property.



Option	Description
<b>Mode</b>	<p>Allows to switch between hardware mode and simulation mode</p> <ul style="list-style-type: none"> <li>▶ <i>Hardware:</i> A connection to the control is established.</li> <li>▶ <i>Simulation - static:</i> No communication between to the control is established, the values are simulated by the driver. In this modus the values remain constant or the variables keep the values which were set by zenon Logic. Each variable has its own memory area. E.g. two variables of the type marker with offset 79 can have different values in the Runtime and do not influence each other. Exception: The simulator driver.</li> <li>▶ <i>Simulation - counting:</i> No communication between to the control is established, the values are simulated by the driver. In this modus the driver increments the values</li> </ul>

Option	Description
	<p>within a value range automatically.</p> <ul style="list-style-type: none"> <li>▶ <i>Simulation - programmed:</i> No communication is established to the PLC. The values are calculated by a freely programmable simulation project. The simulation project is created with the help of the zenon Logic Workbench and runs in a zenon Logic Runtime which is integrated in the driver. For details see chapter Driver simulation (main.chm::/25206.htm).</li> </ul>
<b>Keep update list in the memory</b>	<p>Variables which were requested once are still requested from the control even if they are currently not needed. This has the advantage that e.g. multiple screen switches after the screen was opened for the first time are executed faster because the variables need not be requested again. The disadvantage is a higher load for the communication to the control.</p>
<b>Output can be written</b>	<ul style="list-style-type: none"> <li>▶ <i>Active:</i> Outputs can be written.</li> <li>▶ <i>Inactive:</i> Writing of outputs is prevented.</li> </ul> <p><b>Note:</b> Not available for every driver.</p>
<b>Variable image remanent</b>	<p>This option saves and restores the current value, time stamp and the states of a data point.</p> <p>Fundamental requirement: The variable must have a valid value and time stamp.</p> <p>The variable image is saved in hardware mode if one of these statuses is active:</p> <ul style="list-style-type: none"> <li>▶ User status <i>M1 (0)</i> to <i>M8 (7)</i></li> <li>▶ <i>REVISION(9)</i></li> <li>▶ <i>AUS(20)</i></li> <li>▶ <i>ERSATZWERT(27)</i></li> </ul> <p>The variable image is always saved if:</p> <ul style="list-style-type: none"> <li>▶ the variable is of the object type <b>Driver variable</b></li> <li>▶ the driver runs in simulation mode. (not</li> </ul>

Option	Description
	<p>programmed simulation)</p> <p>The following states are not restored at the start of the Runtime:</p> <ul style="list-style-type: none"> <li>▶ <i>SELECT(8)</i></li> <li>▶ <i>WR-ACK(40)</i></li> <li>▶ <i>WR-SUC(41)</i></li> </ul> <p>The mode <b>Simulation - programmed</b> at the driver start is not a criterion in order to restore the remanent variable image.</p>
<b>Stop on Standby Server</b>	<p>Setting for redundancy at drivers which allow only one communication connection. For this the driver is stopped at the Standby Server and only started at the upgrade.</p> <p><b>Attention:</b> If this option is active, the gapless archiving is no longer guaranteed.</p> <ul style="list-style-type: none"> <li>▶ <i>Active:</i> Sets the driver at the not-process-leading Server automatically in a stop-like state. In contrast to stopping via driver command, the variable does not receive status <b>switched off</b> (<b>statusverarbeitung.chm::/24150.htm</b>) but an empty value. This prevents that at the upgrade to the Server irrelevant values are created in the AML, CEL and Historian.</li> </ul> <p>Default: <i>inactive</i></p> <p><b>Note:</b> Not available if the CE terminal serves as a data server. You can find further information in the zenon Operator manual in the CE terminal as a data server chapter.</p>
<b>Global Update time</b>	<p>Setting for the global update times in milliseconds:</p> <ul style="list-style-type: none"> <li>▶ <i>Active:</i> The set <b>Global update time</b> is used for all variables in the project. The priority set at the variables is not used.</li> <li>▶ <i>Inactive:</i> The set priorities are used for the individual variables.</li> </ul>

Option	Description
	<b>Exceptions:</b> Spontaneous drivers ignore this option. They generally use the shortest possible update time. For details, see the <b>Spontaneous driver update time</b> section.
<b>Priority</b>	<p>The polling times for the individual priority classes are set here. All variables with the according priority are polled in the set time.</p> <p>The variables are allocated separately in the settings of the variable properties.</p> <p>The communication of the individual variables can be graded according to importance or required topicality using the priority classes. Thus the communication load is distributed better.</p> <p><b>Attention:</b> Priority classes are not supported by each driver, e.g. spontaneously communicating zenon drivers.</p>

## CLOSE DIALOG

Option	Description
<b>OK</b>	<b>Applies all changes in all tabs and closes the dialog.</b>
<b>Cancel</b>	Discards all changes in all tabs and closes the dialog.
<b>Help</b>	Opens online help.

## UPDATE TIME FOR SPONTANEOUS DRIVERS

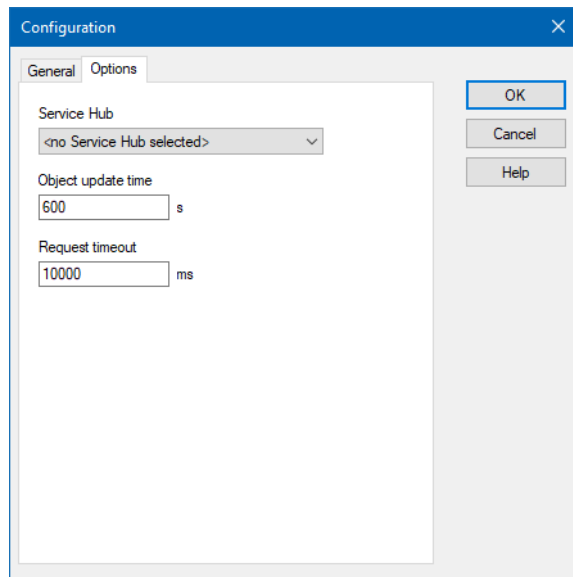
With spontaneous drivers, for **Set value, advising** of variables and **Requests**, a read cycle is triggered immediately - regardless of the set update time. This ensures that the value is immediately available for visualization after writing. The update time is generally 100 ms.

Spontaneous drivers are **ArchDrv**, **BiffiDCM**, **BrTcp32**, **DNP3**, **Esser32**, **FipDrv32**, **FpcDrv32**, **IEC850**, **IEC870**, **IEC870\_103**, **Otis**, **RTK9000**, **S7DCOS**, **SAIA\_Slave**, **STRATON32** and **Trend32**.



### 4.4.2.2 Options

Configuration of the connection settings to the **Service Hub**:



Option	Description
<b>Service Hub</b>	<p>Selection of the connection to the <b>Service Hub</b> from a drop-down list.</p> <p>All connections that have been created with the <b>Service Node Configuration Tool</b> and for which there is a <b>Certificate Bundle</b> are offered:</p> <ul style="list-style-type: none"> <li>▶ <i>&lt;no Service Hub selected&gt;</i>: Empty string, no selection.</li> <li>▶ <i>&lt;configured connections&gt;</i>: List of all available connections.</li> </ul> <p>Default: <i>&lt;no Service Hub selected&gt;</i></p> <p>For access via API, the connection information is entered directly.</p>
<b>Object Update Time</b>	<p>Entry of the object refresh time in seconds.</p> <p><b>Behavior:</b></p> <ul style="list-style-type: none"> <li>▶ If the databases of an Analyzer Server have been read successfully, there is a wait for this time until the existence of the database is checked again with a further call.</li> <li>▶ If the prediction model and schedules of a database have been read successfully, there is a</li> </ul>

Option	Description
	<p>wait for this time until the existence of the objects and their properties have been checked with a further call.</p> <p><b><u>Input time period:</u></b></p> <ul style="list-style-type: none"> <li>▶ Minimum: 60 seconds (1 minute)</li> <li>▶ Maximum: 86400 seconds (1 day)</li> </ul> <p>Default:600 s</p>
<b>Request Timeout</b>	<p>Entry of the query timeout for polling and object updates in milliseconds:</p> <ul style="list-style-type: none"> <li>▶ Polling is aborted after this time if there is no response.</li> <li>▶ Object refreshes are aborted after this time if no response is received and then started again afterwards once this time has elapsed again.</li> </ul> <p>Possible values:</p> <ul style="list-style-type: none"> <li>▶ Minimum: 1000 (1 second)</li> <li>▶ Maximum: 3600000 (1 hour)</li> </ul> <p>Default:10000 (10 seconds).</p>

#### CLOSE DIALOG

Options	Description
<b>OK</b>	<b>Applies settings and closes the dialog.</b>
<b>Cancel</b>	Discards all changes and closes the dialog.
<b>Help</b>	Opens online help.

## 4.5 Creating variables

This is how you can create variables in the zenon Editor:

### 4.5.1 Creating variables in the Editor

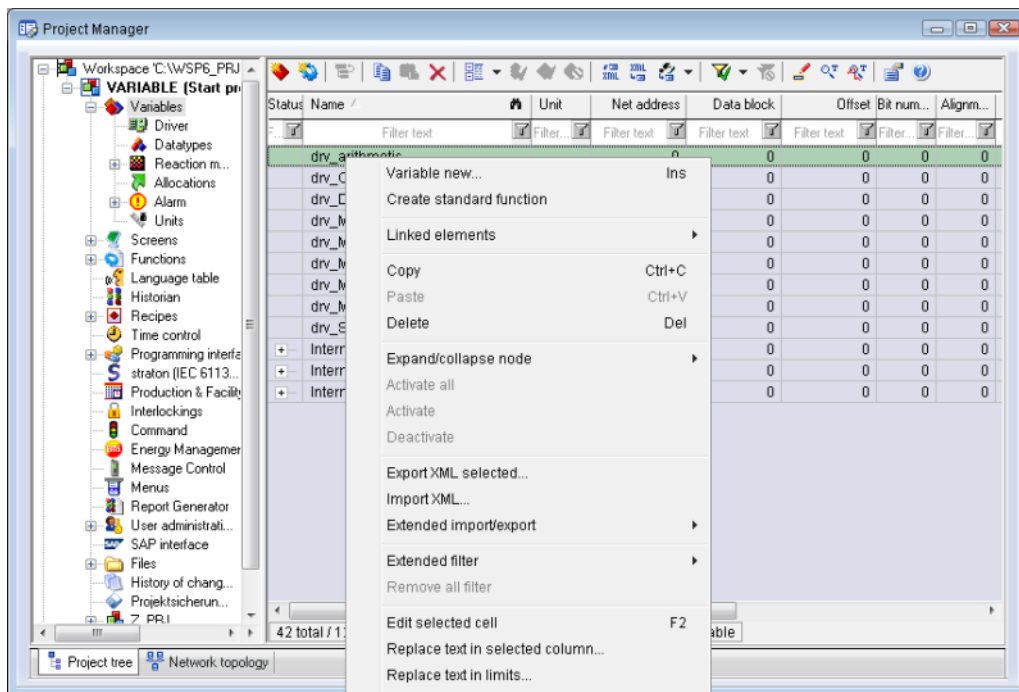
Variables can be created:

- ▶ as simple variables
- ▶ in arrays (main.chm::/15262.htm)
- ▶ as structure variables (main.chm::/15278.htm)

## VARIABLE DIALOG

To create a new variable, regardless of which type:

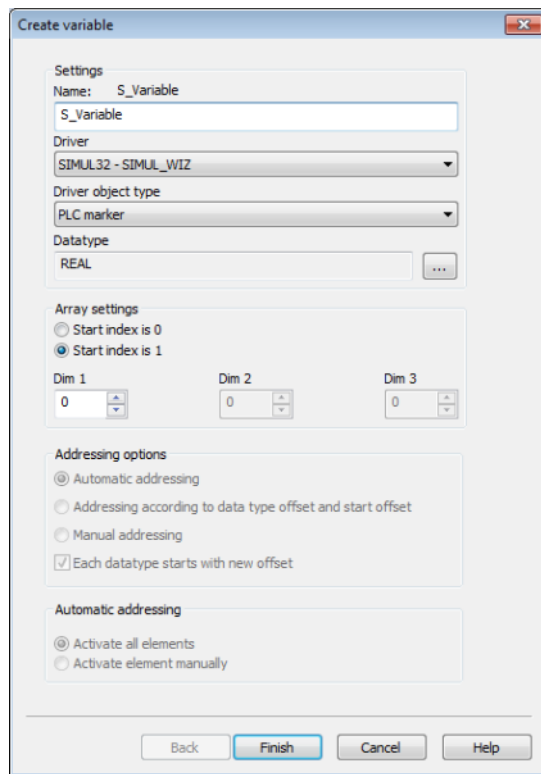
1. Select the **New variable** command in the **Variables** node in the context menu



The dialog for configuring variables is opened

2. Configure the variable
3. The settings that are possible depends on the type of variables

## CREATE VARIABLE DIALOG



Property	Description
<b>Name</b>	<p>Distinct name of the variable. If a variable with the same name already exists in the project, no additional variable can be created with this name.</p> <p>Maximum length: 128 characters</p> <p><b>Attention:</b> the characters # and @ are not permitted in variable names. If non-permitted characters are used, creation of variables cannot be completed and the <b>Finish</b> button remains inactive.</p> <p><b>Note:</b> For some drivers, the addressing is possible over the property <b>Symbolic address</b>, as well.</p>
<b>Drivers</b>	<p>Select the desired driver from the drop-down list.</p> <p><b>Note:</b> If no driver has been opened in the project, the driver for internal variables (<b>Intern.exe (Main.chm::/Intern.chm::/Intern.htm)</b>) is automatically loaded.</p>
<b>Driver Object Type</b> (cti.chm::/28685.htm)	Select the appropriate driver object type from the drop-down list.
<b>Data Type</b>	Select the desired data type. Click on the ... button to open the

Property	Description
	selection dialog.
<b>Array settings</b>	Expanded settings for array variables. You can find details in the Arrays chapter.
<b>Addressing options</b>	Expanded settings for arrays and structure variables. You can find details in the respective section.
<b>Automatic addressing</b>	Expanded settings for arrays and structure variables. You can find details in the respective section.

## SYMBOLIC ADDRESS

The **Symbolic address** property can be used for addressing as an alternative to the **Name** or **Identification** of the variables. Selection is made in the driver dialog; configuration is carried out in the variable property. When importing variables of supported drivers, the property is entered automatically.

Maximum length: 1024 characters.

The following drivers support the **Symbolic address**:

- ▶ 3S\_V3
- ▶ AzureDrv
- ▶ BACnetNG
- ▶ IEC850
- ▶ KabaDPsServer
- ▶ OPCUA32
- ▶ Phoenix32
- ▶ POZYTON
- ▶ RemoteRT
- ▶ S7TIA
- ▶ SEL
- ▶ SnmpNg32
- ▶ PA\_Drv

## INHERITANCE FROM DATA TYPE

**Measuring range**, **Signal range** and **Set value** are always:

- ▶ derived from the datatype
- ▶ Automatically adapted if the data type is changed

**Note for signal range:** If a change is made to a data type that does not support the set **signal range**, the **signal range** is amended automatically. For example, for a change from **INT** to **SINT**, the **signal range** is changed to 127. The amendment is also carried out if the **signal range** was not inherited from the data type. In this case, the **measuring range** must be adapted manually.

## 4.5.2 Addressing

Group/Property	Description
<b>General</b>	
<b>Name</b>	<p>Freely definable name.</p> <p><b>Attention:</b> For every zenon project the name must be unambiguous.</p>
<b>Identification</b>	Freely assignable identification, e.g. for resources label, comment ...
<b>Addressing</b>	<p>Via symbolic address.</p> <p>Syntax: <i>[Identifier for communication type]::[Identifier for model type]::[Analyzer Server ID]::[Database name]::[Model ID]::[Identification]::[Name of the input parameter]</i></p> <p>Description:</p> <ul style="list-style-type: none"> <li>▶ <b>Identifier for communication type:</b> <ul style="list-style-type: none"> <li>▶ <i>S</i>: Spontaneous communication. Only initial values possible. Initial values must be <i>LREAL</i> type.</li> <li>▶ <i>T</i>: Polling communication. Triggered. Initial values must be <i>LREAL</i> type. Input values must be <i>LREAL</i> or <i>DATE_AND_TIME</i> depending on the prediction model used. Trigger variables must be <i>BOOL</i> type.</li> </ul> </li> <li>▶ <b>Identifier for model type:</b> <ul style="list-style-type: none"> <li>▶ <i>T</i>: Time based. Time stamp for which the prediction is created. Is written as an external time stamp of the variable.</li> <li>▶ <i>V</i>: Value-based.</li> </ul> </li> <li>▶ <b>Analyzer Server ID:</b> Unique GUID of the Analyzer server.</li> <li>▶ <b>Database name:</b> Name of the database in which the prediction model is saved.</li> <li>▶ <b>Model ID:</b> Unique ID of the prediction model in the database.</li> </ul>

Group/Property	Description
	<ul style="list-style-type: none"> <li>▶ <b>Identification:</b> Is used depending on communication type: <ul style="list-style-type: none"> <li>▶ <i>Spontaneous:</i> Unique GUID of the schedule on the Analyzer server.</li> <li>▶ <i>Polling:</i> Unique GUID in order to bundle variables for predicted value, input value and controller into a logical structure.</li> </ul> </li> <li>▶ <b>Name of the input parameter:</b> Is only present if the variable is an input value for <i>polling</i>. The following input values are needed for predicted value: <ul style="list-style-type: none"> <li>▶ Simple variables: <b>Input</b> for individual value prediction.</li> <li>▶ Array of simple variables: <b>From</b> and <b>To</b> for value range prediction. The array length determines the number of values in the process. The input values for the prediction models are distributed evenly over the range between <b>From</b> and <b>To</b>.</li> </ul> </li> </ul> <p><b>Recommendation:</b> Use online import to create variables. Background: The <b>Symbolic address</b> primarily uses information from the Analyzer servers and has a complex structure.</p>
Net address	not used for this driver
Data block	not used for this driver
Offset	not used for this driver
Alignment	not used for this driver
Bit number	not used for this driver
String length	not used for this driver
Driver connection/Driver Object Type	<p>Available driver object types:</p> <ul style="list-style-type: none"> <li>▶ <b>Predicted Value:</b> Predicted value (spontaneous and polling).</li> <li>▶ <b>Prediction Input Value:</b> Input value for polling predictions.</li> <li>▶ <b>Prediction Trigger:</b> Control variable for polling forecasts.</li> </ul> <p><b>Note:</b> You can find a precise description of the events in the <b>driver objects</b> (on page 32) chapter.</p>
Driver connection/Data	<p>The driver supports the <i>BOOL</i>, <i>LREAL</i> and <i>DATE_AND_TIME</i> data types.</p> <p><b>Attention:</b> If you change the data type later, all other properties of the</p>

Group/Property	Description
Type	variable must be checked or adjusted, if necessary.

### 4.5.3 Driver objects and datatypes

Driver objects are areas available in the PLC, such as markers, data blocks etc. Here you can find out which driver objects are provided by the driver and which IEC data types can be assigned to the respective driver objects.

#### 4.5.3.1 Driver objects

The following object types are available in this driver:

Driver Object Type	Channel type	Read	Write	Supported data types	Description
<b>Predicted Value</b>	8	x	--	<i>LREAL</i>	<p>Variables that receive the predicted values.</p> <ul style="list-style-type: none"> <li>▶ Simple variable with polling communication: With <b>Input</b> input values, precisely 1 value is obtained.</li> <li>▶ Simple variable with spontaneous communication: The schedule on the Analyzer server must be set with an input value for precisely 1 value.</li> <li>▶ Array of simple variables with polling communication: With the input values <b>From</b> and <b>To</b>, a range prediction is obtained with the array length as the number of values. On the Analyzer server, the predicted values for the set number of</li> </ul>



Driver Object Type	Channel type	Read	Write	Supported data types	Description
					<p>values distributed evenly over the range are determined.</p> <ul style="list-style-type: none"> <li>▶ Array of simple variables with spontaneous communication: The number of values in the schedule on the Analyzer server must correspond to the number of variables in the array. The range of values must also be given. On the Analyzer server, the predicted values for the set number of values distributed evenly over the range are determined.</li> <li>▶ Value-based prediction model The internal time stamp is set to the time of the arrival of the prediction value. The external time stamp remains empty.</li> <li>▶ Time-based prediction model The internal time stamp is set to the time of the arrival of the predicted value. The external time stamp is to set the time for the predicted. And the time stamp external status bit is set.</li> </ul>
<b>Prediction Input Value</b>	64	x	x	<i>DATE_AND_TIME, LREAL</i>	Input variable for polling prediction. If the model is time

Driver Object Type	Channel type	Read	Write	Supported data types	Description
					<p>based, the data type must be <i>DATE_AND_TIME</i>. If the model is value based, the data type must be <i>LREAL</i>.</p> <p>Reading is only supported for status changes here.</p> <p>The name of the parameter is specified in <b>Symbolic address</b>. Possible names are:</p> <ul style="list-style-type: none"> <li>▶ <b>Input:</b> Input value for individual value prediction.</li> <li>▶ <b>From:</b> Lower limit for input value for range prediction.</li> <li>▶ <b>To:</b> Upper limit for input value for range predictions.</li> </ul>
<b>Prediction Trigger</b>	10	x	x	<i>BOOL</i>	<p>Control variable for polling communication.</p> <p>If the variable is set from 0 to 1 by Runtime, polling is started. If the polling has ended, (regardless of whether due to loss of connection, error or arrival of value), the variable is set back to 0 by the driver.</p>
<i>Communication details</i>	35	x	x	<i>BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING</i>	<p>Variables for the static analysis of the communication; Is transferred between driver and the Runtime. (Not to the PLC!)</p> <p><b>Note:</b> The addressing and the behavior is the same for most zenon drivers.</p> <p>You can find detailed</p>

Driver Object Type	Channel type	Read	Write	Supported data types	Description
					information on this in the Communication details (Driver variables) (on page 54) chapter.

**Key:**

X: supported

--: not supported

### 4.5.3.2 Mapping of the data types

All variables in zenon are derived from IEC data types. The following table compares the IEC datatypes with the datatypes of the PLC.

PLC	zenon	Data type
Not available Pure control variable on the driver.	<i>BOOL</i>	8
Prediction value for time-based or value-based prediction model or Input value for value-based prediction model	<i>LREAL</i>	6
Input value for time-based prediction model	<i>DATE_AND_TIME</i>	20

## DATA TYPE

The term **data type** is the internal numerical identification of the data type. It is also used for the extended DBF import/export of the variables.

### 4.5.4 Creating variables by importing

Variables can also be imported by importing them. The XML and DBF import is available for every driver.



#### Information

You can find details on the import and export of variables in the Import-Export (main.chm::/13028.htm) manual in the Variables (main.chm::/13045.htm) section.

#### 4.5.4.1 XML import

During XML import of variables or data types, these are first assigned to a driver and then analyzed. Before import, the user decides whether and how the respective element (variable or data type) is to be imported:

- ▶ *Import:*  
The element is imported as a new element.
- ▶ *Overwrite:*  
The element is imported and overwrites a pre-existing element.
- ▶ *Do not import:*  
The element is not imported.

**Note:** The actions and their durations are shown in a progress bar during import. The import of variables is described in the following documentation. Data types are imported along the same lines.

### REQUIREMENTS

The following conditions are applicable during import:

- ▶ **Backward compatibility**  
At the XML import/export there is no backward compatibility. Data from older zenon versions can be taken over. The handover of data from newer to older versions is not supported.
- ▶ **Consistency**  
The XML file to be imported has to be consistent. There is no plausibility check on importing the file. If there are errors in the import file, this can lead to undesirable effects in the project.  
Particular attention must be paid to this, primarily if not all properties exist in the XML file and these are then filled with default values. E.g.: A binary variable has a limit value of 300.
- ▶ **Structure data types**  
Structure data types must have the same number of structure elements.  
Example: A structure data type in the project has 3 structure elements. A data type with the same name in the XML file has 4 structure elements. Then none of the variables based on this data type in the file are imported into the project.



