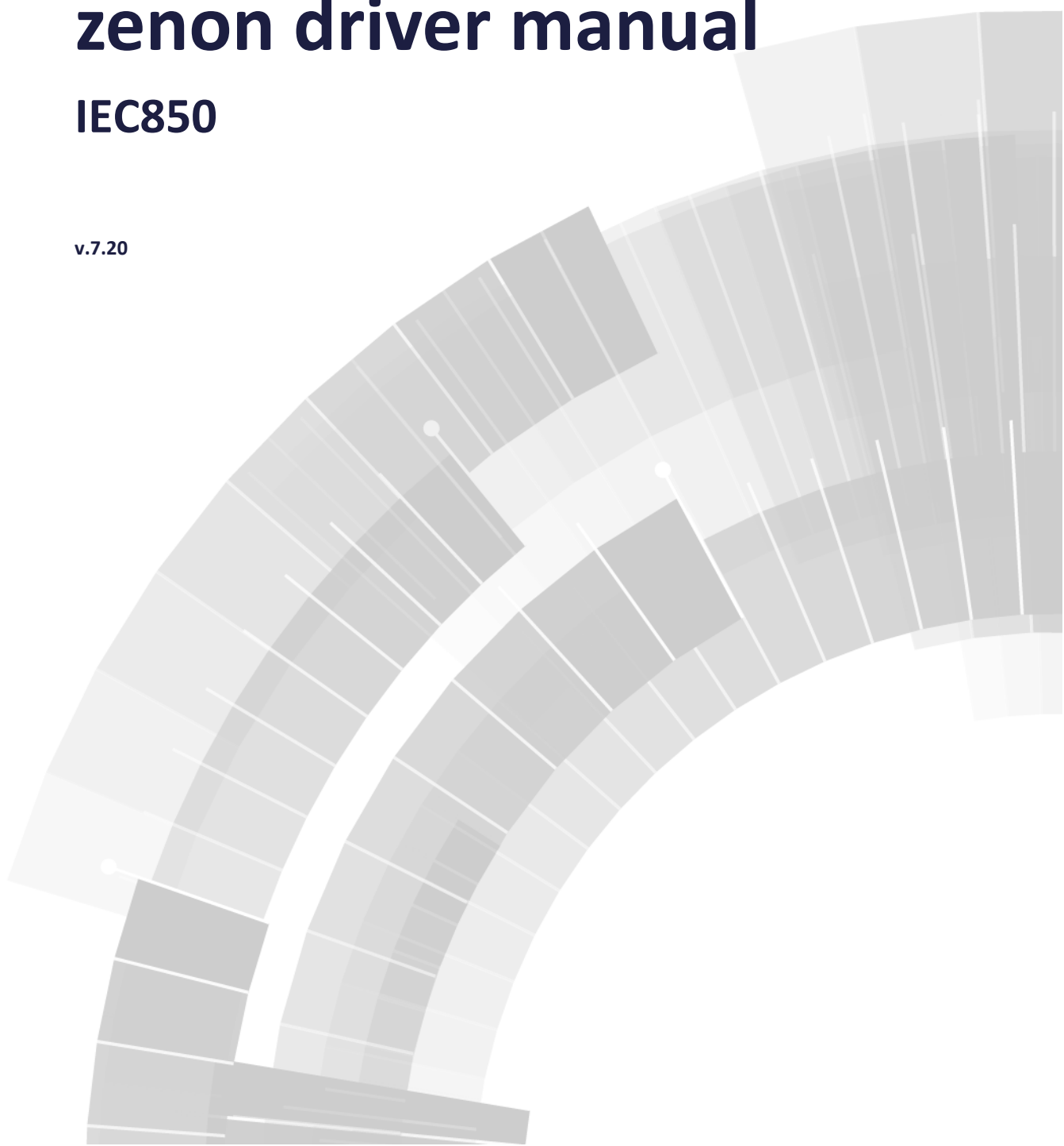


zenon driver manual

IEC850

v.7.20





©2015 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

1. Welcome to COPA-DATA help	5
2. IEC850	5
3. IEC850 - Data sheet.....	6
4. Driver history	7
5. Requirements.....	9
5.1 PC	9
6. Configuration	10
6.1 Creating a driver.....	10
6.2 Settings in the driver dialog	12
6.2.1 General	12
6.2.2 Driver dialog basic settings.....	15
6.2.3 Connections.....	21
7. Creating variables.....	36
7.1 Creating variables in the Editor.....	36
7.2 Addressing.....	38
7.3 Driver objects and datatypes	41
7.3.1 Driver objects	41
7.3.2 Assignment of data types	43
7.4 Creating variables by importing	44
7.4.1 Online import	45
7.4.2 Offline import	48
7.4.3 XML import.....	51
7.4.4 DBF Import/Export	51
7.5 Driver variables	56
8. Driver-specific functions	63
8.1 Double Point Value Mapping	72
8.2 Service Tracking	74

9. Reporting	76
9.1 Unbuffered Reporting	78
9.2 Buffered Reporting.....	80
9.3 Dynamic Data Sets	82
10. Driver commands	83
11. Error analysis.....	86
11.1 Analysis tool	86
11.2 Check list	87
12. Appendix A - Description of the 'Functional Constraints' (FCs):	88
13. Appendix B - Abbreviations for data object/data attribute.....	89
14. Appendix C - Conformance statement	99
14.1 Protocol implementation conformance statement (PICS)	100
14.2 Model implementation conformance statement (MICS)	106
15. Appendix D - PIXIT for COPA-DATA IEC61850 Client	107

1. Welcome to COPA-DATA help

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. IEC850

Communication between the driver and the PLC is based on the IEC 61850 protocol with client/server services via TCP/IP (A1/T1 profile). The driver acts as a client (master) when communicating.



Attention

Configurations that have been edited with a driver from version 3400 cannot be opened with older drivers anymore.

3. IEC850 - Data sheet

General:	
Driver file name	IEC850.exe
Driver name	IEC 61850 driver
PLC types	IEC 61850 compatible PLCs
PLC manufacturer	ABB; Siemens; IEC; SAT; Sprecher Automation; Kalki; GE Multilin;

Driver supports:	
Protocol	IEC 61850;
Addressing: Address-based	-
Addressing: Name-based	x
Spontaneous communication	x
Polling communication	x
Online browsing	x
Offline browsing	x
Real-time capable	x
Blockwrite	-
Modem capable	-
Serial logging	-
RDA numerical	-
RDA String	-

Requirements:	
Hardware PC	standard Network Adapter
Software PC	XML-Lite is required, it is included in Microsoft Internet Explorer 7.0; xmllite.dll, can also be downloaded from Microsoft Homepage.
Hardware PLC	-
Software PLC	-
Requires v-dll	-

Platforms:	
Operating systems	Windows CE 6.0, Embedded Compact 7; Windows 7, 8, 8.1 Server 2008R2, Server 2012, Server 2012R2;
CE platforms	x86; ARM;

4. Driver history

A list of the main changes that were made to the driver documentation. The driver version (driver exe file version) corresponds to pre-compilations for test and documentation purposes and not the approved version of the driver with the described functions.

If you have questions about your project, please contact the Consulting Team by email. support@copadata.com (mailto:support@copadata.com).

Date	Driver version	Change
8/31/2006	100	Driver and documentation created
4/11/2007	400	Support for all models (SBO) of the control model
10/4/2007	700	Unbuffered Reporting
1/9/2008	1100	Version for Windows CE
1/10/2008	1100	File Transfer
1/28/2008	1200	Buffered Reporting
2/4/2008	1300	Auto reset of ctlNum
5/29/2008	1900	Alternative use of identification as external name
9/11/2008	2300	Check condition by a command from Check attribute
12/12/2008	2500	reedited driver documentation
12/23/2008	2500	recognition of duplicated datasets in Reports; log messages for reported or polled variables
12/29/2008	2500	configurable names of BRCBs
5/12/2009	2900	bitstrings also as STRING variables
3:11 AM 09	3300	dynamic Data Sets option: Use preconfigured TrgOps
12/30/2009	3600	option: Do not purge BRCB buffer at start
1/7/2010	3600	Reports not only from LLN0
2/1/2010	3700	off-line import also from *.cid and *.scd files
2/17/2010	3700	Origin is sent to RT before all other attributes of a data object from Report
3/24/2010	3800	Additional Cause Diagnosis
3/24/2010	3900	possibility to split up Select and Operate for Command Processing module
3/24/2010	3900	mapping of Time quality bits ClockFailure and ClockNotSynchronized
11/8/2010	4000	Off-line Import: LNs embedded directly into IEDs without LDs are ignored
11/12/2010	4100	support of data type "Check" (Tissue 10); PICS reviewed; PIXIT enclosed (KEMA Certification)
5/4/2011	4200	UNICODE STRING in UTF-8 character encoding; consideration of all possible TimeAccuracy values; support of Authentication (ISO 8650-1)

5/6/2011	4300	support of alternative IP address
5/19/2011	4400	local time stamp for variables with FC not ST or MX
10/31/2011	4500	individual TrgOps bits setting for Unbuffered Reports
11/2/2011	4600	driver setting for Origin orCat
11/8/2011	4700	individual TrgOps bits setting for Buffered Reports
7/16/2012	4230	support of ARRAY OF

DRIVER VERSIONING

The versioning of the drivers was changed with zenon 7.10. There is a cross-version build number as of this version. This is the number in the 4th position of the file version,
For example: 7.10.0.4228 means: The driver is for version 7.10 service pack 0, and has the build number 4228.

Expansions or error rectifications will be incorporated into a build in the future and are then available from the next consecutive build number.



Example

A driver extension was implemented in build 4228. The driver that you are using is build number 8322. Because the build number of your driver is higher than the build number of the extension, the extension is included. The version number of the driver (the first three digits of the file version) do not have any significance in relation to this. The drivers are version-agnostic

5. Requirements

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

5.1 PC

XML Lite, which is part of Microsoft Internet Explorer 7.0, is necessary; xmllite.dll can also be downloaded separately from the Microsoft website.

This driver supports a connection via the standard network card of the PC. Make sure that the PLC and the PC are in the same network range and that the subnet masks are set accordingly on both devices.

6. Configuration

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



Information

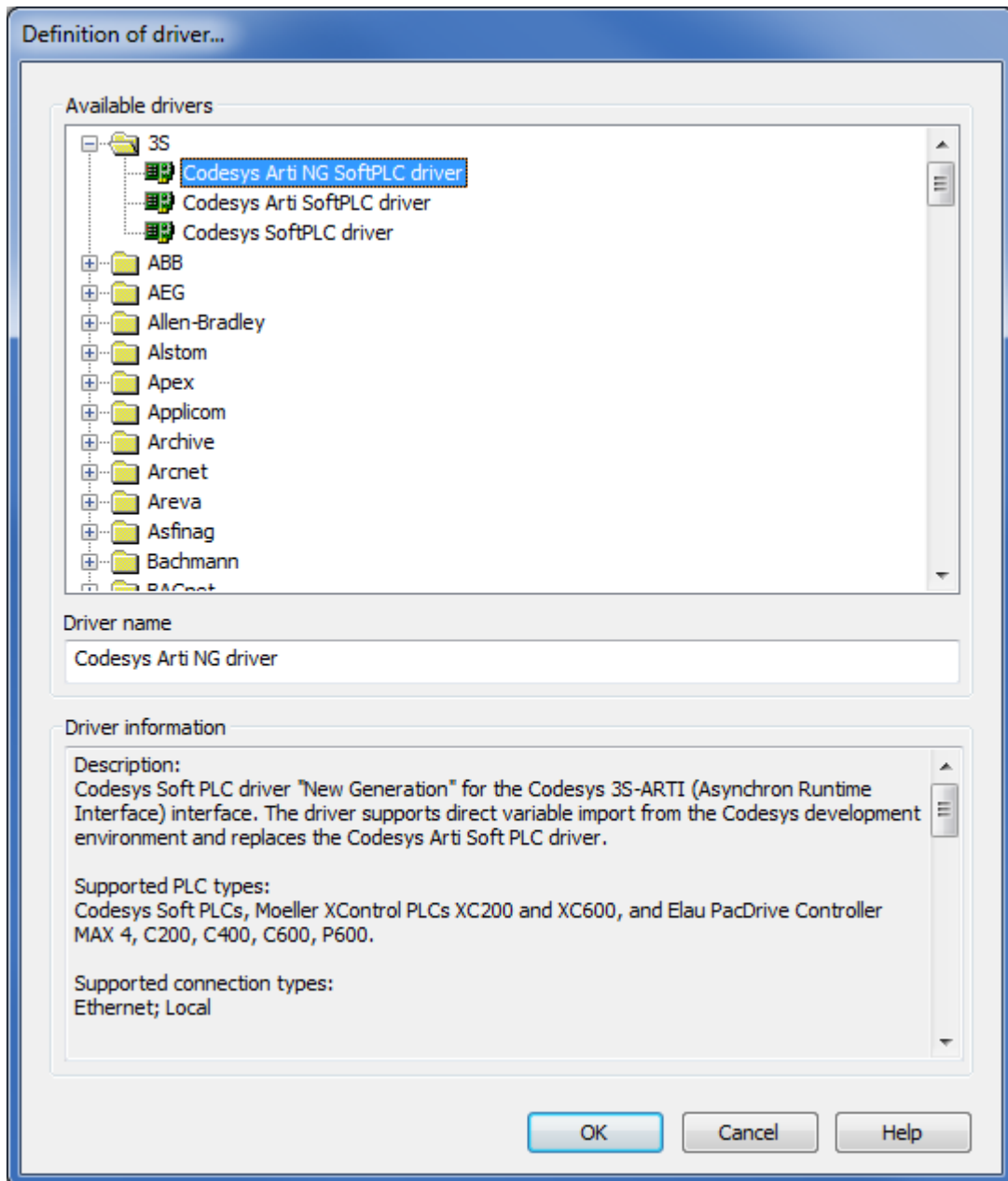
Find out more about further settings for zenon variables in the chapter Variables (main.chm::/15247.htm) of the online manual.

6.1 Creating a driver

In order to create a new driver:

1. Right-click on **Driver** in the Project Manage and select **Driver new** in the context menu.

2. In the following dialog the control system offers a list of all available drivers.



3. Select the desired driver and give it a name:
 - The driver name has to be unique, i.e. if one and the same driver is to be used several times in one project, a new name has to be given each time.
 - 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 with **OK**. In the following dialog the single configurations of the drivers are defined.

Only the respective required drivers need to be loaded for a project. Later loading of an additional driver is possible without problems.



Information

For new projects and for existing projects which are converted to version 6.21 or higher, the following drivers are created automatically:

- ▶ Internal
- ▶ MathDr32
- ▶ SysDrv.

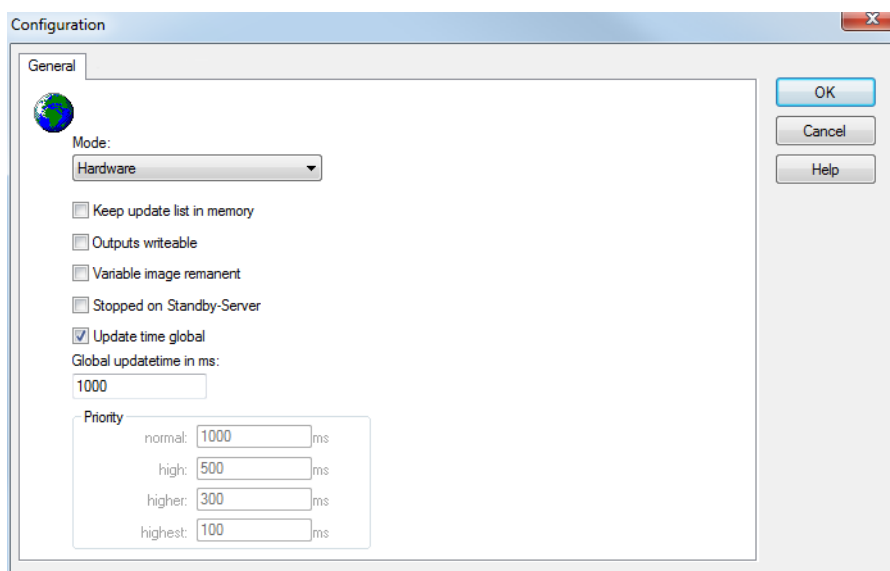
▶

6.2 Settings in the driver dialog

You can change the following settings of the driver:

6.2.1 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.



Parameters	Description
Mode	<p>Allows to switch between hardware mode and simulation mode</p> <ul style="list-style-type: none"> ▶ Hardware: <p>A connection to the control is established.</p> ▶ Simulation static <p>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.</p> ▶ Simulation - counting <p>No communication between to the control is established, the values are simulated by the driver. In this modus the driver increments the values within a value range automatically.</p> ▶ Simulation - programmed <p>N 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).</p>
Keep update list in the memory	<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>
Output can be written	<p>Active: Outputs can be written.</p> <p>Inactive: Writing of outputs is prevented.</p> <p>Note: Not available for every driver.</p>
Variable image remanent	<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 mode hardware if:</p> <ul style="list-style-type: none"> ▶ one of the states S_MERKER_1(0) up to S_MERKER8(7), REVISION(9), AUS(20) or ERSATZWERT(27) is active <p>The variable image is always saved if:</p> <ul style="list-style-type: none"> ▶ the variable is of the object type Driver variable ▶ the driver runs in simulation mode. (not programmed simulation) <p>The following states are not restored at the start of the Runtime:</p> <ul style="list-style-type: none"> ▶ SELECT(8) ▶ WR-ACK(40) ▶ WR-SUC(41) <p>The mode Simulation - programmed at the driver start is not a criterion in order to restore the remanent variable image.</p>
Stop on Standby Server	<p>Setting for redundancy at drivers which allow only on communication connection. For this the driver is stopped at the Standby Server and only started at the upgrade.</p> <p>Attention: If this option is active, the gapless archiving is no longer guaranteed.</p> <p>Active: 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 switched off (statusverarbeitung.chm::/24150.htm) but an empty value. This prevents that at the upgrade to the Server irrelevant values are created in the AML, CEL and Historian.</p> <p>Note: 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>
Global Update time	<p>Active: The set Global update time in ms is used for all variables in the project. The priority set at the variables is not used.</p> <p>Inactive: The set priorities are used for the individual variables.</p>
Priority	<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 allocation to the variables takes place separately in the settings of the variable properties.</p> <p>The communication of the individual variables are graduated in respect of importance or necessary topicality using the priorities.</p>

Thus the communication load is distributed better.

Attention: Priority classes are not supported by each driver. For example, drivers that communicate spontaneously do not support it.

CLOSE DIALOG

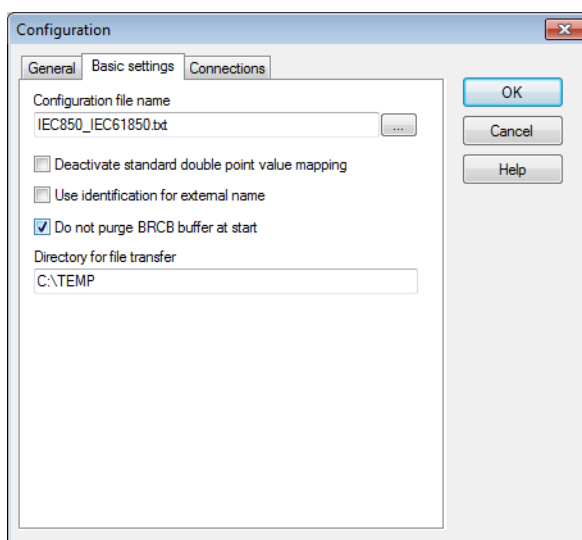
Parameters	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

UPDATE TIME FOR CYCLICAL DRIVERS

The following applies for cyclical drivers:

For **Set value**, **Advising** of variables and **Requests**, a read cycle is immediately triggered for all drivers - regardless of the set update time. This ensures that the value is immediately available for visualization after writing. Update times can therefore be shorter than pre-set for cyclical drivers.

