zenon Handbuch

Process Gateway

v.8.00

©2018 Ing. Punzenberger COPA-DATA GmbH

Alle Rechte vorbehalten.

Die Weitergabe und Vervielfältigung dieses Dokuments ist - gleich in welcher Art und Weise - nur mit schriftlicher Genehmigung der Firma COPA-DATA gestattet. Technische Daten dienen nur der Produktbeschreibung und sind keine zugesicherten Eigenschaften im Rechtssinn. Änderungen - auch in technischer Hinsicht - vorbehalten.

Inhaltsverzeichnis

1. Willkommen bei der COPA-DATA Hilfe 7

2. Process Gateway 7

3. Allgemeines 9

3.1 Variablenzuordnung über Namen 9

4. Voraussetzungen 11

5. Start 11

5.1 Start mit Startup Tool 13

5.2 Start mit zenon Funktion 14

5.3 Start mit Kommandozeile 16

5.4 Modulauswahl 16

5.4.1 Erstmaliger Start 17

5.4.2 Bestehende Modulauswahl ändern 17

5.4.3 Start mit konfiguriertem Modul 18

5.5 INI-Datei 19

6. DEC 21

6.1 Modulkonfiguration 21

6.2 Logging 22

7. DNP3_SG 22

7.1 Installation 23

7.2 Outstation 24

7.3 Datalink 27

7.3.1 Application Layer 31

7.3.2 Events/Unsolicited 35

7.3.3 File Transfer 39

7.4 Binary Input 41

7.5 Double Bit Binary Input 50

7.6 Binary Output 59

7.7 Running Counter 70

7.8 Frozen Counter ... 78
7.9 Analog Input ... 86
7.10 Analog Output ... 96
7.11 Octet String ... 105
7.12 Variablenauswahldialog ... 112
7.13 Warndialoge ... 113
7.14 Statusmapping zenon - DNP3 ... 116
7.15 Select before Operate ... 118
7.16 Kommando-Bearbeitung - Command Routing ... 118
7.17 LATCH_ON und LATCH_OFF ... 121

8. DNP3 Slave - AccessDNP3 veraltet ... 123

9. ICCP-TASE.2 ... 124

9.1 Modulkonfiguration ... 126
9.1.1 General ... 128
9.1.2 Server variables ... 135
9.1.3 Client variables ... 139
9.2 Statusvariablen der Verbindung ... 144
9.3 Nicht vorhandene Variablen in Runtime anzeigen ... 145
9.4 Direct control (non SBO) für ICCP control objects ... 146
9.5 Konfigurationsdatei: INI-Einträge für ICCP-Tase.2 ... 148
9.6 LOG-Einträge ... 152
9.7 Interoperability ... 155

10. IEC870 Slave ... 156

10.1 Modulkonfiguration ... 157
10.1.1 Algemeine Einstellungen ... 161
10.1.2 Devices Liste ... 164
10.1.3 Device - Konfiguration ... 166
10.1.4 Sector - Konfiguration ... 170
10.1.5 104 settings ... 192
10.1.6 DPI/DCS Mapping ... 194
10.1.7 Konfiguration via XML-Datei ... 195
10.2 Sequence of Events (SOE) ... 205
10.2.1 Warnungen ... 207

10.2.2 Fehlermeldungen 208
10.3 Kommunikation 209
10.3.1 Start 209
10.3.2 Generalabfrage 210
10.3.3 Verbindungsende und Resets 211
10.3.4 Übertragung der Wertänderungen 213
10.3.5 Nachladen der Runtime 213
10.3.6 Datenprioritäten 214
10.3.7 Interner Typ T00 (Statusvariablen) 214
10.3.8 Mapping von Statusbits in Melderichtung 219
10.3.9 Befehle 222
10.3.10 Read - T102 - Anforderung 224
10.3.11 Zeit-Synchronisierung - T103 - Anforderung 225
10.3.12 Test - T104 und T107 - Anforderungen 225
10.3.13 Dateitransfer 226
10.4 Broadcasts 229
10.5 Fehleranalyse 229
10.5.1 Analysetool 230
10.5.2 Fehlercodes 236
10.5.3 Checkliste 237
10.6 Interoperability 237

11. MODBUS Slave 255
11.1 Modulkonfiguration 256
11.2 Holding Registers 258
11.3 Coils 260
11.4 Address Settings 262
11.5 Hardware-Adresse des MODBUS-Systems 263
11.6 Konfigurationsdatei: spezifische Einträge für AccessMODBUS 263

12. MS Azure 265
12.1 Modulkonfiguration 266
12.1.1 MS Azure Connection Settings 271
12.2 Nachrichtenformate 275
12.3 Fehlermeldungen 277
12.4 Konfigurationsdatei: spezifische Einträge für MS Azure 279

13. OPC UA Server 280

13.1 Vergleich OPC Server und OPC UA Server 281

13.2 Modulkonfiguration 283

13.2.1 Server 284

13.2.2 Variables 291

13.2.3 Archives 295

13.3 Start 297

13.4 Verbindung 298

13.5 Sicherheit 298

13.6 Alarms and Conditions 298

13.7 Service Sets 301

13.8 Kommunikation 302

13.9 Konfigurationsdatei: INI-Einträge für OPC UA 305

13.10 Fehlermeldungen 308

13.11 Status Mapping 311

14. SNMP Agent 312

14.1 SNMP Architektur 312

14.2 Management Information Base (MIB) 313

14.3 MIB Aufbau 314

14.4 Modulkonfiguration 316

14.5 Konfigurationsdatei: spezifische Einträge für Access SNMP 317

14.6 Konfiguration des Windows SNMP Agent 317

15. SQL 319

15.1 Modulkonfiguration 319

15.2 Variablenauswahl 320

15.3 Tabellenerzeugung 321

15.4 Betrieb im Redundanzsystem 321

15.5 Konfigurationsdatei: spezifische Einträge für AccessSQL 322

15.6 Tabellenformat 323

15.6.1 Report für Prozessabbild 323

15.6.2 Tabelle für Variablen-IDs 323

1. Willkommen bei der COPA-DATA Hilfe

ZENON VIDEO-TUTORIALS

Praktische Beispiele für die Projektierung mit zenon finden Sie in unserem YouTube-Kanal (https://www.copadata.com/tutorial_menu). Die Tutorials sind nach Themen gruppiert und geben einen ersten Einblick in die Arbeit mit den unterschiedlichen zenon Modulen. Alle Tutorials stehen in englischer Sprache zur Verfügung.