#### Hint

You can find further information on XML import in the **Import - Export** manual, in the **XML import (main.chm::/13046.htm)** chapter.

#### 4.5.4.2 DBF Import/Export

Data can be exported to and imported from dBase.



##### Information

Import and Export via CSV or dBase supported; no driver specific variable settings, such as formulas. Use export/import via XML for this.

#### IMPORT DBF FILE

To start the import:

1. right-click on the variable list
2. in the drop-down list of **Extended export/import...** select the **Import dBase** command
3. follow the import assistant

The format of the file is described in the chapter File structure.



##### Information

Note:

- ▶ Driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.
- ▶ dBase does not support structures or arrays (complex variables) at import.

#### EXPORT DBF FILE

To start the export:

1. right-click on the variable list
2. in the drop-down list of **Extended export/import...** select the **Export dBase...** command
3. follow the export assistant



### Attention

DBF files:

- ▶ must correspond to the 8.3 DOS format for filenames (8 alphanumeric characters for name, 3 character suffix, no spaces)
- ▶ must not have dots (.) in the path name.  
e.g. the path `C:\users\John.Smith\test.dbf` is invalid.  
Valid: `C:\users\JohnSmith\test.dbf`
- ▶ must be stored close to the root directory in order to fulfill the limit for file name length including path: maximum 255 characters

The format of the file is described in the chapter File structure.



### Information

dBase does not support structures or arrays (complex variables) at export.

## FILE STRUCTURE OF THE DBASE EXPORT FILE

The dBaseIV file must have the following structure and contents for variable import and export:



### Attention

dBase does not support structures or arrays (complex variables) at export.

DBF files must:

- ▶ conform with their name to the 8.3 DOS format (8 alphanumeric characters for name, 3 characters for extension, no space)
- ▶ Be stored close to the root directory (Root)

## STRUCTURE

Identification	Type	Field size	Comment
KANALNAME	Character	128	Variable name.  The length can be limited using the <b>MAX_LAENGE</b> entry in the <b>project.ini</b> file.
KANAL_R	C	128	The original name of a variable that is to be replaced by the new name entered under "VARIABLENNAME" (variable name) (field/column must be entered

Identification	Type	Field size	Comment
			manually).  The length can be limited using the <b>MAX_LAENGE</b> entry in the <b>project.ini</b> file.
<b>KANAL_D</b>	Log	1	The variable is deleted with the 1 entry (field/column has to be created by hand).
<b>TAGNR</b>	C	128	Identification.  The length can be limited using the <b>MAX_LAENGE</b> entry in the <b>project.ini</b> file.
<b>EINHEIT</b>	C	11	Technical unit
<b>DATENART</b>	C	3	Data type (e.g. bit, byte, word, ...) corresponds to the data type.
<b>KANALTYP</b>	C	3	Memory area in the PLC (e.g. marker area, data area, ...) corresponds to the driver object type.
<b>HWKANAL</b>	Num	3	Net address
<b>BAUSTEIN</b>	N	3	Datablock address (only for variables from the data area of the PLC)
<b>ADRESSE</b>	N	5	Offset
<b>BITADR</b>	N	2	For bit variables: bit address For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters)
<b>ARRAYSIZE</b>	N	16	Number of variables in the array for index variables ATTENTION: Only the first variable is fully available. All others are only available for VBA or the Recipegroup Manager
<b>LES_SCHR</b>	L	1	Write-Read-Authorization 0: Not allowed to set value. 1: Allowed to set value.
<b>MIT_ZEIT</b>	R	1	time stamp in zenon (only if supported by the driver)
<b>OBJEKT</b>	N	2	Driver-specific ID number of the primitive object comprises TREIBER-OBJEKTTYP and DATENTYP

Identification	Type	Field size	Comment
<b>SIGMIN</b>	Float	16	Non-linearized signal - minimum (signal resolution)
<b>SIGMAX</b>	F	16	Non-linearized signal - maximum (signal resolution)
<b>ANZMIN</b>	F	16	Technical value - minimum (measuring range)
<b>ANZMAX</b>	F	16	Technical value - maximum (measuring range)
<b>ANZKOMMA</b>	N	1	Number of decimal places for the display of the values (measuring range)
<b>UPDATERATE</b>	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables
<b>MENTIEFE</b>	N	7	Only for compatibility reasons
<b>HDRATE</b>	F	19	HD update rate for historical values (in sec, one decimal possible)
<b>HDTIEFE</b>	N	7	HD entry depth for historical values (number)
<b>NACHSORT</b>	R	1	HD data as postsorted values
<b>DRRATE</b>	F	19	Updating to the output (for zenon DDE server, in [s], one decimal possible)
<b>HYST_PLUS</b>	F	16	Positive hysteresis, from measuring range
<b>HYST_MINUS</b>	F	16	Negative hysteresis, from measuring range
<b>PRIOR</b>	N	16	Priority of the variable
<b>REAMATRIZE</b>	C	32	Allocated reaction matrix
<b>ERSATZWERT</b>	F	16	Substitute value, from measuring range
<b>SOLLMIN</b>	F	16	Minimum for set value actions, from measuring range
<b>SOLLMAX</b>	F	16	Maximum for set value actions, from measuring range
<b>VOMSTANDBY</b>	R	1	Get value from standby server; the value of the variable is not requested from the server but from the Standby Server in redundant networks
<b>RESOURCE</b>	C	128	Resources label. Free string for export and display in lists.



Identification	Type	Field size	Comment
			The length can be limited using the MAX_LAENGE entry in <b>project.ini</b> .
ADJWVBA	R	1	Non-linear value adaption: 0: Non-linear value adaption is used 1: Non-linear value adaption is not used
ADJZENON	C	128	Linked VBA macro for reading the variable value for non-linear value adjustment.
ADJWVBA	C	128	ed VBA macro for writing the variable value for non-linear value adjustment.
ZWREMA	N	16	Linked counter REMA.
MAXGRAD	N	16	Gradient overflow for counter REMA.



### Attention

When importing, the driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.

## LIMIT VALUE DEFINITION

Limit definition for limit values 1 to 4, or status 1 to 4:

Identification	Type	Field size	Comment
AKTIV1	R	1	Limit value active (per limit value available)
GRENZWERT1	F	20	technical value or ID number of a linked variable for a dynamic limit value (see VARIABLEx) (if VARIABLEx is 1 and here it is -1, the existing variable linkage is not overwritten)
SCHWWERT1	F	16	Threshold value for limit value
HYSTERESE1	F	14	Is not used
BLINKEN1	R	1	Set blink attribute
BTB1	R	1	Logging in CEL
ALARM1	R	1	Alarm

Identification	Type	Field size	Comment
<b>DRUCKEN1</b>	R	1	Printer output (for CEL or Alarm)
<b>QUITTIER1</b>	R	1	Must be acknowledged
<b>LOESCHE1</b>	R	1	Must be deleted
<b>VARIABLE1</b>	R	1	Dyn. limit value linking the limit is defined by an absolute value (see field GRENZWERTx).
<b>FUNC1</b>	R	1	Functions linking
<b>ASK_FUNC1</b>	R	1	Execution via Alarm Message List
<b>FUNC_NR1</b>	N	10	ID number of the linked function (if "-1" is entered here, the existing function is not overwritten during import)
<b>A_GRUPPE1</b>	N	10	Alarm/Event Group
<b>A_KLASSE1</b>	N	10	Alarm/Event Class
<b>MIN_MAX1</b>	C	3	Minimum, Maximum
<b>FARBE1</b>	N	10	Color as Windows coding
<b>GRENZTXT1</b>	C	66	Limit value text
<b>A_DELAY1</b>	N	10	Time delay
<b>INVISIBLE1</b>	R	1	Invisible

Expressions in the column "Comment" refer to the expressions used in the dialog boxes for the definition of variables. For more information, see chapter Variable definition.

#### 4.5.4.3 Online import

Online import is divided into 4 steps:

1. Preparation for the import.
2. Configuration of the variables to be imported in the dialog.
3. Checking of the entries in the dialog confirmation.
4. Execution of the import.

During each step, details on actions that have been carried out and errors that occurred can be read in the LOG.

## PREPARATION FOR THE IMPORT

To prepare the import:

1. The connection to the **Service Hub** is established.  
If an error occurs in the process, the online import is aborted with an error message.  
Errors are, for example, missing communication or a timeout when waiting for a response from the service hub.
2. Once the connection has been established, there is a search for available Analyzer servers.  
If, within the configured timeout, there is not at least 1 available Analyzer server found, the online import is canceled with an error message.
3. The necessary API editor objects are obtained.  
The following is read in the process:
  - ▶ The names of all variables that exist in the project.
  - ▶ The **Symbolic address** of all the variables that exist in the project for this driver.

This information is used for input validation in the import dialog. If an error occurs in the process, the online import is ended with an error. An error is, for example, if one of the necessary objects is not found.

The preparation is completed and the online variable import dialog is displayed.

## CONFIGURATION OF THE VARIABLES TO BE IMPORTED IN THE DIALOG

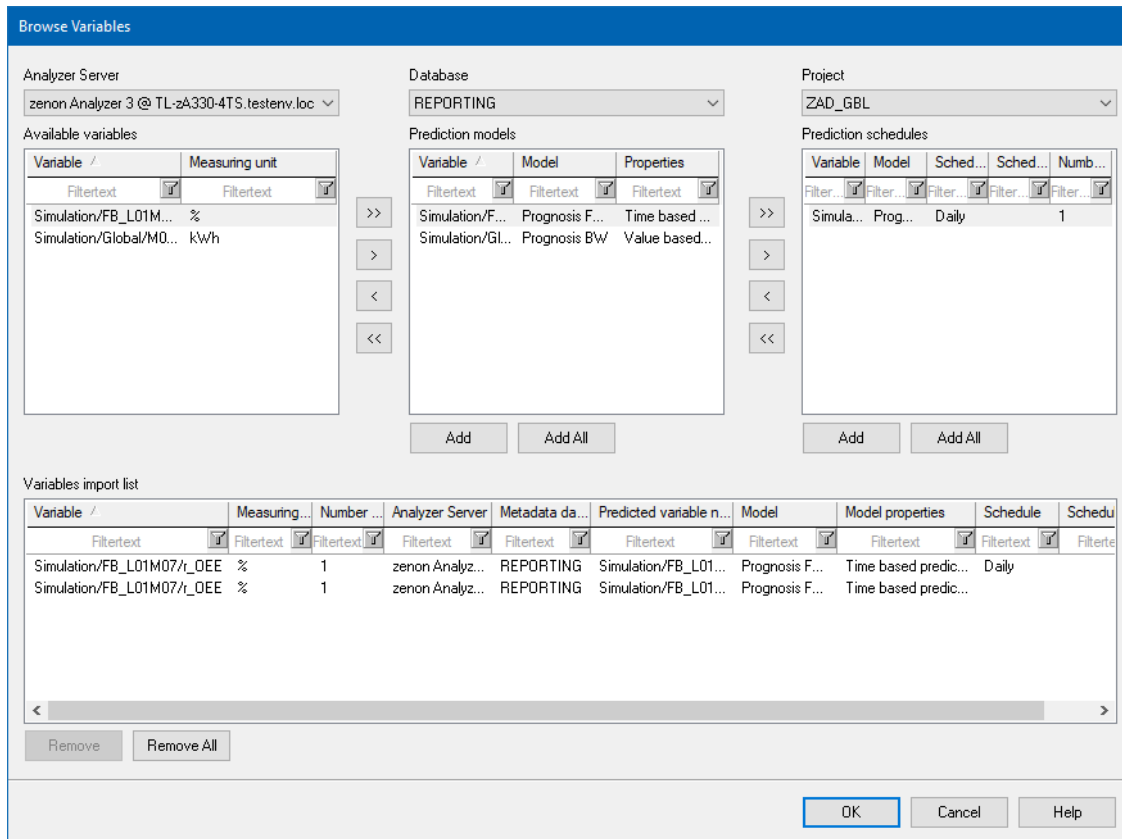
The import dialog allows the browsing of information to all available Analyzer servers.

Contents:

- ▶ Databases per Analyzer server
- ▶ Projects per database
- ▶ Variables with prediction models for each project
- ▶ Prediction model per variable
- ▶ Schedules for automated creation of predictions per prediction model

Variables for the driver can be derived and imported from the prediction models.

## BROWSE VARIABLES DIALOG



### This dialog consists of:

1. Pre-filter selection
2. List of project details plus buttons
3. Action symbols for list operations between the list views
4. List of variables to be imported
5. Buttons to close the dialog and start the help

## PRE-FILTER SELECTION

Selection list	Description
<b>Analyzer Server</b>	<p>This list contains all available Analyzer servers.</p> <p>Only Analyzer servers that are online are available. If the online status of an Analyzer server changes, this is inserted into the selection list or removed from it. If the Analyzer server that is currently selected switches to offline status, an error message is displayed and the server selection is reset to <i>None</i>. If the name of the Analyzer server that is currently</p>

Selection list	Description
	<p>selected changes, an error message is shown and the server selection is reset to <i>None</i>.</p> <p>After selecting an Analyzer server, the available databases of the selected Analyzer server are queried. The database selection list is updated and the selected database is reset to <i>None</i>.</p>
<b>Database</b>	<p>This selection list contains all databases that are available on the selected Analyzer server. The list can only be operated if an Analyzer server has been selected.</p> <p>After selecting a database, the database content is queried by the Analyzer server, the project selection list is updated and the project selection is reset to <i>None</i>.</p>
<b>Project</b>	<p>This list contains all projects contained in the selected database for which there is at least one prediction model. The list can only be operated if a database has been selected.</p> <p>After selecting a project, the list views for the project details are emptied and the variable view list is populated with all variables of the project that exist for the prediction model.</p>

## PROJECT DETAILS

The project details show all three filtering in each column in three views. They support multiple selection. All list views can only be operated if a project has been selected.

With all insertion operations, the filter of the list view for variables to be imported is reset so that newly-inserted variables to be imported are guaranteed to be visible. If an object to be inserted is already in the list view, a new object is inserted as a copy. As a result, different variables with different numbers of values can be created in an online import call.

To remove all variables to be imported that correspond to the filter from the list view:

1. Configure the filter.
2. Select all variables to be imported.
3. Click on the button to remove the selected variables.

## VARIABLES

Allows the selection of variables of the selected project:

Column	Description
Variable	Name of the variables in the zenon Analyzer database.
Measuring unit	Unit of the variables in the zenon Analyzer database.

## PREDICTION MODELS

Allows the selection of the prediction models:

Column	Description
Variable	Name of the variables in the zenon Analyzer database. <b>Note:</b> Prediction models for different variables can have the same name.
Model	Name of the prediction model. Is configured in the <b>Prediction Model Manager</b> of the zenon Analyzer.
Properties	Properties of the prediction model. Are configured in the <b>Prediction Model Manager</b> of the zenon Analyzer and contain: <ul style="list-style-type: none"> <li>▶ An indication of whether the prediction model is time based or value based.</li> <li>▶ Base cycle time of the prediction model if it is time based. It is the cycle time with which the variable is archived that is predicted.</li> </ul>
Button	Description
Add	Adds variables for all prediction models selected in the list view for prediction models for variables to be imported.  Only operable if at least one entry in the list has been selected.
Add all	Adds variables for all prediction models visible in the list view for prediction models for variables to be imported.  Only operable if at least one entry in the list is present.

## PREDICTION SCHEDULES

Allows the selection of schedules for automatic prediction:

Column	Description
<b>Variable</b>	Name of the variables in the zenon Analyzer database.  <b>Note:</b> Prediction models for different variables can have the same name.
<b>Model</b>	Name of the prediction model.  <b>Note:</b> Schedules for different prediction models can have the same name.
<b>Schedule</b>	Name of the schedule. Is configured in the <b>ZAMS</b> of the zenon Analyzer.
<b>Schedule description</b>	Description of the schedule. Is configured in the <b>ZAMS</b> of the zenon Analyzer.
<b>Number of values</b>	Number of values that are contained in each of the predictions triggered by this schedule. Is configured in the <b>ZAMS</b> of the zenon Analyzer.
<b>Button</b>	<b>Description</b>
<b>Add</b>	Adds variables for all schedules selected to the list view for schedules in the list view for variables to be imported.  Only operable if at least one entry in the list has been selected.
<b>Add all</b>	Adds variables for all schedules visible (that correspond to the filter) to the list view for schedules for variables to be imported.  Only operable if at least one entry in the list is present.

## ACTION SYMBOLS

Allows actions between the lists.

Symbols	Description
>>	Inserts sub-objects of all currently-displayed elements of the list view to the left of the button on the right in the list view.  Only operable if, in the list view left of the button, at least one element is displayed.
>	Adds sub-objects of all elements in the list view to the left

Symbols	Description
	of the button into the list view to the right of the button.  Only operable if, in the list view to the left of the button, at least one element is displayed.
<	Removes selected objects from the list view to the right of the button.  This button can only be operated if at least one element is selected in the list view to the right of the button.  Only operable if, in the list view to the right of the button, at least one element is displayed.
<<	Removes all objects from the list view to the right of the button. This is applicable regardless of whether they correspond to the filter or not.-

**Sub-objects are:**

- ▶ For a variable: Prediction models of the variables.
- ▶ For a prediction model: Schedules of the prediction model.

When inserting into a list view, all filters of the list view are reset in order for the newly-inserted objects to be guaranteed to be displayed in the list. If an object is already in the selection list in which it is to be inserted, it is not inserted again.

When removing prediction models, all schedules of the prediction models to be removed are also removed from the schedule list view.

In order to remove the objects of a list view that correspond to the filter:

1. Configure the filter.
2. Select all displayed objects.
3. Click on the symbol to remove all selected objects from the list view.

**LIST OF VARIABLES TO BE IMPORTED**

You can find details for editing and for additionally-created parameter and command variables at the end of the table and in the section about **execution of the import**.

**VARIABLES TO IMPORT**

Column	Description
Variable	Name of variables to be imported. The target variable is created with this name. Any parameter or command



Column	Description
	<p>variables that may be necessary receive a name that consists of the name set here and a suffix.</p> <p>With newly-inserted variables, the name of the variables is always entered into the zenon Analyzer database for which the prediction model was created (for which the schedule was created).</p> <p>The cells of this column can always be edited.</p>
<b>Measuring unit</b>	<p>Measuring unit of the variables to be imported. This is only used for target variables, not for parameter or command variables.</p> <p>With newly-inserted variables, the measuring unit of the variables is always entered into the zenon Analyzer database for which the prediction model was created (for which the schedule was created).</p> <p>The cells of this column can always be edited.</p>
<b>Number of values</b>	<p>Number of values for the target variable. This is only used for target variables, not for parameter or command variables.</p> <p>With newly-inserted variables, 1 is initially entered here if the variable has been created for a prediction model or the number of the values from the schedule if the variable has been created for a schedule.</p> <p>The cells of this column can only be edited for variables that have been created for prediction models.</p>
<b>Analyzer Server</b>	<p>Name of the zenon Analyzer instance from which the predictions come.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Metadata Database</b>	<p>Name of the database in which the prediction model is saved.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>

Column	Description
<b>Predicted variable name</b>	<p>Name of the variables in the database for which the prediction model has been created.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Model</b>	<p>Name of the prediction model used.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Model properties</b>	<p>Properties of the prediction model used.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Schedule</b>	<p>Name of the schedule used.</p> <p>If the variable has been created for a prediction model and not for a schedule, it remains empty.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Schedule description</b>	<p>Description of the schedule used.</p> <p>If the variable has been created for a prediction model and not for a schedule, it remains empty.</p> <p>This column is a column for information only. Its content is not relevant for the import and thus the cells of this column cannot be edited.</p>
<b>Button</b>	<b>Description</b>
<b>Remove</b>	<p>Removes all selected entries from the list view for variables to be imported.</p> <p>Only operable if at least one entry in the list has been selected.</p>
<b>Remove all</b>	<p>Removes all entries from the list view for variables to be imported.</p>

Column	Description
	Only operable if at least one entry in the list has been selected.

When clicking on a cell in this list, the cell is switched to edit mode, if the cell can be edited. Editing can be ended in different ways:

- ▶ Click on another element in the dialog: Editing is confirmed.
- ▶ Press the Enter key: Editing is confirmed.
- ▶ Press the Escape key: Editing is canceled and the previous value is restored.
- ▶ Press the Tab key: Editing is confirmed and the next editable cell is switched to edit mode. The next editable cell can be the cell to the right of the current cell ("**Measuring unit**" if "**Variable name**" is currently being edited; "**Number of values**" if "**Measuring unit**" is currently being edited and the variable has not been created for a schedule) or the "**Variable name**" column in the next line (if there is still a line underneath the cell currently being edited; if "**Measuring unit**" is currently being edited and the variable for a schedule has been created; if "**Number of values**" is currently being edited and the variable for a prediction model has been created).

## CLOSING THE DIALOG

When confirming the editing, the currently-entered value is applied if it is not empty and can also be converted into a figure for the "**Number of values**" column. Otherwise the previous value is restored.

### CLOSE DIALOG

Option	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

## CHECKING THE INPUTS WHEN CONFIRMING THE DIALOG

The entries are validated by clicking on **OK**.

The following error states can occur:

- ▶ Two or more variables in the list have the same name.  
Here, parameter and command variables to be explicitly created (see **executing the import** section) are taken into account.
- ▶ One or more of the variables present in the list has a name that has already been used for another variable in this project.

Here, parameter and command variables to be explicitly created (see **executing the import** section) are taken into account.

- ▶ Two or more variables present in the list have the same symbolic address.
  - ▶ With variables that are derived from schedules, this means that 2 or more variables should be created for the same schedule. This is an error, because several variables for the same schedule are not permitted. They would only provide redundant information.
  - ▶ Polling variables that are derived from a prediction model each receive a new unique GUID. With polling access, several variables are permitted for the same prediction model. This is because these different value figures can be executed with different input values.
- ▶ One or more of the variables in the list have a symbolic address that is already used for another variable for this driver instance in this project. The same detail indicators, such as two or more variables present in the list with the same symbolic address, are applicable.

If errors are discovered, notification is given by means of a pop-up message and the dialog is not closed. The error message contains information for each error category discovered. The names of the variables to be imported are contained in each paragraph. The dialog is only closed if no errors are discovered during this check.

Checks for clashes are always made without taking upper-case/lower-case letters into account. Only unique variable names are contained in the error message, but taking capitalization into account. It can therefore happen that several variable names are included in the error message where only the upper-case and lower-case letters differ.

## EXECUTION OF THE IMPORT

All necessary variables are created during the import.

In doing so, the following applies:

- ▶ Only the target variables are created for spontaneous communication.
- ▶ For polling variables, the target variable, the necessary input variables and the command variables are created.
- ▶ Target variables are always the *Predicted Value* driver object type and *LREAL* data type. They always contain the measuring unit set in the dialog, the variable name set in the dialog and the number of values set in the dialog:
  - ▶ 1 value: A simple variable is created.
  - ▶ More than 1 value: An array variable with the set number of values in dimension 1 is created.  
The array is always set for *Startindex 1*, automatic addressing and new offset for each data type. All elements of the array are activated. In doing so, all array content variables get the measuring unit and the symbolic address.
- ▶ Input variables are always *Prediction Input Value* driver object type.  
The data type depends on the model:

▶ *DATE\_AND\_TIME*, if the model is time based.

▶ *LREAL*, if the model is value based.

Variable names always have the format **[name of the target variables]\_[name of the parameters]**.

The unit of such variables always remains empty.

The symbolic address for these variables is always **[symbolic address of the target variables]::[name of the parameter]**.