6.2.2 Driver dialog basic settings



Parameters	Description
Configuration file name	<p>Name of the file in which the driver-specific configurations are saved. Click on the ... button to open the dialog for selecting a drop folder.</p> <p>Note: You can read about the structure of this configuration file in the Driver configuration file (on page 17) chapter.</p>
Deactivate standard double point value mapping	<ul style="list-style-type: none"> ▶ Inactive: The values of Double Point Values are adjusted to the operating elements of zenon. Use this configuration if you want to use the modules of zenon Energy. You can find details in the Double Point Value Mapping (on page 72) chapter.. ▶ Active: The values of the Double Point Values are forwarded to zenon as they are. In this case, you cannot use the command processing function of zenon Energy or ALC for example. <p>Default: Inactive</p>
Use identification for external name:	<ul style="list-style-type: none"> ▶ Inactive: The variable name is used for addressing. ▶ Active: The driver uses the entries of the 'identification' variable property for addressing. <p>Default: Inactive</p>
Do not purge BRCB buffer at start	<ul style="list-style-type: none"> ▶ Active: The buffer of the BRCBs are not deleted when the connection is first established after Runtime has started. Requirement: For this BRBC, the last entry ID received has been saved. <p>Only affects the first attempt after the start. Is ignored on repeated attempts.</p> <p>More in the Buffered reporting (on page 80) chapter.</p>
Originator category (orCat)	<p>The driver sends configured orCat values as commands.</p> <p>Possible settings:</p> <ul style="list-style-type: none"> ▶ 1 - Bay-control ▶ 2 - Station-control ▶ 3 - Remote-control ▶ 4 - Automatic-bay ▶ 5 - Automatic-station ▶ 6- Automatic-remote ▶ 7 - Maintenance <p>Default: 2- Station-control</p>

Directory for file transfer	Directory for file transfer. All files loaded are saved in this folder.
------------------------------------	---

Structure of a driver's configuration file as defined in **Configuration file name/**.

CONFIGURATION FILE

Line	Description
1	Number of configured servers
2 – (m-1)	Server configuration (see server configuration)
m – n	Possible further server configuration(s)

SERVER CONFIGURATION

Line	Description	Example
1	Start of a server configuration (= *** SERVER ***)	*** SERVER ***
2	Number of items in this server configuration, abbreviated CNTSRVITEMS	44
3	Net address	1
4	Server name	UP-W7x86-2.testenv.local
5	Primary IP address	192.168.250.22
6	Primary IP port	102
7	Read interval in milliseconds (polling rate)	1000
8	Calling AE qualifier	12
9	Called AE qualifier	12
10	Calling AP title[0]	1
11	Called AP title[0]	1
12	Calling AP title[1]	1
13	Called AP title[1]	1
14	Calling AP title[2]	1
15	Called AP title[2]	999
16	Calling AP title[3]	999
17	Called AP title[3]	1
18	Calling AP title[4]	-1
19	Called AP title[4]	1
20	Calling AP title[5]	-12851
21	Called AP title[5]	-1
22	Calling AP title[6]	-12851
23	Called AP title[6]	-12851
24	Calling AP title[7]	-12851

25	Called AP title[7]	-12851
26	Calling AP title[8]	-12851
27	Called AP title[8]	-12851
28	Calling AP title[9]	-12851
29	Called AP title[9]	-12851
30	Maximum number of Unbuffered Reports (URCBs) that the driver activates with automatic assignment per Logical Device	10
31	* - in newer configurations With old configuration: name of the assigned, separated with commas Buffered Reports (BRCBs)	*
32	0 = Subsequently configured triggers are used as an option 1 = Trigger options which are pre-configured in the IEC61850 Server are used.	0
33	0 = The RCB instances of the server are identified by name. 1 = Report ID that is used instead of the report name in the dialog for RCB assignment .	0
34	0 = no ISO-Authentication used 1 = If active, the driver sends the Authentication String at establishing the connection.	0
35	Authentication String	
36	Alternative IP address	
37	Alternative IP port	0
38	TrgOp data-change : 0 = inactive ; 1 = active	1
39	TrgOp quality-change : 0 = inactive ; 1 = active	1
40	TrgOp data-update : 0 = inactive ; 1 = active	0
41	TrgOp integrity : 0 = inactive ; 1 = active	0
42	TrgOp general-interrogation : 0 = inactive ; 1 = active	1
43	0 = Normal GetNameList 1 = The driver reads the object model by requesting data objects (DO) for each available Logical Node , in every server in the IEC61850 standard defined Functional Constraint (FC) (FC).	0
44	Integrity period	7000
45	Buffer time	500
46	Optional fields	73
47	Cycle in seconds in which an attempt is made to activate RCBs that were not activated successfully again. Only present if CNTRSVITEMS >=	

	is 45	
48	Number of client configurations	
49 – (m-1)	Client configuration (see Client configuration)	
m – n	Possible further client configurations	

CLIENT CONFIGURATION

Line	Description	Beispiel
1	Start of a client configuration (= *** CLIENTCFG ***)	*** CLIENTCFG ***
2	Number of Items in this client configuration	1
3	Name of the computer on which the driver is running that receives the reports	CDSBG104
4	Number of RCB configurations	2
5 – (i-1)	RCB Configuration (see RCB configuration)	
i – (j-1)	Possible further RCB configurations	
y	Number of dynamic dataset configurations	
k – (l-1)	Dynamic dataset configuration (see dynamic dataset configuration)	
l – m	Possible further dynamic dataset configuration	

RCB CONFIGURATION

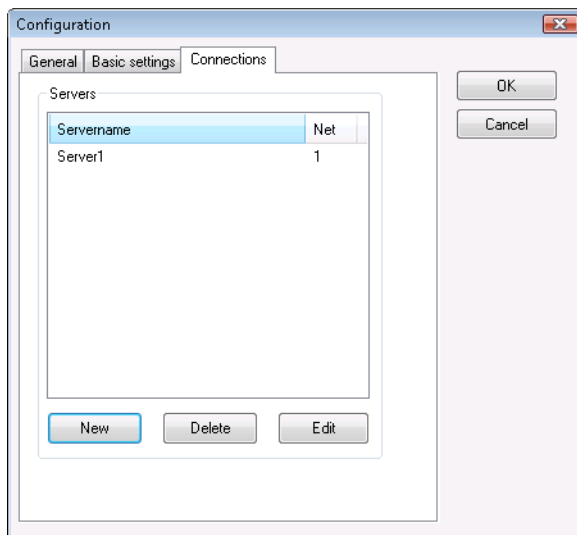
Line	Description	Example
1	Start of an RCB configuration (= *** RCBCFG ***)	*** RCBCFG ***
2	Number of items in this RCB configuration	2
3	RCB Name or ID	UP_CTRL/LLN0/urcb_QxCsWI1_Pos02[RP]
4	Name of the dynamic data set.	NEW_DYN_DATASET

DYNAMIC DATASET CONFIGURATION

Line	Description	Example
1	Start of a dynamic dataset (= *** DATASET ***)	*** DATASET ***
2	Number of items in this dynamic dataset configuration	1
3	Name of the dynamic data set	NEW_DYN_DATASET
4	Number of Object References of the dynamic dataset	2
5	Object Reference	UPCTRL/Q1CSWI1\$CF\$Pos
6 – n	Possible further Object Reference .	UPCTRL/Q1CSWI1\$ST\$Pos

6.2.3 Connections

The connections of the IEC850 driver are set in the Connections tab of the driver dialog.



Parameters	Description
Server	List of connections to the IEC 61850 server that have already been configured (to the control objects).
New	Creates a new connection to an IEC 61850 server. Opens the 'Server' dialog.
Delete	Removes the selected connection from the list.
Edit	Opens the 'Server' dialog to edit the selected connection

Server

Clicking on **New** or **Edit** in the dialog connection (on page 21) opens the dialog for the configuration of a connection to an IEC 61850 server:

Server

General

Server name

Net address

Polling rate

S1

1

1000 ms

Link Layer (TCP)

IP address (primary)

IP Port (primary)

192.168.0.55

102

IP address (redundant)

IP Port (redundant)

0

Application Layer (ACSE)

Calling AP title

Calling AE qualifier

1.1.1.999

12

Called AP title

Called AE qualifier

1.1.999.1.1

12

☐ Use Authentication

☐ GetNameList on DO

Reporting

Max. auto used URCBs

RCBs enable retries

10

7 s

☐ Use preconfigured (SCL) options

☐ Use Report-ID for RCB assignment

☒ TrgOp data-change

☒ TrgOp quality-change

☐ TrgOp data-update

☐ TrgOp integrity

☒ TrgOp general-interrogation

RCB assignment/dynamic datasets

Hostname

Integrity Period

Buffer Time

7000 ms

500 ms

MYWORKSTATION01

OptFlds

New

Edit

Delete

OK

Cancel

GENERAL

Parameters	Description
Server name	Freely definable name. Is used for the names of variables when variables are imported.
Net address	Corresponds to the Net address property in variable configuration. Maximum value: 65535
Polling rate	<p>Defines the update frequency in milliseconds. The driver supports this rate to poll data that is not in active reports. At this a possible failure of the IEC61850 server is also considered.</p> <p>Maximum value: 4294967295 (2³²-1) If a higher value is entered, it is automatically changed to 1 when saved.</p> <p>Default: 1000</p> <p>Note: At least one variable should always be polled even if all data of the project is present in the report. Otherwise a problem with the IEC61850 server cannot be detected if the TCP/IP connection still exists. Create a *LLN0/Mod/c1tModel [CF] variable for this and ensure that the variable is always signed in with the driver.</p> <p>Attention: Exchange of data in Runtime does not depend on this setting or on the global update time. This is set to 100ms and fixed.</p>

LINK LAYER (TCP)

Parameters	Description
Link Layer (TCP)	Connection settings
IP address (primary)	IP address of the server to which a connection is to be made.
IP Port (primary)	Port via which the connection will be established. Default: 102
IP address (redundant)	<p>Alternative IP address. If the connection to the first IP address fails, the alternative IP address will be used after the net error waiting time has passed (20-30 sec., depending on the network). The alternative address will be kept until the driver is restarted (via driver functions or restart of the zenon Logic Runtime) or until the connection fails. Then, the first IP address will be used again.</p> <p>Attention: At least one variable, such as *LLN0/Mod/stVal1 [ST] should always be polled to detect a connection failure.</p>
IP Port (redundant)	Alternative IP port.

APPLICATION LAYER (ACSE)

Parameters	Description
Application Layer (ACSE)	Settings for the ACSE.
Calling AP title	Settings according to ISO 8650-1(ACSE), the value for the <code>OSI ACSE AP Title</code> of the client in the <code>Universal Identifier</code> notation. Should not normally be amended.
Calling AE qualifier	Settings according to ISO 8650-1(ACSE). Should not be used in standard cases.
Called AP title	Settings according to ISO 8650-1(ACSE), <code>OSI-AP-Title</code> of the server in the format for <code>Universal Identifier</code> . Should not normally be amended.
Called AE qualifier	Settings according to ISO 8650-1(ACSE). Should not be used in standard cases.
Use Authentication	Activate this checkbox if you want to use authentication according to ISO 8650-1. Active: If active, the driver sends the <code>Authentication String</code> at establishing the connection.
Authentication String	If Use Authentication is active, enter your <code>Authentication String</code> . Take care about the length restriction of 55 characters according to ISO 8327-1 OSI Session Protocol.
GetNameList on DO	Active: The driver reads the <code>Objekt Model</code> by requesting data objects (DO) for each available <code>Logical Node</code> pro in every server in the IEC61850 standard defined <code>Functional Constraint</code> (FC). Therefore <code>Objekt Models</code> can then be read when some <code>Logical Nodes</code> are larger than the PDU size supported by the server. Attention: If this option is active, it can take much longer to establish the connection. Only activate this option if reading in cannot be carried out in full due to server limitations. Default: <code>inactive</code>

REPORTING

Parameters	Description
Reporting	Settings for the reports (on page 76).
Max. auto used URCBs	Maximum number of Unbuffered Reports which the driver activates at automatic allocation per <code>Logical Device</code> , i.e. in addition to RCBs that were configured in the <code>RCB assignment/dynamic datasets</code> dialog. Default: 10

	<p>Entry is only valid for unbuffered reports.</p> <p>you can find more information on automatic allocation in the Unbuffered Report (on page 78) chapter.</p>
RCBs enable retries	<p>Configuration of the time in which the driver reattempts to register the RCBs that were not registered successfully.</p> <p>Entry of the time in seconds.</p> <p>Entry range: 0 – 999999</p> <p>Default: 7</p> <p>Note: If 0 is entered, there is no attempt to register an RCB again. These settings only affect RCBs that are listed in the RCB assignment section.</p>
Use preconfigured (SCL) options	<p>Active:</p> <p>The driver activates a report without overwriting the data attributes of the RCB. The content of the SCL file of the server is defined as a result of this. The following data attributes are affected by this:</p> <ul style="list-style-type: none"> ▶ IntgPd ▶ BufTime ▶ TrgOps ▶ OptFields <p>Inactive:</p> <p>The driver writes the data attributes of the RCB during activation.</p> <p>You can activate/deactivate the following trigger options regardless of one another.</p> <ul style="list-style-type: none"> ▶ TrgOp: data-change ▶ TrgOp: quality-change ▶ TrgOp: data-update ▶ TrgOp: integrity ▶ TrgOp: general-interrogation <p>Default: all active</p> <p>Note: Not all servers support TrgOps data-change and data-update together. TrgOp integrity can also lead to an unnecessary overload of communication if a an IntgPd (Integrity Period) that is too short was defined in the server for RCB. In case of doubt, set TrgOps: data-change + quality-change + general-interrogation.</p>
Integrity Period	<p>Time interval (IntgPd) in milliseconds in which the server sends an Integrity Report.</p> <p>Default: 7000 ms</p>

	<p>Note: not active if <code>TrgOp integrity</code> is deactivated or Use preconfigured (SCL) options is activated.</p> <p>Because an <code>Integrity Report</code> does not normally contain value changes, it is expressly recommended that only one report on the server is activated with <code>TrgOp: integrity</code>. With an activated integrity report, the server can detect a connection failure more quickly. However zenon does not need this report.</p>
Buffer Time	<p>Time interval (<code>BufTime</code>) in milliseconds in which the server collects the data for a report.</p> <p>Default: 500 ms</p> <p>Note: not active if Use preconfigured (SCL) options is activated.</p>
OptFlds	<p>Opens dialog to configure <code>Optional Fields</code> (on page 27). of the report.</p> <p>Note: not active if Use preconfigured (SCL) options is activated.</p>
Use report ID for RCB assignment	<p>Active: The report ID is used instead of the report name in the <code>client configuration</code> dialog for static RCB assignment.</p> <p>Inactive: The RCB instances of the server are identified by name.</p> <p>Default: <code>inactive</code></p> <p>Attention: If this option is subsequently changed, all RCB assignments in the <code>client configuration</code> must be deleted and created again.</p>
RCB assignment/dynamic datasets	<p>Static <code>Report Control Block</code> allocations and settings for <code>Dynamic Data Sets</code>.</p> <p>Note: If you have activated 'Use redundancy' in your zenon project, (Network active property activated and Server 1 and Server 2 configured), you must create two entries here: one for the server and one for the standby.</p>
New	Creates a new entry in the list. Opens the <code>client configuraton</code> dialog.
Del	Removes the selected entry from the list.
Edit	Opens the <code>Client configuration</code> dialog to edit the selected entry.



Information

OSI value (ACSE)

The driver uses the following OSI selector values:

- ▶ Presentation Selector (OSI-PSEL) = "00000001"
- ▶ Session Selector (OSI-SSEL) = "0001"
- ▶ Transport Selector (OSI-TSEL) = "0001"

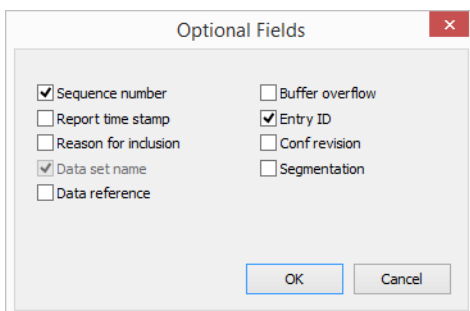
In a server SCL file, it corresponds to the following entries:

- ▶ <P type="OSI-AP-Title">41,999,1,1</P>
- coded in accordance with [Basic Encoding Rules](#) (IEC 8825) for [Object Identifier](#) - corresponds to standard values of the [Universal Identifiers](#) in the **Called AP title**.
- ▶ <P type="OSI-AE-Qualifier">12</P>
- corresponds to standard values in the **Called AE qualifier**.
- ▶ <P type="OSI-PSEL">00000001</P>
set in the driver; cannot be changed
- ▶ <P type="OSI-SSEL">0001</P>
set in the driver; cannot be changed
- ▶ <P type="OSI-TSEL">0001</P>
set in the driver; cannot be changed

Optional Fields

The [Optional Fields](#) are written on the server when a report is activated. These correspond to the bits in the [OptFlds](#) data attribute of the [RCB](#).

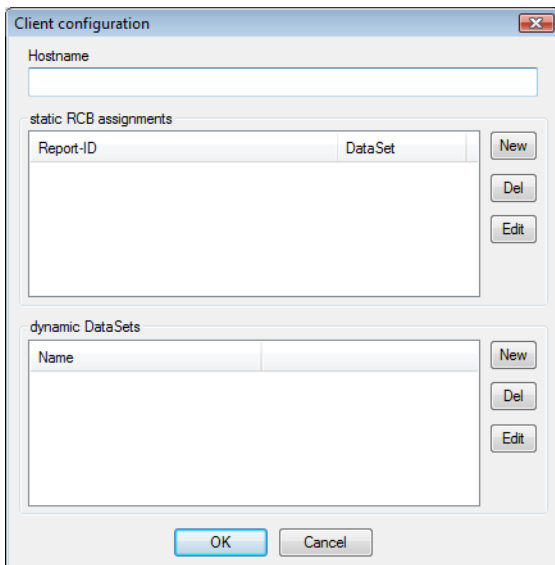
"Sequence number", "Data set name" and "Entry ID" are activated by default. "Data set name" cannot be changed, because without this option the driver cannot evaluate the reports received.



Optional Fields	
<input checked="" type="checkbox"/> Sequence number	<input type="checkbox"/> Buffer overflow
<input type="checkbox"/> Report time stamp	<input checked="" type="checkbox"/> Entry ID
<input type="checkbox"/> Reason for inclusion	<input type="checkbox"/> Conf revision
<input checked="" type="checkbox"/> Data set name	<input type="checkbox"/> Segmentation
<input type="checkbox"/> Data reference	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Client configuration

Clicking on **New** or **Edit** in the **RCB assignments/dynamic datasets** list in the server dialogs (on page 22) opens the dialog for the client configuration settings:



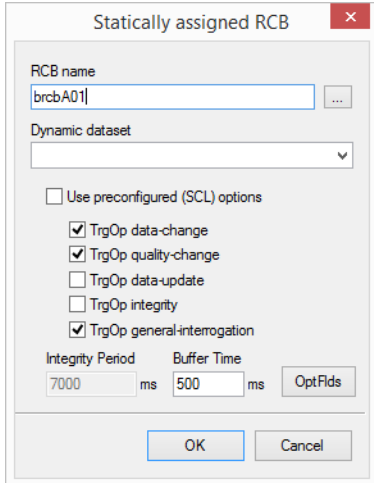
The 'Client configuration' dialog box contains the following elements:

- Hostname:** A text input field at the top.
- static RCB assignments:** A section containing a table with two columns: 'Report-ID' and 'DataSet'. To the right of the table are three buttons: 'New', 'Del', and 'Edit'.
- dynamic DataSets:** A section containing a table with one column: 'Name'. To the right of the table are three buttons: 'New', 'Del', and 'Edit'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom center.