ALLGEMEINE HILFE

Falls Sie in diesem Hilfekapitel Informationen vermissen oder Wünsche für Ergänzungen haben, wenden Sie sich per E-Mail an documentation@copadata.com.

PROJEKTUNTERSTÜTZUNG

Unterstützung bei Fragen zu konkreten eigenen Projekten erhalten Sie vom Support-Team, das Sie per E-Mail an support@copadata.com erreichen.

LIZENZEN UND MODULE

Sollten Sie feststellen, dass Sie weitere Module oder Lizenzen benötigen, sind unsere Mitarbeiter unter sales@copadata.com gerne für Sie da.

2. Process Gateway

*Das **Process Gateway** dient zur Kopplung mit übergeordneten Systemen, z. B. einem Control Center. Dies bedeutet, dass Teile des zenon Prozessabbildes für andere Anwendungen zur Verfügung gestellt und von diesen aktualisiert werden.*

- Prozessdaten aus der zenon Runtime können an übergeordnete Systeme weitergeleitet werden.
- Übergeordnete Systeme können ihre Werte oder Befehle zur zenon Runtime schreiben.

Info

*Die Benutzeroberfläche für das **Process Gateway** steht nur in englischer Sprache zur Verfügung.*

Alle genutzten Komponenten müssen lizenziert sein.

Hinweis: Sowohl Editor und/oder Runtime, welche die Lizenzinformationen zur Verfügung stellen, sowie die Anwendungen, die darauf zugreifen, müssen im gleichen Benutzerkontext laufen. Sie müssen also vom gleichen Benutzer gestartet werden. Eine Anwendung als Administrator und eine andere als lokaler Benutzer zu starten funktioniert nicht.

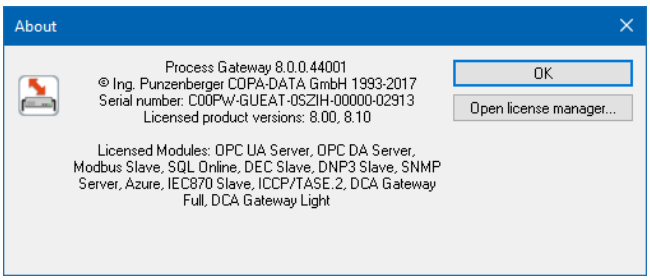
LIZENZINFORMATIONEN ANZEIGEN

Um Lizenzinformationen im **Process Gateway** anzuzeigen:

1. Öffnen Sie das **Process Gateway**.
2. Klicken Sie die Schaltfläche **About**.

   Der Dialog **About** wird geöffnet.

DIALOG ABOUT



Option	Beschreibung
Lizenzinformationen	Informationen zu lizenzierten Modulen inklusive Seriennummer.
OK	Schließt den Dialog.
Open license manager	Öffnet die **Lizenzverwaltung**. Mit diesem Tool können Lizenzen aktiviert, zurückgegeben und verwaltet werden. Dazu wird die Seriennummer benötigt.

Wird das **Process Gateway** ohne bestehende Modulkonfiguration geöffnet, muss zuerst ein Modul gewählt werden. Nach Auswahl eines Moduls wird der entsprechende Startdialog für das **Process Gateway** geöffnet. Dabei ist die Schaltfläche **Settings** so lange inaktiv, bis das **Process Gateway** lizenziert ist.

3. Allgemeines

Für das **Process Gateway** gilt:

- Es versteht sich als Add-On zur Runtime und ist nur in Verbindung mit dieser lauffähig.
  Das bedeutet: Ist keine zenon Runtime gestartet, startet das **Process Gateway** nicht!
- Auf einem Rechner können mehrere **Process Gateways** gestartet werden.
  Weitere Informationen dazu finden Sie im Kapitel Start mit Kommandozeile (auf Seite 16).
- Das **Process Gateway** kann sowohl auf einem Einzelplatz-Rechner wie auch im zenon Netzwerk (Server/Standby/Client) gestartet werden. Alle **Process Gateways**, die im Netzwerk gestartet sind, liefern das gleiche Prozessabbild.
- Die Auswahl und Konfiguration des zu startenden Moduls erfolgt in einer eigenen INI-Datei (auf Seite 11) des **Process Gateways**, nicht in der für zenon sonst üblichen zenon6.ini!
  Die INI-Datei für das **Process Gateway** heißt `zenProcGateway.INI` und befindet sich im Ordner `%CD_SYSTEM%`. Bei einer Standardinstallation ist der Ablageordner per default `C:\ProgramData\COPA-DATA\System`.

PROJEKTAUSWAHL

Wird in der Konfigurationsdatei (`zenProcGateway.INI`) kein spezieller Projektname eingestellt, so gelten alle Konfigurationen des **Process Gateway** auf das erste von der Runtime geladene Projekt.

In einem Mehrprojektsystem kann auf ein beliebiges Projekt in der Hierarchie zugegriffen werden. Tragen Sie dazu in der INI-Datei den Projektname im Eintrag **Project**= der Sektion **[GENERAL]** ein. Ist dieser Eintrag leer, gilt das Integrationsprojekt.

3.1 Variablenzuordnung über Namen

Im **Process Gateway** erfolgt die Zuordnung jeweils über den Variablennamen.

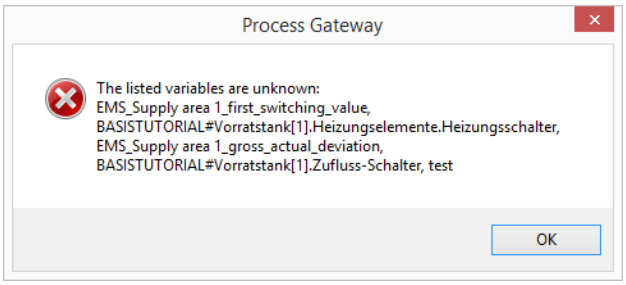
Dies ist besonders wichtig, wenn:

- Variablenbenennungen in der Projektierung im zenon Editor geändert werden.
- Variablen in der Projektierung im Editor gelöscht werden.
- Die zenon Runtime ein neues Startprojekt aufruft.