The parameter variables that are created depend on the number of values of the target variables:

▶ 1 value: Input variable for the **Input** parameter.

▶ More than 1 value: Input variables for the parameters **From** and **To**.

▶ Command variables are always *Prediction Trigger* driver object type and *BOOL* data type.

The unit of such variables always remains empty.

They always receive their variable name as **[name of the target variables]\_command**.

The symbolic address is always identical to the symbolic address of the target variables.

The following variables are created for the following setups:

▶ Variable was derived from a prediction model and has precisely 1 value:

▶ Target variable (simple)

▶ Input variable for **Input** (data type depends on whether the model is time based or value based)

▶ Command variable

▶ Variable was derived from a prediction model and has more than 1 value:

▶ Target variable (array)

▶ Input variable for **From** (data type depends on whether the model is time based or value based)

▶ Input variable for **To** (data type depends on whether the model is time based or value based)

▶ Command variable

▶ Variable was derived from a schedule and has precisely 1 value:

▶ Target variable (simple)

▶ Variable was derived from a schedule and has more than 1 value:

▶ Target variable (array)

### 4.5.5 Communication details (Driver variables)

The driver kit implements a number of driver variables. These variables are part of the driver object type *Communication details*. These are divided into:

- ▶ Information
- ▶ Configuration
- ▶ Statistics and
- ▶ Error message

The definitions of the variables implemented in the driver kit are available in the import file **DRVVAR.DBF** and can be imported from there.

Path to file: %ProgramData%\COPA-DATA\zenon<Versionsnummer>\PredefinedVariables

**Note:** Variable names must be unique in zenon. If driver variables of the driver object type *Communication details* are to be imported from **DRVVAR.DBF** again, the variables that were imported beforehand must be renamed.



#### Information

Not every driver supports all driver variables of the driver object type *Communication details*.

For example:

- ▶ Variables for modem information are only supported by modem-compatible drivers.
- ▶ Driver variables for the polling cycle are only available for pure polling drivers.
- ▶ Connection-related information such as **ErrorMSG** is only supported for drivers that only edit one connection at a time.

### INFORMATION

Name from import	Type	Offset	Description
MainVersion	UINT	0	Main version number of the driver.
SubVersion	UINT	1	Sub version number of the driver.
BuildVersion	UINT	29	Build version number of the driver.
RTMajor	UINT	49	zenon main version number
RTMinor	UINT	50	zenon sub version number

Name from import	Type	Offset	Description
RTSp	UINT	51	zenon Service Pack number
RTBuild	UINT	52	zenon build number
LineStateIdle	BOOL	24.0	TRUE, if the modem connection is idle
LineStateOffering	BOOL	24.1	TRUE, if a call is received
LineStateAccepted	BOOL	24.2	The call is accepted
LineStateDialtone	BOOL	24.3	Dialtone recognized
LineStateDialing	BOOL	24.4	Dialing active
LineStateRingBack	BOOL	24.5	While establishing the connection
LineStateBusy	BOOL	24.6	Target station is busy
LineStateSpecialInfo	BOOL	24.7	Special status information received
LineStateConnected	BOOL	24.8	Connection established
LineStateProceeding	BOOL	24.9	Dialing completed
LineStateOnHold	BOOL	24.10	Connection in hold
LineStateConferenced	BOOL	24.11	Connection in conference mode.
LineStateOnHoldPendConf	BOOL	24.12	Connection in hold for conference
LineStateOnHoldPendTransfer	BOOL	24.13	Connection in hold for transfer
LineStateDisconnected	BOOL	24.14	Connection terminated.
LineStateUnknow	BOOL	24.15	Connection status unknown
ModemStatus	UDINT	24	Current modem status
TreiberStop	BOOL	28	Driver stopped  For <i>driver stop</i> , the variable has the value <i>TRUE</i> and an <b>OFF</b> bit. After the driver has started, the variable has the value <i>FALSE</i> and no <b>OFF</b> bit.
SimulRTState	UDINT	60	Informs the status of Runtime for driver simulation.
ConnectionStates	STRING	61	Internal connection status of the driver to

Name from import	Type	Offset	Description
			<p>the PLC.</p> <p>Connection statuses:</p> <p>0: Connection OK</p> <p>1: Connection failure</p> <p>2: Connection simulated</p> <p>Formating:</p> <p>&lt;Netzadresse&gt;:&lt;Verbindungszustand&gt;;...;;</p> <p>A connection is only known after a variable has first signed in. In order for a connection to be contained in a string, a variable of this connection must be signed in once.</p> <p>The status of a connection is only updated if a variable of the connection is signed in. Otherwise there is no communication with the corresponding controller.</p>

## CONFIGURATION

Name from import	Type	Offset	Description
ReconnectInRead	<i>BOOL</i>	27	If TRUE, the modem is automatically reconnected for reading
ApplyCom	<i>BOOL</i>	36	Apply changes in the settings of the serial interface. Writing to this variable immediately results in the method SrvDrvVarApplyCom being called (which currently has no further function).
ApplyModem	<i>BOOL</i>	37	Apply changes in the settings of the modem. Writing this variable immediately calls the method SrvDrvVarApplyModem. This closes the current connection and opens a new one according to the settings <b>PhoneNumberSet</b> and <b>ModemHwAdrSet</b> .
PhoneNumberSet	<i>STRING</i>	38	Telephone number, that should be used
ModemHwAdrSet	<i>DINT</i>	39	Hardware address for the telephone



Name from import	Type	Offset	Description
			number
GlobalUpdate	<i>UDINT</i>	3	Update time in milliseconds (ms).
BGlobalUpdaten	<i>BOOL</i>	4	TRUE, if update time is global
TreiberSimul	<i>BOOL</i>	5	TRUE, if driver in sin simulation mode
TreiberProzab	<i>BOOL</i>	6	TRUE, if the variables update list should be kept in the memory
ModemActive	<i>BOOL</i>	7	TRUE, if the modem is active for the driver
Device	<i>STRING</i>	8	Name of the serial interface or name of the modem
ComPort	<i>UINT</i>	9	Number of the serial interface.
Baudrate	<i>UDINT</i>	10	Baud rate of the serial interface.
Parity	<i>SINT</i>	11	Parity of the serial interface
ByteSize	<i>USINT</i>	14	Number of bits per character of the serial interface  Value = 0 if the driver cannot establish any serial connection.
StopBit	<i>USINT</i>	13	Number of stop bits of the serial interface.
Autoconnect	<i>BOOL</i>	16	TRUE, if the modem connection should be established automatically for reading/writing
PhoneNumber	<i>STRING</i>	17	Current telephone number
ModemHwAdr	<i>DINT</i>	21	Hardware address of current telephone number
RxIdleTime	<i>UINT</i>	18	Modem is disconnected, if no data transfer occurs for this time in seconds (s)
WriteTimeout	<i>UDINT</i>	19	Maximum write duration for a modem connection in milliseconds (ms).
RingCountSet	<i>UDINT</i>	20	Number of ringing tones before a call is accepted
ReCallIdleTime	<i>UINT</i>	53	Waiting time between calls in seconds (s).

Name from import	Type	Offset	Description
ConnectTimeout	<i>UINT</i>	54	Time in seconds (s) to establish a connection.

## STATISTICS

Name from import	Type	Offset	Description
MaxWriteTime	<i>UDINT</i>	31	The longest time in milliseconds (ms) that is required for writing.
MinWriteTime	<i>UDINT</i>	32	The shortest time in milliseconds (ms) that is required for writing.
MaxBlkReadTime	<i>UDINT</i>	40	Longest time in milliseconds (ms) that is required to read a data block.
MinBlkReadTime	<i>UDINT</i>	41	Shortest time in milliseconds (ms) that is required to read a data block.
WriteErrorCount	<i>UDINT</i>	33	Number of writing errors
ReadSucceedCount	<i>UDINT</i>	35	Number of successful reading attempts
MaxCycleTime	<i>UDINT</i>	22	Longest time in milliseconds (ms) required to read all requested data.
MinCycleTime	<i>UDINT</i>	23	Shortest time in milliseconds (ms) required to read all requested data.
WriteCount	<i>UDINT</i>	26	Number of writing attempts
ReadErrorCount	<i>UDINT</i>	34	Number of reading errors
MaxUpdateTimeNormal	<i>UDINT</i>	56	Time since the last update of the priority group <b>Normal</b> in milliseconds (ms).
MaxUpdateTimeHigher	<i>UDINT</i>	57	Time since the last update of the priority group <b>Higher</b> in milliseconds (ms).
MaxUpdateTimeHigh	<i>UDINT</i>	58	Time since the last update of the priority group <b>High</b> in milliseconds (ms).
MaxUpdateTimeHighest	<i>UDINT</i>	59	Time since the last update of the priority group <b>Highest</b> in milliseconds (ms).
PokeFinish	<i>BOOL</i>	55	Goes to 1 for a query, if all current pokes were

Name from import	Type	Offset	Description
			executed

## ERROR MESSAGE

Name from import	Type	Offset	Description
ErrorTimeDW	UDINT	2	Time (in seconds since 1.1.1970), when the last error occurred.
ErrorTimeS	STRING	2	Time (in seconds since 1.1.1970), when the last error occurred.
RdErrPrimObj	UDINT	42	Number of the PrimObject, when the last reading error occurred.
RdErrStationsName	STRING	43	Name of the station, when the last reading error occurred.
RdErrBlockCount	UINT	44	Number of blocks to read when the last reading error occurred.
RdErrHwAdresse	DINT	45	Hardware address when the last reading error occurred.
RdErrDatablockNo	UDINT	46	Block number when the last reading error occurred.
RdErrMarkerNo	UDINT	47	Marker number when the last reading error occurred.
RdErrSize	UDINT	48	Block size when the last reading error occurred.
DrvError	USINT	25	Error message as number
DrvErrorMsg	STRING	30	Error message as text
ErrorFile	STRING	15	Name of error log file

## 4.6 Driver-specific functions

The driver supports the following functions:

- ▶ Receipt of predicted values that are created and sent by the Analyzer server in an automated manner by means of a schedule.
- ▶ Requests for predicted values from the Analyzer server with specifiable input values.

- ▶ Predictions can be received and queried for time-based and value-based prediction models.
- ▶ Each prediction can be an individual prediction or a range prediction.
- ▶ Immediate detection of changes of the status of the Analyzer server: If this goes offline, all dependent variables are highlighted as *INVALID*.
- ▶ Checking the existence and properties of databases, prediction models and schedules in the event of a change to the status of a higher-level object. This check is carried out after expiry of a defined time after the last check: If an object cannot be found or its properties differ from the variable settings, all dependent variables are marked as *INVALID*. Hierarchy of the check: Analyzer instance -> Database -> Models and schedules.  
Examples of differing properties:
  - ▶ The model is time based, but it is given as value based in the symbolic address.
  - ▶ The number of variables in the array does not correspond to the number of the predicted values in the schedule.

### PROCESS FOR SPONTANEOUS COMMUNICATION (FROM DRIVER START):

1. The variables are sent to the driver on starting. The administration structure for the required objects on the Analyzer server is built up from the symbolic addresses of all existing variables in the driver.
2. The Analyzer server is detected as online.
3. The database is found when checking the status.
4. The prediction model and the schedule are found when checking the status and the properties correspond to the configuration.
5. As soon as the variable is signed in, the receipt structure for schedule-based data is created and activated.
6. As soon as the Analyzer server sends a prediction value message, it is received in the driver via the receipt structure.
7. The values from the message are read and forwarded to the variables.

### PROCESS FOR POLLING COMMUNICATION (FROM DRIVER START):

1. The variables are sent to the driver on starting. The administration structure for the required objects on the Analyzer server is built up from the symbolic addresses of all existing variables in the driver. In doing so, all variables required for polling communication - prediction value variable, input variables, control variable - are connected to one another via the GUID from the symbolic address.
2. The Analyzer server is detected as online.
3. The database is found when checking the status.

4. The prediction model is found when checking the status and the properties correspond to the configuration.
5. The necessary input values are written to the variables in the Runtime.
6. The control variable is set to 1 in the Runtime.
7. The polling process is started in the driver by changing the control variable.
8. The input values are read in the driver from the variables.
9. A query-response structure is created in the driver and the query is sent with the desired input values and number of values (see **driver objects** (on page 32)) via the structure.
10. The query is received in the Analyzer server. The predictions are created and sent as a response.
11. The response is received on the driver.
12. The values are read from the response and forwarded to the prediction variables.
13. The control variable is reset to 0 by the driver.

## 4.7 Driver command function

The zenon **Driver commands** function is to influence drivers using zenon. You can do the following with a driver command:

- ▶ Start
- ▶ Stop
- ▶ Shift a certain driver mode
- ▶ Instigate certain actions

**Note:** This chapter describes standard functions that are valid for most zenon drivers.

Not all functions described here are available for every driver. For example, a driver that does not, according to the data sheet, support a modem connection also does not have any modem functions.



### Attention

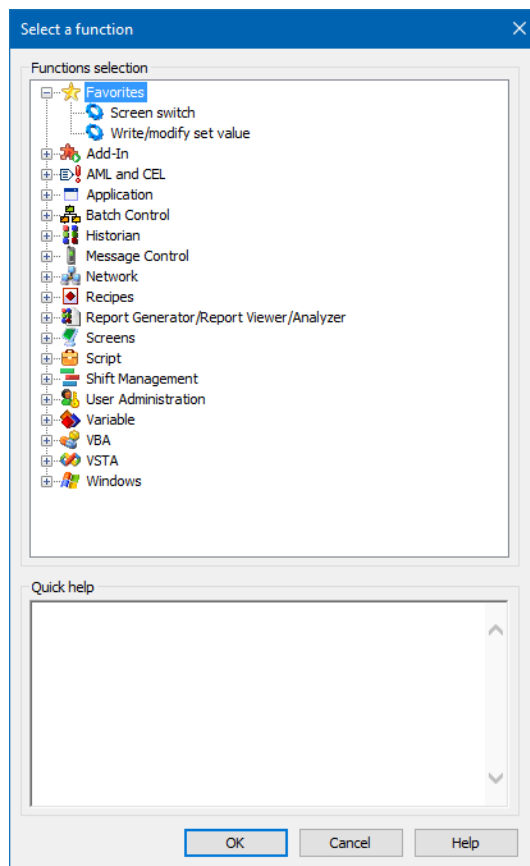
The zenon **Driver commands** function is not identical to driver commands that can be executed in the Runtime with Energy drivers!

## CONFIGURATION OF THE FUNCTION

Configuration is carried out using the **Driver commands** function. To configure the function:

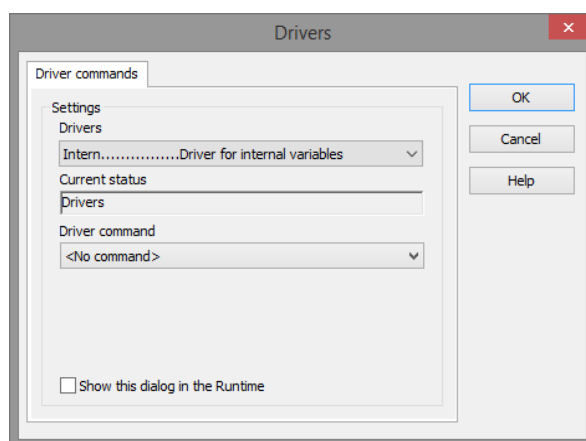
1. Create a new function in the zenon Editor.

The dialog for selecting a function is opened



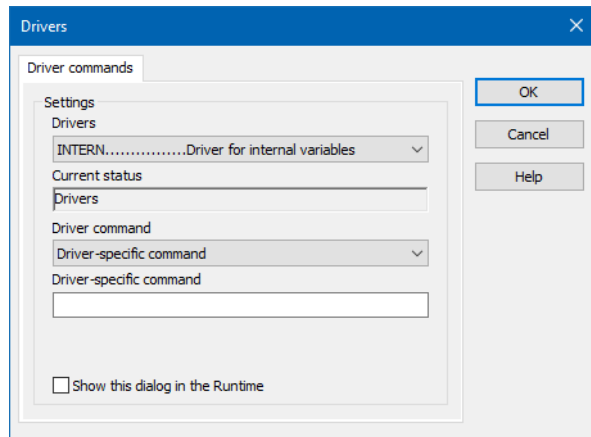
2. Navigate to the node **Variable**.
3. Select the **Driver commands** entry.

The dialog for configuration is opened



4. Select the desired driver and the required command.
5. Close the dialog by clicking on **OK** and ensure that the function is executed in the Runtime. Heed the notices in the **Driver command function in the network** section.

## DRIVER COMMAND DIALOG



Option	Description
<b>Driver</b>	Selection of the driver from the drop-down list. It contains all drivers loaded in the project.
<b>Current condition</b>	Fixed entry that is set by the system. Has no function in the current version.
<b>Driver command</b>	Selection of the desired driver command from a drop-down list.  For details on the configurable driver commands, see the <b>available driver commands</b> section.
<b>Driver-specific command</b>	Entry of a command specific to the selected driver.  <b>Note:</b> Only available if, for the <b>driver command</b> option, the <i>driver-specific command</i> has been selected.
<b>Show this dialog in the Runtime</b>	Configuration of whether the configuration can be changed in the Runtime: <ul style="list-style-type: none"> <li>▶ <i>Active</i>: This dialog is opened in the Runtime before executing the function. The configuration can thus still be changed in the Runtime before execution.</li> <li>▶ <i>Inactive</i>: The Editor configuration is applied in the Runtime when executing the function.</li> </ul> Default: <i>inactive</i>

## CLOSE DIALOG

Options	Description
OK	<b>Applies settings and closes the dialog.</b>
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

## AVAILABLE DRIVER COMMANDS

These driver commands are available - depending on the selected driver:

Driver command	Description
<No command>	No command is sent. A command that already exists can thus be removed from a configured function.
<i>Start driver (online mode)</i>	Driver is reinitialized and started. <b>Note:</b> If the driver has already been started, it must be stopped. Only then can the driver be re-initialized and started.
<i>Stop driver (offline mode)</i>	Driver is stopped. No new data is accepted. <b>Note:</b> If the driver is in offline mode, all variables that were created for this driver receive the status <i>switched off</i> (OFF; Bit 20).
<i>Driver in simulation mode</i>	Driver is set into simulation mode. The values of all variables of the driver are simulated by the driver. No values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
<i>Driver in hardware mode</i>	Driver is set into hardware mode. For the variables of the driver the values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
<i>Driver-specific command</i>	Entry of a driver-specific command. Opens input field in order to enter a command.
<i>Activate driver write set value</i>	Write set value to a driver is possible.
<i>Deactivate driver write set value</i>	Write set value to a driver is prohibited.
<i>Establish connection with modem</i>	Establish connection (for modem drivers)



Driver command	Description
	Opens the input fields for the hardware address and for the telephone number.
<i>Disconnect from modem</i>	Terminate connection (for modem drivers)
<i>Driver in counting simulation mode</i>	Driver is set into counting simulation mode. All values are initialized with 0 and incremented in the set update time by 1 each time up to the maximum value and then start at 0 again.
<i>Driver in static simulation mode</i>	No communication to the controller is established. All values are initialized with 0.
<i>Driver in programmed simulation mode</i>	The values are calculated by a freely-programmable simulation project. The simulation project is created with the help of the zenon Logic Workbench and runs in the zenon Logic Runtime.

## DRIVER COMMAND FUNCTION IN THE NETWORK

If the computer on which the **Driver commands** function is executed is part of the zenon network, further actions are also carried out:

- ▶ A special network command is sent from the computer to the project server. It then executes the desired action on its driver.
- ▶ In addition, the Server sends the same driver command to the project standby. The standby also carries out the action on its driver.

This makes sure that Server and Standby are synchronized. This only works if the Server and the Standby both have a working and independent connection to the hardware.

## 4.8 Error analysis

Should there be communication problems, this chapter will assist you in finding out the error.

### 4.8.1 Analysis tool

All zenon modules such as Editor, Runtime, drivers, etc. write messages to a joint log file. To display them correctly and clearly, use the Diagnosis Viewer (main.chm::/12464.htm) program that was also installed with zenon. You can find it under **Start/All programs/zenon/Tools 8.10 -> Diagviewer**.

zenon driver log all errors in the LOG files. LOG files are text files with a special structure. The default folder for the LOG files is subfolder **LOG** in the folder **ProgramData**. For example:

**%ProgramData%\COPA-DATA\LOG.**

**Attention:** With the default settings, a driver only logs error information. With the Diagnosis Viewer you can enhance the diagnosis level for most of the drivers to "Debug" and "Deep Debug". With this the driver also logs all other important tasks and events.

In the Diagnosis Viewer you can also:

- ▶ Follow newly-created entries in real time
- ▶ customize the logging settings
- ▶ change the folder in which the LOG files are saved

**Note:**

1. The Diagnosis Viewer displays all entries in UTC (coordinated world time) and not in local time.
2. The Diagnosis Viewer does not display all columns of a LOG file per default. To display more columns activate property **Add all columns with entry** in the context menu of the column header.
3. If you only use **Error-Logging**, the problem description is in the column **Error text**. For other diagnosis level the description is in the column **General text**.
4. For communication problems many drivers also log error numbers which the PLC assigns to them. They are displayed in **Error text** or **Error code** or **Driver error parameter (1 and 2)**. Hints on the meaning of error codes can be found in the driver documentation and the protocol/PLC description.
5. At the end of your test set back the diagnosis level from **Debug** or **Deep Debug**. At **Debug** and **Deep Debug** there are a great deal of data for logging which are saved to the hard drive and which can influence your system performance. They are still logged even after you close the Diagnosis Viewer.



**Attention**

In Windows CE errors are not logged per default due to performance reasons.

You can find further information on the Diagnosis Viewer in the Diagnose Viewer (main.chm::/12464.htm) manual.

## 4.8.2 Check list

### GENERAL TROUBLESHOOTING

Check the following in the event of errors:

- ▶ Is the computer switched on with the zenon Analyzer and are the Analyzer server services running?

- ▶ Are the **Service-Hub** services running on the computer envisaged for this?
- ▶ Analysis with the **Diagnosis Viewer** (on page 65):  
-> Which messages are displayed?
- ▶ Is the computer available with the zenon Analyzer and the **Service Hub** in the **TCP/IP** network?  
Can it be contacted via the *Ping* command?
- ▶ Was the **Symbolic address** of the variable set correctly?
- ▶ Did you use the right object type for the variable

**Example:** Driver variables are purely statistics variables. They do not communicate with the PLC.

zenon driver log all errors in the LOG files. LOG files are text files with a special structure. The default folder for the LOG files is subfolder **LOG** in the folder **ProgramData**. For example:

**%ProgramData%\COPA-DATA\LOG.**

**Attention:** With the default settings, a driver only logs error information. With the Diagnosis Viewer you can enhance the diagnosis level for most of the drivers to "Debug" and "Deep Debug". With this the driver also logs all other important tasks and events. You can find more information on the Diagnosis Viewer in the Diagnosis Viewer (main.chm::/12464.htm) manual.

The following is required for further analysis of errors:

- ▶ The project backup
- ▶ LOG files

Send these to your support person after agreement with the customer service department.

## SOME VARIABLES REPORT INVALID.

- ▶ **Prediction Trigger** and **Prediction Input Value** driver object type variables go to INVALID if the desired prediction model is not available.
- ▶ **Predicted Value** driver object type values go to INVALID if the desired prediction model is no longer available or no prediction value has been received.

## VALUES ARE NOT DISPLAYED, NUMERIC VALUES REMAIN EMPTY

Driver is not working. Check the:

- ▶ Installation of zenon
- ▶ the driver installation
- ▶ The installation of all components  
-> Pay attention to error messages during the start of the Runtime.

## VARIABLES ARE DISPLAYED WITH A BLUE DOT

The communication in the network is faulty:

- ▶ With a network project:  
Is the network project also running on the server?
- ▶ With a stand-alone project or a network project which is also running on the server:  
Deactivate the property **Read from Standby Server only** in node **Driver connection/Addressing**.