Parameters	Description
Hostname	<p>Name of the computer on which the driver is running that receives the reports:</p> <ul style="list-style-type: none"> ▶ For standalone projects: The computer name of the computer on which zenon Runtime runs; ▶ in the zenon network: The computer name of the project server or project standby. <p>Note: Do not use "localhost"; use the computer names from the operating system instead.</p> <p>Attention: If the current computer name and the configured host name do not correspond, the reports defined under static RCB assignment are ignored.</p>
<code>static RCB assignment</code>	<p>Settings for the static assignment of RCBs (the same for unbuffered and buffered RCBs).</p> <p>Note: Buffered reports must be configured in the driver configuration. Otherwise they are not used.</p> <p>For more information, see driver-specific functions (on page 63).</p>
Report-ID	Displays the report IDs already entered.
DataSet	Displays the dynamic data set (if the RCB was allocated one).
New	Creates a new entry in the list. Opens the 'statically assigned RCB' dialog to create a new entry with a report ID (or a name) - you can either enter it manually or browse the server and if necessary with dynamic data set (is taken from the lower list).
Del	Removes the selected entry from the list.
Edit	Opens the 'statically assigned RCB' dialog to edit the selected entry.
dynamic DataSets	<p>Settings for dynamic data sets.</p> <p>Note: Dynamic data sets are not a requirement for buffered or unbuffered reports. They are optional settings if the IEC61850 server that is used supports this.</p> <p>For more information, see dynamic data sets (on page 82).</p>
Name	Displays the data sets that have already been entered - that are already present in the upper list.
New	Opens the dialog for creating a data set.
Del	Removes the selected entry from the list.
Edit	Opens the 'dynamic DataSet configuration' dialog to edit the selected entry.

statically assigned RCB

Clicking on **New** or **Edit** in the **static RCB assignment** list of the **client configuration** (on page 28) dialog opens the dialog to create an RCB and/or to assign an RCB a dynamic data set. RCBs are either defined by means of the RCB name:



The dialog box is titled "Statically assigned RCB" and contains the following fields and options:

- RCB name:** A text input field containing "brcbA01" and a browse button "...".
- Dynamic dataset:** A dropdown menu.
- Use preconfigured (SCL) options:** A checkbox that is currently unchecked.
- TrgOp options:**
 - ☒ TrgOp data-change
 - ☒ TrgOp quality-change
 - ☐ TrgOp data-update
 - ☐ TrgOp integrity
 - ☒ TrgOp general-interrogation
- Integrity Period:** A text input field containing "7000" followed by "ms".
- Buffer Time:** A text input field containing "500" followed by "ms".
- OptFids:** A button.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

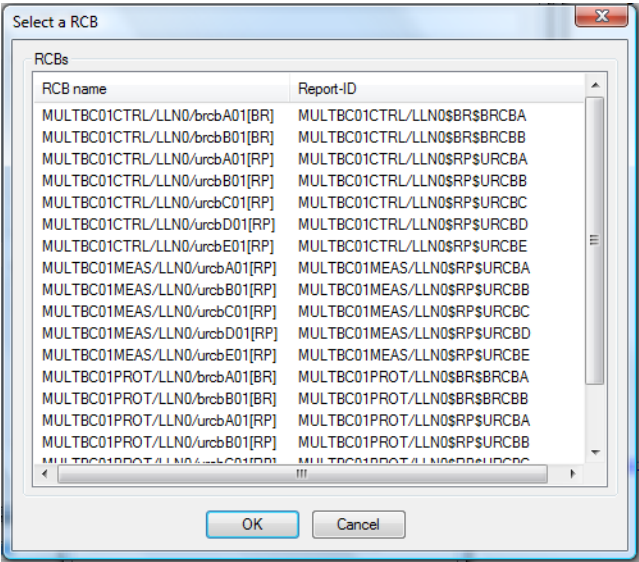
Parameters	Description
RCB name/report ID	<p>The name or the ID of the report (buffered or unbuffered) that is to register the driver (use). Must correspond exactly to the name/report ID in the PLC (IEC61850 server).</p> <p>A click on . . . starts online browsing of the configured IEC 61850 server and shows if the server is running and all reporting control blocks that are currently on the server.</p> <p>Note: RCB name is expected here by default. The RCB name is saved with the full <code>ObjectReference</code> and is thus unique. The use of the RCB name can be changed using the Use report ID for RCB assignment property in the server (on page 22) properties on report ID.</p> <p>Report-ID: the report ID is saved without <code>ObjectReference</code>, so several RCBs can thus be applicable. In Runtime, the driver will search for all RCBs with the respective Report ID and register them.</p> <p>Note: If Runtime is also running at the same time on the same computer, the RCBs cannot be browsed online.</p>
dynamic Dataset	<p>You can also optionally select a data set from the drop-down list of the data sets that have already been entered into the 'dynamic DataSet configuration (on page 34)' dialog.</p> <p>Important: If you do not define a data set here, the data set that is predefined for this RCB in the PLC (IEC61850 server) is used.</p>
Use preconfigured (SCL) options	<p>Active:</p> <p>The driver activates a report without overwriting the data attributes of the RCB. The content of the SCL file of the server is defined as a result of this. The following data attributes are affected by this:</p> <ul style="list-style-type: none"> ▶ <code>IntgPd</code> ▶ <code>BufTime</code> ▶ <code>TrgOps</code> ▶ <code>OptFields</code> <p>Inactive:</p> <p>The driver writes the data attributes of the RCB during</p>

	<p>activation.</p> <p>You can activate/deactivate the following trigger options regardless of one another.</p> <ul style="list-style-type: none"> ‣ TrgOp: data-change ‣ TrgOp: quality-change ‣ TrgOp: data-update ‣ TrgOp: integrity ‣ TrgOp: general-interrogation <p>Default: all active</p> <p>Note: Not all servers support TrgOps data-change and data-update together. TrgOp integrity can also lead to an unnecessary overload of communication if a an IntgPd (Integrity Period) that is too short was defined in the server for RCB. In case of doubt, set TrgOps: data-change + quality-change + general-interrogation.</p>
Integrity Period	<p>Time interval (IntgPd) in milliseconds in which the server sends an Integrity Report.</p> <p>Default: 7000 ms</p> <p>Note: not active if TrgOp integrity is deactivated or Use preconfigured (SCL) options is activated.</p> <p>Because an Integrity Report does not normally contain value changes, it is expressly recommended that only one report on the server is activated with TrgOp: integrity. With an activated integrity report, the server can detect a connection failure more quickly. However zenon does not need this report.</p>
Buffer Time	<p>Time interval (BufTime) in milliseconds in which the server collects the data for a report.</p> <p>Default: 500 ms</p> <p>Note: not active if Use preconfigured (SCL) options is activated.</p>
OptFlds	<p>Opens dialog to configure Optional Fields (on page 27). of the report.</p> <p>Note: not active if Use preconfigured (SCL) options is activated.</p>

CLOSE DIALOG

Parameters	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

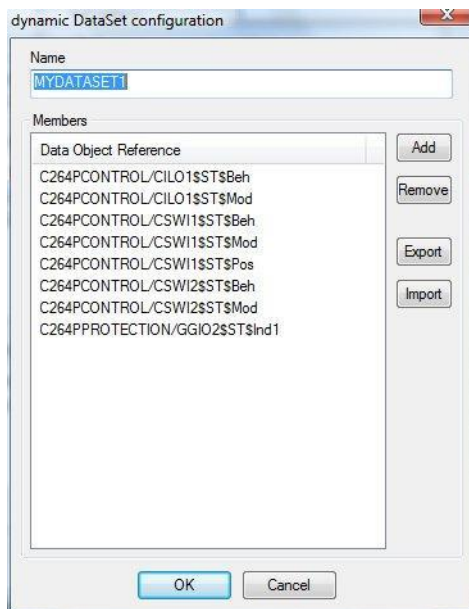
This list is displayed after you have browsed RCBs online on the IEC 61850 server:



Select the desired Reporting Control Block from the list.

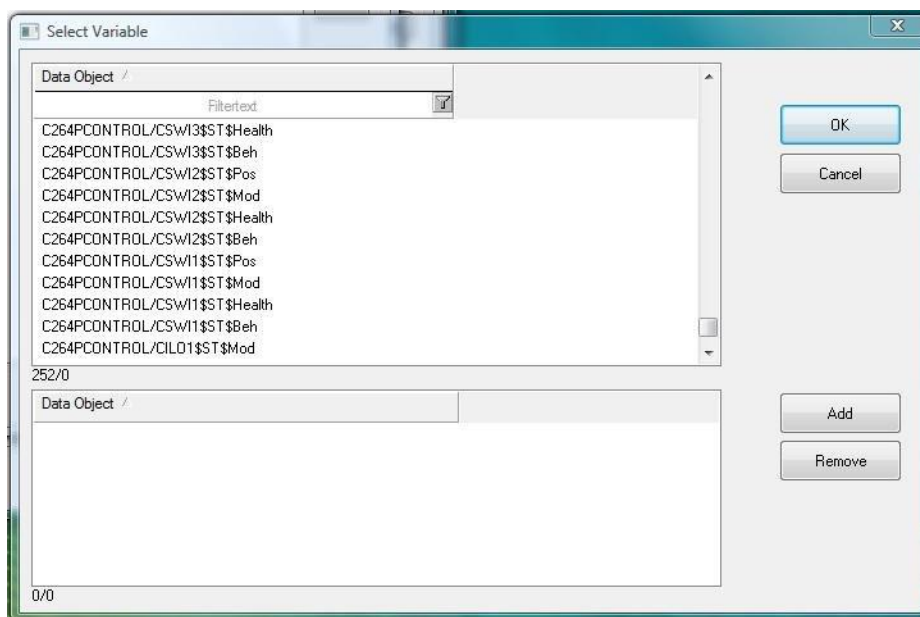
dynamic DataSet configuration

Clicking on **New** or **Edit** in the **dynamic data sets** list of the **client configuration** (on page 28) dialog opens the dialog to define a dynamic data set.



Parameters	Description
Name	Name of the data set.
Members	List of pre-configured data objects.
Add	<p>Starts the browsing of data objects from the variable list of the zenon project and displays all data objects for which a variable has already been created in the project.</p> <p>Note: The driver does not browse the data objects from the IEC61850 server, but from the variable list in the Editor. However, if Runtime is running at the same time on the same computer, the variables cannot be browsed.</p>
Remove	Removes the selected entry from the list.
Export	Exports members to a TXT file.
Import	<p>Imports members to a TXT file.</p> <p>In the open dialog:</p> <ul style="list-style-type: none"> ‣ select the corresponding file ‣ select the desired variables ‣ Confirm the settings by clicking on OK

The list is displayed to you after you have prepared suitable project variables:



The driver displays all data objects - that are appropriate for data reporting - from the variable list of the zenon project, except those that have already been selected for the data set. In the upper list, you can filter and select the variable filters (multiple selection is also possible) and add these to the lower list. Objects that have already been selected are kept. Use the **OK** button to add the selected variables to the **members** of the dynamic data set.

7. Creating variables

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

7.1 Creating variables in the Editor

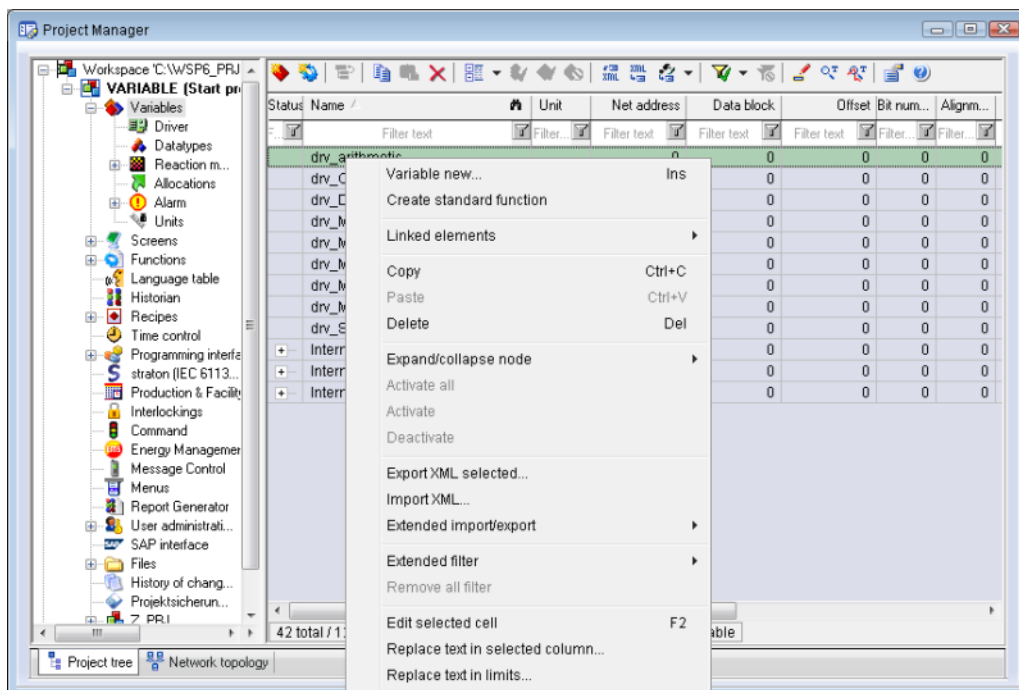
Variables can be created:

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

VARIABLE DIALOG

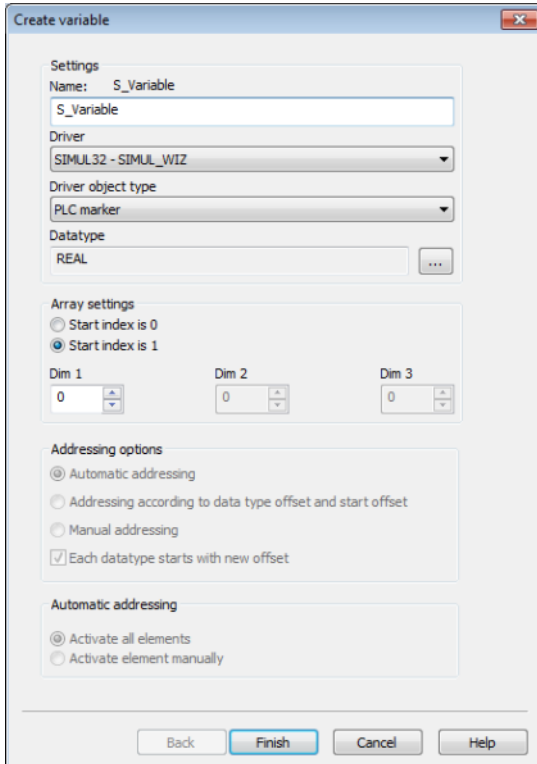
To create a new variable, regardless of which type:

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



2. The dialog for configuring variables is opened
3. configure the variable

4. The settings that are possible depends on the type of variables



The screenshot shows the 'Create variable' dialog box with the following settings:

- Settings**
 - Name: S_Variable
 - Driver: SIMUL32 - SIMUL_WIZ
 - Driver object type: PLC marker
 - Datatype: REAL
- Array settings**
 - ☐ Start index is 0
 - ☒ Start index is 1
 - Dim 1: 0
 - Dim 2: 0
 - Dim 3: 0
- Addressing options**
 - ☒ Automatic addressing
 - ☐ Addressing according to data type offset and start offset
 - ☐ Manual addressing
 - ☒ Each datatype starts with new offset
- Automatic addressing**
 - ☒ Activate all elements
 - ☐ Activate element manually

Buttons at the bottom: Back, Finish, Cancel, Help.

Property	Description
Name	<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 Zeichen</p> <p>Attention: The characters # and @ are not permitted in variable names. If non-permitted characters are used, creation of variables cannot be completed and the Finish button remains inactive.</p> <p>Note: For some drivers, the addressing is possible over the property Symbolic address, as well.</p>
Driver	<p>Select the desired driver from the drop-down list.</p> <p>Note: If no driver has been opened in the project, the driver for internal variables (Intern.exe (Main.chm::/Intern.chm::/Intern.htm)) is automatically loaded.</p>
Driver object type (cti.chm::/28685.htm)	Select the appropriate driver object type from the drop-down list.
Data type	Select the desired data type. Click on the ... button to open the selection dialog.
Array settings	Expanded settings for array variables. You can find details in the Arrays chapter.
Addressing options	Expanded settings for arrays and structure variables. You can find details in the respective section.
Automatic element activation	Expanded settings for arrays and structure variables. You can find details in the respective section.

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.

7.2 Addressing

The addressing is based on the variable names.

Name	Description
Name	Freely definable name Attention: The name is used for addressing the variables and must be clear in each control system project. As a result, the server name is also taken into account if the driver identifies the connection with the help of the Net Address property.
Identification	Enter any desired text; for example resource name, comments, ... Can be used for addressing (optional).
Net address	Network address of variable. This address refers to the network address in the connection list of the driver configuration. It defines the IEC 61850 server on which variables are saved.
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	Only available for String variables: Maximum number of characters from which a variable can exist.
Driver object type	Dependent on the type of variable, the object type is selected when the variable is created; the type can be changed here later. For more information, see also the driver objects (on page 41).
Data type	Data type of variables that were selected when the variable was created; the type can be changed here later. Attention: If you subsequently change the data type here, you must check all variable properties, such as value range for example, and change them if necessary.

SYNTAX NAMING CONVENTION

The naming convention is based on the following syntax: **SERVER!LD/LN/DATA/DataAttr[FC]**

For this, the following applies:

- ▶ The server name (on page 22) is separated by the exclamation mark (!) from the following identifiers.
- ▶ All additional identifiers to **DataAttr** are separated by a slash (/).
- ▶ Only if an identifier is a structure, their elements are separated by a dot (.); for example:
SERVER!LD/LN/DATA/DataAttr.item[FC].

The designations are:

Parameters	Description
SERVER	<p>Freely definable name of the server.</p> <p>Name should make sure that all variable names are identical.</p> <p>Recommendation: Use the same names as defined in driver dialog Dialog Server (on page 22). The driver however does not recognize the used connection via the Server name but via settings the Net address.</p>
LD	Name of the logical device.
LN	Name of the logical node.
DATA	Name of the 'data object', see Appendix B - data objects / data attributes
DataAttr	Name of the 'data attribute', see Appendix B - data objects / data attributes
FC	"functional constraint" of the "data attribute", see Appendix A - Description of the Functional Constraints (FCs)



Attention

The naming convention for variables complies with the **ObjectReference** defined in standard IEC61850-7-1.

In the standard a slash (/) is only used as separator between **LD** and **LD** and otherwise separated by dots (.). If **DATA** or **DataAttr** is a structure, dots make it impossible to differentiate between **DATA** and structure. Therefore an own naming conversion is used for the variable addressing.

EXAMPLE:

The logical node "XCBR1" (circuit breaker 1) has, among other things, a data object "Pos", which is a CDC (Common Data Class) "Controllable Double Point" (DPC). This data point 'Pos' has, in turn, attributes 'stVal', 'q' and 't', the name and semantics of which is specific due to its CDC, which is defined in IEC61850-7-3.

Data attribute	Name	IEC 61850 Standard
XCBR1/Pos	Switching position	In accordance with LN definition XCBR in 61850-7-4
XCBR1/Pos/stVal[ST]	position value	In accordance with CDC definition DPC in 61850-7-3
XCBR1/Pos/q[ST]	Quality	In accordance with CDC definition DPC in 61850-7-3
XCBR1/Pos/t[ST]	time stamp	In accordance with CDC definition DPC in 61850-7-3
XCBR1/Pos/Oper.ctlVal[CO]	Value of the command	In accordance with 'Oper' type definition in 61850-8-1, Table E.9

All objects in the variable model are constructed in this way. For this reason, the driver knows, for example, that the attributes 'q' and 't' contain quality and time stamps for the variables in 'stVal'. This

allows access to all attributes of the each object in the variable model. Existing values for quality and time stamp are automatically assigned to the actual values of the driver.

Operate command: Thanks to the specific variable nomenclature, the driver automatically notices that an operate command is to be created if a variable is written to, for example with the name `*/Oper.ctlVal[CO]` (Operation.ControlValue). It the automatically sends the required sequence of telegrams to the control object.

TIME STAMP AND QUALITY

Time stamp and quality attribute are always parts of variables in zenon. Variables with attributes 'q' and 't' do not need to be created separately. The driver automatically uses the time stamp and the quality of the data object for its variables with `FC = ST and MX` (FC stands for Functional Constraint (on page 88)). Without taking into account whether a variable for attribute 't' was created, the time stamp received is used for all `ST` and `MX` variables, for example the `stval` variable for CEL, AML or Historian. Selected quality bits that are received in attribute 'q' are also assigned the status bits of the `stval` variable (see also Driver-specific functions (on page 63)).

7.3 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.