In diesem Falle erscheinen beim Start des **Process Gateway** modulabhängige Fehlerdialoge.

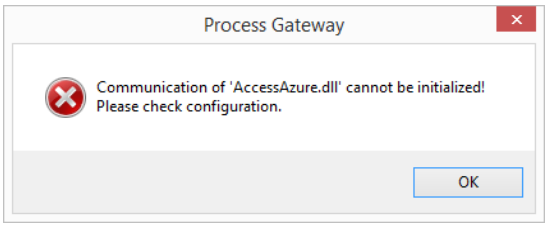
> **Achtung**
>
> *Bei fehlerhafter Variablenzuordnung wird das **Process Gateway** nicht gestartet! Das **Process Gateway** startet erst dann, wenn die fehlerhafte Variablenzuordnung richtig gestellt wurde.*

Der erste Fehlerdialog listet die unbekannten Variablen auf.



- Passen Sie in diesem Falle die Variablenzuordnung neu an.
- Entfernen Sie dazu alle falsch zugeordneten Variablen aus dem jeweiligen Exportbereich und fügen Sie die entsprechenden Variablen neu zu.

Der darauffolgende Dialog beinhaltet modulabhängige Fehlermeldungen.



LOG-EINTRAG

Bei fehlerhafter Variablenzuordnung wird folgender LOG-Eintrag erstellt:

LOG-Eintrag	Beschreibung
`Failed to update value for variable %s.`	Es konnte keine COM-Verbindung zu zenon aufgebaut werden, oder die Variable ist in zenon nicht (mehr) verfügbar oder wurde umbenannt.

4. Voraussetzungen

Auf dem Rechner, auf dem das **Process Gateway** laufen soll, muss vor dessen Start die Runtime mit passender Release-Version und einem der Konfiguration entsprechenden geladenen Projekt gestartet werden. Wird die Runtime beendet, so wird auch das **Process Gateway** automatisch beendet.

Ist die zenon Runtime bei Start des **Process Gateway** nicht geladen, wird ein entsprechender Warndialog angezeigt:

Info

*Das **Process Gateway** benutzt das COM Interface, um auf zenon Daten zuzugreifen. Das COM Interface wird von VBA benutzt und ist nur verfügbar, wenn in der Datei **zenon6.ini** unter **[VBA]** der Eintrag EVENT=1 vorhanden ist.*

5. Start

Das **Process Gateway** besteht aus der Datei **zenProcGateway.exe.** Diese wird bei der Installation in den zenon Programmordner kopiert. Die Datei kann aber auch verwendet werden, wenn Sie in einem anderen Ordner abgelegt wird.
Achtung: Das Modul (Access*.dll), mit dem das **Process Gateway** verwendet werden soll, muss sich ebenfalls im selben Ordner wie die **zenProcGateway.exe** befinden.
Ausnahme: **SNMP** (auf Seite 317).

Die Konfigurationsdatei **zenProcGateway.ini** liegt im Systemordner `%CD_SYSTEM%`.
In dieser INI-Datei wird mit dem Eintrag DLL= (im Abschnitt **[GENERAL]**) das Modul bestimmt, mit dem das **Process Gateway** startet.

Auf einem Rechner können mehrere **Process Gateways** gestartet werden. Die Voraussetzung dafür ist die verwendung unterschiedlichen INI-Dateien. Weitere Informationen dazu finden Sie im Kapitel Start mit Kommandozeile (auf Seite 16).

PROCESS GATEWAY IN ZENON NETZWERK

Das **Process Gateway** kann sowohl auf einem Einzelplatz-Rechner wie auch im zenon Netzwerk (Server/Standby/Client) gestartet werden. Alle **Process Gateways**, die im Netzwerk gestartet sind, liefern das gleiche Prozessabbild.

In redundanten Netzwerk wird empfohlen das **Process Gateway** auf einem (dedizierten) Client zu starten. Bei Redundanzumschaltung verbindet sich ein Netzwerk Client automatisch zu jeweiligen prozessführenden Server und kann weiter die Daten an die übergeordnete Systeme liefern.

PROCESS GATEWAY - START

Das **Process Gateway** kann gestartet werden:

- via Startup-Tool (auf Seite 13)
- via zenon Funktion (auf Seite 14)

In beiden Startmethoden kann das **Process Gateway** mit Parameter in der Kommandozeile (auf Seite 16) gestartet werden. Damit können unterschiedliche INI-Dateien des **Process Gateway** bestimmt werden.

> **Achtung**
>
> Das **Process Gateway** kann nur dann gestartet werden, wenn die zenon Runtime läuft. Ein Start ohne laufender Runtime ist nicht möglich. In diesem Fall wird ein entsprechender Warndialog angezeigt.

KONFIGURATION

Die Konfiguration besteht aus zwei Teilen:

- Modulauswahl (auf Seite 16)
  Im ersten Schritt der Konfiguration wählen Sie das Modul, mit dem das **Process Gateway** kommuniziert.
  Weitere Informationen dazu finden Sie im Kapitel Modulauswahl.
- Modulspezifische Konfiguration
  Abhängig vom ausgewählten Modul können Verbindungsparameter und Variablenzuordnungen konfiguriert werden.
  Die Details dazu entnehmen Sie dem Kapitel Modulkonfiguration im jeweiligen Abschnitt dieses Handbuchs.

Alle Konfigurationen des **Process Gateway** werden in einer .INI-Datei gespeichert. Beachten Sie hierbei, dass nicht alle Einträge der entsprechenden INI-Datei über eine entsprechende Benutzeroberfläche konfiguriert werden können.

Das Modul wird mit dem Eintrag **DLL=** (im Abschnitt **[GENERAL]**) bestimmt.