## VALUES ARE DISPLAYED INCORRECTLY

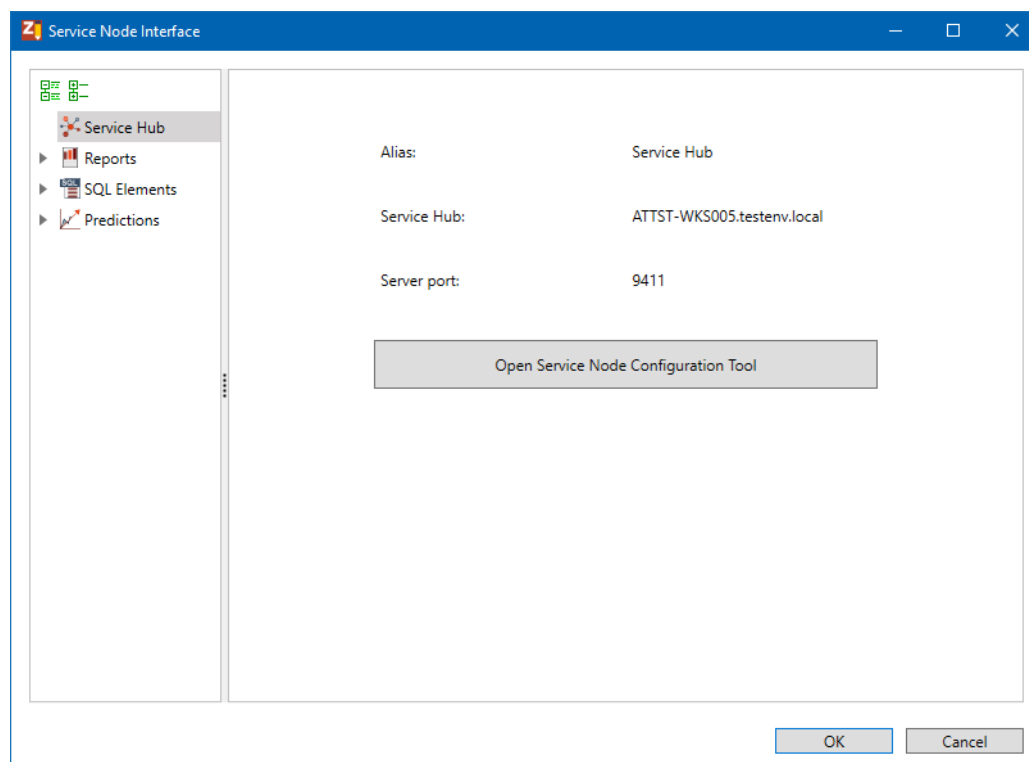
Check the information for the calculation in node **Value calculation** of the variable properties.

## DRIVER FAILS OCCASIONALLY

Analysis with the **Diagnosis Viewer** (on page 65):  
-> Which messages are displayed?

# 5 Analyzer Service Node Interface

The **Analyzer Service Node** forms the interface to the Analyzer server of the **Service Hub** in the Service Grid. It is configured in ZAMS.



The zenon Analyzer provides other prosumers with reports, SQL elements and prediction models.

The **Service Node Interface** is used for the following:

- ▶ Display of the **Service Hub** to which the Analyzer service node connects. The **Service Node Configuration Tool** can be started directly to configure the connection.
- ▶ Configuration of reports that are provided via the **Service Grid** and can be executed.
- ▶ Configuration of schedules for reports. These are automatically executed on the **Analyzer Service Node**.

The **Service Grid** or another **Service Node** can use the **Service Hub** to call up which schedules are currently running on the **Analyzer Service Node** and subscribe to one or more schedules. If a schedule has been subscribed to, the requesting **Service Node** is sent the results from the executed schedule.

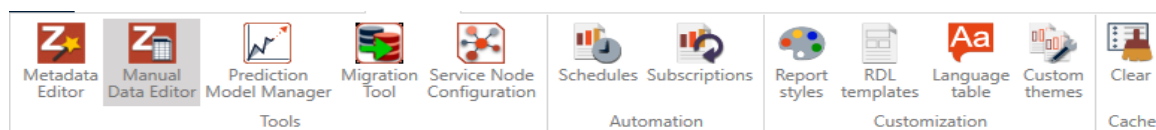
- ▶ Configuration of SQL elements that are provided and can be executed via the **Service Grid**: Stored Procedures, Scalar User Defined Functions, Table Valued Functions.

**Note:** A service (on page 74) that monitors the configuration of the **Analyzer Service Node** runs on the Analyzer Server. The **Service Node Interface** can only be configured if ZAMS is executed on the same computer as the Analyzer server.

## 5.1 Start and set parameters for service node interface

To start the **Service Node Interface**:

1. Navigate to the **Options** ribbon in ZAMS.



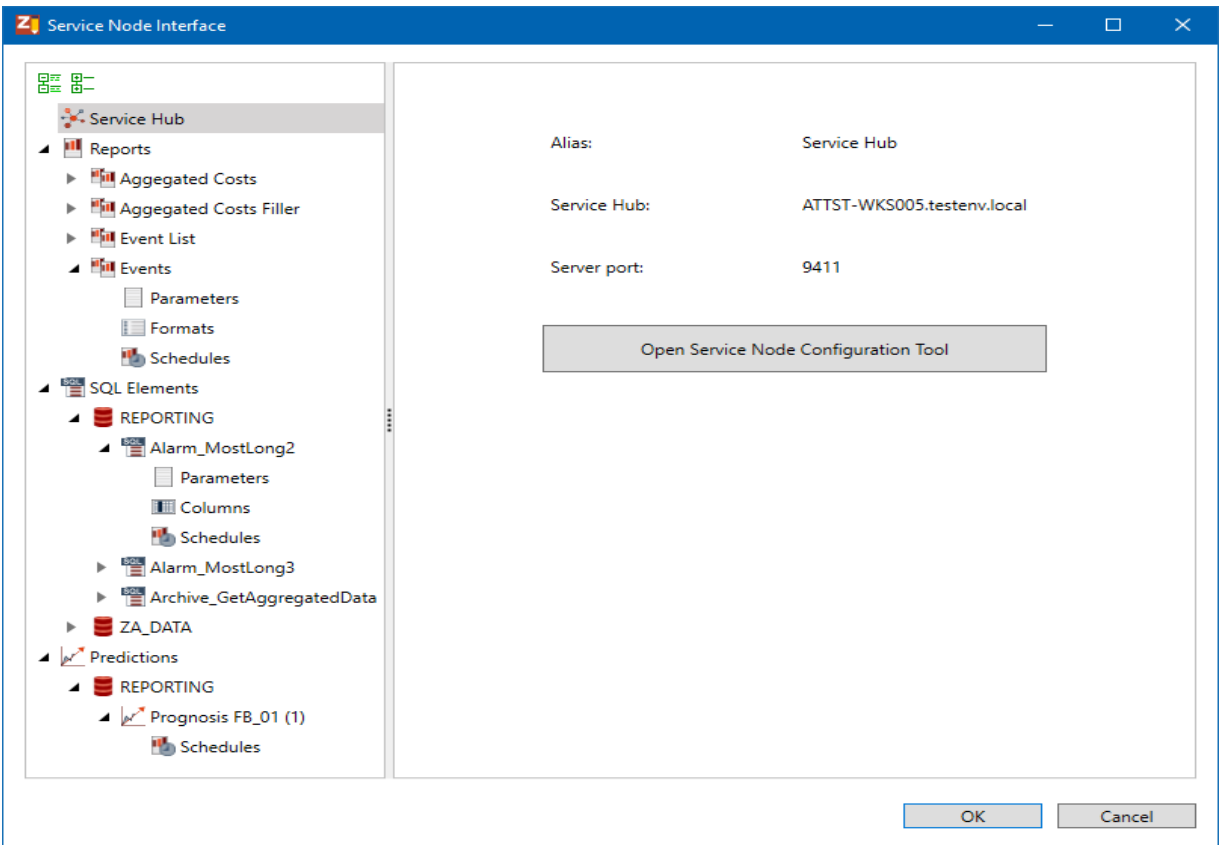
2. In the **Tools** section, click on the **Service Node Interface** entry.

**Note:** The entry is only active if ZAMS is executed on the same computer as the Analyzer server and is connected to a database.

The dialog for configuration is opened. When opening, all existing and required information is loaded. A progress bar is shown in the progress.



SERVICE NODE INTERFACE DIALOG



Option	Description
Object tree	<p>Lists the objects to be configured:</p> <ul style="list-style-type: none"><li>▶ <b>Service Hub</b> (on page 77): Display of the connection to the <b>Service Hub</b>.</li><li>▶ <b>Reports</b> (on page 81): Configuration of the reports that the <b>Analyzer Service Node</b> provides.</li><li>▶ <b>SQL elements</b> (on page 94): Configuration of the SQL elements that the <b>Analyzer Service Node</b> provides.</li><li>▶ <b>Predictions</b> (on page 107): Configuration of the prediction models that the <b>Analyzer Service Node</b> provides.</li></ul> <p>Objects that still need to be configured or contain errors are displayed with a red border.</p> <p>The objects subordinated to the main nodes can be expanded or collapsed using the symbols at the header of the column. Individual nodes are expanded or collapsed</p>

Option	Description
	with a mouse click on the node.  For configuration, select the respective objects and configure them. Clicking on the <b>OK</b> button accepts all configurations and closes the dialog.
<b>Configuration window</b>	Detail view and configuration of the selected objects.

## CLOSE DIALOG

Option	Description
<b>OK</b>	Applies settings and closes the dialog.  Changes are saved locally and provided in the <b>Service Hub</b> .
<b>Cancel</b>	Discards all changes and closes the dialog.

**Note:** The **OK** and **Cancel** buttons are always applicable for the complete dialog, not for individual objects. They are no longer documented separately when describing the configurable objects. When closing the dialog, a request for confirmation appears, which points out that there are unsaved changes.

## 5.1.1 Sort, filter and group lists

Lists in the dialog can be displayed as sorted, filtered and grouped.

### SORT LIST

To sort a list:

1. Click in the header of the column according to which sorting is to take place.
2. Click again to change the sorting.

Three sorting processes are gone through in sequence:

- ▶ *Ascending:* The list is sorted from 0 - 9 and from a - z.  
Display with arrow pointing upwards.
- ▶ *Descending:* The list is sorted from 9 - 0 and from z - a.  
Display by the arrow upwards.
- ▶ *Unsorted:* The list is displayed in the original status, i.e. as it was created.

## FILTER LIST

To filter a list:

1. Click on the filter symbol in the filter line.  
The filter dialog is opened.
2. Configure the filter.
3. Click on the **Filter** button.  
The filter is applied to the list.

To remove the filter again:

1. Open the filter dialog.
2. Click on the **Remove filter** button.

## FILTER OPTIONS

Option	Description
<b>Option</b>	Entry of the character sequence that is to be filtered for.
<b>aA</b>	Switches between: <ul style="list-style-type: none"> <li>▶ <i>Capitalization is not taken into account</i></li> <li>▶ <i>Capitalization is taken into account</i></li> </ul> Default: <i>Capitalization is not taken into account</i>
<b>Select all</b>	<ul style="list-style-type: none"> <li>▶ Active: Selects all existing elements.</li> </ul>
<b>[Report name]</b>	Selection of individual elements that are applied to the filter. Selection by activating the checkbox in front of the element names.
<b>Show lines with the following value</b>	Configuration of the selection by setting the following criteria.
<b>Criterion</b>	First criterion that is applied. <ul style="list-style-type: none"> <li>▶ Selection of the condition from the drop-down list.</li> <li>▶ Entry of the criterion in the text field.</li> </ul> For criteria, see the <b>Criteria</b> section.
<b>Linkage</b>	Linking to a second criterion. Select from drop-down list: <ul style="list-style-type: none"> <li>▶ <i>and</i>: Both criteria must be applicable.</li> </ul>



Option	Description
	<ul style="list-style-type: none"> <li>▶ <i>or</i>: One of the two criteria must be met.</li> </ul>
<b>Criterion</b>	<p>Second criterion that is applied and is linked to the first criterion.</p> <ul style="list-style-type: none"> <li>▶ Selection of the condition from the drop-down list.</li> <li>▶ Entry of the criterion in the text field.</li> </ul> <p>For criteria, see the <b>Criteria</b> section.</p>
<b>Filter</b>	Clicking on the button applies filter criteria to the list.
<b>Remove filter</b>	Removes all filter criteria. All lines are displayed again.

## CRITERIA

The following are available as criteria for filter expressions:

Criterion	Description
<i>Is the same as</i>	All entries with precisely the character sequence entered are displayed.
<i>Is not the same as</i>	All entries that do not precisely correspond to the character sequence entered are displayed.
<i>Starts with</i>	All entries that start with the character sequence are displayed.
<i>Ends with</i>	All entries that end with the character sequence are displayed.
<i>Contains</i>	All entries that contain this character sequence are displayed. (Default setting.)
<i>Does not contain</i>	All entries that do not contain this character sequence are displayed.
<i>Is empty</i>	<p>All empty entries are displayed.</p> <p><b>Note:</b> Language tables must not contain empty cells for existing key words.</p>
<i>is not empty</i>	All entries that contain at least one character are displayed. Spaces are also considered characters.

## GROUP LIST

To group a list:

1. Click on the header of the column according to which the list is to be grouped.
2. Drag the header of the column to the grouping line, with the mouse.  
The list is shown accordingly grouped:
3. Add further grouping columns if required.

To remove the grouping:

1. Move the mouse over the grouping element in the grouping column.
2. Click on the **X** next to the column name or drag the element from the grouping line.  
The grouping is removed.

## 5.1.2 Windows service for Analyzer Service Node

For the configuration and use of the **Analyzer Service Node**, the associated Windows service must be running. This is installed and configured automatically during the installation of the Analyzer Server. The service monitors the configuration of the **Analyzer Service Node** for changes. Availability and status are displayed using a symbol in the system tray of the Windows taskbar (on page 74). A dialog with actions and LOG entries can thus be opened.

### 5.1.2.1 Monitoring of service node status

Availability and status of the Window service for the **Analyzer Service Node** are displayed using a symbol in the system tray of the Windows taskbar. The **Service Node Status** dialog with actions and log entries can thus be opened.

The symbol visualizes the status of the service:

- ▶ Service was not found: The executable file has not been registered.
- ▶ Service is stopped: The service can be started via the **Service Node Status** dialog.
- ▶ The service is running.

The status is updated every second and when commands to administer the service are executed. If the mouse is moved over the symbol, a tooltip with the current status is shown.

## SERVICE NODE STATUS CONTEXT MENU

Right-clicking on the symbol for **Service Node Status** opens a context menu.  
Available actions:

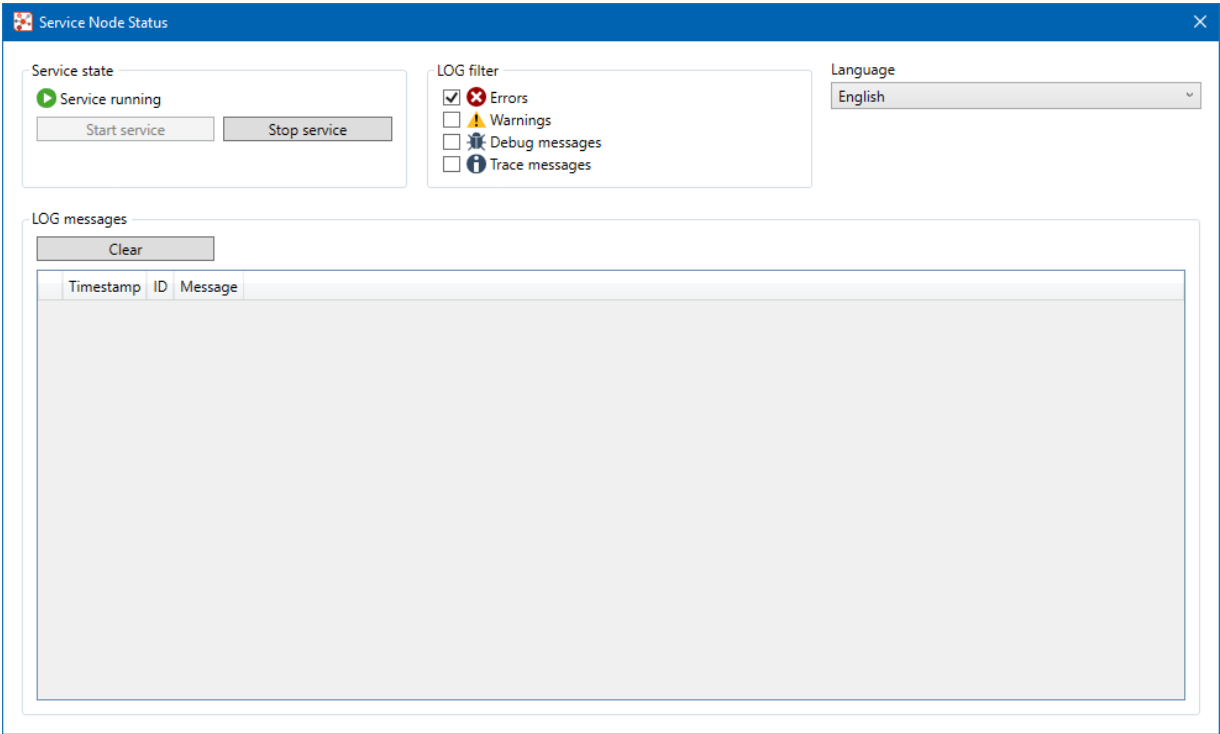
- ▶ **Start service:** Starts the service.  
Only available if the service has been found and has been stopped.



- ▶ **Stop service:** Stops the service.  
Only available if the service has been started.
- ▶ **Display:** Opens the **Service Node Status** dialog.
- ▶ **Exit:** Ends the monitoring of the **Service Node status**.

SERVICE NODE STATUS DIALOG

Double-clicking on the symbol opens the **Service Node status** dialog.



Option	Description
Service status	Display of the status for the <b>Service Node</b> .
Start service	Click on the button to start the service. Only available if the service has been found and has been stopped.
Stop service	Click on the button to stop the service. Only available if the service has been started.
LOG filter	Configuration of which log messages are displayed. Select or deselect a level by clicking in the checkbox in front: <ul style="list-style-type: none"><li>▶ <b>Error</b></li></ul>

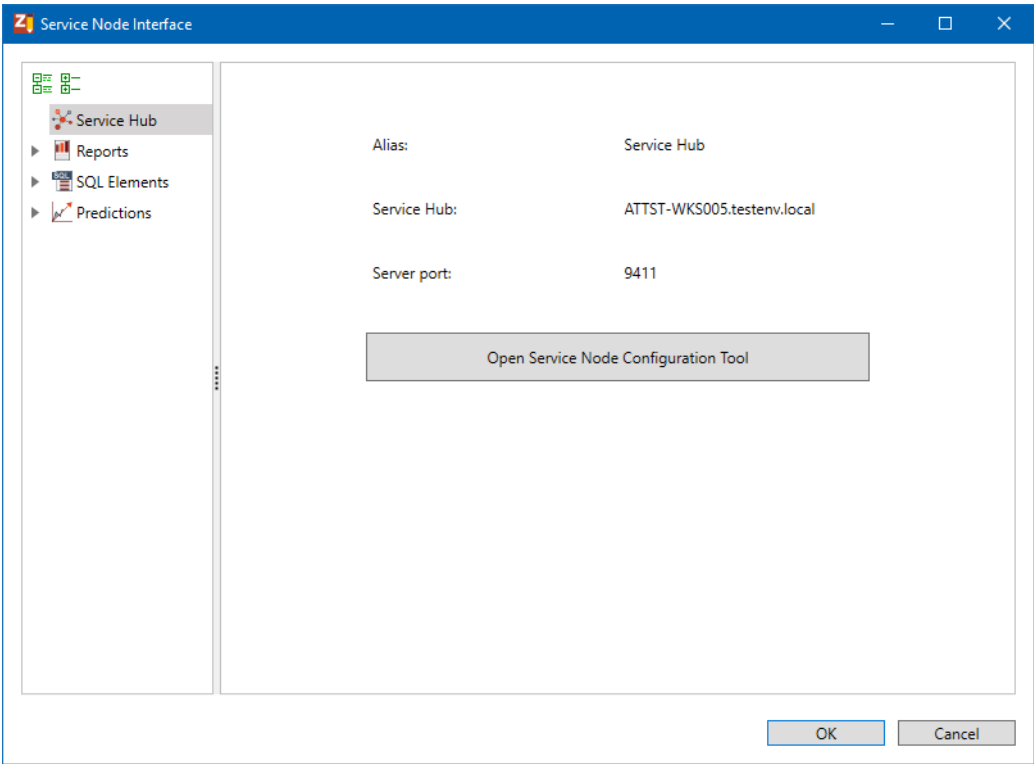
Option	Description
	<ul style="list-style-type: none"> <li>▶ <b>Warnings</b></li> <li>▶ <b>Debug messages</b></li> <li>▶ <b>Trace messages</b></li> </ul>
<b>Language</b>	<p>Selection of the language for dialog and symbol in the info area from a drop-down list.</p> <p>Default: <i>English</i></p>
<b>LOG messages</b>	<p>Display of the LOG messages according to the configuration of the <b>LOG filter</b> option.</p> <p>A maximum of 1000 messages are displayed. If this number is reached, the oldest ones are moved on each restart.</p> <p>Columns for:</p> <ul style="list-style-type: none"> <li>▶ Symbol with display of the level. Corresponds to the display in the <b>LOG filter</b> option. This column cannot be sorted.</li> <li>▶ Time stamp: Shows the time stamp of a message. Format: <i>yyyy-MM-dd HH:mm:ss.fff</i> The column is initially sorted in descending order. The most recent messages are thus shown at the top.</li> <li>▶ Message ID: ID of the message.</li> <li>▶ Message: Contains the text of the message. Texts are not translated. Service messages are generally in English. Operating system messages are generally in the current language of the operating system.</li> </ul> <p>All columns, with the exception of the Symbol column, can be sorted.</p>
<b>Delete</b>	Removes the buffered LOG messages.

To close the dialog, click on the **X** in the top right corner. The application continues to run and is displayed in the info area as a symbol.

## 5.2 Connection to the service hub

To provide objects in the **Service Node**, a connection to the **Service Hub** must be established.

The currently-configured connection is displayed in the **Service Hub** section:.



The connection data is displayed

Option	Description
Alias	Alias for the configured <b>Service Hub</b> . This is saved locally by the <b>Service Node Configuration Tool</b> .
Service Hub	URL of the <b>Service Hub</b> .
Serverport	Number of the port. <b>Service Hub</b> .
Service Node Configuration Tool open	To change the connection, start the <b>Service Node Configuration Tool</b> (on page 77) and select the desired <b>Service Hub</b> there.