7.3.1 Driver objects

The following object types are available in this driver:

Driver object type	Channel type	Read / Write	Supported data types	Comment
PLC marker	8	R / W	REAL, LREAL, WSTRING, DINT, UDINT, BOOL, INT, UINT, STRING, USINT, SINT	Data attribute variables
File Transfer	9	R / W	STRING	<p>Command and return variables for the file transfer.</p> <p>Note: The string length should correspond to the maximum length of the file names (including the path) in the PLC; a maximum of 260 characters for *!Command variable for example, (but more for *!Directory).</p> <p>You can find more information in the chapter on driver-specific functions (on page 63)</p>
Command Info	11	R	UINT	<p>Internal variables show the current status of the command; transferred from the driver to Runtime (not by the PLC).</p> <p>You can find more information in the chapter on driver-specific functions (on page 63)</p>
Service tracking		R	REAL, LREAL, DINT, UDINT, BOOL, INT, UINT, STRING, USINT, SINT	<p>Additional Service tracking (*[SR]) variables.</p> <p>You can find more information in the chapter Service tracking (on page 74)</p>
Driver variable	35	R / W	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	<p>Variables for the static analysis of the communication; is transferred between driver and Runtime (not to the PLC).</p> <p>Note: The addressing and the behavior is the same for most zenon drivers.</p> <p>You can find more information in the chapter on driver variables (on page 56)</p>

7.3.2 Assignment of 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.

ASSIGNMENT OF DATA TYPES FROM THE PLC TO ZENON DATA TYPES

Control	zenon	Data type	Comment
BOOLEAN	BOOL	8	
INT8	SINT	10	
INT8U	USINT	9	
INT16	INT	1	
INT16U	UINT	2	
INT24	DINT	3	
INT24U	UDINT	4	
INT32	DINT	3	
INT32U	UDINT	4	
INT128	DINT	3	
FLOAT32	REAL	5	
FLOAT64	LREAL	6	
ENUMERATED	INT	1	
CODED ENUM	UDINT, STRING	4, 12	MMS Bitstring
OCTET STRING	STRING	12	
VISIBLE STRING	STRING	12	
UNICODE STRING	WSTRING	21	Coding is carried out in UTF-8
PACKED LIST	UDINT, STRING	4, 12	MMS Bitstring, inverted bit order
TIMESTAMP	LREAL	6	Only octets 0..7 (see also TimeQuality Bit mapping (on page 63))
EntryTime	LREAL	6	
TriggerConditions	UDINT, STRING	4, 12	MMS Bitstring, inverted bit order

Data type: The property **Data type** is the internal numerical name of the data type. It is also used for the extended DBF import/export of the variables.

MMS Bitstring: The data attribute types that were assigned to the `MMS Bitstring` in accordance with the standard (see IEC61850-8-1), `CODED ENUM`, `PACKED LIST` und `TrigerConditions` for example, correspond to the `UDINT` data type in zenon. Whereby in the case of `PACKED LIST` and `TrigerConditions`, the bit 0 has the highest value in its octet. The driver also allows bit string variables with the data type `STRING` instead of `UDINT` to be inserted manually. The value of the string only contains the characters "0" and "1" in a sequence that is the same as that of the bits defined in the standard and only in the length of the bit string, for example the value of the `TrgOps` Attribute - `Bitstring(6)` - as String "011111" (in `UDINT=124`) means that all trigger condition bits except bit 0 - 'Reserved' are set and the value "000010" (in `UDINT=8`) - only bit 4 - 'integrity'.

"ARRAY OF" TYPE DATA ATTRIBUTES

If there is an array level in the IEC61850 `Data Classs`, this level is placed at the end of the variable structure for display in zenon. This means that, from an `ARRAY OF` that may have nested structures, there are one or several arrays of scaled values in the zenon variable model. This remapping is carried out at the start of the configuration time during online import and in the runtime environment during the allocation between the zenon variable list and the IEC61850 object model.

As a result of this, there is the restriction that only one array level is supported. The support for arrays is also limited to the data attribute.

Example

The data attribute `HaAmp.har` of the `Common Data Class` (`CDC, -7-3`) `MHAN` - as defined in the IEC61850 standard: `ARRAY[0..numHar]` of `Vector` - becomes the following array in zenon:

[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][0]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][1]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][2]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][3]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][4]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][5]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][6]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][7]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][8]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][9]
[-]	ARI!ArrLD1/MHAN1/HaAmp/har.ang.f[MX][10]

In addition, there are of course also the arrays for `ang.i`, `mag.f` and `mag.i`.

7.4 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.

7.4.1 Online import

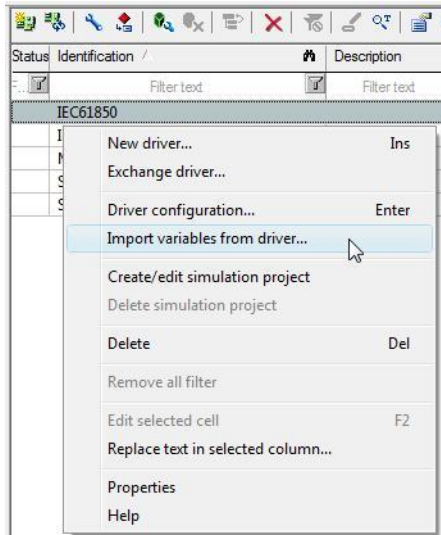
Variables in a zenon project can be created with the import of the driver. The driver can browse the variable list of the IEC 61850 server.



Attention

If Runtime is running on the same computer at the same time, the variables in the editor cannot be imported by the PLC, because the driver in Runtime uses the given settings for communication.

You call up the import using the context menu of the driver in the driver list.



A dialog is opened.

SELECT A SERVER

Parameters	Description
Server	List of the connections to IEC61850 servers created in the driver. You can read further information about the configuration of server connections in the Connections (on page 21) chapter.
Source	Source of the data model: <ul style="list-style-type: none"> ▶ PLC Browsing Variables are imported from an IEC61850 server that is available online. ▶ File Browsing Variables are imported from an SCL file. You can read further information about the SCL file in the IEC61850-6 standard.
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

PROCEDURE FOR PLC BROWSING

- ▶ In the dialog that opens, select the **PLC Browsing** option and select a server from which the variables can be imported.
- ▶ The connection to the server is established and the list of existing data points (LDs, LNs, data objects, data attributes) is read.
- ▶ If the reading in has ended, an important dialog opens in which you can filter the browsed variables (capitalization, for example *XCBR1*stVal*), select them (including multiple selection) and add them from the upper to lower list.
- ▶ You create the selected variables in the project using the OK button.



Attention

The driver creates variables with default value ranges, for example a UDINT with <0...4294967295> and a LREAL with <-1000..1000>, which may not correspond to the value ranges of the attributes, for example stVal-attribute UDINT <0..3> or t-attribute LREAL<0.. 2147483647> with three decimal places for milliseconds. Therefore ensure after import after that all variables have the correct value range properties and the string is the right length.

"ARRAY OF" IMPORT

As a result of the limitation of only one array level being supported (on page 43), the support of arrays is also limited to the data attributes.

These **ARRAY OF** variables are created as zenon arrays when imported.

Example

The data attribute **HaAmp.har** of the **Common Data Class MHAN** - as defined in the IEC61850 standard: **ARRAY[0..numHar] of Vector** - becomes up to four arrays in zenon, with the same respective number of elements:

- ▶ ***/HaAmp.har.ang.f[MX][0..count-1]**
- ▶ ***/HaAmp.har.ang.i[MX][0..count-1]**
- ▶ ***/HaAmp.har.mag.f[MX][0..count-1]**
- ▶ ***/HaAmp.har.mag.i[MX][0..count-1]**

SERVICE TRACKING IMPORT

Service tracking is only offered if one or more elements with **FC = SR** are present on the **Logical Device**. For example, one or more **Logical Nodes LTRK**. For import, only **Service tracking** is automatically offered for **Control Services**. **Control Services** means: only **Service tracking** command execution information.

If there is **Service tracking** on the server, a proposal for the creation of additional service tracking variables is created for each data object that contains the data attributes with **FC = CO**.

Example of a data object with command execution: **S1!Device/Node/Pos/Oper.ctlVal[CO]**

These additional variables no longer exist on the server but are additionally created in the respective data object with a possible command execution.

The variables that are actually created as additional service tracking variables are selected from this list of proposals.

The name of this proposal comprises the following:

Datenobjektreferenz_Servicetracking-Datenattributname[SR]. As a service tracking data attribute name, the data attributes that are present in the [Logical Node LRTK](#) for [Control Services](#) are proposed.

Example of naming: [S1!Device/Node/Pos_errorCode\[SR\]](#).

The data type of the additional service tracking variable corresponds to the data type of the SR data attribute that is present in the [Logical Node LRTK](#).

The driver object type is [Service tracking](#).

In addition, a string variable *_allCTS[SR] is also offered. This string variable contains the complete service tracking Information and can be evaluated with the help of zenon Logic or VBA/VSTA.

7.4.2 Offline import

Variables can also be imported offline. The following file endings are accepted:

- ▶ *.xml
- ▶ *.scl
- ▶ *.icd
- ▶ *.cid
- ▶ *.scd

The driver only imports DOI elements online. The **ReportControl** properties (report control block) and **SettingControl** (Setting Control Block) are not supported during offline import. You can create the variables for these attributes by means of online import or manually.

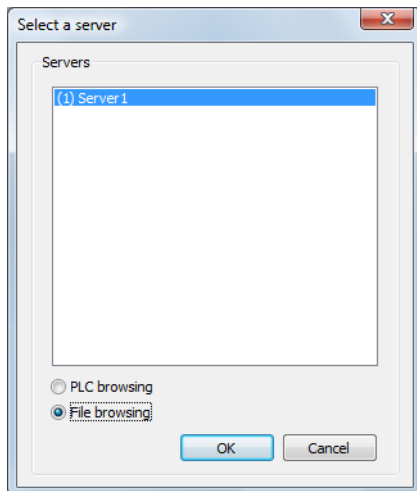


Attention

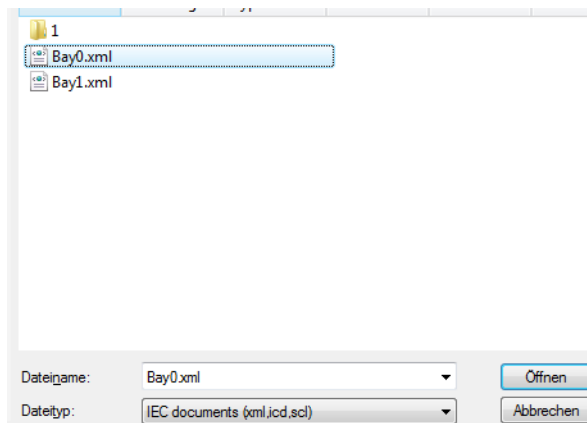
With offline import, logical nodes that are linked directly to the IED - instead of to the logical device - are ignored. The driver can only import attributes from the object model that are fully compliant with the IEC61850 standard.

To import variables from a file:

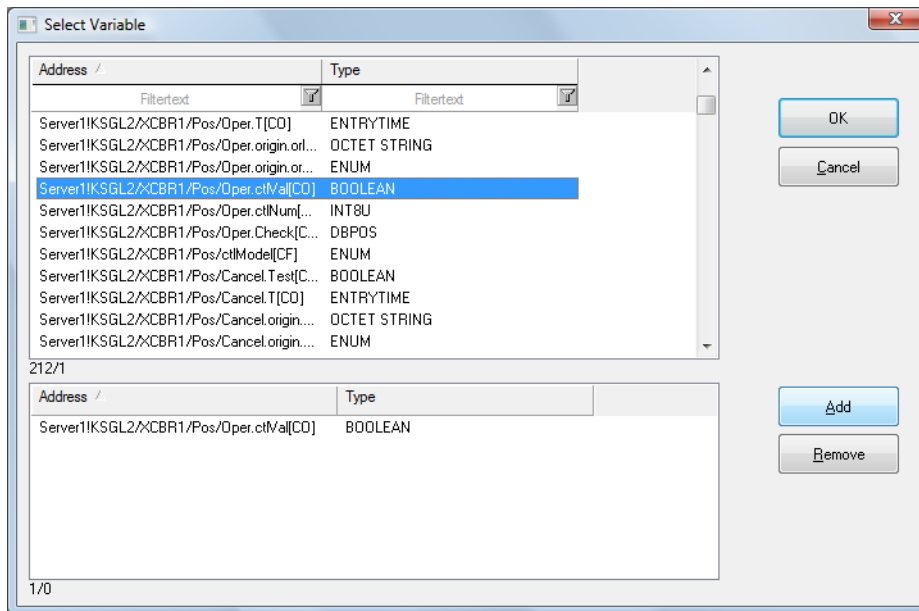
1. Select the **File browsing** option in server selection



2. Select the import file.



3. The following steps for the selection of variables are similar to online import.



SERVICE TRACKING IMPORT

Service tracking is only offered if one or more elements with **FC = SR** are present on the **Logical Device**. For example, one or more **Logical Nodes LTRK**. For import, only **Service tracking** is automatically offered for **Control Services**. **Control Services** means: only **Service tracking** command execution information.

If there is **Service tracking** in the SCL file, a proposal for the creation of additional service tracking variables is created for each data object that contains the data attributes with **FC = CO**.

Example of a data object with command execution: `S1!Device/Node/Pos/Oper.ctiVal[CO]`

These additional variables no longer exist on the server but are additionally created in the respective data object with a possible command execution.

The variables that are actually created as additional service tracking variables are selected from this list of proposals.

The mane of this proposal comprises the following:

Datenobjektreferenz_Servicetracking-Datenattributname[SR]. As a service tracking data attribute name, the data attributes that are present in the **Logical Node LTRK** for **Control Services** are proposed.

Example of naming: `S1!Device/Node/Pos_errorCode[SR]`.

The data type of an additional service tracking variable corresponds to the data type of the SR data attribute, that is in the **Logical Node LTRK**.

The driver object type is **Service tracking**.

In addition, a string variable *_allCTS[SR] is also offered. This string variable contains the complete service tracking Information and can be evaluated with the help of zenon Logic or VBA/VSTA.

7.4.3 XML import

For the import/export of variables the following is true:

- ▶ The import/export must not be started from the global project.
- ▶ The start takes place via:
 - Context menu of variables or data typ in the project tree
 - or context menu of a variable or a data type
 - or symbol in the symbol bar variables



Attention

When importing/overwriting an existing data type, all variables based on the existing data type are changed.

Example:

There is a data type XYZ derived from the type `INT` with variables based on this data type. The XML file to be imported also contains a data type with the name XYZ but derived from type `STRING`. If this data type is imported, the existing data type is overwritten and the type of all variables based on it is adjusted. I.e. the variables are now no longer `INT` variables, but `STRING` variables.

7.4.4 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 there 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

Description	Type	Field size	Comment
KANALNAME	Char	128	Variable name. The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_R	C	128	The original name of a variable that is to be replaced by the new name entered under "VARIABLENNAME" (field/column must be entered manually). The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_D	Log	1	The variable is deleted with the 1 entry (field/column has to be created by hand).
TAGNR	C	128	Identification. The length can be limited using the MAX_LAENGE entry in project.ini .
EINHEIT	C	11	Technical unit
DATENART	C	3	Data type (e.g. bit, byte, word, ...) corresponds to the data type.
KANALTYP	C	3	Memory area in the PLC (e.g. marker area, data area, ...) corresponds to the driver object type.
HWKANAL	Num	3	Bus address
BAUSTEIN	N	3	Datablock address (only for variables from the data area of the PLC)
ADDRESS	N	5	Offset
BITADR	N	2	For bit variables: bit address For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters)
ARRAYSIZE	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 Recipe Group Manager
LES_SCHR	R	1	Write-Read-Authorization

			0: Not allowed to set value. 1: Allowed to set value.
MIT_ZEIT	R	1	time stamp in zenon zenon (only if supported by the driver)
OBJEKT	N	2	Driver-specific ID number of the primitive object comprises TREIBER-OBJEKTYP and DATENTYP
SIGMIN	Float	16	Non-linearized signal - minimum (signal resolution)
SIGMAX	F	16	Non-linearized signal - maximum (signal resolution)
ANZMIN	F	16	Technical value - minimum (measuring range)
ANZMAX	F	16	Technical value - maximum (measuring range)
ANZKOMMA	N	1	Number of decimal places for the display of the values (measuring range)
UPDATERATE	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables
MEMTIEFE	N	7	Only for compatibility reasons
HDRATE	F	19	HD update rate for historical values (in sec, one decimal possible)
HDTIEFE	N	7	HD entry depth for historical values (number)
NACHSORT	R	1	HD data as postsorted values
DRRATE	F	19	Updating to the output (for zenon DDE server, in [s], one decimal possible)
HYST_PLUS	F	16	Positive hysteresis, from measuring range
HYST_MINUS	F	16	Negative hysteresis, from measuring range
PRIOR	N	16	Priority of the variable
REAMATRIZE	C	32	Allocated reaction matrix
ERSATZWERT	F	16	Substitute value, from measuring range
SOLLMIN	F	16	Minimum for set value actions, from measuring range
SOLLMAX	F	16	Maximum for set value actions, from measuring range
VOMSTANDBY	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
RESOURCE	C	128	Resources label. Free string for export and display in lists. The length can be limited using the MAX_LAENGE entry in project.ini .
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 DEFINITION

Limit definition for limit values 1 to 4, and status 1 bis 4:

Description	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 (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
HYSTERESE1	F	14	Is not used
BLINKEN1	R	1	Set blink attribute
BTB1	R	1	Logging in CEL
ALARM1	R	1	Alarm
DRUCKEN1	R	1	Printer output (for CEL or Alarm)
QUITTTIER1	R	1	Must be acknowledged
LOESCHE1	R	1	Must be deleted
VARIABLE1	R	1	Dyn. limit value linking the limit is defined by an absolute value (see field GRENZWERTx).
FUNC1	R	1	Functions linking
ASK_FUNC1	R	1	Execution via Alarm Message List
FUNC_NR1	N	10	ID number of the linked function (if "-1" is entered here, the existing function is not overwritten during import)
A_GRUPPE1	N	10	Alarm/event group
A_KLASSE1	N	10	Alarm/event class
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Color as Windows coding
GRENZTXT1	C	66	Limit text
A_DELAY1	N	10	Time delay
INVISIBLE1	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.

7.5 Driver variables

The driver kit implements a number of driver variables. These are divided into:

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

The definitions of the variables defined in the driver kit are available in the import file `drvvar.dbf` (on the CD in the directory: `CD_Drive:/Predefined/Variables`) and can be imported from there.

Note: Variable names must be unique in zenon. If driver variables are to be imported from `drvvar.dbf` again, the variables that were imported beforehand must be renamed.



Information

Not every driver supports all driver variants.

For example:

- ▶ Variables for modem information are only supported by modem-compatible drivers
- ▶ Driver variables for the polling cycle only for pure polling drivers
- ▶ Connection-related information such as ErrorMessage only 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
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 <code>driver stop</code> , the variable has the value <code>TRUE</code> and an OFF bit. After the driver has started, the variable has the value <code>FALSE</code> and no OFF bit.
SimulRTState	UDINT	60	Informs the status of Runtime for driver simulation.

CONFIGURATION

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

PhoneNumberSet	STRING	38	Telephone number, that should be used
ModemHwAdrSet	DINT	39	Hardware address for the telephone number
GlobalUpdate	UDINT	3	Update time in milliseconds (ms).
BGlobalUpdaten	BOOL	4	TRUE, if update time is global
TreiberSimul	BOOL	5	TRUE, if driver in sin simulation mode
TreiberProzab	BOOL	6	TRUE, if the variables update list should be kept in the memory
ModemActive	BOOL	7	TRUE, if the modem is active for the driver
Device	STRING	8	Name of the serial interface or name of the modem
ComPort	UINT	9	Number of the serial interface.
Baud rate	UDINT	10	Baud rate of the serial interface.
Parity	SINT	11	Parity of the serial interface
ByteSize	USINT	14	Number of bits per character of the serial interface Value = 0 if the driver cannot establish any serial connection.
StopBit	USINT	13	Number of stop bits of the serial interface.
Autoconnect	BOOL	16	TRUE, if the modem connection should be established automatically for reading/writing
PhoneNumber	STRING	17	Current telephone number
ModemHwAdr	DINT	21	Hardware address of current telephone number
RxIdleTime	UINT	18	Modem is disconnected, if no data transfer occurs for this time in seconds (s)

WriteTimeout	UDINT	19	Maximum write duration for a modem connection in milliseconds (ms).
RingCountSet	UDINT	20	Number of ringing tones before a call is accepted
ReCallIdleTime	UINT	53	Waiting time between calls in seconds (s).
ConnectTimeout	UINT	54	Time in seconds (s) to establish a connection.