Mögliche Einträge:

Modul	Eintrag
DEC	DLL=AccessDEC.dll
SQL	DLL=AccessSQL.dll
MODBUS	DLL=AccessMODBUS.dll
DNP3	DLL=AccessDNP3.dll
IEC870SI	DLL=AccessIEC870S1.dll
SNMP	DLL=AccessSNMP.dll
OPCUA	DLL=AccessOPCUA.dll
ICCP/TASE.2	DLL=AccessICCP.dll
MS Azure	DLL=AccessAzure.dll
DNP3_SG	DLL=AccessDNP3_SG.dll

SICHERHEITSKONFIGURATION

Um das **Process Gateway** vor unbefugter Benutzung zu schützen, kann das Programm versteckt ausgeführt werden. Wird es von der Kommandozeile mit dem Parameter `/hide` (oder `-hide`) aufgerufen, startet das **Process Gateway** unsichtbar und kann nicht konfiguriert und beendet werden. Das **Process Gateway** beendet sich automatisch, wenn die zenon Runtime beendet wird.

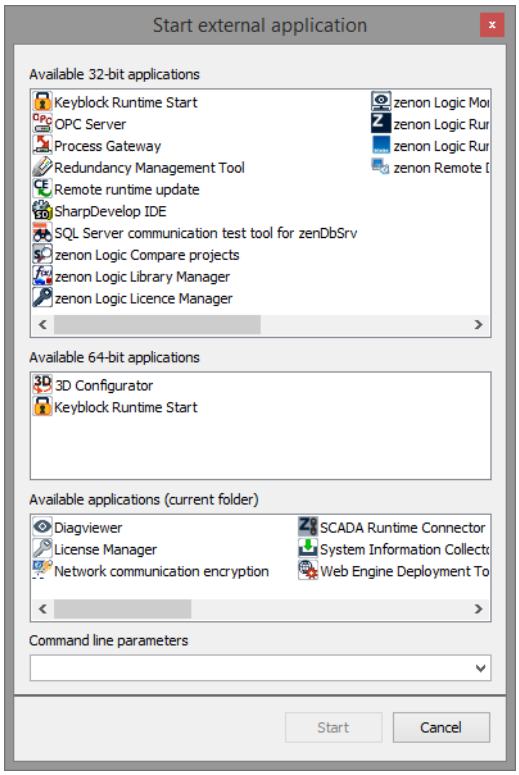
5.1 Start mit Startup Tool

Das **Process Gateway** kann über das zenon **Startup Tool** gestartet werden.

Führen Sie dazu folgende Schritte durch:

1. Öffnen Sie das zenon **Startup Tool**.
2. Klicken Sie auf **Tools**.

Der Dialog zur Konfiguration der zenon Tools wird geöffnet.



3. Wählen Sie **Process Gateway** aus.
4. Klicken Sie den Button Start um das **Process Gateway** mit der Konfiguration aus der aktuellen `zenProcGateway.INI`-Datei zu starten.

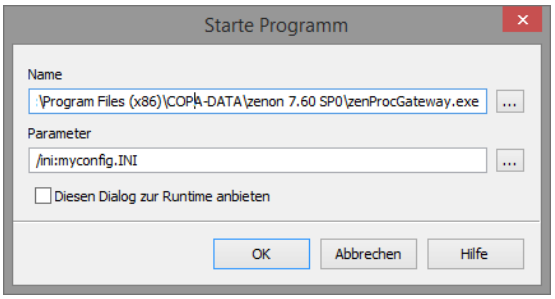
   oder:
   Tragen Sie in das Feld **Command line parameters** den Namen der EXE-Datei und den Namen der INI-Datei als Startparameter ein:
   zenProcGateway.EXE /ini:"MyConfig.INI"

   Weitere Informationen dazu finden Sie im Kapitel Start mit Kommandozeile (auf Seite 16).
   Achtung: Die Parameter alleine genügen nicht. Es muss zwingend der Aufruf **zenProcGateway.EXE** eingetragen werden.
5. Klicken Sie auf **Start**.

5.2 Start mit zenon Funktion

Das **Process Gateway** kann - wie jede andere Windows Anwendung - via der Funktion **Starte Programm** gestartet werden.

PROJEKTIERUNG



Schritte zum Anlegen der Funktion:

1. Erstellen Sie eine neue Funktion:

   Wählen Sie in der Symbolleiste oder im Kontextmenü des Knotens Funktionen **Funktion neu**.
   Der Dialog zur Auswahl einer Funktion wird geöffnet.

2. Navigieren Sie zum Knoten **Windows**.

3. Wählen Sie die Funktion **Starte Programm**.
   Der Dialog zur Auswahl des Programms und Eingabe von Startparameter wird geöffnet.

4. Wählen Sie das gewünschte Programm.

   a) Klicken Sie die Schaltfläche **...**
   Der Dialog zur Auswahl einer Datei wird geöffnet.

   b) Navigieren sie in das zenon Installationsverzeichnis:
   `Program Files (x86)` - 64-Bit Betriebssystem

   `Program Files` - 32-Bit Betriebssystem

   c) Wählen Sie dort die Datei `zenProcGateway.exe`

   d) Bestätigen Sie die Auswahl mit Klick auf die Schaltfläche **Öffnen**.

5. Optional: Konfigurieren Sie die Parameter, die bei Ausführung der Funktion ausgeführt werden. Weitere Informationen dazu finden Sie im Kapitel Start mit Kommandozeile (auf Seite 16).
   Ist dieser Parameter leer, startet das **Process Gateway** mit der `zenProcGateway.INI`-Datei.

6. Schließen Sie den Dialog mit Klick auf die Schaltfläche **OK**.

7. Benennen Sie die Funktion in der Eigenschaft **Name**.

> **Tipp**
>
> *Ist diese Funktion im Skript `AUTOSTART*` verknüpft, startet das **Process Gateway** automatisch mit der Runtime.*

5.3 Start mit Kommandozeile

Auf einem Rechner können mehrere **Process Gateways** gestartet werden. Dazu muss für jedes zu startende **Process Gateway** eine eigene INI-Datei vorhanden sein. Die INI-Datei muss im Systemordner liegen (%CD_SYSTEM%).

Anwendungbeispiele:

- Paralleler Start mehrerer **Process Gateways** mit unterschiedlichen Modulen
- Start mehrere Instanzen des **Process Gateways** mit dem gleichen Modul (gilt nicht für alle Module).
  Voraussetzung: eigene Kommunikationsparameter für jede Instanz, z.B.: keine Konflikte in IP-Ports.

In diesem Fall muss die **zenProcGateway.EXE** mit dem Kommandozeilen-Parameter **/ini:"[file]"** gestartet werden.