### 5.2.1 Dialog: Service Node Configuration Tool

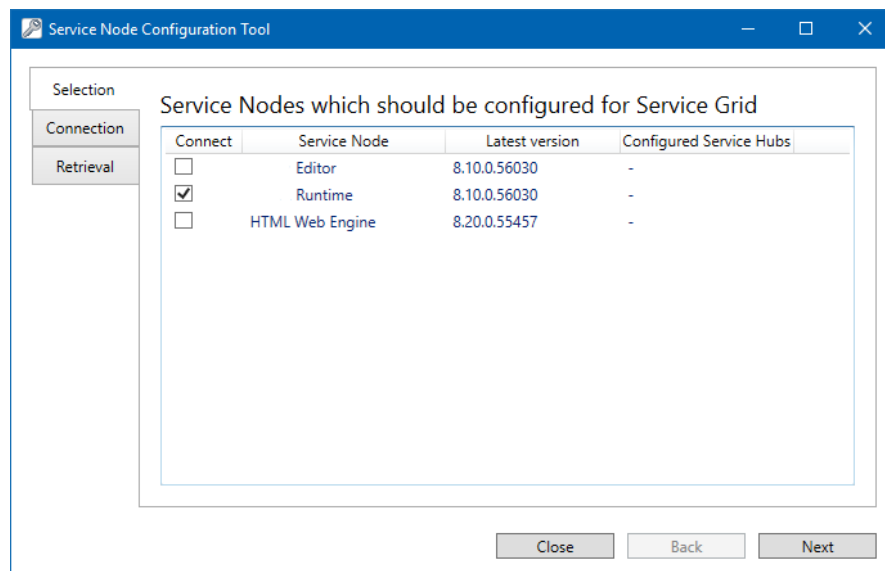
The **Service Node Configuration Tool** consists of 3 tabs:

- ▶ Selection

- ▶ Connection
- ▶ Retrieval

## SELECTION

Here you can configure the **Service Nodes** that are to be connected to the **Service Hub**.



Connect	Service Node	Latest version	Configured Service Hubs
<input type="checkbox"/>	Editor	8.10.0.56030	-
<input checked="" type="checkbox"/>	Runtime	8.10.0.56030	-
<input type="checkbox"/>	HTML Web Engine	8.20.0.55457	-

Column	Description
<b>Connect</b>	Selection of the <b>Service Nodes</b> that are to be connected to the <b>Service Hub</b> .  Click the checkbox to select the respective <b>Service Node</b> for the following steps.
<b>Service Node</b>	Display of the <b>Service Nodes</b> that can be connected to the <b>Service Hub</b> .
<b>Latest version</b>	Display the current version number of the <b>Service Node</b> .
<b>Configured Service Hubs</b>	Display the previously configured connections for the <b>Service Node</b> .

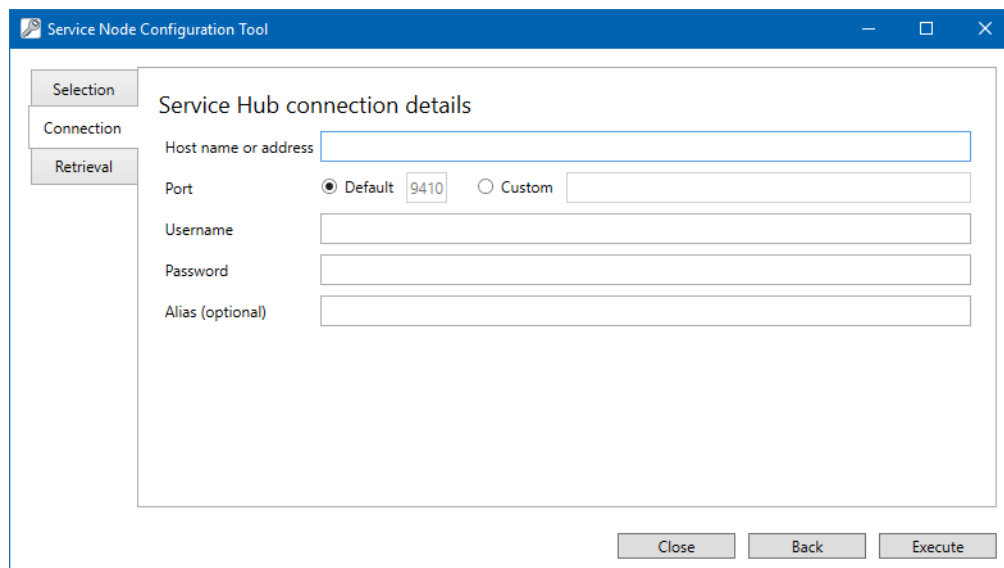
## NAVIGATION

Option	Description
<b>Close</b>	Closes the dialog. Changes that have not been saved are lost.
<b>Back</b>	Goes back one tab.  Not available for <b>Selection</b> .

Option	Description
Next	Moves one step forward to <b>Connection</b> .

## CONNECTION

Here you can configure the connection to the **Hub Controller**.



The screenshot shows the 'Service Node Configuration Tool' window. On the left, there is a sidebar with three tabs: 'Selection', 'Connection' (which is active), and 'Retrieval'. The main area is titled 'Service Hub connection details' and contains the following fields:

- Host name or address:** A text input field.
- Port:** A section with two radio buttons: 'Default' (selected) and 'Custom'. The 'Default' option has a text field containing '9410'. The 'Custom' option has an empty text field.
- Username:** A text input field.
- Password:** A text input field.
- Alias (optional):** A text input field.

At the bottom right of the window, there are three buttons: 'Close', 'Back', and 'Execute'.

Option	Description
<b>Host name or address</b>	Address at which the <b>Hub Controller</b> can be contacted. Input as IP address or host name.
<b>Port</b>	<p>Selection of the port at which the <b>Hub Controller</b> can be contacted:</p> <ul style="list-style-type: none"> <li>▶ <b>Default:</b> The port selected by the system is used. Default: <i>9410</i></li> <li>▶ <b>Custom:</b> The port selected by the user is used. Entry of the port number in the field.</li> </ul> <p><b>Note:</b> Make sure that the user-defined port is not blocked by the firewall.</p>
<b>Username</b>	Entry of the <b>Hub Controller</b> user name.
<b>Password</b>	Entry of the <b>Hub Controller</b> password.
<b>Alias</b>	<p>Optional entry of an alias for the <b>Service Node</b>.</p> <p>This is used, for instance, by the zenon Editor or zenon Analyzer to make it easier for the user to distinguish between</p>

Option	Description
	the configured connections later on.

## NAVIGATION

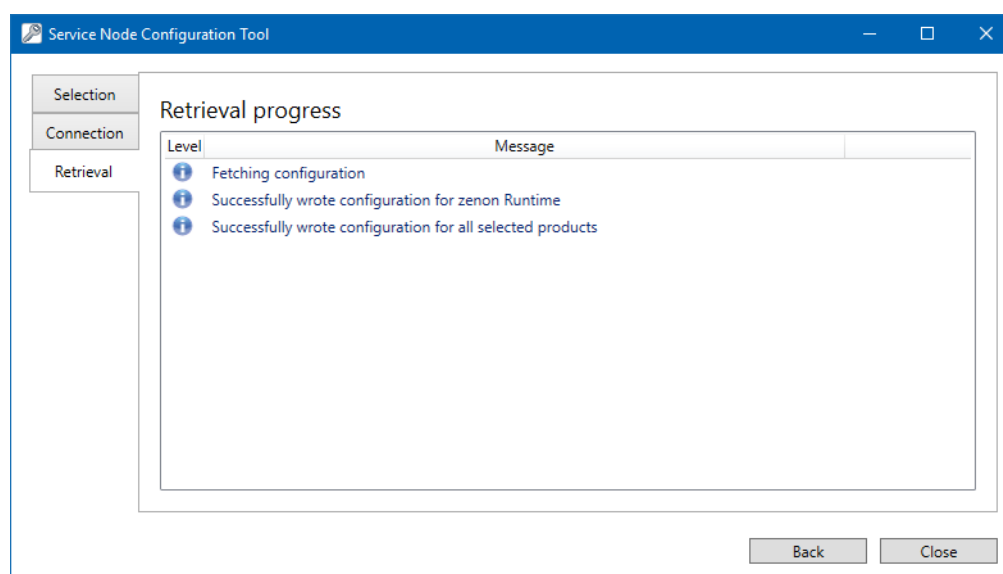
Option	Description
Close	Closes the dialog. Changes that have not been saved are lost.
Back	Goes back one tab.
Execute	Checks the entered connection parameters and starts the configuration of the previously selected <b>Service Nodes</b> .

## RETRIEVAL

The access data for the selected **Service Nodes** are requested and configured here.

All the steps performed are listed.

The successful configuration of the **Service Nodes** and any problems that occur are also displayed. The messages in the window can be highlighted and copied.



Column	Description
Level	Uses a symbol to show the type of message that is displayed.
Message	Information on the individual steps performed. All information can be copied from the window.



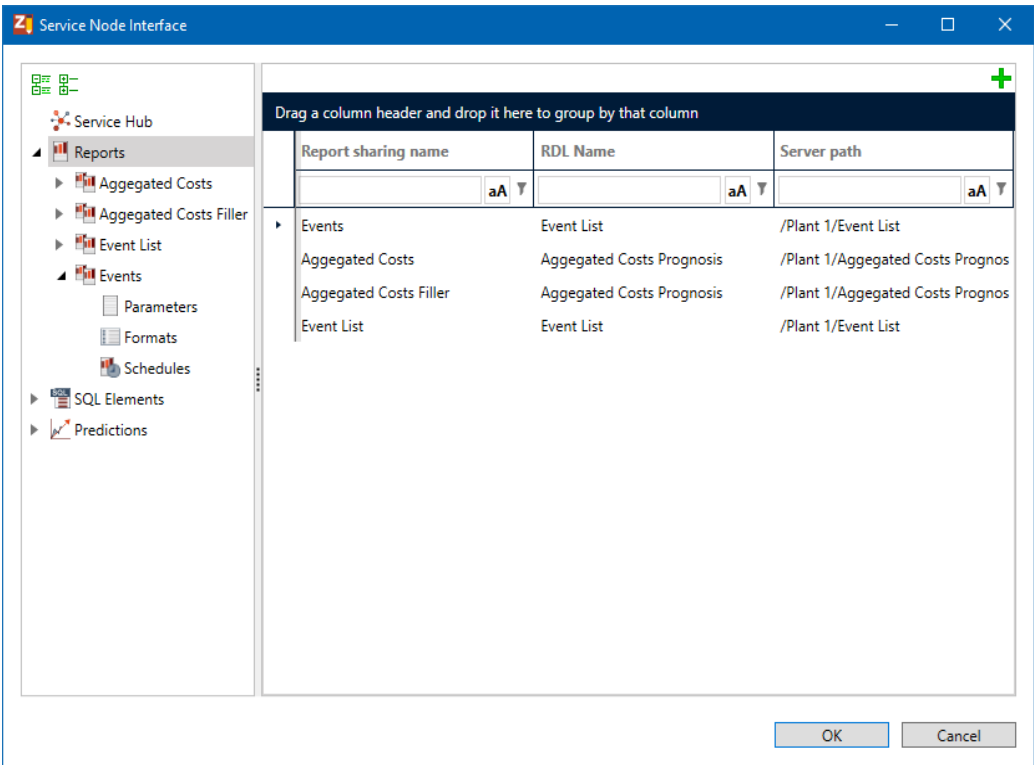


NAVIGATION

Option	Description
Back	Goes back one tab.  Not available for <b>Selection</b> .
Close	Closes the dialog.

5.3 Reports

Display of the configured reports and addition of new reports for the **Service Grid**. Existing reports are administered in the subordinate objects.



Options	Description
+	Clicking on the button opens the dialog to select a report.
List of reports	Displays all reports available in the <b>Analyzer Service Node</b> .



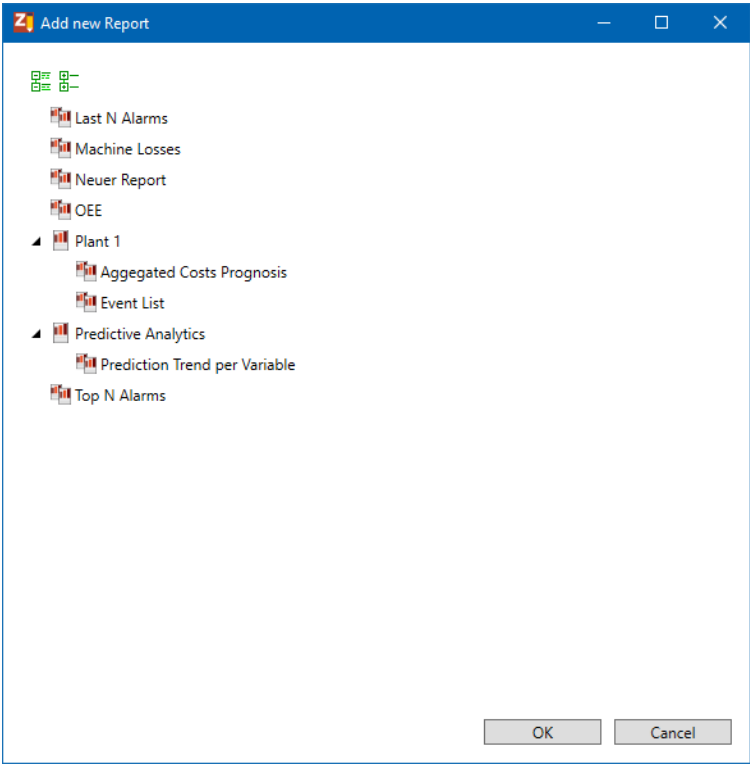
ADD REPORT

To add a report:

- 1. Click on the + symbol in the top right corner of the dialog.  
The selection dialog is opened.  
Only reports that can be provided via the **Service Grid** are displayed. Reports with shifts or lots are not available.
- 2. Select the desired report.
- 3. Click on **OK**.  
The report is added to the list and displayed.
- 4. Configure the report.  
Objects that need inputs are highlighted in red. In doing so, it is always the higher-level objects that are highlighted.

**Note:** The report is only saved in the configuration file and on the service hub if the **Service Node Interface** dialog has been saved and closed with **OK**.

ADD NEW REPORT DIALOG



Option	Description
Collapse/expand node symbol	The objects subordinate to the main nodes can all be displayed as expanded or collapsed.

Option	Description
	Individual nodes are expanded or collapsed with a mouse click.
List of reports	Display of the reports present. Selection via mouse click.
OK	Applies selection, closes the dialog and enters the selected report into the list.
Cancel	Discards selection and closes the dialog.

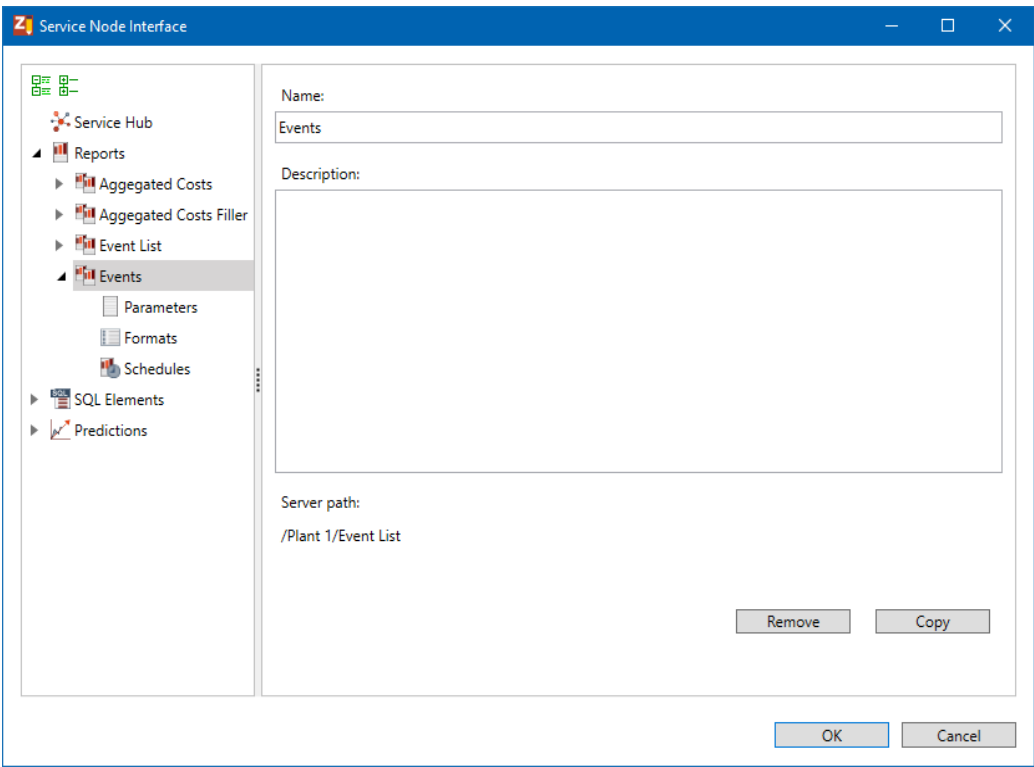
### 5.3.1 Report

Display of a selected report.

You have the possibility:

- ▶ To amend the name and description.
- ▶ To duplicate reports and thus configure variants.
- ▶ To remove reports.
- ▶ To configure parameters (on page 86).
- ▶ To select formats (on page 87) for deployment.
- ▶ To create schedules (on page 87) for deployment.

REPORT CONFIGURATION DIALOG



Option	Description
Name	<p>Name of the report.</p> <p>Can be amended:</p> <ul style="list-style-type: none"><li>▶ Maximum 128 characters.</li><li>▶ Must be unique within the <b>Service Node</b>.</li></ul>
Description	<p>Description of the report.</p> <p>Can be amended:</p> <ul style="list-style-type: none"><li>▶ Maximum 256 characters.</li></ul>
Server path	<p>Save location of the report.</p> <p>Display only.</p>
Delete	<p>Clicking on the button removes the report from the list.</p>
Copy	<p>Clicking on the button creates a duplicate of the report.</p> <p>The following is applicable for duplicates:</p> <ul style="list-style-type: none"><li>▶ Parameters and formats can be amended</li></ul>

Option	Description
	individually. ▶ The name must be changed.

## CREATE AND CONFIGURE VARIANTS

To create a new report on the basis of a pre-existing report:

1. Select the report that is to serve as a basis.
2. Click on **Copy**.  
A duplicate of the report is added.
3. Change the name of the duplicate.
4. Configure parameters, formats and schedules.

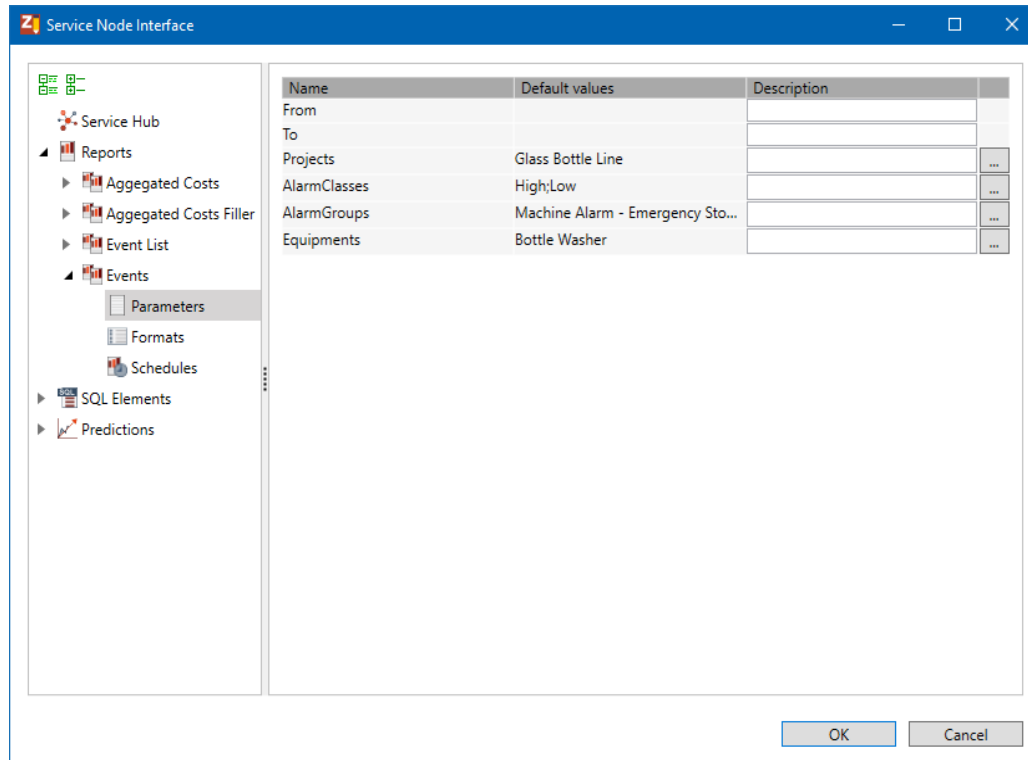
## DELETE REPORT

To remove a new report from the **Service Node Interface**:

1. Select a report in the left window.
2. Click on **Remove** in the detail view.  
The report is removed from the configuration.
3. To confirm the removal, close the **Service Node Interface** by clicking on **OK**.

### 5.3.2 Parameter

Configuration of the report parameters.



Name	Default values	Description
From		
To		
Projects	Glass Bottle Line	...
AlarmClasses	High;Low	...
AlarmGroups	Machine Alarm - Emergency Sto...	...
Equipments	Bottle Washer	...

The parameters that are available depend on the report type. The parameters must be configured consecutively from top to bottom.

**Background:** Each time a parameter is set, a corresponding query is sent to the Analyzer Server. Only after a successful query can valid values be received for the following parameter.

To configure parameters:

1. Start with the uppermost parameter.  
Except: Parameters for time.
2. Open the selection dialog by clicking on the ... button and select the desired values.
3. Configure the following parameters consecutively until no more validation errors are displayed.
4. Enter a description for the respective parameter if required.

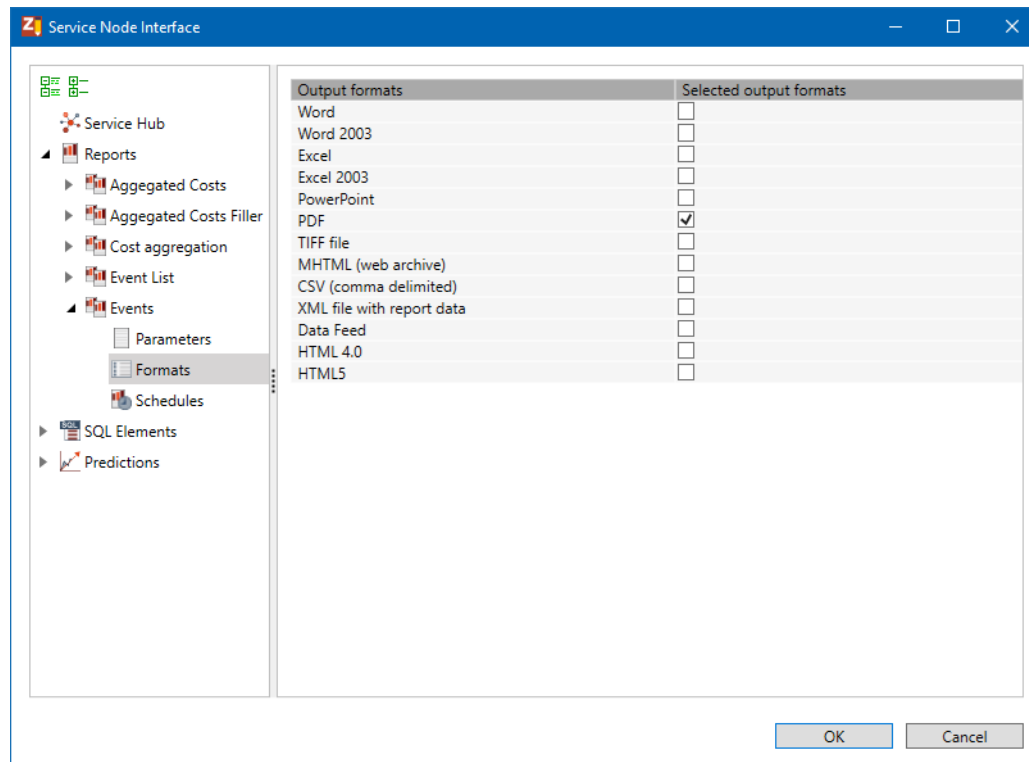


#### Attention

*Timespan* parameters and *Datetime* parameters are only displayed. These are set via the **Service Grid** API when requesting a report.

### 5.3.3 Formats

Selection of the output formats in which the report is to be prepared.