STATISTICS

Name from import	Type	Offset	Description
MaxWriteTime	UDINT	31	The longest time in milliseconds (ms) that is required for writing.
MinWriteTime	UDINT	32	The shortest time in milliseconds (ms) that is required for writing.
MaxBlkReadTime	UDINT	40	Longest time in milliseconds (ms) that is required to read a data block.
MinBlkReadTime	UDINT	41	Shortest time in milliseconds (ms) that is required to read a data block.
WriteErrorCount	UDINT	33	Number of writing errors
ReadSucceedCount	UDINT	35	Number of successful reading attempts

MaxCycleTime	UDINT	22	Longest time in milliseconds (ms) required to read all requested data.
MinCycleTime	UDINT	23	Shortest time in milliseconds (ms) required to read all requested data.
WriteCount	UDINT	26	Number of writing attempts
ReadErrorCount	UDINT	34	Number of reading errors
MaxUpdateTimeNormal	UDINT	56	Time since the last update of the priority group Normal in milliseconds (ms).
MaxUpdateTimeHigher	UDINT	57	Time since the last update of the priority group Higher in milliseconds (ms).
MaxUpdateTimeHigh	UDINT	58	Time since the last update of the priority group High in milliseconds (ms).
MaxUpdateTimeHighest	UDINT	59	Time since the last update of the priority group Highest in milliseconds (ms).
PokeFinish	BOOL	55	Goes to 1 for a query, if all current pokes were 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

8. Driver-specific functions

Special specifications of the IEC 61850 standard were implemented as specific functions in the IEC 850 driver.

ESTABLISHMENT OF A CONNECTION AND DETECTION OF A CONNECTION FAILURE

The driver only opens a connection to a **Logical Device** if at least one variable from the LD is created in the project.

After communication is broken off (due to a network failure, for example), the connection to the **Logical Device** is then only reestablished automatically if at least one variable of the LD is being polled.

Recommendation: In order to detect a connection failure, create, for each LD, at least one variable for a data attribute that always occurs (for example `*/LLN0/Mod/stVal[ST]` or `*/LLN0/Mod/ctlModel[CF]`) and activate the **Permanently read variable** property (**Additional settings** group) for the variable so that the variable remains registered with the driver. This data attribute may not be communicated using **Reports** but should use polling communication.

COMMANDS (CONTROL MODEL)

The driver supports the '**Direct Operate**' and '**Select Before Operate**' commands (SBO) with normal and increased security. The driver automatically checks the actual **Control Model** and executes this command accordingly.

The required command sequences are automatically performed in the right sequence by the driver. No further configuration steps are necessary.

If a value is set to a project variable that corresponds to the `ctlVal` item in a data attribute with the name 'Oper' and it has the **Functional Constraint** CO or SP (`*/Oper.ctlVal[CO|SP]`, `*/Oper.ctlVal.i[CO]` or `*/Oper.ctlVal.f[CO]`), the driver automatically checks which **Control Model** is required for a command with this **Data Object**.

To do this, the data attribute `*/ctlModel[CF]` of this **Data Object** is read off first. Depending on the value of this data attribute, the respective **Control Model** is used and the command is executed in the corresponding procedure:

- ▶ Value 0 - **status only** :
No action, execution of the command not possible
- ▶ Value 1 - **direct control with normal security** :
An 'Operate' service is executed - the value is written to `*/Oper[CO]`.
- ▶ Value 2 - **select before operate with normal security** :
The 'Select' and 'Operate' services are executed - read of `*/SBO[CO]`, write to `*/Oper[CO]`.
- ▶ Value 3 - **direct control with enhanced security** :
An 'Operate' service is executed - the value is written to `*/Oper[CO]`.
- ▶ Value 4 - **select before operate with enhanced security** :
The 'Select' and 'Operate' services are executed - write to `*/SBOw[CO]`, write to `*/Oper[CO]`.



Information

The driver always reads the `ctlModel` data attribute from the server again in order to execute the command correctly. However the driver does not transfer the `ctlModel` to Runtime automatically. This includes situations where a variable was created for the `ctlModel`. If a variable was created, it is updated in accordance with the polling rate that is defined in the driver configuration.

DETAILS ON THE COMMAND FOR ANALOG VALUES

If it is a command to an analog value (`*/Oper.ctlVal.i[CO]` or `*/Oper.ctlVal.f[CO]`), the driver checks to see whether the respective counterpart (f or i) exists for the set value on the object model of the server. If both values are present in the object model, the other respective value is calculated using the previously-read configuration of the data object with the formula $f = ((i * scaleFactor) + offset) / 10 * unitsMultiplier$ and then both values (i and f) are sent as a structure.

MANUAL SELECTION AND CANCELLATION

The driver supports manual **Select** and **Cancel**, but only by means of the **Command Input** module in zenon. If the Command Processing sends a command and the **Select Before Operate** property is active, the driver carries out a select - in accordance with the `ctlModel` of the data object. This only applies for `ctlModel=2` or `4`. The values of the `ctlModel=0, 1` or `3` are treated as a negative select response.

Note: **Select Before Operate** means the property for the `Oper.ctlVal[CO]` variable in zenon and must not be confused with the data attribute `SBO`.

If the IEC 61850 server responds positively to the select, the driver saves the information on the selected data object and in the following command (from the second stage of the command processing), the driver only executes the operate.

If the command processing is canceled from the second stage, the driver sends a cancel.

The **Select Before Operate** property of the `ctlVal` variables has no influence on the execution of commands if the value of the `ctlVal` has been set directly and not by an action of command processing.

CANCEL OPERATE

The driver supports **Cancel operate**, but only for the **Command Processing** module in zenon.

If the **Cancel Operate** property is activated for the command variable, an **Operate** that has not yet been canceled can be canceled with the control model with enhanced safety (3 or 4). If an Operate that is running is canceled in command processing, the driver sends a Cancel command to the controller.

CTLNUM (CONTROL SEQUENCE NUMBER)

The driver automatically creates the values of the `ctlNum` attributes. If a value is set to the project variable and to the data attribute `*/Oper.ctlVal[CO]` accordingly, the driver automatically increases the value of `*/Oper.ctlNum[CO]`. The driver restarts incrementing the `ctlNum` at 0 if the value has reached 127.

The driver also counts `ctlModel=0` (no action). For this **Control Model**, the `ctlNum` variable is not updated in Runtime, because in fact - in accordance with the standard (IEC61850-7-3) - the `ctlVal` attribute should not exist.

ADDITIONAL CAUSE DIAGNOSIS AND 'COMMAND INFO' VARIABLES

The IEC61850 server keeps **AdditionalCauseDiagnosis** ready for negative responses to a command (**AddCause** in **LastApplError** structure). The possible values of the **AddCause** service parameter are listed in the IEC 61850-7-2. The driver can relay the received **AddCause** to zenon and also the current state of the current command (for example: **Select/Operate/Cancel**) and provide information on whether the command was successful.

The driver provides this information in 'Command Info' driver object variables (see Driver objects (on page 41)) if these have been created. Variables that relay the stage of the current command can be created for each `*/Oper` structure.

- ▶ `*/Oper.AddCause`: After a failed command, the value displays error information that it has received from the PLC.
- ▶ `*/Oper.ControlRun`: The status or the result of the current command:
 - 1 = command is running

- 0 = command successful
- -1 = command failed
- `*/Oper.ControlState`: The state of the command that is currently running (processed sequentially):
 - -1 = command ended
 - 0 = read `ctlModel`
 - 1 = 'Select' sent
 - 2 = 'Operate' sent
 - 3 = 'Cancel of Select' sent
 - 4 = 'Operate response' received (only in `ctlModel=3` o. 4)
 - 5 = 'Cancel of Operate' sent (only in `ctlModel=3` o. 4)

Example:

You can create the following status variables for the variable 'PLC-Marker'-BOOL 'GE650!GEDeviceF650/XCBR1/Pos/Oper.ctlVal[CO]':

Command Info - SINT: `S1!GEDeviceF650/XCBR1/Pos/Oper.AddCause`

Command Info - SINT: `S1!GEDeviceF650/XCBR1/Pos/Oper.ControlRun`

Command Info - SINT: `S1!GEDeviceF650/XCBR1/Pos/Oper.ControlState`

Case 1:

If you set the value to `*/Oper.ctlVal` directly or the variable does not have the **Select Before Operate** property, the sequence of states is as follows:

1. `*/Oper.ControlRun = 1` and `*/Oper.ControlState = 0`
2. `*/Oper.ControlState = 1` and then `*/Oper.ControlState = 2`
3. `*/Oper.ControlRun = 0` and `*/Oper.ControlState = -1`

Case 2:

If you use the two-stage command processing for `*/Oper.ctlVal` with the **Select Before Operate** property, the sequence of states for each command serve (for select, operate and cancel) is the following:

1. `*/Oper.ControlRun = 1` and `*/Oper.ControlState = 0`
2. `*/Oper.ControlState = x`
3. `*/Oper.ControlRun = 0` and `*/Oper.ControlState = -1`

For this, the following applies: $x=1$ or 2 or 3 , depending on the command service.



Attention

The **Command Info** variables do not have any value if no command was given. To evaluate these variables, you must first set a value to the `*/Oper.ctlVal[CO|SP]` variable.

'CHECK' SERVICE PARAMETER FOR COMMANDS

There are two methods for setting **Check** conditions with the following Operate or SelectWithValue commands:

1. **CheckCondition** can be pre-defined by setting a value for `*/Oper.Check[CO]`. The current value of this attribute is used for the command service parameter.
Attention: The **Check** attribute is a PACKED LIST (MMS bit string) with 2 bits. In accordance with IEC61850-8-1 8.1.3.5, the first member of the PACKED LIST is mapped to bit(0) and this bit should be the highest value (**most significant**) bit of the octet. This means that the valid set values for the **Check** data attribute 0x80 (**synchrocheck**), 0x40 (**interlock-check**), are 0xC0 (both check bits) and 0x00.
2. In zenon by setting in the command processing: Action - **Qualifier of Command** property. The valid set values are 2 - **synchrocheck** (inverted 01b), 1 - **interlock-check** (inverted 10b), 3 (both bits) and 0.

ORIGINATOR

The **Originator** data attribute **orCat** (**Category**) and **orIdent** (**Identification**) show which originator (for example a client that is at **bay level**) has caused the last change of data (issued a command for example).

The **Origin** attributes are an obligatory part of Oper, SBOw and Cancel services and optional in some cases, such as the 'Pos' data object. If this is the case then, in order to analyze the **Originator** of the value changes `*/Pos.stVal[ST]` in Runtime, you can create project variables with corresponding names in zenon: `*/Pos/origin.orCat[ST]` and `*/Pos/origin.orIdent[ST]`.

Important: To be able to analyze the origin of a data object, the following must be implemented in an IEC61850 server: the server must write the origin of the command received to this attribute, **for example** `*/Oper.origin.*[CO]`.

If a **Reports** is received, the driver forwards the **Originator** to Runtime before it forwards the other data attributes of the received data object (if only one **Origin** with **Functional Constraint** ST exists in the reported data object).

Note: You can use **orCat** or **orIdent** variables to compile a dynamic limit value text for value changes to the **stVal** variables.



Information

When a command is executed, the driver sets the following value for the Originator data attributes in SBOw, Oper and Cancel:

orCat = 2 (station control)

You can adapt the value for **orCat** in the Basic Settings (on page 15).

orIdent = 'zenon: <Computer name> (string between quotation marks - CHAR(27))', e.g. zenon: MyPC'

MAPPING OF DOUBLE POINT VALUES

In accordance with the IEC61850-7-3 standard, in [Common Data Classes](#), [Double Point Status \(DPS\)](#) and [Controllable Double Point \(DPC\)](#), the `stVal` attribute has the data type CODED ENUM with the value range: [intermediate-state](#) (00) | [off](#) (01) | [on](#) (10) | [bad-state](#) (11). The driver therefore waits until a [Double Point](#) attribute of the data type CODED ENUM is mapped to the communication protocol as an MMS of the `bit string` data type and thus can have the following values: 0x00, 0x40, 0x80 and 0xC0. This basic data type is permitted in the SCL language with the name 'Dbpos' (IEC61850-6). In zenon, it corresponds to a variable with UDINT data type (see Assignment of data types (on page 43)).

In the driver configuration, the 'Deactivate standard double point value mapping' option should remain inactive by default in order to use the modules from zenon Energy, for example the functions of `command input` and `ALC`. With the default setting, the driver assigns values of the `stVal` data attributes of DPS and DPC in accordance with the following table:

position value	zenon Dbpos value - STRING	zenon Dbpos value - UDINT unmapped	zenon Wert - value mapped
Intermediate (00)	'00'	0	2
Off (01)	'01'	64 (0x40)	0
On (10)	'10'	128 (0x80)	1
Invalid (11)	'11'	192 (0xC0)	3

The driver only converts values for variables with the name */`stVal` and numerical data type.

You can also find further information about [Double Point Values](#) in the Double Point Value Mapping (on page 72) chapter.

QUALITY BITS, TIME STAMP AND TIMEQUALITY BITS MAPPING

The driver can support received data attributes 'q' and 't' in corresponding project variables if these have been created.

However the driver always assigns the selected [Quality Bits](#) (of the 'q' attribute) and [TimeQuality Bits](#) (of the 't' attribute) of the corresponding status bits to each zenon variable that was created for an attribute of the data object and for which `FC` (on page 88) = `ST` or `FC=MX`, for example status bits of `stVal` variables.

The driver also assigns the value of the 't' attribute to the time stamp of all **st/mx** variables of the data object.

For example: Receiving the time stamp as a 't' attribute is used for the corresponding **stVal** variable in the **CEL**. The driver automatically uses the time stamp and the quality of the data object for all its variables that were created in the project. For this reason, no variables need to be created separately for the attributes 'q' and 't', except if you want to analyze quality bits that have not been assigned any status bits.



Information

The driver takes into account the precision value received from the *TimeQuality bits of accuracy*. If this value is not *unspecified* (all bits), the driver shortens the time value accordingly.

For example: Precision value < 10 bit: the driver shortens the time value received down to the milliseconds.

The selected *Quality* and *TimeQuality* bits are assigned to the status bits of the variables of the data objects that respond as follows:

IEC61850 Quality or TimeQuality bits	Status bit in zenon	Notes
Validity = Invalid, Reserved or Questionable	INVALID	Identified by a red square on the linked element.
Overflow	OV_870 (Overflow)	Also an IEC 60870 status
Source = Substituted	SB_870 (substituted)	Also an IEC 60870 status
Test	TEST (test bit)	Also an IEC 60870 status
OperatorBlocked	BL_870 (Blocked)	Also an IEC 60870 status
ClockFailure	T_INVALID (invalid time) and T_INTERN (real time internal)	The value of the 't' attribute is ignored - stVal has the time stamp of the local PC time.
ClockNotSynchronized	T_INVALID (invalid time) and T_EXTERN (real time external)	

The *TimeQuality* bit **LeapSecondsKnown** is not assigned T_INVALID.

The variables of the data object without 't' attribute (or with 't' value of 0) and the variables with other FC (on page 88) as **st/mx** receive the time stamp of the local PC time.



Information

You can evaluate status bits (such as `T_INVALID`) by using a combined element or reaction matrix ('multi' type) and you can also filter in the CEL or AML.

CAUSE OF TRANSMISSION (COT)

For each variable, the driver provides information about whether the value of the variable was received from polling or by means of a report. The information is written to the status bits `COT`.

COT value	Description
1	The value was received by means of polling
2	The value was retained using an Integrity Report . Note: This only works if Reason for inclusion is activated in the optional fields (on page 27) dialog. Otherwise the COT=3
3	The value was received by means of a report Note: If Reason for inclusion is activated in the optional fields (on page 27) dialog, only reports with the trigger option data-change , data-update or quality-change .
20	The value was retained using an General Interrogation Report . Note: This only works if Reason for inclusion is activated in the optional fields (on page 27) dialog. Otherwise the COT=3



Information

The COT value for the IEC850 driver is orientated towards the IEC60870 standard:

1 - periodic, cyclic

2 - background scan

3 - spontaneous

20 - general interrogation

FILETRANSFER

There are three functions implemented for the file transfer:

1. Request folder information
2. Get file from server
3. Delete file

1. REQUEST FOLDER INFORMATION

To request folder information:

1. Create, in your zenon project, two `file transfer driver object type` string variables with the corresponding network address:
 - a) The first variable, hereafter called the command variable, with the name "**SERVER!Command**" can also be used to get files and to delete them.
 - b) The second variable with the name "**SERVER!Directory**" is only used for the result of the folder query. It receives the folder content as legible text. For this reason, its size should correspond to the maximum size of the file name (including the folder) in the PLC.
2. For the command variable, set the value "DIR" (for the root directory) or "DIR <file_spec>".
 <file_spec> is a string that is sent to the server as a transfer parameter.

If the folder has been successfully received:

- ▶ The command variable changes its value to "DIR OK"
- ▶ The folder variable contains the received folder content as legible text. One line of this text has the following format:
 <File name>;<File length>;<Time stamp>.

2. GET FILE FROM SERVER

To get a file from the server, set the value "GET <File name>" for the command variable".

If the file has been successfully received:

- ▶ It is saved in the folder that was defined as **Directory for file transfer** in the driver dialog basic settings (on page 15).
Files from a subfolder are also stored in the main folder.
- ▶ The command variable changes its value to "GET OK"

3. DELETE FILE

To delete a file from the server, set the value "DEL <File name>". for the command variable".

If the file was deleted successfully, the command variable changes its value to "DEL OK"

TROUBLESHOOTING

If errors occur when carrying out the file transfer, such as the server responding negatively, the command variable changes its value to "XXX ERROR" (XXX = DIR, GET or DEL) and the driver optionally writes an entry into the log file (see also error analysis (on page 86)).

LIMITATIONS

Only one array level is supported. The support for arrays is limited to the data attributes. It is not possible to write array elements. When reading via polling or reporting, the whole **Data Object** that contains the array attribute is read, as with other data attributes.

You can find the information on supported or required services in Appendix C (on page 99).

8.1 Double Point Value Mapping

Double Point Value Mapping Is a standard function of the zenon Energy driver. It only influences zenon Runtime and has no influence on the driver communication with a device. Configuration is carried out in the driver settings in the **Basic Settings** tab.

The driver uses **Double Point Value Mapping** to convert values so that they are displayed in a user-friendly manner. However this only applies to HMIs.

DOUBLE POINT VALUES

The driver always communicates with one device with values for **Double Points** as defined in the Energy Standards with 2-bit information. That means:

Parameters	Double Point	Value	Meaning
Intermediate	00b	0	Switches are neither open nor closed, for example the end position has not yet been reached
Off	01b	1	Switch open
On	10b	2	Close switch/switch closed
Fault	11b	3	Error

Double Points Are coded with 2-bits in the energy sector for historical reasons: The transmission of a telegram to a serial connection (RS232) with a series of values that only contain 0 was not safeguarded against transmission errors. In order to increase the certainty, it was decided in the first standards that the value for OFF is not to be sent as 0 but as 01b, which corresponds to decimal 1. These **Double Point Values** also precisely reflect the type of how two sensors record the physical position of a switch.

However, the values sent this way may be confusing for people:

- ▶ OFF = 1
- ▶ ON = 2

Humans are used to all other devices and systems:

- ▶ OFF = 0

- ▶ ON = 1

At the same time, single point values are also defined with OFF = 0 in the same standard.

Thus everybody who is involved must always be aware of the technical level on which they are acting and receiving and sending information. In stressful situations, this can very easily lead to serious mistakes. For example, if ON is sent instead of OFF.

In order to avoid this dangerous error, the zenon Energy driver offers its own **Double Point Value Mapping**.