Hinweis: der Name der INI-Datei kann auch beim Start mit zenon Funktion (auf Seite 14) konfiguriert werden.

Syntax Kommandozeile:

- im Startup Tool:
  `zenProcGateway.EXE /ini:"[Dateiname].INI"`
- in der zenon Funktion **Starte Programm**
  Eingabefeld **Parameter**: `/ini:"[Dateiname].INI"`

`[Dateiname].ini` muss durch den richtigen Namen der INI-Datei ersetzt werden.

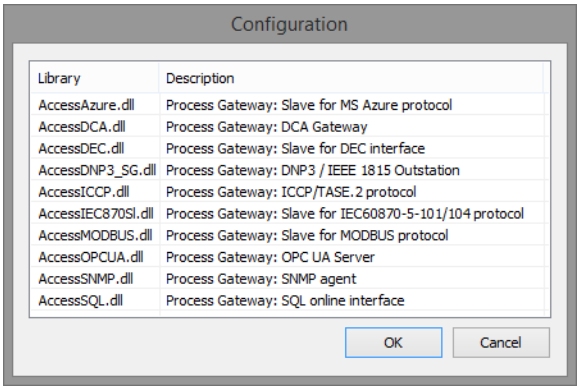
5.4 Modulauswahl

Für die Auswahl eines Moduls für das **Process Gateway** stehen folgende Möglichkeiten zur Verfügung:

- Erstmaliger Start
  Bei erstmaligen Start des **Process Gateway** wird ein Dialog zur Auswahl des Moduls angezeigt. Dieser Auswahldialog wird bei Start des **Process Gateway** auch dann angeboten, wenn keine INI-Datei vorhanden ist oder in der ausgewählten INI-Datei kein Modul konfiguriert ist.
  Per Default wird die Datei `zenProcGateway.INI` verwendet.
- Modul bereits konfiguriert
  Wurde bereits ein Modul für das **Process Gateway** konfiguriert, startet das Process Gateway mit dem Kommunikationsdialog des konfigurierten Moduls.
  Mit dem Start des **Process Gateway** beginnt bereits die Kommunikation, wenn alle notwendigen Verbindungsparameter für das Modul konfiguriert und gültig sind.

5.4.1 Erstmaliger Start

Das Modul muss beim ersten Start konfiguriert werden.



Um das **Process Gateway** mit einem auswählbaren Modul zu starten:

1. Starten Sie das **Process Gateway** mit nicht vorhandener INI-Datei (in der INI-Datei darf noch kein anderes Modul konfiguriert sein).
   Per Default nutzt das **Process Gateway** die Datei `zenProcGateway.INI`. In der Kommandozeile (`/ini:"[Dateiname].INI"`) können Sie eine andere Datei definieren.
   Mehr Informationen finden Sie im Kapitel **Start mit Kommandozeile** (auf Seite 16).
2. Der Dialog zur Modul-Auswahl wird geöffnet.
3. Wählen Sie das gewünschte Modul.
4. Bestätigen Sie Ihre Auswahl mit Klick auf die Schalftäche **OK**.
   Das **Process Gateway** erstellt dabei eine INI-Datei.
5. Das **Process Gateway** startet mit dem Kommunikationsdialog (auf Seite 18) des ausgewählten Moduls.

5.4.2 Bestehende Modulauswahl ändern

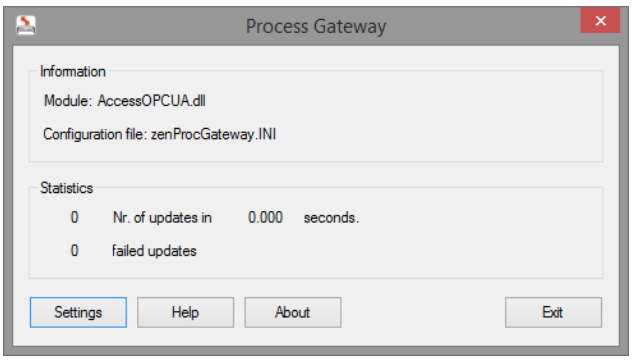
Startet das Process Gateway mit einem bereits konfigurierten Modul für ein anderes Kommunikationsprotokoll, ohne den Dialog zur Auswahl eines Moduls anzubieten, dann:

1. Schließen Sie das **Process Gateway**.
2. Bearbeiten Sie die bestehende zenProcGateway.INI-Datei.
   a) Navigieren Sie dazu in das Verzeichnis %CD_SYSTEM%.
   b) Öffnen Sie die Datei in einem beliebigen Texteditor und kommentieren Sie das Modul in der Datei aus; oder benennen Sie die INI-Datei um.

3. Starten Sie das **Process Gateway** neu:
   Wurde die INI-Datei entfernt, wird bei Start des **Process Gateway** der Dialog zur Modulauswahl (auf Seite 17) erneut angeboten.

5.4.3 Start mit konfiguriertem Modul

Starten Sie das **Process Gateway** mit einem bereits konfigurierten Modul, wird ein Kommunikationsdialog mit statistischen Daten geöffnet.



INFO

Minimale Informationen über das konfigurierte Modul.

Parameter	Beschreibung
Module:	Konfiguriertes und gestartetes Modul des **Process Gateway**.
Config File:	Name der Konfigurationsdatei `zenProcGateway.INI` des **Process Gateways**.

STATISTICS

Informationen zur Kommunikation mit der Gegenstelle.

Parameter	Beschreibung
Nr. of updates in xx seconds	Anzahl der erfolgreichen Kommunikationsversuche inkl. Gesamtlaufzeit.
failed updates	Anzahl der fehlerhaften Kommunikationsversuche.
Exit	Schließt das **Process Gateway**.
Settings	Öffnet Konfigurationsdialog des Moduls.
Help	Öffnet die Online-Hilfe.

Achtung

*Wir die Anwendung als `unsichtbar` gestartet, wird auch kein Konfigurationsdialog angezeigt. Ist die Konfiguration ungültig oder nicht vorhanden, wird das **Process Gateway** automatisch geschlossen.*

5.5 INI-Datei

Das **Process Gateway** speichert in der INI-Datei die Modulauswahl. Diese `zenProcGateway.INI` Datei finden Sie im Ordner `%CD_SYSTEM%`. Die Konfiguration wird pro Modul in einer XML-Datei gespeichert. Diese Konfigurationsdatei wird ebenfalls im Ordner %CD_SYSTEM% gespeichert.