To select output formats:

1. Activate the checkbox behind each desired format.  
These are then available in the schedule.  
**Note:** Schedules can only be configured if at least one output format as been defined.
2. For each schedule, select the format in which the respective report is to be prepared.

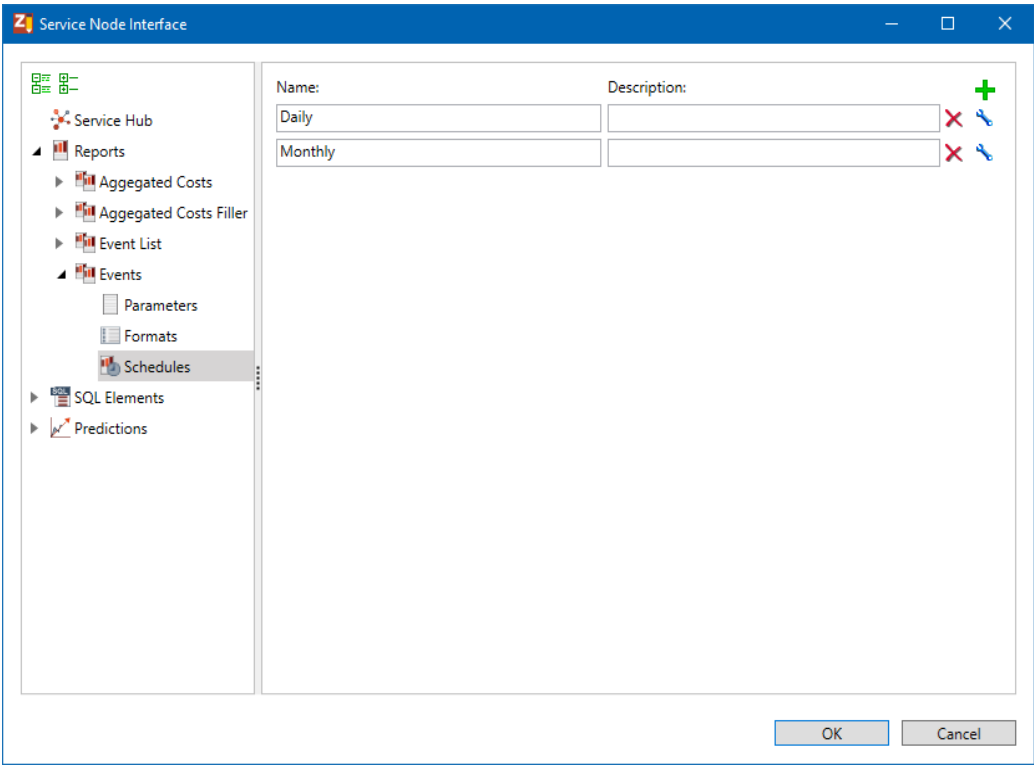
### 5.3.4 Schedules

Reports that have been prepared for the **Service Grid** can be executed automatically by means of a schedule. Each report has a **Schedules** node for this.

You can do the following here:

- ▶ Create new schedules
- ▶ Edit existing schedules
- ▶ Delete existing schedules

SCHEDULES DIALOG



Option	Description
List of schedules	Shows all configured schedules.
Symbol +	Clicking on the symbol opens the dialog to create and configure a schedule.
Symbol X	Clicking on the symbol deletes the selected schedule without a request for confirmation.
Wrench symbol	Clicking on the symbol opens the dialog to configure the selected schedule.

CREATE NEW SCHEDULE

To create a new schedule:

- Click on the symbol +.  
The dialog for configuring the schedule (on page 113) is opened
- Configure the schedule with:
  - Schedule duration
  - Date of execution



- ▶ **Execution time**
  - ▶ **Parameter**
  - ▶ **Output format**
3. Confirm the configuration by clicking on **OK**.  
The schedule is entered into the list.
  4. Please define a name!  
The name must be unique within the report.

## EDIT SCHEDULE

To edit a schedule:

1. Select the schedule.
2. Click on the symbol with the wrench symbol.  
The dialog for configuring the schedule is opened.
3. Configure the schedule.
4. Confirm the configuration by clicking on **OK**.

## DELETE SCHEDULE

To delete a schedule:

1. Select the schedule.
2. Click on the symbol **X**.  
The schedule is deleted.

### 5.3.4.1 Configure a schedule

Here you configure schedules for the preparation of reports with:

- ▶ Schedule duration
- ▶ Date of execution
- ▶ Execution time
- ▶ Parameter

► Output format

Schedules

Schedule duration

Schedule start:

1/30/2019

☐ Schedule expires

Schedule end:

12/31/9999

Execution day

Recurrence:

Daily

Interval:

1

Execution time

Recurrence:

Set times

Hour:

0

Minute:

0

Second:

0

Parameters

From

Now

0

Minute

To

Now

0

Minute

Output format:

PDF

OK

Cancel

SCHEDULE DURATION

Configuration of the time range for which the schedule is applied.

Option	Description
Schedule start	<p>Start time of the schedule. The schedule is valid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i> or selection via calendar element.</p>
Schedule expires	<p>Configuration of whether the schedule has an expiry date.</p> <ul style="list-style-type: none"><li>► <i>Active</i>: Schedule has a defined end.</li><li>► <i>Inactive</i>: Schedule is always valid from the start time.</li></ul> <p>Default: <i>inactive</i></p>
Schedule end	<p>End time of the schedule. The schedule is invalid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i></p>

90 | 128

Option	Description
	or selection via calendar element.

## DATE OF EXECUTION

Configuration of the days on which the schedule is executed.

Option	Description
<b>Recurrence</b>	<p>Selection of the recurrence configuration from drop-down list:</p> <ul style="list-style-type: none"> <li>▶ <i>Daily</i></li> <li>▶ <i>Weekly</i></li> <li>▶ <i>Monthly</i></li> </ul> <p>Default: <i>Daily</i></p>
<b>Interval</b>	<p>Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys.</p> <ul style="list-style-type: none"> <li>▶ Minimum: <i>1</i></li> <li>▶ Maximum: <i>2147483647</i></li> </ul> <p>Default: <i>1</i></p>
<b>Weekday</b>	<p>Configuration of the weekdays on which the schedule is executed.</p> <p>Selection by activating the checkbox in front of the day.</p> <p>Default: <i>Monday</i></p> <p><b>Note:</b> Only available for weekly recurrence.</p>
<b>Day</b>	<p>Configuration of the days in the month on which the plan is executed:</p> <ul style="list-style-type: none"> <li>▶ Entry in the text field directly or configuration using the arrow keys. Minimum: <i>1</i> Maximum: <i>28</i></li> <li>▶ Addition of further days: Click on the <b>+</b> symbol.</li> <li>▶ Deletion of days: Click on the <b>X</b> symbol.</li> </ul>

Option	Description
	<b>Note:</b> Only available for monthly recurrence.

## EXECUTION TIME: STIPULATED TIMES

Configuration for schedules with fixed times of execution.

Option	Description
<b>Recurrence</b>	Type of recurrence. Must be set to <i>stipulated times</i> .
<b>Hour</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 23</li> </ul> Default: 0
<b>Minute</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0
<b>Second</b>	Entry of the second. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0

## EXECUTION TIME: RECURRENCE PATTERN

Configuration for schedules with recurrence pattern.

Option	Description
<b>Recurrence</b>	Type of recurrence.  Must be set to <i>recurrence pattern</i> .
<b>Interval</b>	Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys.  <ul style="list-style-type: none"> <li>▶ Minimum: 1</li> <li>▶ Maximum: Depending on unit: Hour: 23 Minute: 1439 Second: 86399</li> </ul> Default: 1
<b>Time unit</b>	Selection of the scaling from a drop-down list:  <ul style="list-style-type: none"> <li>▶ <i>Hour</i></li> <li>▶ <i>Minute</i></li> <li>▶ <i>Second</i></li> </ul> Default: <i>Hour</i>
<b>Start time</b>	Definition of the start time via:  <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>
<b>End time</b>	Definition of the end time via:  <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>

## PARAMETER

Configuration of the execution times.

Option	Description
<b>From</b>	Selection of the start of the execution time from a drop-down list.

Option	Description
	<b>Note:</b> Not available for prediction models with a <b>value count</b> of 1.
<b>To</b>	Selection of the end of the execution time from a drop-down list.  <b>Note:</b> Not available for prediction models with a <b>value count</b> of 1.

## OUTPUT FORMAT

Configuration of the output format.

Option	Description
<b>Output format</b>	Selection of the output format from a drop-down list:  All output formats (on page 87) configured in the <b>Formats</b> area are available here.  <b>Note:</b> Only available for reports.

## 5.4 SQL elements

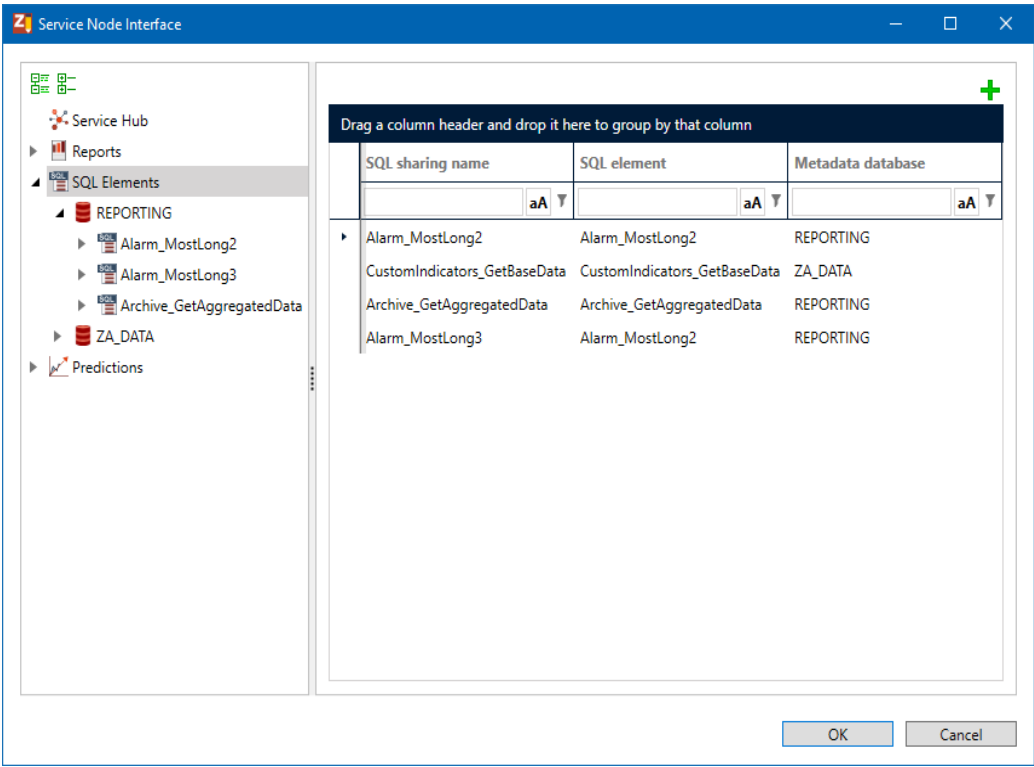
Display of the SQL elements to be configured for the **Service Grid** and addition of new SQL elements:

- ▶ Stored Procedures
- ▶ Scalar User Defined Functions
- ▶ Table Valued Functions

Existing SQL elements are administered in the subordinate objects.



**Note:** The SQL elements **System** and **zrs** are not available.



Options	Description
+	Clicking on the button opens the dialog to select an SQL element.
List of SQL elements	Shows all SQL elements available in the <b>Analyzer Service Node</b> .

ADD SQL ELEMENT

To add an SQL element:

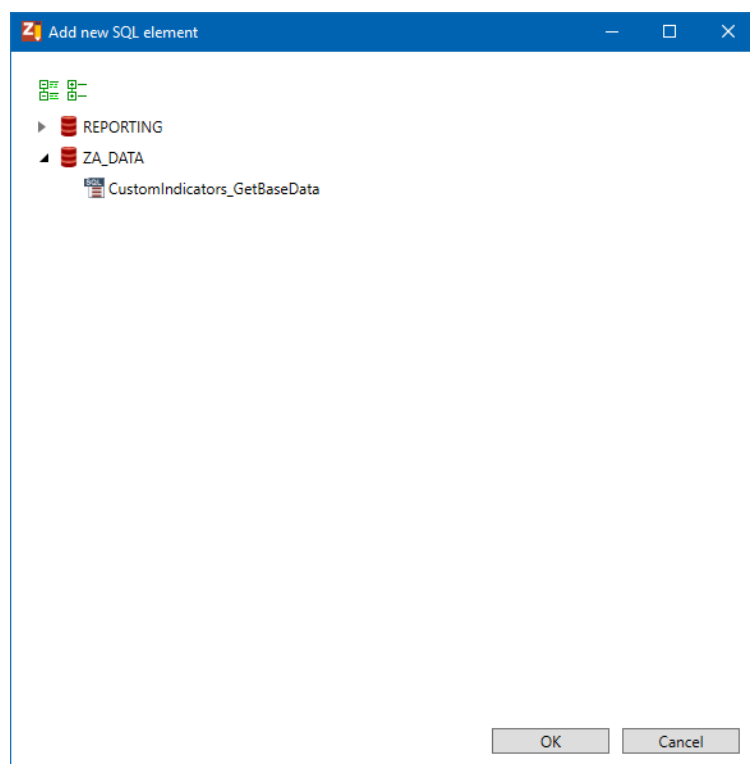
1. Click on the + symbol in the top right corner of the dialog.  
The selection dialog is opened.
2. Select the desired SQL element.
3. Click on **OK**.

The SQL element is added to the list.

Objects that require entries are highlighted in red. In doing so, it is always the higher-level objects that are highlighted.

**Note:** The report is only saved in the configuration file and on the service hub if the **Service Node Interface** dialog has been saved and closed with **OK**.

## ADD NEW SQL ELEMENT DIALOG



Option	Description
<b>Collapse/expand node symbol</b>	The objects subordinate to the main nodes can all be expanded or collapsed. Individual nodes can be expanded or collapsed by clicking with the mouse on the nodes.
<b>List of the SQL elements</b>	Display of the existing SQL elements. Selection via mouse click.
<b>OK</b>	Applies selection, closes the dialog and enters selected SQL element into the list.
<b>Cancel</b>	Discards selection and closes the dialog.

### 5.4.1 SQL element

Display of a selected SQL element.

You have the possibility:

- ▶ To amend the name and description.





- ▶ To duplicate SQL elements and thus configure variants.
- ▶ To remove SQL elements.
- ▶ To configure parameters.
- ▶ To display the column configuration.
- ▶ To create schedules (on page 87) for deployment.

CONFIGURE SQL ELEMENT DIALOG

Service Node Interface

Service Hub

▶ Reports

▶ SQL Elements

- ▶ REPORTING
  - ▶ Alarm\_MostLong2
  - ▶ Alarm\_MostLong3
  - ▶ Archive\_GetAggregatedData
- ▶ ZA\_DATA
- ▶ Predictions

Name:  
Alarm\_MostLong2

Description:

Database name:  
REPORTING

SQL element name:  
AlarmMostLong2

SQL element type:  
StoredProcedure

Remove

Copy

OK

Cancel

Option	Description
Name	Name of the SQL element.  Can be amended: <ul style="list-style-type: none"><li>▶ Maximum 128 characters.</li><li>▶ Must be unique within the <b>Service Node</b>.</li></ul>
Description	Description of the SQL element.  Can be amended: <ul style="list-style-type: none"><li>▶ Maximum 256 characters.</li></ul>
Database name	Name of the database from which the SQL element comes.

97 | 128

Option	Description
	Display only.
<b>SQL element name</b>	Name of the SQL element. Display only.
<b>SQL element type</b>	Type of the SQL element. Display only.
<b>Delete</b>	Clicking on the button removes the SQL element from the list.
<b>Copy</b>	Clicking on the button creates a duplicate of the SQL element. The following is applicable for duplicates: <ul style="list-style-type: none"> <li>▶ Parameters can be amended individually.</li> <li>▶ The name must be changed.</li> </ul>

## CREATE AND CONFIGURE VARIANTS

To create a new SQL element on the basis of a pre-existing SQL element:

1. Select the SQL element that is to serve as the basis.
2. Click on **Copy**.  
A duplicate of the SQL element is added.
3. Change the name of the duplicate.
4. Configure parameters and schedules.

### 5.4.2 Parameter

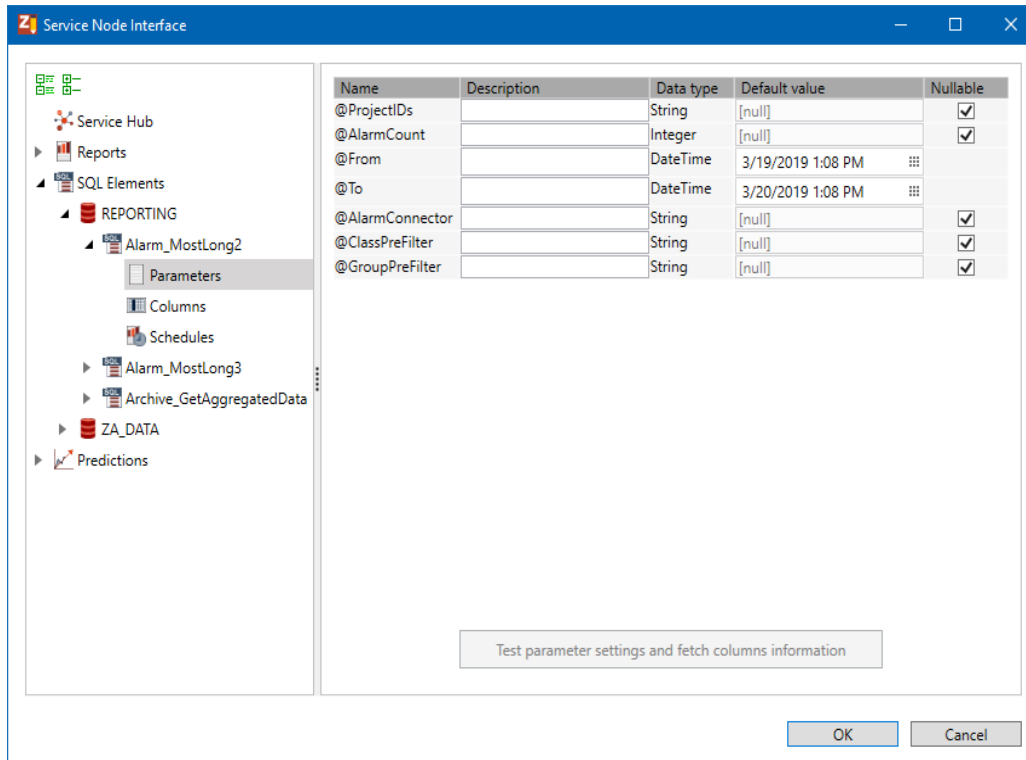
Configuration of the report parameters.



#### Attention

Parameter values are obtained via the connector. Therefore, at the time of the configuration, the appropriate zenon Runtime must be running with the correct zenon project and it must be contactable.

An error message is shown if there is no connection.



The screenshot shows the 'Service Node Interface' window. On the left, a tree view shows the hierarchy: Service Hub > Reports > SQL Elements > REPORTING > Alarm\_MostLong2 > Parameters. The 'Parameters' tab is selected. The main area displays a table with the following data:

Name	Description	Data type	Default value	Nullable
@ProjectIDs		String	[null]	<input checked="" type="checkbox"/>
@AlarmCount		Integer	[null]	<input checked="" type="checkbox"/>
@From		DateTime	3/19/2019 1:08 PM	
@To		DateTime	3/20/2019 1:08 PM	
@AlarmConnector		String	[null]	<input checked="" type="checkbox"/>
@ClassPreFilter		String	[null]	<input checked="" type="checkbox"/>
@GroupPreFilter		String	[null]	<input checked="" type="checkbox"/>

Below the table is a button labeled 'Test parameter settings and fetch columns information'. At the bottom right are 'OK' and 'Cancel' buttons.

The parameters that are available depend on the SQL element type.

To configure parameters:

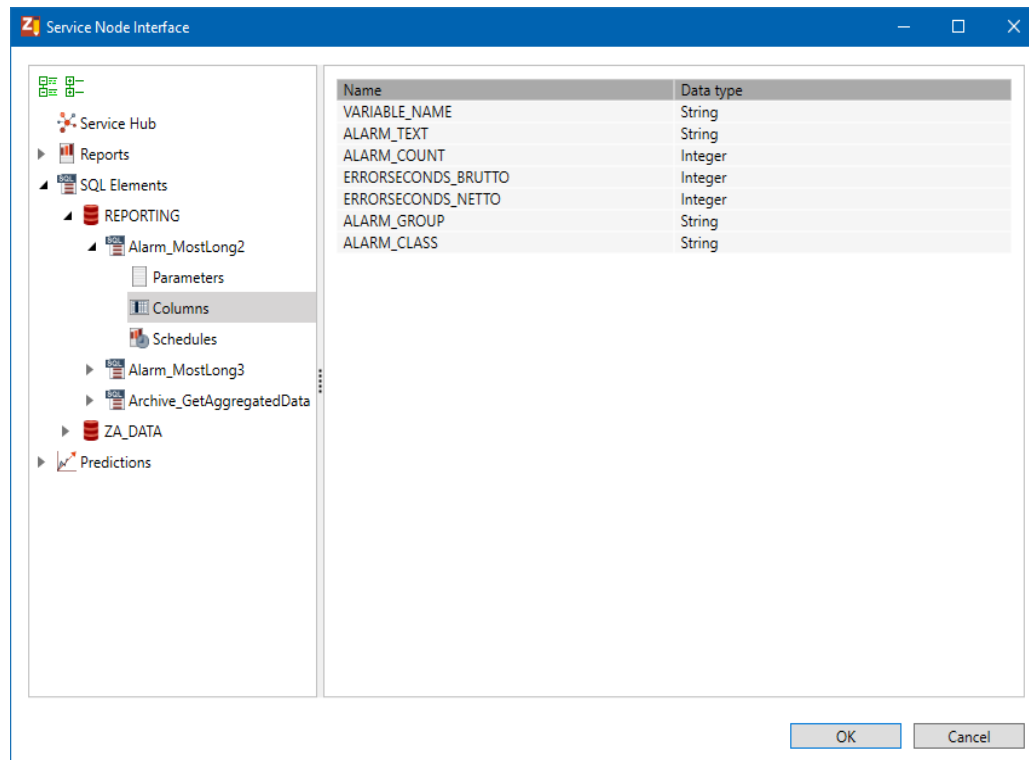
1. Select the respective parameters.
2. Enter the value for the parameter in the field or activate the checkbox for **NULL**.  
Use the calendar input dialog for date entries.
3. Enter a description for the respective parameter if required.
4. Click on the **Test Parameter Settings and Call Up Column Information** button.

A query is thus sent to the SQL server. If no error message is returned, all values are valid. Otherwise a corresponding error message is shown. This query must be carried out each time the parameter values are changed.

**Hint:** First check the respective SQL element in **SQL Management Studio** and determine the correct parameters.

### 5.4.3 Columns

Display of the columns used by the SQL element.



The columns are filled the first time the **Test Parameter Settings and Call Up Column Information** button is clicked on. Changes are applied each further time it is clicked on.

**Attention:** There is no column information for **Scalar User Defined Functions**. Only the data type is returned. You can find information about this in the main node (on page 96) of the SQL element.