MAPPING

With the **Double Point Value Mapping**, all **Double Points** in zenon have the following values:

- ▶ Intermediate = 2
- ▶ Off = 0
- ▶ On = 1
- ▶ Fault = 3



Information

This function can be deactivated in the driver settings. However some functions such as command processing or ALC can no longer be used then.

Recommendation: Do not use numerical elements and numerical values to display OFF/ON or OPEN/CLOSE. Use combined elements with graphic symbols or text elements instead.

TABLE **DOUBLE POINT VALUES**

The following is applicable to the **Inactive** setting of **Deactivate standard double point value mapping**:

The driver depicts the double point values for Runtime according to the zenon column (stVal : **CODED ENUM** in **DPS** and **DPC Classes**, IEC 61850-7-3).

Position	Position Decimal	position value Binary-string	Transferred Dbpos value (unmapped)	zenon (mapped)
intermediate state	0	00	0x00	2
off	1	01	0x40	0
on	2	10	0x80	1
bad state	3	11	0xC0	3

This means in Runtime (for example):

Parameters	mapped	unmapped
off	0	64
on	1	128

You can find further information in the chapter entitled: **Driver-specific functions**

8.2 Service Tracking

Service tracking was introduced with edition 2 of the IEC61850 standard. As a basic requirement, the IEC61850 server must contain at least one logical node **LTRK** in the data model.

In order to use this functionality, corresponding additional service tracking variables must be created.

You can find detailed information on creating these variables in the Online import of an IEC 61850 server (on page 45) and Offline import (on page 48) chapters.



Attention

The additional service tracking variables - "**Service tracking**" driver object type - only get the values in Runtime if at least one ***[SR]** "PLC marker" driver object type variable of **Logical Node LTRK** in this **Logical Device** has been created in the project.

A PLC marker of **LTRK** need only exist and does not need to be requested nor shown in a screen.

If a PLC marker is created, the driver automatically requests all relevant data attributes of the following data objects with **Service tracking** from the controller: **SpcTrk, DpcTrk, IncTrk, EncTrk, ApcFTrk, ApcIntTrk, BscTrk, IscTrk, BacTrk**. These data objects all belong to the CTS common data class.

In order to be able to receive further data objects with service tracking information - that belongs to a different common data class (different than CTS):

- ▶ Create the **LTRK*[SR]** PLC marker driver object variables.
By means of online or offline import.
- ▶ Create additional "**service tracking**" driver object type service tracking variables.
This is to be carried out manually.

EXAMPLE:

To receive service tracking by means of value changes of the RptEna[BR] attribute in report control blocks 'brcbA01' and 'brcbB02', create the following variables:

- ▶ A PLC marker variable ***LTRK*BrcbTrk/rptEna[SR]**
- ▶ The "service tracking" variables: ***brcbA01_rptEna[SR]** and ***brcbB02_rptEna[SR]**

A service tracking object with FC = SR can be received from polling or by means of a report. The driver also supports data models with more than one instance of the LTRK node per logical device.

Note: If a screen contains service tracking variables that have not previously been registered, the variables remain empty until service tracking information is received by the Server.

ERRORCODE[SR] VARIABLES - ACTION FOR STATUS BIT M1

If additional service tracking variables are created for a data object, there is no pre-defined sequence in which the values are transferred from the driver to Runtime.

The consistency of data, for example for the creation of dynamic limit texts for the CEL, is however ensured by the following process:

- ▶ Each time service tracking is received, the driver sets the status bit **M1** with the service tracking variable `*_errorCode[SR]`.
- ▶ After all service tracking values have been transferred to Runtime, the driver resets the status bit M1 of the `*_errorCode[SR]` variable (`M1 = 0`).
- ▶ This resetting of the M1 status bit is a suitable trigger for the creation of a CEL/AML entry.



Attention

If you evaluate the status bit M1 with a reaction matrix and the service tracking values are received from polling and not from a report, the initial value also triggers a CEL/AML entry.

ALLCTS[SR] VARIABLE

A `*_allCTS[SR]` variable contains the received service tracking information, consisting of the data attributes of the [Common Data Class CTS](#).

- ▶ This variable is a string variable.
- ▶ The value of the variable is a text, consisting of data attributes of the [Common Data Class CTS](#) in the following sequence: `objRef, serviceType, errorCode, originatorID, t, ctlVal, origin.orCat, origin.orIdent, ctlNum, T, Test, Check, respAddCause`.
- ▶ The individual data attributes are each separated by a `;`.
- ▶ The status bits of the variable result from the sum of all status bits of the data attributes.
- ▶ The time stamp results from the time stamp of the last attribute of the service tracking object. If there is no time stamp, the current time is used.

If the service tracking object received does not contain a [Control Service \(CTS\)](#), but instead [Common \(CST\)](#) or [Control Blocks \(UTS, BTS etc.\)](#), then the `*_allCTS` variable only consists of data attributes of the [Common Data Classe CST](#): `objRef, serviceType, errorCode, originatorID, t`.

LOG ENTRIES

LOG entry	Debug Level	Description
Variable %s has wrong format for a service tracking variable.	Error	The variable addressing used does not correspond to the expected format. The following is expected: „Server!Device/Node/Object_Attribute“
Variable %s cannot be advised since no Service Tracking Variable from Driver Object Typ SPS-Merker was found.	Error	The driver cannot request the variable from the PLC because no "PLC marker" [SR] variables were created in the Editor. Please create at least one corresponding variable.

9. Reporting

Reporting was introduced by the IEC 61850 standard for spontaneous communication. The variables that are linked with a successfully activated **Report** are no longer polled, but are reported by the PLC if there is a value change. The variables that are reported if a client registers (activates) a special report control block (RCB) are saved in the DataSet (Data Set) attribute of the RCB. The DataSet attribute is a visible string for the name of the data object, the value change of which is reported spontaneously by the IEC 61850 server.

The IEC850 driver supports **Buffered** and **Unbuffered Reporting** with the data predefined in the PLC and with **Dynamic Data Sets** - with the DataSet attribute, which was dynamically defined during activation of the RCB.

As envisaged by the IEC 61850 Standard, for a **Report** data object with **Functional Constraints** (FC) the following is available:

- ▶ **Status** - ST
- ▶ **Measurands** - MX
- ▶ **Service tracking** - SR
- ▶ **Configuration** - CF
- ▶ and further: **SP, SV**

In a data set:

- ▶ Data objects with different FCs can be used together if the IEC 61850 server supports this.
- ▶ Data objects from different **Logical Devices** can be contained.

To make activation of reporting possible, the following conditions should be met:

- ▶ The data set and the **Reporting Control Block** that uses the data set must be in the same **Logical Device**.

- At least one variable of the **Logical Node**, in which the RCB is included, must have been created in the project. Otherwise the driver does not take this **Logical Node** into account at all.
Exception: LLNO. The RCBs defined in the logical node LLNO are automatically taken into account for each logical device if the LD has at least one variable that was created in the project variable list.



Attention

If you configure the data sets in the PLC - in the SCL file, you cannot add data attributes (no 'daName'), only whole data objects ('doName'). Otherwise the time stamp and quality information in the driver cannot be edited, because they are not sent by the PLC (see also standard 61850-7-2 14.2.3.2.2.9, figure 23)
The driver ignores data set definitions that contain the 'daName' data attribute.

The basic configuration of the reporting is carried out in driver configuration - server (on page 22). The (**Buffered** and **Unbuffered**) **Reports** configured in Statically assigned RCB (on page 30) have a higher priority than any **Unbuffered Reports** that may be searched for automatically. If the activation of the **Buffered Reports** is not successful, the driver will instead activate **Unbuffered Reports**, which contain the required objects as well as the data sets. If you do not want the driver to activate the **Unbuffered Reports** automatically, set the **max. auto used URCBs** property in the driver configuration to 0.

If the driver cannot activate the reports for any reason, polling is used instead.



Information

When activating reporting, the driver can (optionally) write log entries, in which it can be seen which variables are no longer polled but were successfully activated in reporting (in the 'general text' column).

TRIGGER OPTIONS

The driver supports all **Trigger Conditions**, that are used to trigger the sending of the report. The **Trigger Options** (TrgOps) define which value change of a data instance is reported. The driver attempts to activate all configured TrgOps when the report is activated, but ignores write errors. If writing is unsuccessful, the **Trigger Options** that were already configured in the SCL file (or by the IED program) of the IEC61850 server are used.

You can activate/deactivate the individual **Trigger Options** by means of the check box if you deactivate the **Use preconfigured (SCL) options** check box. This settings is true for all RCBs.

The **TrigerConditions** data attribute type that was assigned to the **MMS Bitstring(6)** in accordance with the standard if the corresponding ***TrgOps[RP|BR]** variable was created during import, corresponds to the UDINT data type in zenon by default (Motorola format, bit 0 highest value in its octet). In addition, the driver allows such variables with the data type **STRING** instead of UDINT. The value of the string has a length of 6 and only contains the characters "0" and "1" in a sequence that is the same as that of the bits defined in the standard, for example the value as a string "011111" (as UDINT=124=01111100b) means that all trigger condition bits except bit 0 - 'Reserved' are set and the value "000010" (as UDINT=8=00001000b) - only bit 4 - 'integrity'.

9.1 Unbuffered Reporting

The IEC850 driver supports two variants of **Unbuffered Reports**:

1. Automatic allocation: all existing **Reporting Control Blocks** are checked in terms of content (data sets). Reports that contain most variables of a project are registered first;
2. Static allocation: the driver uses the assignment of **Reporting Control Blocks** to the client computer as it is defined in the driver configuration - client configuration (on page 28).

AUTOMATIC ASSIGNMENT

Automatic assignment works with pre-defined data sets and pre-configured assignments of data sets to **Reporting Control Blocks** in the PLC.

Whilst the driver starts, all data sets that are present in the server are read off and ordered according to their usability (number of project variables included). This sorting indicates the sequence in which the driver attempts to activate free URCBs for reporting. At most, only as many URSBs per **Logical Device** are activated as are defined in Driver configuration (on page 22) in the **max. auto used URCBs** property. All data objects of the data sets that are successfully linked to an active RCB are no longer polled but are reported in the event of a value change.

RULES FOR CONFLICTS THAT OCCUR

When activating reporting, the driver also recognizes reports with the same content; the name of these reports can be different. Before the RCBs are sorted for the zenon project according to their usability, the driver first sorts the data objects of the data sets of each RCB and compares them in order to avoid duplicates. The following applies for conflicts when activating the **Reporting**:

- ▶ For the same reports - with identical or different sequences of the same data objects: only one report is activated.
- ▶ If a report is already occupied by another client, the next free report is activated.
- ▶ If all reports are already occupied by other clients, the data points are polled.

EXAMPLE OF AUTOMATIC ALLOCATION

The following variables are available in your project

Name	Datentyp
Server1!GDeviceF650/LLN0/urcbMXA01/BufTm[RP]	UDINT
Server1!GDeviceF650/LLN0/urcbMXA01/ConfRev[RP]	UDINT
Server1!GDeviceF650/LLN0/urcbMXA01/DatSet[RP]	STRING
Server1!GDeviceF650/LLN0/urcbMXA01/GI[RP]	BOOL
Server1!GDeviceF650/LLN0/urcbMXA01/IntgPd[RP]	UDINT
Server1!GDeviceF650/LLN0/urcbMXA01/OptFlds[RP]	UDINT
Server1!GDeviceF650/LLN0/urcbMXA01/Resv[RP]	BOOL
Server1!GDeviceF650/LLN0/urcbMXA01/RptEna[RP]	BOOL
Server1!GDeviceF650/LLN0/urcbMXA01/RptID[RP]	STRING
Server1!GDeviceF650/LLN0/urcbMXA01/SqNum[RP]	USINT
Server1!GDeviceF650/LLN0/urcbMXA01/TrgOps[RP]	UDINT
Server1!GDeviceF650/MMXU1/Hz/mag.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsA/cVal.ang.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsA/cVal.mag.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsB/cVal.ang.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsB/cVal.mag.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsC/cVal.ang.f[MX]	REAL
Server1!GDeviceF650/MMXU1/PhV.phsC/cVal.mag.f[MX]	REAL
Server1!GDeviceF650/MSQI1/SeqA.c1/mag.f[MX]	REAL
Server1!GDeviceF650/MSQI1/SeqA.c2/mag.f[MX]	REAL
Server1!GDeviceF650/MSQI1/SeqA.c3/mag.f[MX]	REAL
Server1!GDeviceF650/XCBR1/Pos/Oper.ctVal[CO]	BOOL
Server1!GDeviceF650/XCBR1/Pos/stVal[ST]	UDINT

The */urcbMXA01/* variables are for information only. The variables for the RCB attributes need not necessarily exist in the zenon project in order for reporting to work. The PLC also contains a data set GEDeviceF650/LLN0\$MMXU1\$MX with the following content.

Delete DataSet: OK	
Member	Value
GEDeviceF650/MMXU1\$MX\$TotW	{ { f0.00000i
GEDeviceF650/MMXU1\$MX\$TotVAr	{ { f0.00000i
GEDeviceF650/MMXU1\$MX\$TotVA	{ { f0.00000i
GEDeviceF650/MMXU1\$MX\$TotPF	{ { f1.00000i
GEDeviceF650/MMXU1\$MX\$Hz	{ { f0.00000i
GEDeviceF650/MMXU1\$MX\$PPV	{ { { f0.000
GEDeviceF650/MMXU1\$MX\$PhV	{ { { f0.000
GEDeviceF650/MMXU1\$MX\$A	{ { { f0.000
GEDeviceF650/MMXU1\$MX\$AuxV	{ { { f0.0000
GEDeviceF650/MMXU1\$MX\$BusHz	{ { f0.00000i

This data set is assigned to RCB LLN0\$RP\$urcbMXA01 in the PLC. To do this, enter the name of the data set in the DataSet attribute of this RCB.

LLN0	
FC ST	
FC CF	
FC DC	
FC RP	
DO urcbSTA01	{ "UnbufInps1", F, F, "GEDeviceF650/LLN0\$eveGGIO1\$ST", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbSTB01	{ "UnbufContInputs1", F, F, "GEDeviceF650/LLN0\$GGIO_F_Contacts", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbSTC01	{ "UnbufContInputs2", F, F, "GEDeviceF650/LLN0\$GGIO_G_Contacts", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbSTD01	{ "UnbufContInputs3", F, F, "GEDeviceF650/LLN0\$GGIO_H_Contacts", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbSTE01	{ "UnbufContInputs4", F, F, "GEDeviceF650/LLN0\$GGIO_I_Contacts", +0, [0111110010], +0, +0, [010010], +0, F }
DO ubrbSTF01	{ "UnbufPulseCounters1", F, F, "GEDeviceF650/LLN0\$MMTR1\$ST", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbMXA01	{ "UnbufMeas1", F, F, "GEDeviceF650/LLN0\$MMXU1\$MX", +0, [0111110010], +0, +0, [010010], +0, F }
DO urcbMXB01	{ "UnbufMeas2", F, F, "GEDeviceF650/LLN0\$MSQI1\$MX", +0, [0111110010], +0, +0, [010010], +0, F }
FC BR	
FC GO	

The RCBs LLN0\$RP\$urcbMXA01 and LLN0\$RP\$urcbMXB01 are activated when the driver starts (if they are still free).

Because there are no further suitable RCBs that have project variables in their datasets, the remaining RCBs are not used. If the project also contains a variable that is included in the data set GEDeviceF650/LLN0\$veveGGIO1\$ST, this would mean the LLN0\$RP\$urcbSTA01 is also activated.

STATIC ASSIGNMENT

For static assignment, the **Reporting Control Blocks** that are used are defined and fixed in the driver configuration - statically assigned RCB (on page 30). The driver attempts to activate all listed **Reporting Control Blocks**, regardless of their content.

If a **Dynamic Data Set** was also entered in the driver configuration, the driver in the PLC of this data set allocates this set to the RCB, in that it changes the value of this block before the report is activated. If no **Dynamic Data Set** was entered, the data set pre-defined in the PLC is valid (as with automatic allocation).

If a block was activated successfully, all data objects that are contained in its data set are no longer polled but are reported if a value is changed.

9.2 Buffered Reporting

For **Buffered Reports**, an IEC61850 server saves the changes in the time in which a connected client cannot be reached. If the client (the IEC850 driver) is online again (if the connection is reestablished, for example), the driver reads all changes and processes them; it creates omitted alarms, for example.

There is no automatic allocation for **Buffered Reporting**. The allocation of BRCBs to the driver is always static:

- ▶ Either using the driver configuration as described for static allocation with unbuffered reporting;
- ▶ Or for reasons of compatibility with older drivers that were used to configure the SCL file using the **Report ID** of the BRCBs (the **Use Report-ID for RCB assignment** should be activated in the driver configuration).

If the assignments are to be configured in the SCL file, the assigned report ID (RptID) of the BRCB must contain a string with the following nomenclature: 'zenon_Computername_*'; for example: zenon_MyPC_Measurands1. The driver then uses the data set of RCB, which has a fixed definition in the SCL file.

A **Buffered Report** can only be activated by a host (a client). To prevent the driver from activating all **Buffered Reports** and blocking access for other IEC61850 clients, they must be explicitly defined in the driver configuration (on page 10).

Irrespective of this, the basic behavior is the same as with **Unbuffered Reporting**.

BRCB BUFFER DELETE

With `report enable`, the driver sends the `purge buffer` command in the following constellations

- ▶ The first connection after Runtime has been started was established and the **Do not purge BRCB buffer at start** option in the basic settings (on page 15) is deactivated.
- ▶ It is a repeated attempt to enable a report. The command is always sent with a repeated attempt. This causes data duplicates, because data has already been received from polling. The **Do not purge BRCB buffer at start** option has no effect on the repetitions.

Hint: If the **Do not purge BRCB buffer at start** option is activated at the first attempt to enable a report, the driver sends the value of the `EntryID`. The driver saves the last-received `entry ID` for each BRCB every minute or, if the driver is closed, in a file in the Runtime data directory.



Information

After Runtime starts, the driver first reads the RCB attributes and only writes to enable if the RCB:

- ▶ Is free
- ▶ Is not yet enabled
- ▶ Is not reserved

The driver no longer reads the RCB in the event of repetitions. In this case, it always writes to enable and then checks whether the PLC confirms success or returns an error.

BUFFERED REPORTS AND ZENON REDUNDANCY

Because zenon supports redundancy (see zenon network) and a BRCB can only be activated once by a PC, you must define separately, for the project server and the Standby Server, which PC is to activate which report. The driver is started on both the server and the standby and checks the driver configuration (**Host name** property), to see if it finds a configuration that corresponds to the computer names that is running on Runtime.

For this reason, you need two copies of the **Buffered Report** in the event of a redundant configuration; one for the driver that runs on the project server and one for the driver that runs on the Standby Server. You also create two entries in the RCB assignment (on page 30) list in the driver configuration. One with the computer name of the project server and one with the computer name of the Standby Server. When entering the configuration of the project server, the first copy of the BRCB is entered and the second copy when entering the configuration of the Standby-Server.



Attention

Even if you do not use redundancy, you must nevertheless define which reports are used for the PC (**Host name** property) in the driver configuration on which zenon Runtime runs.

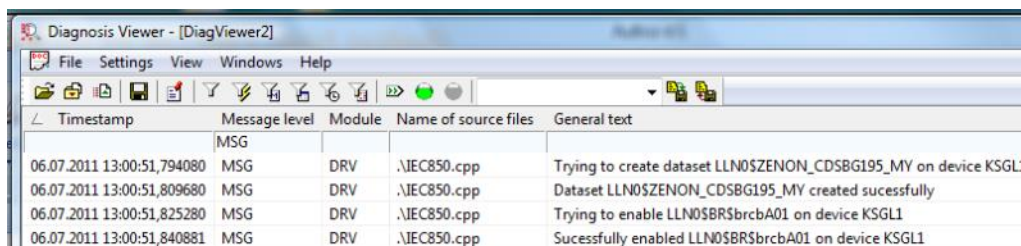
9.3 Dynamic Data Sets

To use [Dynamic Data Sets](#), the two services `CreateDataSet()` and `DeleteDataSet()` must be supported by the IEC 61850 server and the `DatSet` attribute must be able to be written to in the [Reporting Control Blocks](#) (SCL: `ReportSettings.datSet="Dyn"`). `ReportSettings.datSet="Dyn"`). When configuring, you must take into account all possible server restrictions such as the maximum number of elements in the data set (SCL: `DynDataSet.max` and `maxAttributes`).