Die modulspezifischen Konfigurationen sind zwischen den Modulen sehr unterschiedliche. Beachten Sie die detaillierte Dokumentation der modulabhängigen Einträgen in diesem Handbuch.

Zum Lesen der INI-Dateien werden systembedingt nur ANSI und Unicode unterstützt.

Achtung

UTF-8 Speicherformat wird nicht unterstützt!

Sichern Sie ihre INI-Dateien deshalb immer als Textdatei im ANSI- oder Unicode-Format.

MODULUNABHÄNGIGE INHALTE

[GENERAL]

Eintrag	Beschreibung
[GENERAL]	Allgemeine Einstellung für das Process Gateway, unabhängig vom gewählten Modul.
DLL=	Auswahl der DLL-Datei, welche für das Process Gateway verwendet wird. Die gewählte DLL-Datei bestimmt die Modulauswahl ▶ **AccessAzure.dll** ▶ **AccessDEC.dll** ▶ **AccessDNP3_SG.dll** ▶ **AccessICCP.dll** ▶ **AccessIEC870SI.dll** ▶ **AccessMODBUS.dll** ▶ **AccessOPCUA.dll** ▶ **AccessSNMP.dll** ▶ **AccessSQL.dll**

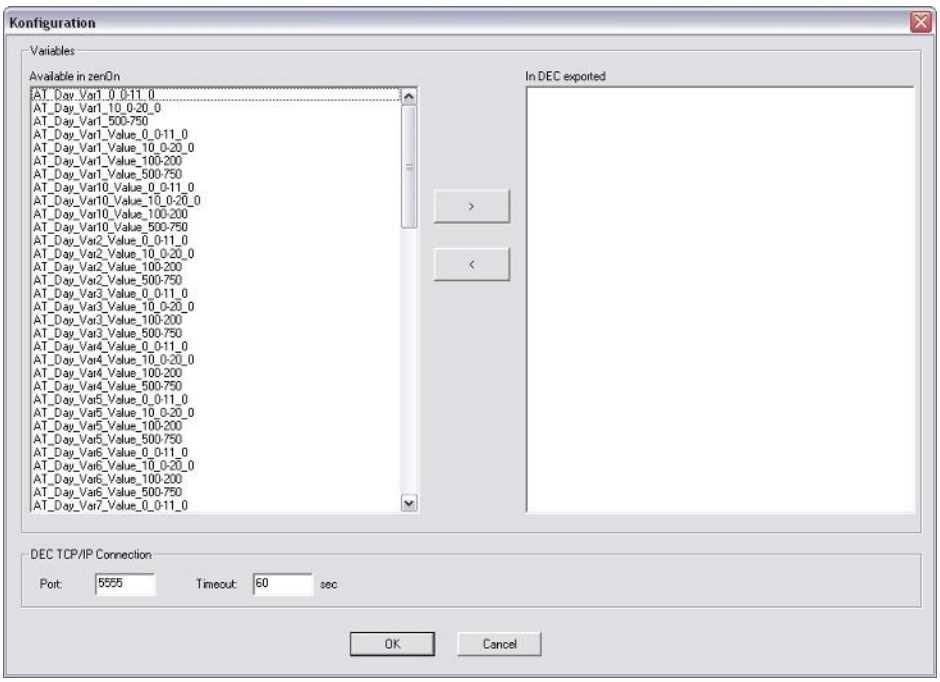
[DATABASE]

Eintrag	Beschreibung
PROJECT=	Name des zenon Projekts, mit dem das **Process Gateway** zur Runtime kommuniziert. In einem Mehrprojektsystem kann auf ein beliebiges Projekt in der Hierarchie zugegriffen werden. Ist dieser Eintrag leer, gilt das Integrationsprojekt. Default: `leer` (in der Runtime aktuell geladenes Projekt)

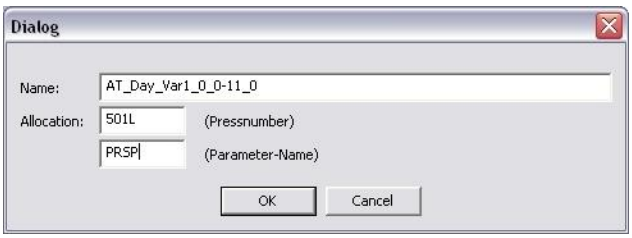
6. DEC

6.1 Modulkonfiguration

Nach Klick auf die Schaltfläche **Settings** erscheint folgender Dialog:



Hier können zum einen die Variablen, die gekoppelt werden sollen, ausgewählt werden. In der linken Liste werden alle in zenon verfügbaren Variablen angezeigt. Diese können ausgewählt werden und mit der '**>**'-Schaltfläche in die In DEC exportiert Liste verschoben werden. Mit der '**<**'-Schaltfläche können sie wieder entfernt werden. Wird eine Variable in die Exportiert-Liste verschoben, erscheint ein Dialog zur Namenszuordnung der zenon Variable zur DEC-Variable:



Die Zuordnung erfolgt über die vierstellige Pressennummer und den vierstelligen Parameter-Name. Eingegebene Zeichen werden automatisch in Großbuchstaben umgewandelt. Die so ausgewählten Variablen werden dann in zenon oder im angebundenen DEC-System laufend synchronisiert.

Im unteren Teil des Konfigurations-Dialogs kann die TCP/IP-Verbindung zum DEC-System konfiguriert werden.

Die Portnummer bezeichnet den Port, auf dem das Process Gateway auf eingehende Verbindungen wartet.

Das Timeout ist die Zeit ohne Antwort, nach der eine Verbindung abgebrochen wird.

6.2 Logging

Alle vom DEC-System empfangenen Telegramme und die Antwort vom Process Gateway können mitprotokolliert werden.

Zum Logging wird der Diagnosis Viewer verwendet, nähre Informationen finden Sie im Kapitel Diagnosis Viewer.

7. DNP3_SG

Das Modul **AccessDNP3_SG** für das **Process Gateway** basiert auf einen von COPA-DATA entwickelten DNP3 Stack.