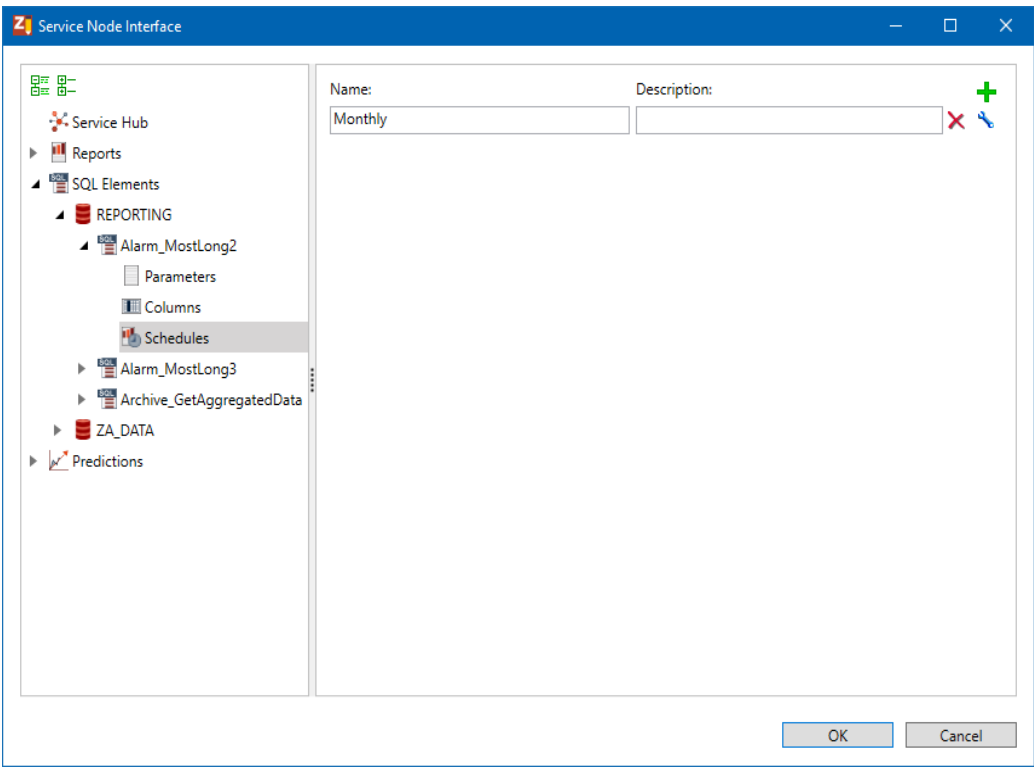
### 5.4.4 Schedules

SQL elements that have been provided for the **Service Grid** can be executed automatically via a schedule. To do this, each SQL element contains a **Schedules** node.

You can do the following here:

- ▶ Create new schedules
- ▶ Edit existing schedules
- ▶ Delete existing schedules

SCHEDULES DIALOG



Option	Description
List of schedules	Shows all configured schedules.
Symbol +	Clicking on the symbol opens the dialog to create and configure a schedule.
Symbol X	Clicking on the symbol deletes the selected schedule without a request for confirmation.
Wrench symbol	Clicking on the symbol opens the dialog to configure the selected schedule.

CREATE NEW SCHEDULE

To create a new schedule:

1. Click on the symbol +.  
The dialog for configuring the schedule (on page 113) is opened
2. Configure the schedule with:
  - ▶ **Schedule duration**
  - ▶ **Date of execution**

- ▶ **Execution time**

- ▶ **Parameter**

3. Confirm the configuration by clicking **OK**.  
The schedule is entered into the list.
4. Please define a name!  
The name must be unique within the report.

## EDIT SCHEDULE

To edit a schedule:

1. Select the schedule.
2. Click on the symbol with the wrench symbol.  
The dialog for configuring the schedule is opened.
3. Configure the schedule.
4. Confirm the configuration by clicking on **OK**.

## DELETE SCHEDULE

To delete a schedule:

1. Select the schedule.
2. Click on the symbol **X**.  
The schedule is deleted.

### 5.4.4.1 Configure a schedule

You can configure the schedules for SQL elements here:

- ▶ Schedule duration
- ▶ Date of execution
- ▶ Execution time

► Parameter

Schedules

Schedule duration

Schedule start:

1/31/2019

☐ Schedule expires

Schedule end:

12/31/9999

Execution day

Recurrence:

Daily

Interval:

1

Execution time

Recurrence:

Set times

Hour:

0

Minute:

0

Second:

0

+

×

Parameters

@From

Now

0

Minute

@To

Now

0

Minute

OK

Cancel

SCHEDULE DURATION

Configuration of the time range for which the schedule is applied.

Option	Description
Schedule start	<p>Start time of the schedule. The schedule is valid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i> or selection via calendar element.</p>
Schedule expires	<p>Configuration of whether the schedule has an expiry date.</p> <ul style="list-style-type: none"><li>► <i>Active</i>: Schedule has a defined end.</li><li>► <i>Inactive</i>: Schedule is always valid from the start time.</li></ul> <p>Default: <i>inactive</i></p>
Schedule end	<p>End time of the schedule. The schedule is invalid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i></p>

Option	Description
	or selection via calendar element.

## DATE OF EXECUTION

Configuration of the days on which the schedule is executed.

Option	Description
<b>Recurrence</b>	<p>Selection of the recurrence configuration from drop-down list:</p> <ul style="list-style-type: none"> <li>▶ <i>Daily</i></li> <li>▶ <i>Weekly</i></li> <li>▶ <i>Monthly</i></li> </ul> <p>Default: <i>Daily</i></p>
<b>Interval</b>	<p>Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys.</p> <ul style="list-style-type: none"> <li>▶ Minimum: <i>1</i></li> <li>▶ Maximum: <i>2147483647</i></li> </ul> <p>Default: <i>1</i></p>
<b>Weekday</b>	<p>Configuration of the weekdays on which the schedule is executed.</p> <p>Selection by activating the checkbox in front of the day.</p> <p>Default: <i>Monday</i></p> <p><b>Note:</b> Only available for weekly recurrence.</p>
<b>Day</b>	<p>Configuration of the days in the month on which the plan is executed:</p> <ul style="list-style-type: none"> <li>▶ Entry in the text field directly or configuration using the arrow keys. Minimum: <i>1</i> Maximum: <i>28</i></li> <li>▶ Addition of further days: Click on the <b>+</b> symbol.</li> <li>▶ Deletion of days: Click on the <b>X</b> symbol.</li> </ul>



Option	Description
	<b>Note:</b> Only available for monthly recurrence.

## EXECUTION TIME: STIPULATED TIMES

Configuration for schedules with fixed times of execution.

Option	Description
<b>Recurrence</b>	Type of recurrence. Must be set to <i>stipulated times</i> .
<b>Hour</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 23</li> </ul> Default: 0
<b>Minute</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0
<b>Second</b>	Entry of the second. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0

## EXECUTION TIME: RECURRENCE PATTERN

Configuration for schedules with recurrence pattern.

Option	Description
<b>Recurrence</b>	Type of recurrence.  Must be set to <i>recurrence pattern</i> .
<b>Interval</b>	Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys.  <ul style="list-style-type: none"> <li>▶ Minimum: 1</li> <li>▶ Maximum: Depending on unit: Hour: 23 Minute: 1439 Second: 86399</li> </ul> Default: 1
<b>Time unit</b>	Selection of the scaling from a drop-down list:  <ul style="list-style-type: none"> <li>▶ <i>Hour</i></li> <li>▶ <i>Minute</i></li> <li>▶ <i>Second</i></li> </ul> Default: <i>Hour</i>
<b>Start time</b>	Definition of the start time via:  <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>
<b>End time</b>	Definition of the end time via:  <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>

## PARAMETER

Configuration of the execution times.

Option	Description
<b>From</b>	Selection of the start of the execution time.  <b>Note:</b> Not available for prediction models with a

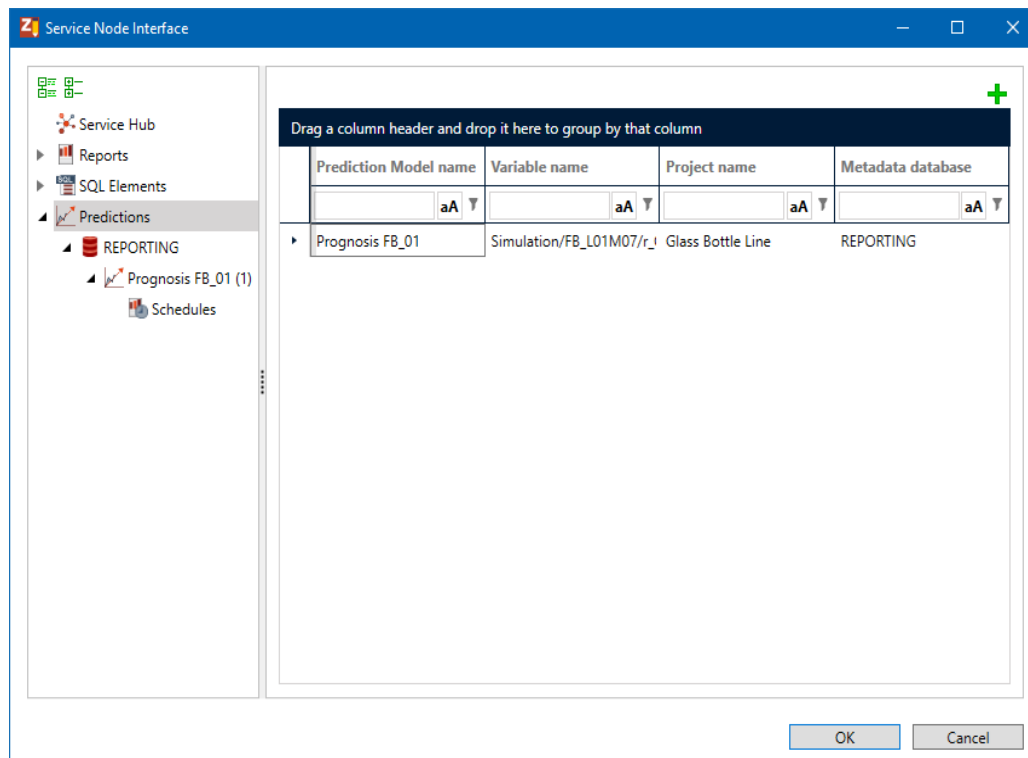
Option	Description
	<b>value count</b> of 1.
<b>To</b>	Selection of the end of the execution time.  <b>Note:</b> Not available for prediction models with a <b>value count</b> of 1.

## 5.5 Prediction

Display and addition of prediction models. Existing prediction models are administered in the subordinate objects.

Structure:

- ▶ Level 1: Main nodes for the predictions.
- ▶ Level 2: Database in which the predictions are created.
- ▶ Level 2: Selected prediction model. The ID of the prediction model in the database is shown in brackets.
- ▶ Level 3: Schedules for the prediction model.



Options	Description
+	Clicking on the button opens the dialog to select a prediction model.
List of prediction models	Displays all prediction models available in the <b>Service Node</b> .

## ADD PREDICTION MODEL

To add a prediction model:

1. Click on the + symbol in the top right corner of the dialog.

The selection dialog is opened.

2. Select the desired prediction model.
3. Click on OK.

The prediction model is added to the list.

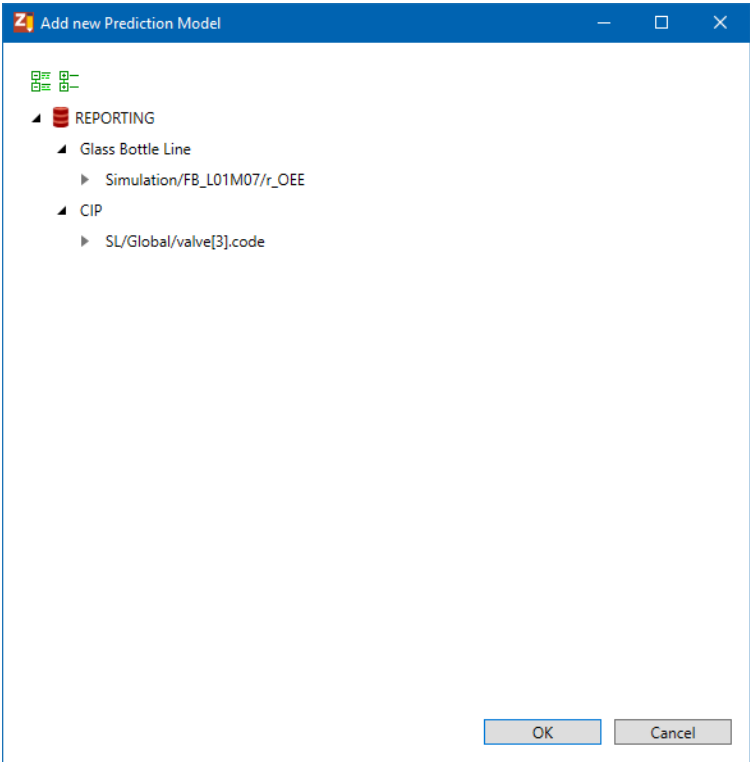
Objects that require entries are highlighted in red. In doing so, it is always the higher-level objects that are highlighted.

4. Configure at least one schedule.

**Note:** The prediction model is only saved in the configuration file and on the service hub if the **Service Node Interface** dialog is saved and closed with **OK**.



ADD NEW PREDICTION MODEL DIALOG



Option	Description
Collapse/expand node symbol	The objects subordinate to the main nodes can all be displayed as expanded or collapsed. Individual nodes are expanded or collapsed with a mouse click.
Selection tree of the prediction models	Display of the prediction models present. Selection via mouse click.  Structure: <ul style="list-style-type: none"><li>▶ 1. Node: Database name.</li><li>▶ 2. Node: Name zenon project.</li><li>▶ 3. Node: Variable name.</li><li>▶ 4. Node: Name of prediction model.</li></ul>
OK	Applies selection, closes the dialog and enters tthe selected prediction model into the list.
Cancel	Discards selection and closes the dialog.



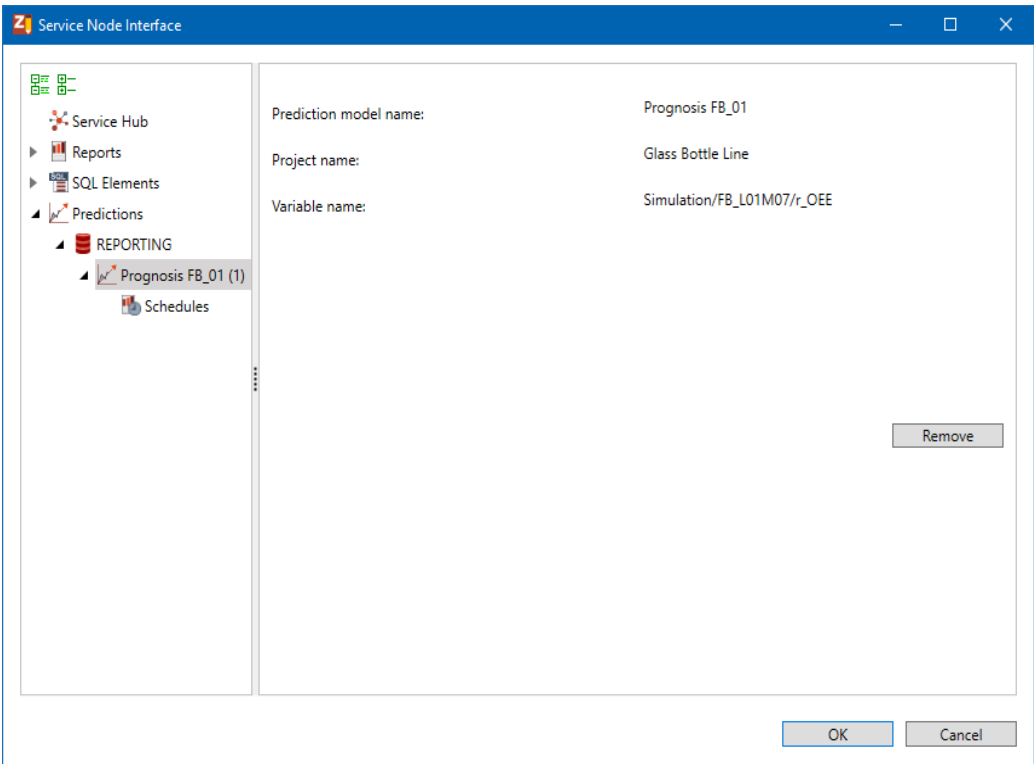
### 5.5.1 Prediction model

Display of a selected prediction model.

You have the possibility:

- ▶ To see the prediction model name, zenon project name and variable name.
- ▶ To remove prediction models.

### CONFIGURE THE PREDICTION MODEL DIALOG



Option	Description
Prediction model name	Name of the prediction model. Display only.
Project name	Name of the zenon Logic project Display only.
Variable name	Name of the variable. Display only.
Delete	Clicking on the button removes the prediction model from the list.

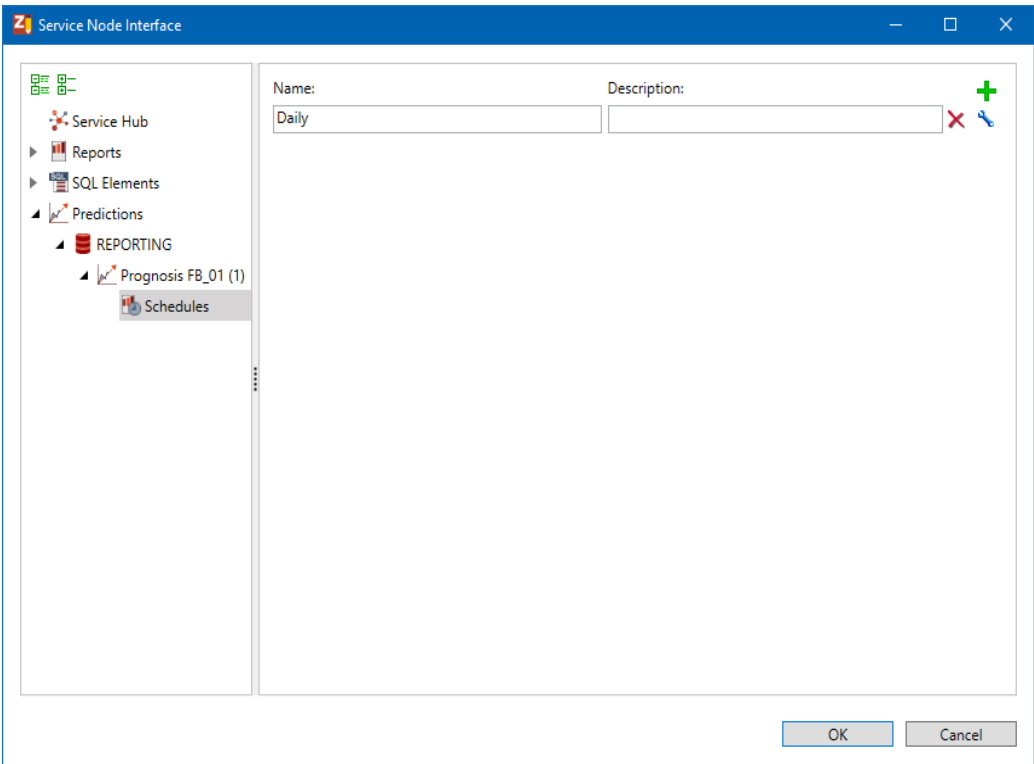
### 5.5.2 Schedules

Prediction models that have been prepared for the **Service Grid** are executed automatically by means of a schedule. Each prediction model has a **Schedules** node for this. Each prediction model must have at least one schedule.

You can do the following here:

- ▶ Create new schedules
- ▶ Edit existing schedules
- ▶ Delete existing schedules

#### SCHEDULES DIALOG



Option	Description
List of schedules	Shows all configured schedules.
Symbol +	Clicking on the symbol opens the dialog to create and configure a schedule.
Symbol X	Clicking on the symbol deletes the selected

Option	Description
	schedule without a request for confirmation.
<b>Wrench</b> symbol	Clicking on the symbol opens the dialog to configure the selected schedule.

## CREATE NEW SCHEDULE

To create a new schedule:

1. Click on the symbol **+**.  
The dialog for configuring the schedule (on page 113) is opened
2. Configure the schedule with:
  - ▶ **Schedule duration**
  - ▶ **Date of execution**
  - ▶ **Execution time**
  - ▶ **Parameter**
3. Confirm the configuration by clicking **OK**.  
The schedule is entered into the list.
4. Please define a name!  
The name must be unique within the report.

## EDIT SCHEDULE

To edit a schedule:

1. Select the schedule.
2. Click on the symbol with the wrench symbol.  
The dialog for configuring the schedule is opened.
3. Configure the schedule.
4. Confirm the configuration by clicking on **OK**.

## DELETE SCHEDULE

To delete a schedule:

1. Select the schedule.
2. Click on the symbol **X**.  
The schedule is deleted.



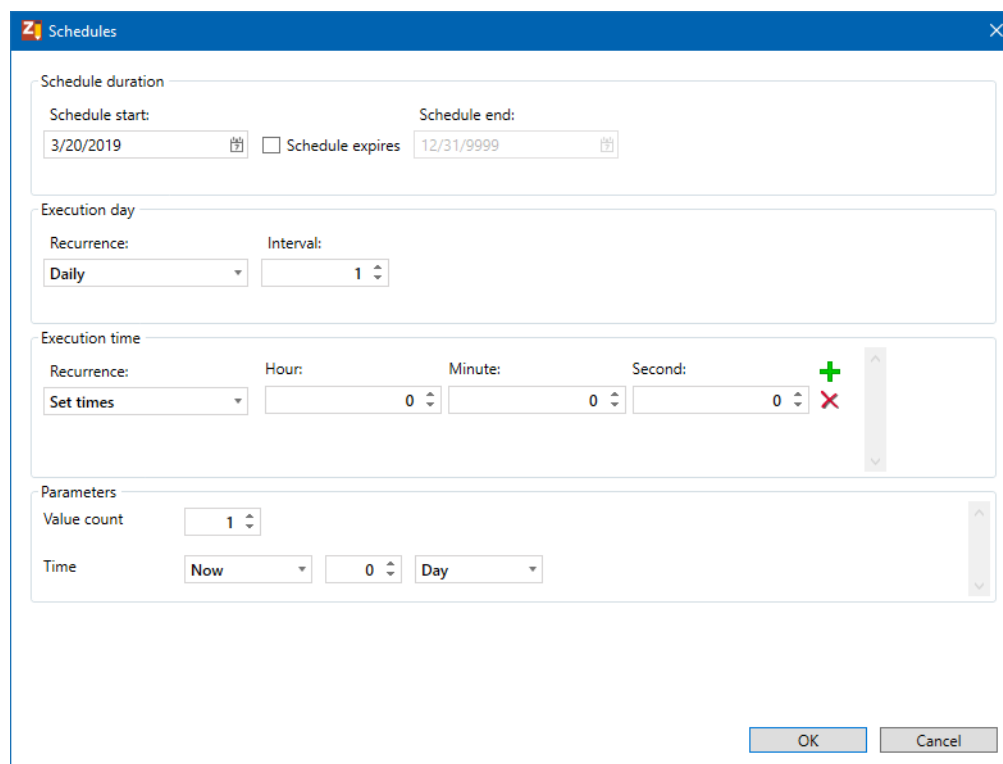
### 5.5.2.1 Configure a schedule

Here you configure the schedules for predictions with:

- ▶ Schedule duration
- ▶ Date of execution
- ▶ Execution time
- ▶ Parameter

The schedule configuration differs depending on whether a value is configured or several values are used.

#### One value:



**Schedules**

**Schedule duration**

Schedule start: 3/20/2019 ☐ Schedule expires Schedule end: 12/31/9999

**Execution day**

Recurrence: Daily Interval: 1

**Execution time**

Recurrence: Set times Hour: 0 Minute: 0 Second: 0 + X

**Parameters**

Value count: 1

Time: Now 0 Day

OK Cancel

Several values:

Schedules

Schedule duration

Schedule start:

5/31/2019

☐ Schedule expires

Schedule end:

12/31/9999

Execution day

Recurrence:

Daily

Interval:

1

Execution time

Recurrence:

Set times

Hour:

0

Minute:

0

Second:

0

Parameters

Value count

2

From

Now

0

Day

To

Now

1

Day

OK

Cancel

SCHEDULE DURATION

Configuration of the time range for which the schedule is applied.