[Dynamic Data Sets](#) lists data objects and are configured in the dynamic `DataSet` configuration (on page 34) dialog. They can be used for [Unbuffered Reporting](#) and for [Buffered Reporting](#), but only if static assignments with RCBs were configured in the driver configuration - client configuration (on page 28).

If, in the driver configuration, a [Dynamic Data Set](#) was entered for an RCB, the driver in the PLC assigns this data set to the RCB, by changing the value of the `DatSet` attribute of this block before the [Report](#) is activated.

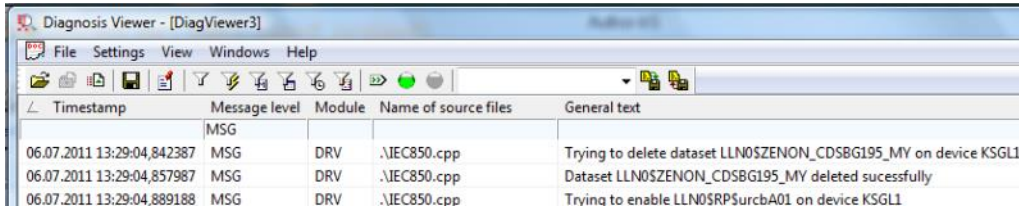
The creation of a dynamic data set can be checked in the Diagnosis Viewer (on page 86):



Timestamp	Message level	Module	Name of source files	General text
06.07.2011 13:00:51,794080	MSG	DRV	.\IEC850.cpp	Trying to create dataset LLN0\$ZENON_CDSBG195_MY on device KSGL1
06.07.2011 13:00:51,809680	MSG	DRV	.\IEC850.cpp	Dataset LLN0\$ZENON_CDSBG195_MY created successfully
06.07.2011 13:00:51,825280	MSG	DRV	.\IEC850.cpp	Trying to enable LLN0\$BRSbrcbA01 on device KSGL1
06.07.2011 13:00:51,840881	MSG	DRV	.\IEC850.cpp	Successfully enabled LLN0\$BRSbrcbA01 on device KSGL1

Before the statically assigned [Reporting Control Blocks](#) is activated, all dynamic data sets that are linked to them via driver configuration - client configuration (on page 28) are compared to those on the IEC 61850 server. All [Dynamic Data Sets](#) available on the server that are not (or no longer) required are deleted. All [Dynamic Data Sets](#) not available on the server are created and all data sets that are available on both sides are checked to see that they have the same content. If the content is different, they are deleted and created again. However, the driver only deletes the data sets that it has created itself. If the driver cannot delete a data set, the assigned RCB is activated with the pre-existing data set.

The deletion of a dynamic data set can be checked in the Diagnosis Viewer (on page 86):



Timestamp	Message level	Module	Name of source files	General text
06.07.2011 13:29:04,842387	MSG	DRV	.\IEC850.cpp	Trying to delete dataset LLN0\$ZENON_CDSBG195_MY on device KSGL1
06.07.2011 13:29:04,857987	MSG	DRV	.\IEC850.cpp	Dataset LLN0\$ZENON_CDSBG195_MY deleted successfully
06.07.2011 13:29:04,889188	MSG	DRV	.\IEC850.cpp	Trying to enable LLN0\$SRP\$urcbA01 on device KSGL1

The dynamic data sets that were created in Runtime (using the `CreateDataSet()` service) receive a client-dependent name on the server, in order for them to be easily found again. The nomenclature of the **Dynamic Data Set** name is as follows:

LLN0\$ZENON_hostname_datasetname. Here:

Hostname is the computer name of the clients as defined in the driver configuration - client configuration (on page 28) under **Hostname** and

datasetname is the name that was defined in the driver configuration - Dynamic DataSet configuration (on page 30).

If no **Dynamic Data Set** was entered in the RCB configuration of the driver, the pre-defined data set in the PLC is valid.

10. Driver commands

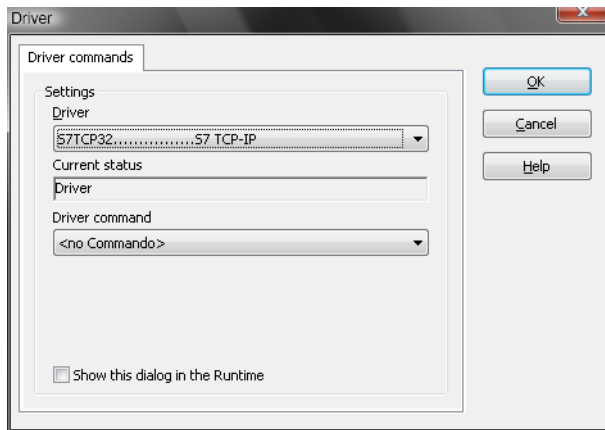
This chapter describes standard functions that are valid for most zenon drivers. Not all functions described here are available for every driver. For example, a driver that does not, according to the data sheet, support a modem connection also does not have any modem functions.

Driver commands are used to influence drivers using zenon; start and stop for example.

The engineering is implemented with the help of function **Driver commands**. To do this:

- ▶ create a new function
- ▶ select *Variables -> Driver commands*

- The dialog for configuration is opened



Parameters	Description
Drivers	Drop-down list with all drivers which are loaded in the project.
Current state	Fixed entry which has no function in the current version.
Driver commands	Drop-down list for the selection of the command.
▶ Start driver (online mode)	Driver is reinitialized and started.
▶ Stop driver (offline mode)	Driver is stopped. No new data is accepted. Note: If the driver is in offline mode, all variables that were created for this driver receive the status <code>switched off (OFF; Bit 20)</code> .
▶ Driver in simulation mode	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.
▶ Driver in hardware mode	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.
▶ Driver-specific command	Enter driver-specific commands. Opens input field in order to enter a command.
▶ Activate driver write set value	Write set value to a driver is allowed.
▶ Deactivate driver write set value	Write set value to a driver is prohibited.
▶ Establish connection with modem	Establish connection (for modem drivers) Opens the input fields for the hardware address and for the telephone number.
▶ Disconnect from modem	Terminate connection (for modem drivers)
Show this dialog in the Runtime	The dialog is shown in Runtime so that changes can be made.

DRIVER COMMANDS IN THE NETWORK

If the computer, on which the **driver command** function is executed, is part of the zenon network, additional actions are carried out. A special network command is sent from the computer to the project server, which 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.

11. Error analysis

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

11.1 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 7.20 -> Diagviewer*.

zenon driver log all errors in the log files. The default folder for the log files is subfolder `LOG` in directory `ProgramData`, example:

`C:\ProgramData\COPA-DATA\LOG`. Log files are text files with a special structure.

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 currently created entries live
- ▶ customize the logging settings
- ▶ change the folder in which the log files are saved

Note:

1. In Windows CE even errors are not logged per default due to performance reasons.
2. The Diagnosis Viewer displays all entries in UTC (coordinated world time) and not in local time.
3. 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.
4. If you only use **Error logging**, the problem description is in column **Error text**. For other diagnosis level the description is in column **General text**.
5. For communication problems many drivers also log error numbers which the PLC assigns to them. They are displayed in **Error text** and/or **Error code** and/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.
6. 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**.

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

11.2 Check list

Check your system for errors using the driver documentation; for example correct driver configuration, addressing of the variables in zenon projects, report settings in the control unit, etc.

Check most of all:

- ▶ Is the device (IEC 61850 server) connected to the power grid and ready for use?
 - Can a TCP connection to the device be created (check this using a 'ping') via the defined port (check this via Telnet)?
 - Is communication between this device and another IEC 61850 client possible?
 - Is the driver in hardware mode and not in simulation mode?
- ▶ Have you evaluated the errors that are logged in the log file of the driver?
 - Which errors occurred?
 - Was the diagnosis logging activated, at least for the error level and for all driver modules?
 - Have you taken into account that the times in the Diagnosis Viewer are always UTC?
 - Is the Diagnose Viewer set so, that all registered and all columns are displayed with the content (filtering)?
- ▶ Variable addressing:
 - Do all variables have the right addresses?
 - Are the names (or identifications) of the variables valid (SERVER!LD/LN/DO/DA[FC])?
 - Do the variables have the right driver object type, data type and value range?
- ▶ General information and reporting:
 - Is at least one variable from **Logical Device** polled?
 - Does the RCB contain the whole data objects (not just the data attributes */stVal, but also */q and */t)?
 - Does the **max. auto used URCBs** property have the correct value?
 - Is reporting available in the device?
 - If defined in the SCL file: Does the report ID of **Buffered RCB** contain the string "zenon_Computername*"?
 - For **Dynamic Data Set**: Is the data set describable in the **Reporting Control Block**?
 - Have you taken into account limitations of the device, such as the maximum number of elements in the data set etc. ?

12. Appendix A - Description of the 'Functional Constraints' (FCs):

FC	Semantics	Description
ST	Status information	Stands for status information, the value of which can be read, replaced, reported and logged, but not written.
MX	Measurands (analogue values)	Stands for status information, the measured size of which can be read, replaced, reported and logged, but not written.
CO	Control	represents a control information whose value may be operated (control model) and read.
SP	Setpoint	represent a set-point information whose value may be controlled (control model) and read.
SV	Substitution	represents a substitution information whose value may be written to substitute the value attribute and read.
CF	Configuration	represents a configuration information whose value may be written and read.
DC	Description	represents a description information whose value may be written and read
EX	Extended definition	represents an extension information providing a reference to a name space. Extensions are used in conjunction with extended definitions of LNs, DATA, and DataAttributes in 61850-7-3 and IEC 61850- 7-4. Value may be read.

13. Appendix B - Abbreviations for data object/data attribute

Term	Description
f	floating point (analog value)
i	integer (analog value)
d	Visible String(255)
q	Quality
t	Timestamp
A	Current
Acs	Access
ACSI	Abstract Communication Service Interface
Acu	Acoustic
Age	Ageing
Alm	Alarm
Amp	Current non phase related
To	Analogue
Ang	Angle
Auth	Authorization
Auto	Automatic
Aux	Auxiliary
Av	Average
B	Bushing
Bat	Battery
Beh	Behaviour
Bin	Binary
Blk	Block, blocked
Bnd	Band
Bo	Bottom
Cancel	Cancel

Cap	Capability
Capac	Capacitance
Car	Carrier
CB	Circuit Breaker
CDC	Common Data Class
CE	Cooling Equipment
Cf	Crest factor
Cfg	Konfiguration
CG	Core Ground
Ch	Channel
Cha	Charger
Chg	Change
Chk	Check
Chr	Characteristic
Cir	Circulating
Clc	Calculate
Clk	Clock, clockwise
Cls	Close
Cnt	Counter
Col	Coil
Cor	Correction
Crd	Coordination
Crv	Curve
CT	Current Transducer
Ctl	Control
Ctr	Center
Cyc	Cycle
Dea	Dead
Den	Density
Det	Detected
DExt	De-excitation
Diag	Diagnostics

Dif	Differential, difference
Dir	Richtung
Dis	Distance
DI	Delay
Dlt	Delete
Dmd	Demand
Dn	Down
DPCSO	Double point controllable status output
DQ0	Direct, Quadrature, and zero axis quantities
Drag	Drag hand
Drv	Drive
DS	Device State
Dsch	Discharge
Dur	Duration
EC	Earth Coil
EE	External Equipment
EF	Earth Fault
Ena	Enabled
Eq	Equalization, Equal
Ev	Evaluation
Ex	External
Exc	Exceeded
Excl	Exclusion
Ext	Excitation
FA	Fault Arc
Fact	Factor
Fan	Fan
Flt	Fault
Flw	Flow

FPF	Forward Power Flow
Fu	Fuse
Fwd	Forward
Gen	General
Gn	Generator
Gnd	Ground
Gr	Group
Grd	Guard
Gri	Grid
H	Harmonics (phase related)
H2	Hydrogen
H2O	Water
Ha	Harmonics (non phase related)
Hi	High, highest
HP	Hot point
Hz	Frequency
IEEE	Institute of Electrical and Electronic Engineers
Imb	Imbalance
Imp	Impedance non phase related
In	Input
Ina	Inactivity
Incr	Increment
Ind	Indication
Inh	Inhibit
Ins	Insulation
Int	Integer
ISCSO	Integer status controllable status output

Km	Kilometer
L	Lower
LD	Logical Device
LDC	Line Drop Compensation
LDCR	Line Drop Compensation Resistance
LDCX	Line Drop Compensation Reactance
LDCZ	Line Drop Compensation Impedance
LED	Light Emitting Diode
Len	Length
Lev	Level
Lg	Lag
Lim	Limit
Lin	Line
Liv	Live
Lo	Low
LO	Lockout
Loc	Local
Lod	Load, loading
Lok	Locked
Los	Loss
Lst	List
LTC	Load Tap Changer
M	minutes
M/O	Data Object is Mandatory or Optional
Max	Maximum
Mem	Memory
Min	Minimum
Mod	Mode
Mot	Motor

Ms	Milliseconds
Mst	Moisture
MT	Main Tank
N	Neutral
Nam	Name
Net	Net sum
Ng	Negative
Nom	Nominal, Normalizing
Num	Number
Ofs	Offset
Op	Operate, Operating
Oper	Operate*
Opn	Open
Out	Output
Ov	Over, Override, Overflow
Pa	Partial
Par	Parallel
Pct	Percent
Per	Periodic
PF	Power Factor
Ph	Phase
Phy	Physical
Pls	Pulse
Plt	Plate
Pmp	Pump
Po	Polar
Pol	Polarizing
Pos	Position
POW	Point on wave switching
PP	Phase to phase

PPV	Phase to phase voltage
Pres	Pressure
Prg	Progress, in progress
Pri	Primary
Pro	Protection
Ps	Positive
Pst	Post
Pwr	Power
Qty	Quantity
R	Raise
R0	Zero sequence resistance
R1	Positive sequence resistance
Rat	Winding ratio
Rcd	Record, recording
Rch	Reach
Rcl	Reclaim
Re	Retry
React	Reactance; Reactive
Rec	Reclose
Red	Reduction
Rel	Release
Rem	Remote
Res	Residual
Ris	Resistance
RI	Relation, relative
Rms	Root mean square
Rot	Rotation, Rotor
Rs	Reset, Resettable
Rsl	Result
Rst	Restraint

Rsv	Reserve
Rte	Rate
Rtg	Rating
Rv	Reverse
Rx	Receive, received
S1	Step one
S2	Step two
SBO	Select before operate*
SBOw	Select with value*
Sch	Scheme
SCO	Supply change over
SCSM	Specific Communication Service Mapping
Sec	Security
Seq	Sequence
Set	Setting
Sh	Shunt
Spd	Speed
SPI	Single Pole
SPCSO	Single point controllable status output
Src	Source
St	Status
Stat	Statistics
Stop	Stop
Std	Standard
Str	Start
Sup	Supply
Svc	Service
Sw	Switch
Swg	Swing
Syn	Synchronisation

Tap	Tap
Td	Total distortion
Tdf	Transformer derating factor
Test	Test
Thd	Total Harmonic Distortion
Thm	Thermal
TiF	Telephone influence factor
Tm	Time
	• Tmh = Time in h
	• Tmm = Time in min
	• Tms = Time in s
	• Tmms = Time in ms
Tmp	Temperature (°C)
To	Top
Tot	Total
TP	Three pole
Tr	Trip
Trg	Trigger
Ts	Total signed
Tu	Total unsigned
Tx	Transmit, transmitted
Typ	Type
Un	Under
V	Voltage
VA	Volt Amperes
Vac	Vacuum
Val	Value
VAr	Volt Amperes Reactive
Vlv	Valve

Vol	Voltage non phase related
VT	Voltage Transducer
W	Active Power
Wac	Watchdog
Watt	Active Power non phase related
Wei	Weak End Infeed
Wh	Watt hours
Wid	Width
Win	Window
Wrm	Warm
X0	Zero sequence reactance
X1	Positive sequence reactance
Z	Impedance
Z0	Zero sequence impedance
Z1	Positive sequence impedance
Zer	Zero
Zn	Zone
Zro	Zero sequence method

* SBO,SBOw,Oper, and Cancel are defined in IEC 61850-8-1:

Data attribute name	Semantics
SBO	Select – returns ACSI name of control
SBOw	SelectWithValue – receives service parameters
Oper	Operate – receives service parameters and control values
Cancel	Cancel – receives service parameters and control values

14. Appendix C - Conformance statement



Information

The [Conformance Statement](#) of the iec850 driver for the IEC61850 protocol is in English only.

14.1 Protocol implementation conformance statement (PICS)

A.1 - BASIC ACSI CONFORMANCE STATEMENT

		Client / subscriber	Server / publisher	Value / comments
Client-Server roles				
B11	Server side (of TWO-PARTY-APPLICATION-ASSOCIATION)	-	N	for Server see zenon Logic
B12	Client side of (TWO-PARTY-APPLICATION-ASSOCIATION)	Y	-	
SCSMs supported				
B21	SCSM: IEC 61850-8-1 used	Y		
B22	SCSM: IEC 61850-9-1 used			not supported
B23	SCSM: IEC 61850-9-2 used			not supported
B24	SCSM: other			not supported
Generic substation event model (GSE)				
B31	Publisher side	-		
B32	Subscriber side	N		
Transmission of sampled value model (SVC)				
B41	Publisher side	-		
B42	Subscriber side	N		
- = not applicable Y = supported N or empty = not supported				

A.2 - ACSI MODELS CONFORMANCE STATEMENT

		Client / subscriber	Server / publisher	Value / comments
If Server or Client side (B11/12) is supported				
M1	Logical device	Y		

M2	Logical node	Y		
M3	Data	Y		
M4	Data set	Y		
M5	Substitution	Y		
M6	Setting group control	Y		
Reporting				
M7	Buffered report control	Y		
M7-1	sequence-number	Y		
M7-2	report-time-stamp	Y		
M7-3	reason-for-inclusion	Y		
M7-4	data-set-name	Y		
M7-5	data-reference	Y		
M7-6	buffer-overflow	Y		
M7-7	entryID	Y		
M7-8	Buflm	Y		
M7-9	IntgPd	Y		
M7-10	GI	Y		
M8	Unbuffered report control	Y		
M8-1	sequence-number	Y		
M8-2	report-time-stamp	Y		
M8-3	reason-for-inclusion	Y		
M8-4	data-set-name	Y		
M8-5	data-reference	Y		
M8-6	Buflm	Y		
M8-7	IntgPd	Y		
M8-8	GI	Y		
Logging				
M9	Log control	N		
M9-1	IntgPd	N		
M10	Log	N		
M11	Control	N		

If GSE (B31/B23) is supported				
M12	GOOSE	N		
M13	GSSE	N		
If SVC (B41/B42) is supported				
M14	Multicast SVC	N		
M15	Unicast SVC	N		
If Server or Client side (B11/12) is supported				
M16	Time	N		Time of Operating System must be available with required accuracy
M17	File Transfer	Y		
Y = service is supported				
N or empty = service is not supported				

A.3 - ACSI SERVICE CONFORMANCE STATEMENT

ACSI Service Conformance		AA: TP/MC	Client (C)	Server (S)	Comments
Server					
S1	GetServerDirectory	TP	Y		
Application association					
S2	Associate		Y		
S3	Abort		Y		supported on TCP/IP
S4	Release		N		
Logical device					
S5	GetLogicalDeviceDirectory	TP	Y		
Logical node					
S6	GetLogicalNodeDirectory	TP	Y		
S7	GetAllDataValues	TP	N		not supported

Data					
S8	GetDataValues	TP	Y		
S9	SetDataValues	TP	Y		
S10	GetDataDirectory	TP	Y		
S11	GetDataDefinition	TP	Y		= GetDataDirect ory
Data set					
S12	GetDataSetValues	TP	N		not supported
S13	SetDataSetValues	TP	N		not supported
S14	CreateDataSet	TP	Y		
S15	DeleteDataSet	TP	Y		
S16	GetDataSetDirectory	TP	Y		
Substitution					
S17	SetDataValues	TP	Y		
Setting group control					
S18	SelectActiveSG	TP	Y		
S19	SelectEditSG	TP	Y		
S20	SetSGValues	TP	Y		
S21	ConfirmEditSGValues	TP	Y		
S22	GetSGValues	TP	Y		
S23	GetSGCBValues	TP	Y		
Reporting					
Buffered report control block (BRCB)					
S24	Report	TP	Y		
S24-1	data-change (dchg)		Y		
S24-2	quality-change (qchg)		Y		
S24-3	data-update (dupd)		Y		
S25	GetBRCBValues	TP	Y		
S26	SetBRCBValues	TP	Y		
Unbuffered report control block (URCB)					