- Das Modul **AccessDNP3_SG** ist konform mit Subset Level 1, 2 und 3 und unterstützt Dateitransfer.
- Die Konfiguration ist nicht kompatibel zum bestehenden **AccessDNP3 Process Gateway** von zenon Versionen vor 8.00.
- Das bisherige Modul **AccessDNP3** im **Process Gateway** steht für neue Konfigurationen nicht mehr zur Auswahl.
- Bestehende Konfigurationen mit dem bisherigen **AccessDNP3** Modul sind weiterhin lauffähig.
- In eine zukünftige Version wird das **AccessDNP3**-Modul nicht mehr im Lieferumfang enthalten sein. Das alte Modul wird vollständig durch das Modul **AccessDNP3_SG** ersetzt werden.

Im Lieferumfang für das neue Modul ist ein XML-Device Profil enthalten. Dieses Dokument enthält auch eine Beschreibung der Fähigkeiten des **AccessDNP3_SG** Moduls.

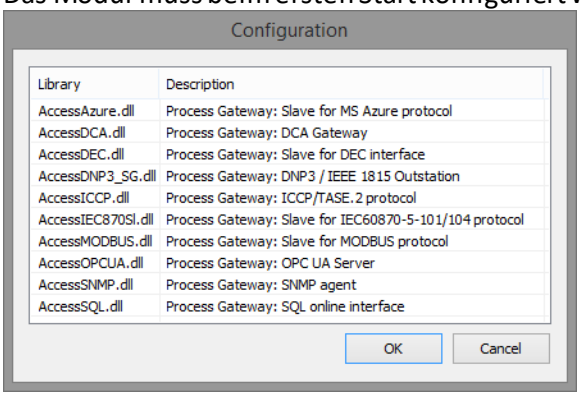
Installationspfad:

`C:\ProgramData\COPA-DATA\zenon8.00\CommunicationProfiles\Dnp3\ProcessGateway`

7.1 Installation

Um das DNP3_SG Process Gateway zu installieren haben sie folgende Möglichkeiten:

1. Erstmaliger Start des Process Gateway. Es wurde noch kein Modul konfiguriert:
   Das Modul muss beim ersten Start konfiguriert werden.



   a) Starten Sie das **Process Gateway** mit einer leeren oder nicht vorhandener INI-Datei (in der INI-Datei darf noch kein anderes Modul konfiguriert sein).
      Per Default nutzt das **Process Gateway** die Datei `zenProcGateway.INI`. In der Kommandozeile (`/ini:"[Dateiname].INI"`) können Sie eine andere Datei definieren.
      Mehr Informationen finden Sie im Kapitel **Start mit Kommandozeile** (auf Seite 16).

   b) Der Dialog **Configuration** zur Modul-Auswahl wird geöffnet.

   c) Wählen Sie **AccessDNP3_SG.dll** als Modul aus.

   d) Bestätigen Sie Ihre Auswahl mit Klick auf die Schalftäche **OK**.
      Das **Process Gateway** erstellt dabei eine INI-Datei.

   e) Das **Process Gateway** startet mit dem Kommunikationsdialog des Process Gateway.

   f) Klicken Sie die Schaltfläche **Settings** um das **DNP3_SG** Modul zu konfigurieren.

2. Start bei bereits konfiguriertem **Process Gateway** Modul.

   Startet das **Process Gateway** mit einem bereits konfigurierten Modul für ein anderes Kommunikationsprotokoll, ohne den Dialog zur Auswahl eines Moduls anzubieten, dann führen Sie folgende Schritte aus:

   a) Schließen Sie das **Process Gateway**.

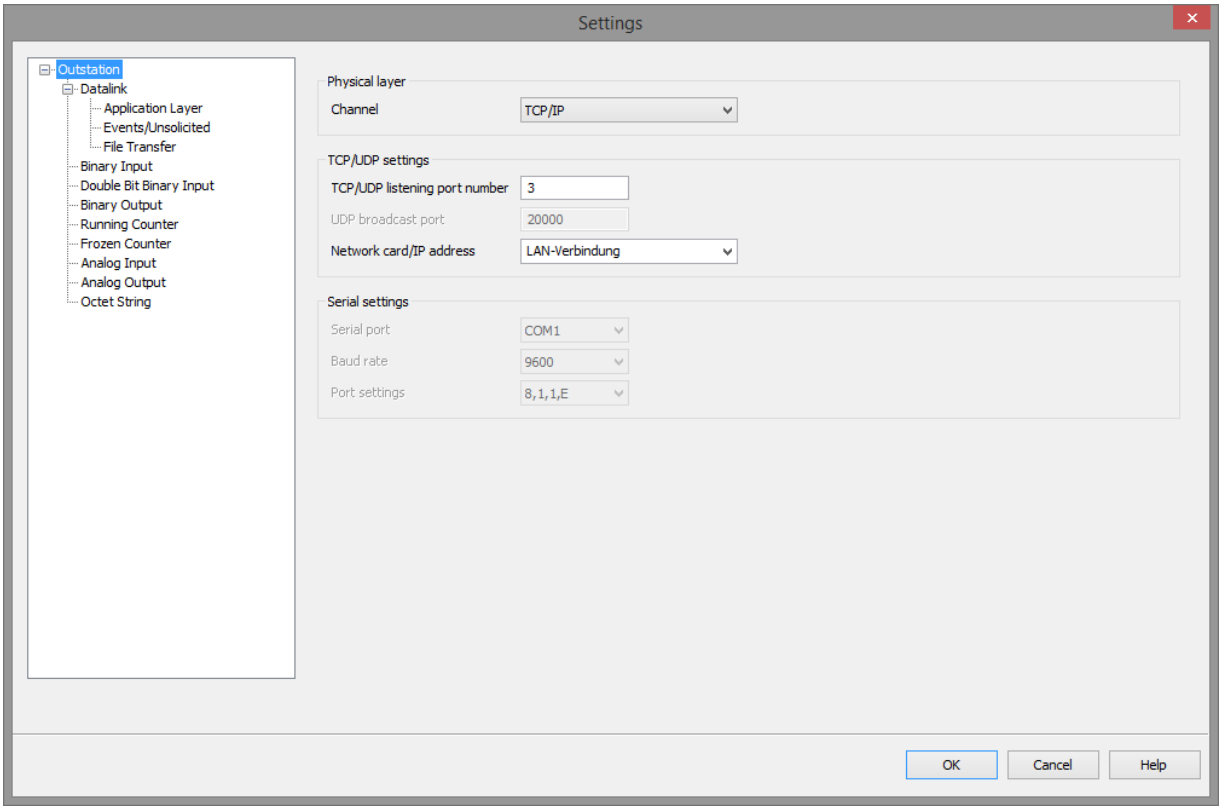
   b) Bearbeiten Sie die bestehende zenProcGateway.INI-Datei.