Option	Description
Schedule start	<p>Start time of the schedule. The schedule is valid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i> or selection via calendar element.</p>
Schedule expires	<p>Configuration of whether the schedule has an expiry date.</p> <ul style="list-style-type: none"><li>▶ <i>Active</i>: Schedule has a defined end.</li><li>▶ <i>Inactive</i>: Schedule is always valid from the start time.</li></ul> <p>Default: <i>inactive</i></p>
Schedule end	<p>End time of the schedule. The schedule is invalid from this point.</p> <p>Entry in the field directly in the format <i>dd.mm.yyyy</i></p>

114 | 128

Option	Description
	or selection via calendar element.

## DATE OF EXECUTION

Configuration of the days on which the schedule is executed.

Option	Description
<b>Recurrence</b>	<p>Selection of the recurrence configuration from drop-down list:</p> <ul style="list-style-type: none"> <li>▶ <i>Daily</i></li> <li>▶ <i>Weekly</i></li> <li>▶ <i>Monthly</i></li> </ul> <p>Default: <i>Daily</i></p>
<b>Interval</b>	<p>Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys.</p> <ul style="list-style-type: none"> <li>▶ Minimum: <i>1</i></li> <li>▶ Maximum: <i>2147483647</i></li> </ul> <p>Default: <i>1</i></p>
<b>Weekday</b>	<p>Configuration of the weekdays on which the schedule is executed.</p> <p>Selection by activating the checkbox in front of the day.</p> <p>Default: <i>Monday</i></p> <p><b>Note:</b> Only available for weekly recurrence.</p>
<b>Day</b>	<p>Configuration of the days in the month on which the plan is executed:</p> <ul style="list-style-type: none"> <li>▶ Entry in the text field directly or configuration using the arrow keys. Minimum: <i>1</i> Maximum: <i>28</i></li> <li>▶ Addition of further days: Click on the <b>+</b> symbol.</li> <li>▶ Deletion of days: Click on the <b>X</b> symbol.</li> </ul>

Option	Description
	<b>Note:</b> Only available for monthly recurrence.

## EXECUTION TIME: STIPULATED TIMES

Configuration for schedules with fixed times of execution.

Option	Description
<b>Recurrence</b>	Type of recurrence. Must be set to <i>stipulated times</i> .
<b>Hour</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 23</li> </ul> Default: 0
<b>Minute</b>	Entry of the hour. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0
<b>Second</b>	Entry of the second. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 0</li> <li>▶ Maximum: 59</li> </ul> Default: 0

## EXECUTION TIME: RECURRENCE PATTERN

Configuration for schedules with recurrence pattern.

Option	Description
<b>Recurrence</b>	Type of recurrence.  Must be set to <i>recurrence pattern</i> .
<b>Interval</b>	Configuration of the intervals of the recurrence. Direct entry in the field or configuration using arrow keys. <ul style="list-style-type: none"> <li>▶ Minimum: 1</li> <li>▶ Maximum: Depending on unit: Hour: 23 Minute: 1439 Second: 86399</li> </ul> Default: 1
<b>Time unit</b>	Selection of the scaling from a drop-down list: <ul style="list-style-type: none"> <li>▶ <i>Hour</i></li> <li>▶ <i>Minute</i></li> <li>▶ <i>Second</i></li> </ul> Default: <i>Hour</i>
<b>Start time</b>	Definition of the start time via: <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>
<b>End time</b>	Definition of the end time via: <ul style="list-style-type: none"> <li>▶ <b>Hour</b></li> <li>▶ <b>Minute</b></li> <li>▶ <b>Second</b></li> </ul>

## PARAMETER

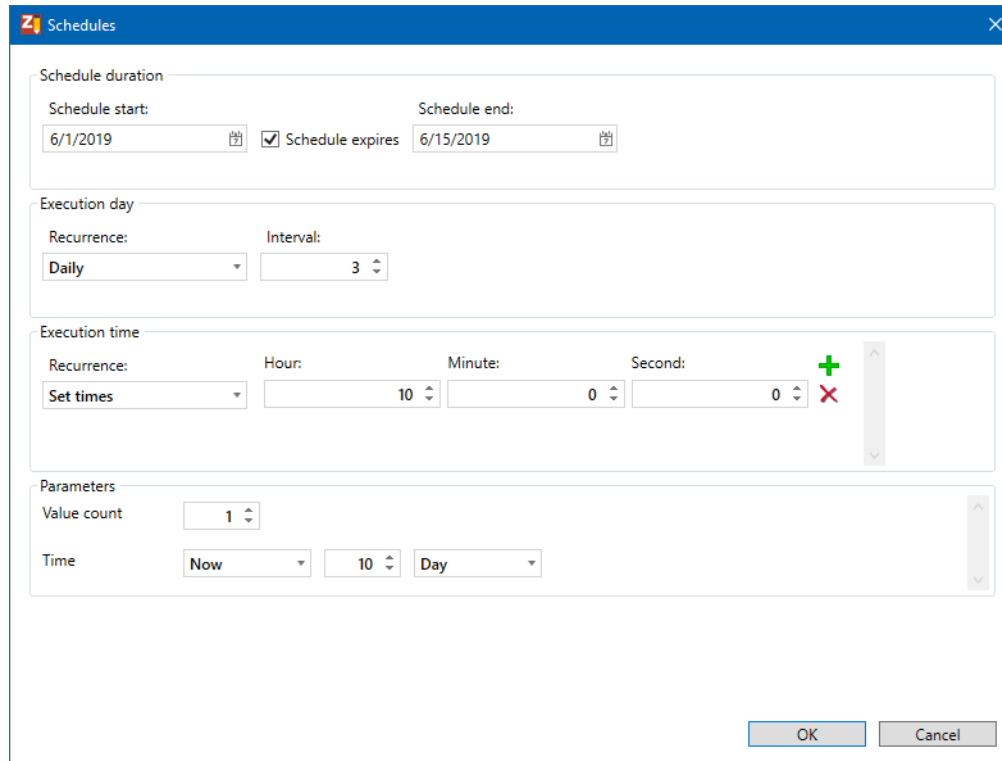
Configuration of the execution times.

Option	Description
<b>Number of values</b>	Number of values that are set for the time period. Direct entry in the field or selection using arrow keys.

Option	Description
	<ul style="list-style-type: none"><li>▶ Minimum: <i>1</i></li><li>▶ Maximum: <i>1000</i></li></ul> Default: <i>1</i>
Enter	Configuration of the prediction time. <b>Note:</b> Only available for prediction models with a <b>value count</b> of <i>1</i> .
From	Selection of the start of the execution time. <b>Note:</b> Not available for prediction models with a <b>value count</b> of <i>1</i> .
To	Selection of the end of the execution time. <b>Note:</b> Not available for prediction models with a <b>value count</b> of <i>1</i> .

## 6 Example schedule configuration

Configuration of a schedule for prediction models. Schedules for reports and SQL elements are configured along the same lines.



**Schedules**

**Schedule duration**

Schedule start: 6/1/2019 ☐ ☐ ☒ Schedule expires ☐ 6/15/2019 ☐

**Execution day**

Recurrence: Daily Interval: 3

**Execution time**

Recurrence: Set times Hour: 10 Minute: 0 Second: 0

**Parameters**

Value count: 1

Time: Now 10 Day

OK Cancel

### Configuration:

- ▶ Start of schedule: 01. 06. 2019
- ▶ End of schedule: 15. 06. 2019
- ▶ Time interval: Every 3 days
- ▶ Time of execution: 10:00:00 AM
- ▶ Parameter:
  - ▶ Number of values: 1
  - ▶ Entry: now, 10 days

### Execution:

Execution on/at	Result
6/1/2019, 10:00:00 AM	Calculates prediction for June 11, 2019
6/4/2019, 10:00:00 AM	Calculates prediction for 6/14/2019
6/7/2019, 10:00:00 AM	Calculates prediction for 6/17/2019

Execution on/at	Result
6/10/2019, 10:00:00 AM	Calculates prediction for 6/20/2019
5/13/2019, 10:00:00 AM	Calculates prediction for 6/23/2019

## 7 Service Node Configuration Tool

You use the **Service Node Configuration Tool** to establish a connection to the **Hub Controller**. The **Service Node** to be configured gets the required access data in the form of a **Certificate Bundles** from it. Before starting the **Service Node Configuration Tool** for the first time, you must install the necessary certificate.

### INSTALL CERTIFICATE FROM THE HUB CONTROLLER

In order to install the certificate:

1. Log in to the **Hub Controller**.
2. Download the CA certificate.
3. Install the certificate in the Windows Certificate Store **Trusted Root Certification Authorities**.
4. Start the SNCT.

### START SERVICE NODE CONFIGURATION TOOL

To start the **Service Node Configuration Tool**:

1. Open the zenon **Startup Tool**.
2. Click on the **Tools** button.  
The **Start external application** dialog is opened.
3. In the **Available applications (current folder)** section, select the **Service Node Configuration Tool** entry.
4. Click **Start**.  
The **Service Node Configuration Tool** is opened.

### OPERATE SERVICE NODE CONFIGURATION TOOL

You use the **Service Node Configuration Tool** to configure the connections to **Service Nodes** via the tabs:

- ▶ **Selection**: Selection of the **Service Nodes** that are to be connected to the **Hub Controller**.

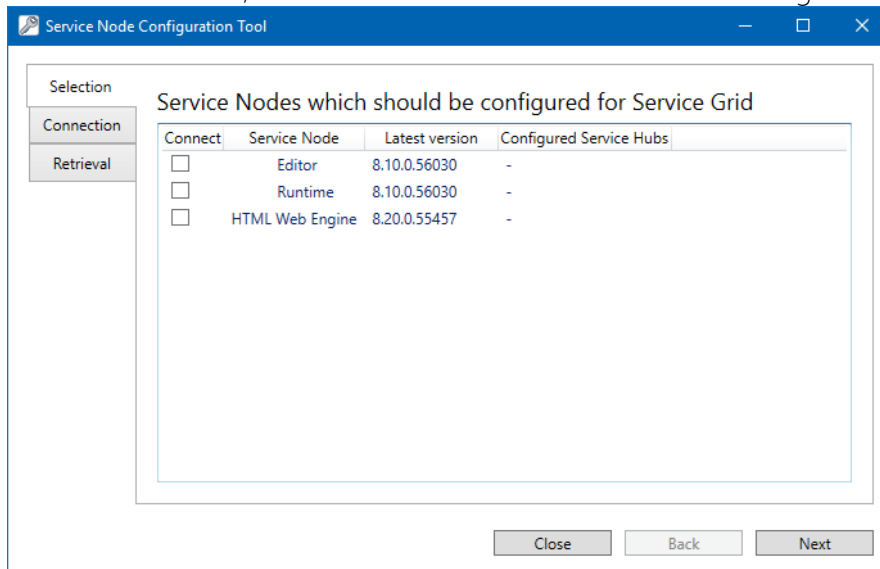


- ▶ **Connection:** Entry of the connection parameters to the **Hub Controller**.
- ▶ **Retrieval:** Check of the connection and configuration of the **Service Nodes**.

These can only be edited in order. In order to be possible to switch to the next step, the current tab must be configured correctly. Errors are highlighted with a red frame the next time there is an attempt to switch further.

To configure the connection from a **Service Node** to the **Hub Controller**:

1. In the **Selection** tab, select the **Service Nodes** that are to be configured.



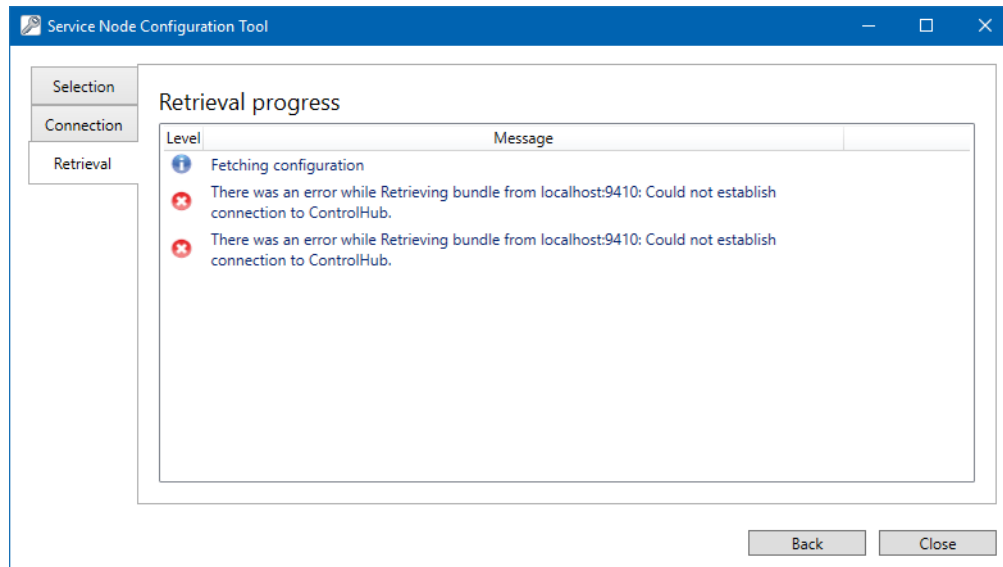
2. Click **Next**.
3. Enter the address under which the hub controller can be contacted.
4. If the default port is not used:
  - a) Activate the **Custom** option.
  - b) Enter the port to be used.
5. Enter a user name and password.
6. Optional: Enter an alias.
7. Click on **Execute**.

The connection data entered is validated.

The **Retrieval** tab is opened.

A connection to the hub controller is established and the access data for the previously-selected **Service Nodes** are queried from the **Hub Controller**. These **Service Nodes** are then configured with their respective access data.

The individual steps are displayed in the form of LOG messages. These inform you of successful execution, as well as problems with the establishment of a connection to the **Hub Controller**, the data query or the configuration of the **Service Nodes**.



8. Close the dialog by clicking on **Close**.

## 8 Metadata Synchronizer

The **Metadata Synchronizer** sends metadata from zenon to a zenon Analyzer metadata database.

Requirements: zenon Analyzer 3.30 or higher and zenon 8.10 or higher.

In contrast to the **Analyzer Export Wizard**, the **Metadata Synchronizer** is implemented in zenon and zenon Analyzer directly. This results in many benefits, most of all:

- ▶ The transfer runs much more quickly.
- ▶ Increased stability and tolerance of errors.
- ▶ Version independence starting from zenon 8.10 and zenon Analyzer 3.30.

### DATA TRANSFER

The **Metadata Synchronizer** transfers from zenon to zenon Analyzer:

- ▶ Alarm/event classes and alarm groups
- ▶ Users
- ▶ Equipment models
- ▶ Network:  
If the **Network** property is active, configurations for **Server 1** and **Server 2**.

- ▶ Projects
- ▶ Project contents:
  - ▶ Variables
  - ▶ Archives
  - ▶ Shifts
  - ▶ Status texts
  - ▶ Efficiency class models
- ▶ Sankey models
- ▶ Waterfall models

**The following is applicable for the target during transfer:**

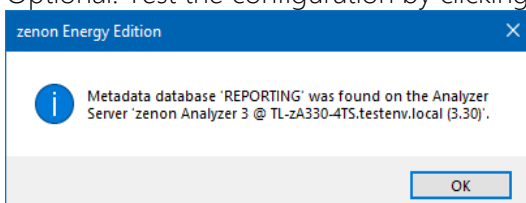
- ▶ Objects that no longer exist are deleted.  
Exception: Projects  
During deletion, insofar as possible, all dependent objects are also deleted.
- ▶ Existing objects are updated.
- ▶ New objects are added.

**Note:** Objects that have been created by **ZAMS** or the **Metadata Editor** are not changed.

## 8.1 Configuration

To transfer data for zenon Analyzer in the Service Grid:

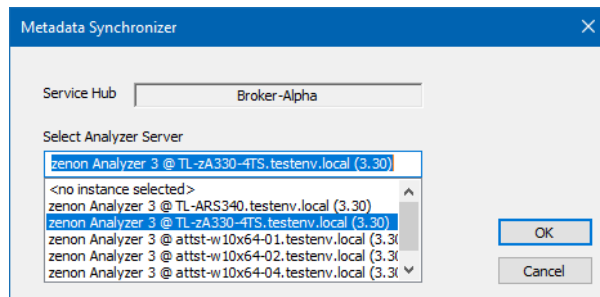
1. Ensure that a valid connection has been selected in the zenon Editor for the **Service Hub** project property in the **Network/Service Grid** node.
2. Navigate to the **Metadata Synchronizer** node in the project properties.
3. Select a **Analyzer instance**.  
click on the ... button to open the dialog to select an Analyzer server (on page 124).
4. To do this, select a **Metadata database**.  
Click on the ... button to open the dialog to select and configure a database (on page 125).
5. Optional: Test the configuration by clicking on the ... button in the **Test connection** property.



The **Metadata Synchronizer** can now be executed in the zenon Editor.

### 8.1.1 Analyzer Server selection dialog

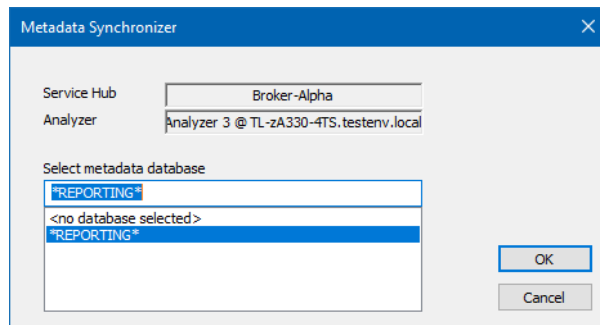
You select the Analyzer server in the service hub with this dialog.



Option	Description
Service Hub	Display of the service hub configured in the <b>Service Hub</b> property.
Select Analyzer server	<p>Direct entry of the Analyzer servers or selection from a drop-down list:</p> <ul style="list-style-type: none"> <li>▶ Selection of an Analyzer server: Applies selected instance.</li> <li>▶ <i>&lt;No Analyzer server selected&gt;</i>: Removes the configured Analyzer server.</li> <li>▶ <i>&lt;Apply from the global project&gt;</i>: Applies the configuration selected in the global project.</li> </ul> <p><b>Hinweis:</b> In order for an Analyzer server to be able to be selected, a valid connection to the <b>Service Hub</b> must be configured. This is established with the <b>Service Node Configuration Tool</b> (on page 120).</p>
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

## 8.1.2 Database selection dialog

You select the metadata database with this dialog.



Option	Description
<b>Service Hub</b>	Display of the service hub configured in the <b>Service Hub</b> property.
<b>Analyzer Server</b>	Display of the analyzer server configured in the <b>Analyzer instance</b> property.
<b>Select metadata database</b>	<p>Direct entry of the metadata database or selection from drop-down list:</p> <ul style="list-style-type: none"> <li>▶ Selection of a metadata database: Applies selected metadata database.</li> <li>▶ <i>&lt;No database selected&gt;</i>: Removes the configured database.</li> <li>▶ <i>&lt;Apply from the global project&gt;</i>: Applies the configuration selected in the global project.</li> </ul> <p><b>Note:</b> In order for a metadata database to be selected, a valid connection to the <b>Service Hub</b> and to the <b>Analyzer Server</b> must be configured.</p>
<b>OK</b>	Applies settings and closes the dialog.
<b>Cancel</b>	Discards all changes and closes the dialog.

## 8.2 Execution

The **Metadata Synchronizer** can be executed and stopped.

To transfer metadata to a database:

1. Go to the **Extras** menu in the zenon Editor.

2. Select the **Execute Metadata Synchronizer** entry.

The **Metadata Synchronizer** is started. Metadata is collated and transferred to the configured database.

The actions and the result are displayed in the output window.

To stop the **Metadata Synchronizer**:

1. Go to the **Extras** menu in the zenon Editor.
2. Select the **Stop Metadata Synchronizer** entry.

The **Metadata Synchronizer** is stopped.

## SYNCHRONIZATION RULES

### VISUAL NAME

Behavior when synchronizing the Metadata Editor and the **Metadata Synchronizer**:

1. A variable does not exist in the Metadata database:
  - ▶ **Visual name** in the zenon Editor is empty: The variable name is entered as the **Visual name** in the Metadata database.
  - ▶ **Visual name** in the zenon Editor is configured: The **Visual name** in the zenon Editor is entered as the **Visual name** in the Metadata database.
2. A variable already exists in the Metadata database and the visual name corresponds with the variable name:
  - ▶ **Visual name** in the zenon Editor is empty: The variable name is entered as the **Visual name** in the Metadata database.
  - ▶ **Visual name** in the zenon Editor is configured: The **Visual name** in the zenon Editor is entered as the **Visual name** in the Metadata database.
3. A variable already exists in the Metadata database and the visual name does not correspond with the variable name:
  - ▶ **Visual name** in the zenon Editor is empty: The visual name in the Metadata database remains unchanged.
  - ▶ **Visual name** in the zenon Editor is configured: The **Visual name** in the zenon Editor is entered as the Visual name in the Metadata database. Visual names changed in the Metadata Editor are overwritten.

The name in the zenon Editor is always used as the visual name for projects. When updating renamed projects (if the **Visual name** property remains empty), the zenon Analyzer overwrites none of the changes made with the **Metadata Editor**.

## DESCRIPTIONS

If descriptions for objects applied from the **Metadata Synchronizer** from zenon are empty, the descriptions present in the database remain unchanged.

This applies for:

- ▶ Equipment groups
- ▶ Alarm/Event classes
- ▶ Alarm/Event groups
- ▶ User
- ▶ Projects
- ▶ Archives
- ▶ Variables (**Identification** is used)

## NORMALISATION

Data for efficiency classes must be normalised for use in the zenon Analyzer. Data from the zenon Editor are never normalised. Normalisation can only be configured in the Metadata Editor.

During synchronization, the **Metadata Synchronizer** checks whether the efficiency class model already exists in the Metadata database:

- ▶ efficiency class model is not present: no normalisation is present. This must be configured in the Metadata Editor.
- ▶ Efficiency class model is present: The normalisation present in the Metadata database remains unchanged.

## SANKEY DIAGRAMS AND WATERFALL MODELS

Sankey diagrams and waterfall models are validated after checking the variables and before sending the data.

In doing so, the following applies:

- ▶ Connections in Sankey diagrams may only use variables that are contained in project, archive, variable and compression during synchronization.
- ▶ Waterfall models may only use variables that are contained in project and variable, but not archives during synchronization. For waterfall charts, it is sufficient if the variable is contained in any archive.

## VARIABLES

All variables are checked before synchronization to see if they have to be synchronized. A variable is only synchronized if it meets at least one of these conditions:

- ▶ The variable has an assigned reaction matrix:  
In addition to the default entry, this reaction matrix contains at least one other entry that generates the alarms (AML) or events (CEL).
- ▶ The variable does not have an assigned reaction matrix:  
It has at least one activated limit value that generates an alarm (AML) or an event (CEL).
- ▶ The variable is contained in at least one archive.

## VALIDATION

Entries from zenon are largely validated before transfer. Errors are corrected. If correction is not possible, the respective object is excluded from synchronization. All validation errors are logged in the diagnosis viewer as a warning and displayed in the output window in the zenon Editor.

This applies for:

- ▶ Archives
- ▶ Reaction matrices
- ▶ Alarm classes
- ▶ Alarm groups
- ▶ Equipment models
- ▶ User
- ▶ Projects
- ▶ Sankey diagrams
- ▶ Shift models
- ▶ Variables
- ▶ References to equipment models
- ▶ Waterfall charts