S27	Report	TP	Y		
S27-1	data-change (dchg)		Y		
S27-2	quality-change (qchg)		Y		
S27-3	data-update (dupd)		Y		
S28	GetURCBValues	TP	Y		
S29	SetURBCValues	TP	Y		
Logging					
Log control block					
S30	GetLCBValues	TP	Y		
S31	SetLCBValues	TP	Y		
Log					
S32	QueryLogByTime	TP	N		not supported
S33	QueryLogAfter	TP	N		not supported
S34	GetLogStatusValues	TP	Y		

ACSI Service Conformance		AA: TP/MC	Client/ subscriber	Server/ publisher	Comments
Generic substation event model (GSE)					
GOOSE-CONTROL-BLOCK					
S35	SendGOOSEMessage	MC	C8	C8	not supported
S36	GetGoReference	TP	O	C9	not supported
S37	GetGOOSEElementNumber	TP	O	C9	not supported
S38	GetGoCBValues	TP	Y	O	
S39	SetGoCBValues	TP	Y	O	
GSSE-CONTROL-BLOCK					
S40	SendGSSEMessage	MC	C8	C8	not supported
S41	GetGsReference	TP	O	C9	not supported
S42	GetGSSEDataOffset	TP	O	C9	not supported
S43	GetGsCBValues	TP	Y	O	
S44	SetGsCBValues	TP	Y	O	
C-8 shall declare support for at least one (SendGOOSEMessage or SendGSSEMessage)					
C-9 shall declare support if TP association is available					
Transmission of sampled value model (SVC)					
Multicast SVC					
S45	SendMSVMessage	MC	N	C10	not supported
S46	GetMSVCBValues	TP	N	O	not supported
S47	SetMSVCBValues	TP	N	O	not supported
Unicast SVC					
S48	SendUSVMessage	TP	N	C10	not supported
S49	GetUSVCBValues	TP	N	O	not supported
S50	SetUSVCBValues	TP	N	O	not supported
C10 – shall declare support for at least one (SendMSVMessage or SendUSVMessage)					

ACSI Service Conformance		AA: TP/MC	Client (C)	Server (S)	Comments
Control					
S51	Select		Y		
S52	SelectWithValue	TP	Y		
S53	Cancel	TP	Y		
S54	Operate	TP	Y		
S55	Command-Termination	TP	Y		
S56	TimeActivated-Operate	TP	N		not supported
File Transfer					
S57	GetFile	TP	Y		
S58	SetFile	TP	N		not supported
S59	DeleteFile	TP	Y		
S60	GetFileAttributeValues	TP	Y		

ACSI Service Conformance		Client (C)	Server (S)	Comments
Time				
T1	Time resolution of internal clock	Operating System		
T2	Time accuracy of internal clock	Operating System	T1	T0 (10ms), T1 (1ms), T2 (100µs), T3 (25µs), T4 (4µs), T5 (1µs)
T3	Supported TimeStamp resolution	T1 (1ms)		or LREAL decimal place when t-attribute created as variable (T3)

14.2 Model implementation conformance statement (MICS)

All logical nodes, common data classes and data attribute types - as defined in IEC 61850-7-3 and IEC 61850-7-4 (Standard Edition 1) - are supported.

Introduced by Edition 2 for common data class APC the data attribute ctIVal as AnalogValue (structure) with FC=CO is supported in read-only Services; ctIVal[SP] in APC from Edition1 is supported in all Services.

15. Appendix D - PIXIT for COPA-DATA IEC61850 Client



Info

The PIXIT document for the certification test of the IEC850 driver is only available in English.

PIXIT based on a template provided by KEMA.

INTRODUCTION

This document specifies the protocol implementation extra information for testing (PIXIT) of the IEC 61850 interface in the client system: “COPA-DATA IEC850 - Driver for IEC61850 protocol”, further referred to as “Client”.

Together with the PICS and the MICS the PIXIT forms the basis for a conformance test according to IEC 61850-10.

The following chapters specify the PIXIT for each applicable ACSI service model as structured in IEC 61850-10 and the “Conformance Test Procedures for Client System with IEC 61850-8-1 interface” by KEMA.

PIXIT FOR CONFIGURATION

Description	Value / Clarification
Describe how the client handles nameplate configuration revision mismatches	<i>The Client reads out data model at each associate and therefore does not need to check configuration revision.</i>
Describe how the client handles report control block configuration revision mismatches	<i>RCB configuration is read out at each associate.</i>
<additional items>	<i>The Client does not use items of pre-defined data model until reading them from data model from an online server.</i> <i>The SCL file is used only by offline import of variables to HMI project in the Editor for pre-configuration.</i> <i>Please refer Client documentation for more details.</i>

PIXIT FOR ASSOCIATION MODEL

Description	Value / Clarification
Guaranteed number of servers that can set-up an association simultaneously (one association per server)	<i>Per design: 65535</i> <i>Tested: 1000</i>
Lost connection detection time range (default range of TCP_KEEPAIVE is 1 – 20 seconds)	<i>The current TCP/IP timeout of Operating System</i> <i>The connection loss to LD is only detected if at least one its attribute is currently polled.</i> <i>Please refer Client documentation for more details</i>
Lost (abort) connection retry time	<i>30 seconds</i>
Is authentication supported	<i>Y</i>
What is the maximum and minimum MMS PDU size	<i>Max MMS PDU size 65535</i> <i>Min MMS PDU size depends on largest DO</i>
What is the typical startup time after a power supply interrupt	<i>The Client works only together with the zenon Runtime and both are Windows applications.</i> <i>1. Windows does not guarantee the files will be not corrupted after power failure. Please use an UPS to prevent loss of data stored on HDD.</i> <i>2. There is no typical start-up time for the Client as the start-up time of the Runtime depends of HMI project design and PC performance. In practice, the Client opens TCP/IP session to a server after few (1-4) seconds since application start.</i>

PIXIT FOR SERVER MODEL

Description	Value / Clarification
Maximum object identification length	<i>LD: 128 chars/octetets LN: 16 chars/octetets DO: 128 chars/octetets DA: 128 chars/octetets FC: 2 chars/octetets</i>
Does client support autodescription	<i>In the Runtime the Client reads data model only via MMS. Then the Client does not uses data not responded via MMS by association.</i> <i>In the HMI Editor (by pre-configuration of the HMI project) the Client can read the data model from SCL file (off-line import) or from the server via MMS (on-line import) or data can be configured manually (variable creation).</i> <i>Please refer HMI documentation for more details.</i>
What analogue value (MX) quality bits are used in the client	Y Good, Y Invalid, Y Reserved, Y Questionable Y Overflow Y OutofRange Y BadReference Y Oscillatory Y Failure Y OldData Y Inconsistent Y Inaccurate Y Process Y Substituted Y Test Y OperatorBlocked
Which status value (ST) quality bits are used in the client	Y Good, Y Invalid, Y Reserved, Y Questionable Y BadReference Y Oscillatory Y Failure Y OldData Y Inconsistent Y Inaccurate Y Process Y Substituted Y Test Y OperatorBlocked
Describe how to view/display quality values	<i>1.the status bits of */stVal variable in HMI, if created (only selected bits; please refer Client</i>

	<i>documentation for more details)</i> <i>2.the value of */q variable in HMI, if created</i>
Describe how to force a SetDataValues request	<i>Set value to the variable created in HMI.</i> <i>Please refer HMI documentation for more details.</i>
Describe how to force a GetAllDataValues request	<i>not used</i>
Describe how the client behaves in case of: <ul style="list-style-type: none"> ▶ GetDataDefinition response- ▶ GetLogicalDeviceDirectory response- ▶ GetAllDataValues response- ▶ GetDataValues response- ▶ SetDataValues response- 	<ul style="list-style-type: none"> ▶ <i>I-Bit* + log error entry</i> ▶ <i>I-Bit* + log error entry</i> ▶ <i>not supported</i> ▶ <i>I-Bit* + log error entry</i> ▶ <i>log error entry</i> <i>*Status bit INVALID of variables mirroring DO/DA.</i> <i>Please refer HMI documentation for more details.</i>

PIXIT FOR DATA SET MODEL

Description	Value / Clarification
Describe how to force a GetDataSetValues request	<i>not used</i>
Describe how to force a SetDataSetValues request	<i>not used</i>
Describe how to force a DeletaDataSet request	<i>would be done automatically according Client configuration for Dynamic Data Sets. Please refer Client documentation for more details.</i>
Describe how the client handles following dataset mismatches between the SCL and the data sets exposed via MMS: <ol style="list-style-type: none"> 1. new dataset element 2. missing dataset element 3. Reordered dataset members in a dataset of a different data type 4. Reordered dataset members in a dataset of the same data type 	<i>No mismatch possible - SCL file is not used in the Client during communication; the Client only uses data sets exposed via MMS.</i>
Describe how the client behaves in case of: <ul style="list-style-type: none"> ▶ GetLogicalNodeDirectory(DATA-SET) response- ▶ GetDataSetDirectory response- 	<i>log entry + data will be polled (no reporting)</i>
Does the client support the creation of: <ul style="list-style-type: none"> ▶ persistent datasets ▶ non-persisten datasets 	<ul style="list-style-type: none"> ▶ Y ▶ N
Describe how the client behaves in case of: <ul style="list-style-type: none"> ▶ CreateDataSetDirectory response- ▶ DeleteDataSet response- 	<ul style="list-style-type: none"> ▶ <i>log entry + data will be polled (not reported)</i> ▶ <i>log entry + old dataset used for reporting</i>

PIXIT FOR SUBSTITUTION MODEL

Description	Value / Clarification
Describe how to substitute a value	<i>For attributes with FC = SV (* /subEna, subQ, subID and e.g. subVal) create variables in the HMI project. Then first set values to be substituted (* /subVal, subQ, subID) and then set value TRUE to variable * /subEna. Please refer HMI documentation for more details about setting variable values</i>
<additional items>	<i>There is an alternative method in the HMI for use instead of substitution - e.g. when an IED does not support substitution. Any variable created in the HMI project can be turn to alternative value and than the HMI simulates it. Please refer description of status bit ALT_VAL in the HMI manual for more details.</i>

PIXIT FOR SETTING GROUP CONTROL MODEL

Description	Value / Clarification
Describe how to change the active setting group	<i>set value of adequate variable in HMI project</i>
Describe how to get the actual setting group values	<i>create variables in the HMI project for attributes with these values</i>
Describe how to edit setting group values	<i>set value of adequate variables in HMI project</i>
Describe how the client behaves in case of: <ul style="list-style-type: none"> ▶ GetSGCBValues response- ▶ The configured SG is different then the actual setting group 	<ul style="list-style-type: none"> ▶ <i>log entry</i> ▶ <i>not tested</i>
<additional items>	

PIXIT FOR REPORTING MODEL

Description	Value / Clarification
Does the client search for RCB in all logical nodes? when not specify the logical nodes	<p>Y - All logical nodes*</p> <p>Y - The following logical nodes**:</p> <ul style="list-style-type: none"> ▶ LLNO ▶ RCBs in any LN when defined in the Client configuration ▶ BRCBs in any LD with RptID=zenon_computername_* <p>* for automatic allocation of URCBs ** for static allocation of URCBs and BRCBs Please refer Client documentation for more details.</p>
Which dynamic RCB attributes are/can be configured by the client	<p>N RptID</p> <p>Y DataSet</p> <p>Y Optional fields</p> <p>Y Trigger conditions</p> <p>Y Buffer time</p> <p>Y Integrity period</p> <p><i>For attributes with FC = BR RP variables in the HMI project can be created, read, written.</i></p>
Does the client supports IED's with indexed and non-indexed report control blocks (RCB)	<p>Y Buffered RCB indexed</p> <p>Y Buffered RCB not indexed</p> <p>Y Unbuffered RCB indexed</p> <p>Y Unbuffered RCB not indexed</p>
The supported trigger conditions are	<p>Y integrity</p> <p>Y data change</p> <p>Y quality change</p> <p>Y data update</p> <p>Y general interrogation</p>
The minimum required optional fields are	<p>N sequence-number</p> <p>N report-time-stamp</p> <p>N reason-for-inclusion</p> <p>Y* data-set-name</p> <p>N data-reference</p> <p>N buffer-overflow</p> <p>Y* entryID</p> <p>N conf-rev</p> <p><i>The Client can receive reports with any additional optional field, not only the minimum; the optional fields not required are then ignored.</i></p>
Does the client support segmented reports	Y
Does the client support pre-assigned RCB	Y
Does the client support reported data set containing structured data objects or data	<p>Y reporting of data objects</p> <p>N* reporting of data attributes</p>

attributes?	<i>*the Client process report but ignores single data attributes while they do not have time and quality. Skipped daName items of data set will be polled.</i>
Describe how the client does respond when an URCB is already reserved	<i>log entry + trying to use another one if dynamically allocated, fallback polling</i>
Describe how the client does respond when a BRCB is already reserved	<i>log entry, data will be polled; optionally the Client can periodically try to enable the RCB</i>
Describe how the client does respond on a SetBRCBValues(EntryID) respond-	<i>log entry; as next the Client tries to enable BRCB anyway</i>
Describe how the client does respond when a report has an unknown: dataset, RptId, unexpected number of dataset entries, and/or unexpected data type format entries	<i>mismatches are not possible - dataset directory and RCB configuration is read out via MMS at each associate</i>
Describe how the client detect reporting configuration changes (mismatches). Does it check the "configuration revision" attributes and/or does it check the dataset members?	<i>RCB configuration is read out at each associate, "configuration revision" doesn't care</i>
Describe how to force the client to change the RCB buffertime	<i>The setting in the Client configuration For attributes with FC = BR RP variables in the HMI project can be created, read, written.</i>

PIXIT FOR LOGGING MODEL

Description	Value / Clarification
Does the client search for LCB in all logical nodes? when not specify the logical nodes	<i>not supported</i>
Describe how to change LOG and LCB attributes	<i>not supported</i>

PIXIT FOR GENERIC SUBSTATION EVENTS MODEL

Description	Value / Clarification
<p>What elements of a subscribed GOOSE header are checked to decide the message is valid and the allData values are accepted?</p> <p>Ignored = element value is not checked, message will be accepted</p> <p>SCL match = element value should match with the configuration, otherwise the GOOSE message will be ignored</p>	<i>not supported</i>
<p>For the checked GOOSE header elements describe the checking conditions in more detail when necessary</p>	<i>not supported</i>
<p>What is the behavior when one subscribed GOOSE message isn't received or syntactically incorrect (missing GOOSE)</p>	<i>not supported</i>
<p>What is the behavior when one subscribed GOOSE message exceeds the previous time Allowed to Live (TAL)</p>	<i>not supported</i>
<p>What is the behavior when a subscribed GOOSE message is out-of-order</p>	<i>not supported</i>
<p>What is the behavior when a subscribed GOOSE message is duplicated</p>	<i>not supported</i>
<p>May the GOOSE data set contain structured data objects?</p>	Y/N

PIXIT FOR CONTROL MODEL

Description	Value / Clarification
What control modes are supported	<p>Y status-only Y direct-with-normal-security Y sbo-with-normal-security Y direct-with-enhanced-security Y sbo-with-enhanced-security</p> <p><i>For each use of */Oper.ctlVal[CO SP], the Client automatically checks current */ctlModel[CF]. Depending on the value of that attribute, the corresponding control model is executed. Please refer Client documentation for more details.</i></p>
Is Time activated operate (operTm) supported	N
Is "operate-many" supported	N
Can the client set the test flag?	Y
What check conditions can be set	<p>Y synchrocheck Y interlock-check</p> <p><i>There are two methods to set check conditions for Operate and SelectWithValue:</i></p> <p><i>1. Check condition can be predefined by setting a value to */Oper.Check[CO] variable. The current variable value is used for command service parameter;</i></p> <p><i>2. in the IEC 61850 interface (HMI): by setting in Command processing: Action its property Qualifier of Command (Please refer HMI Help)</i></p>
Which originator categories are supported and what is the originator identification?	<p>orCat =2 (station control)* orIdent ='zenon: nnn' where: nnn - computer name ' - apostrophe (CHAR(27))</p> <p><i>*per default orCal=2 - setting in the Client configuration</i></p>
Describe if and how the client sets/increments the ctlNum	<p><i>for each use of any ctlVal[CO SP] the Client increments* ctlNum from 0 to 127 and next turns to 0.</i></p> <p><i>* also by ctlModel=0 (no control possible)</i></p>
What does the client when its receives a LastApplicationError and describe how to view the additional cause?	<p><i>The Client gives possibility to create (pro */Oper structure) an internal "State Info" variable for received AddCause item value and variables ControlRun and ControlState showing current state and success/fail of command. These internal variables can be display and store in the HMI project.</i></p> <p><i>Please refer Client documentation for more details.</i></p>
What does the client when its receives a Select,	<p><i>The Client sets corresponding values to intern variables ControlRun (-1=error) and AddCause.</i></p>

SelectWithValue or Operate respond negative ?	<i>Optionally writes a log entry. Please refer Client documentation for more details.</i>
Can the client change the control model via online services?	Y
What does the client when the ctlModel is not initialized in the SCL?	<i>out of scope The Client does not read ctlModel from SCL, only via MMS</i>
<additional items>	

PIXIT FOR TIME AND TIME SYNCHRONIZATION MODEL

Description	Value / Clarification
Described how to view the internal time & quality or how to expose the timestamp and timestamp quality via the IEC 61850 interface	<i>The Client uses Windows time (local PC clock), thus the Client does not cooperate directly with any SNTP application. The Client does not send timestamp quality to control object.</i>
What time quality bits are supported	<i>N LeapSecondsKnown Y* ClockFailure Y* ClockNotSynchronised *only time quality bit received in t attribute</i>
What is the behavior when the time synchronization signal/messages are lost	<i>Not applicable</i>
When is the quality bit "Clock failure" set?	<i>The Client does not set any time quality bits, it supports reception of ClockFailure</i>
When is the quality bit "Clock not synchronised" set?	<i>The Client does not set any time quality bits, it supports reception of ClockNotSynchronised</i>
<additional items>	<i>The Client maps time quality bits received in t attribute into Status bits of variables responding same DO in the HMI project:</i> <ul style="list-style-type: none"> ▶ <i>ClockNotSynchronised</i> - status bits T_INVALID & T_EXTERN ▶ <i>ClockFailure</i> - status bits T_INVALID & T_INTERN (and changes time attribute to HMI time). <i>Please refer Client documentation for more details.</i>

PIXIT FOR FILE TRANSFER MODEL

Description	Value / Clarification
Describe when or how to force the client to request GetServerDirectory(FILE) and what it does with the responded filenames	<i>The Client gives possibility to create two internal string variables: a Command and a Directory variable. Setting value of Command to: "DIR <fileSpec>" results with the name or list of files in Directory variable. Please refer Client documentation for more details.</i>
Does the client uses a wildcard in the GetServerDirectory(FILE) request	No
Does the client support IED's that include the path in the file name in the GetServerDirectory(FILE) respond?	Y path included Y path not included Y list of files with a char separator
Does the client support IED's that use the fileseparator	N "/" Y "\"
What is the maximum file name size including path	OS limit
Can the client read a file with size 0	Y
Are directory/file name case sensitive	Not case sensitive only Command value is case sensitive: "DIR <fileSpec>" or "GET <fileSpec>", the <fileSpec> is not case sensitive
Maximum file size	per design 4 GB practical* 25 MB * transfer of 25 MB takes about 40 seconds, bigger transfers wasn't tested
Describe how the client behaves in case of: GetFileAttributes response-	Command Variable value changes to "DIR ERROR" (or "GET ERROR" for GetFile response-)
<additional items>	For DeleteFile set Command variable value to "DEL <fileSpec>"