   c) Navigieren Sie dazu in das Verzeichnis %CD_SYSTEM%.

   d) Öffnen Sie die Datei in einem beliebigen Texteditor und kommentieren Sie das Modul in der Datei aus; oder benennen Sie die INI-Datei um.

   e) Starten Sie das **Process Gateway** neu:
      Wurde die INI-Datei entfernt, wird bei Start des **Process Gateway** der Dialog zur Modulauswahl (auf Seite 17) erneut angeboten.

7.2 Outstation



In diesem Dialog konfigurieren Sie die Verbindungsparameter zur Outstation.

PHYSICAL LAYER

Parameter	Beschreibung
Channel	Art der Verbindung zur Outstation. Auswahl aus Dropdownliste: ▶ `Serial` Serielle Verbindung mit der Outstation. ▶ `TCP/IP` Verbindung mit der Outstation via TCP/IP Protokoll. ▶ `UDP` Verbindung mit der Outstation via UDP Protokoll. ▶ `TCP with UDP broadcast` Verbindung mit der Outstation via TCP/IP Protokoll mit UDP broadcast. Default: `TCP/IP`

TCP/UDP SETTINGS

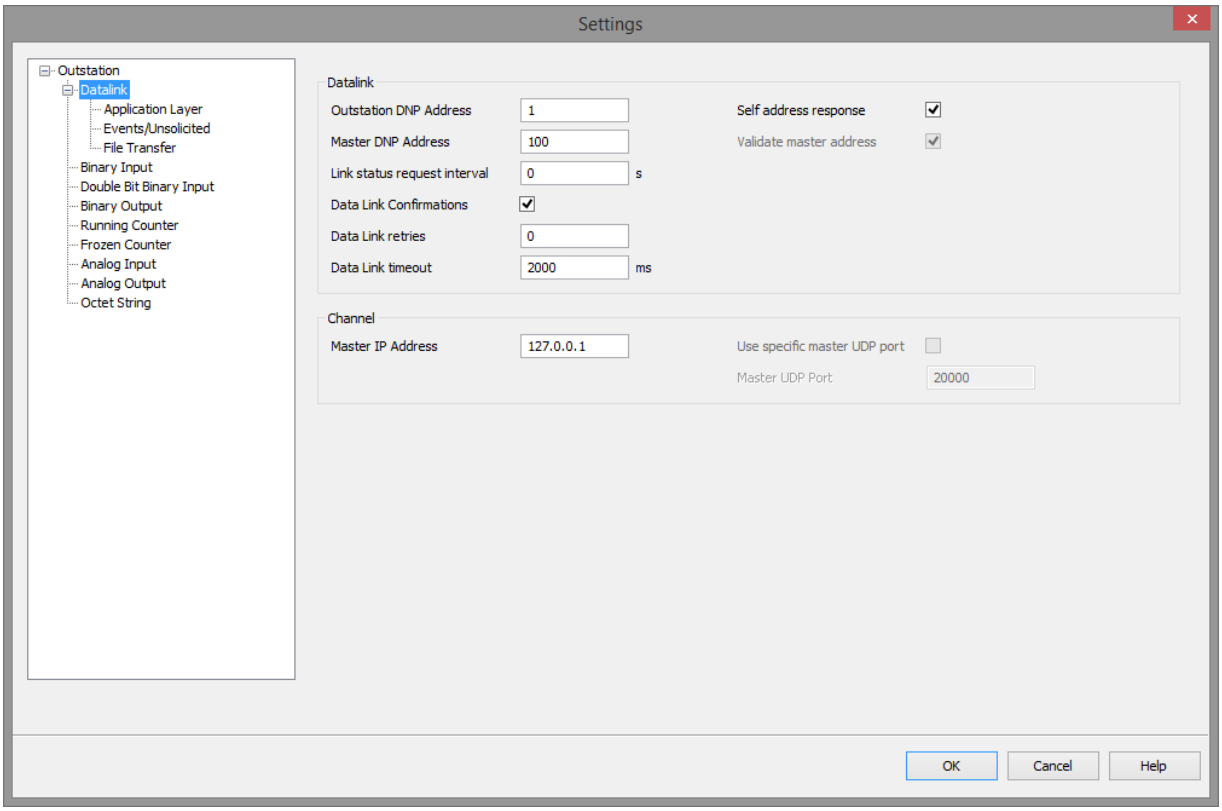
Parameter	Beschreibung
TCP/UDP listening port number	Portnummer für die TCP oder UDP Kommunikation mit der Outstation. Default: `20000`
UDP Broadcast port	Portnummer für den Empfang von UDP broadcasts. Default: `20000` Hinweis: nur aktiv, wenn bei der Option **Channel** der Wert `TCP with UDP broadcast` konfiguriert ist.
Network card/IP address	Netzwerk-Schnittstelle für die Kommunikation mit der Outstation. Auswahl aus Dropdownliste. Die Outstation empfängt Netzwerktelegramme nur auf der konfigurierten Schnittstelle. Ist keine Schnittstelle ausgewählt empfängt die Outstation die Telegramme auf allen Schnittstellen. Default: `leer` Hinweis: Nicht aktiv, wenn bei der Option **Channel** `Serial` konfiguriert ist.

SERIAL SETTINGS

Die Optionen dieser Gruppe sind nur dann aktiv, wenn bei der Option **Channel** die Verbindungsart `Serial` konfiguriert ist.

Parameter	Beschreibung
Serial port	Auswahl der seriellen Schnittstelle. Auswahl aus Dropdownliste. Default: `COM1`
Baud rate	Auswahl der Baudrate für die serielle Kommunikation. Auswahl aus Dropdownliste. Default: `9600`
Port settings	Anzahl der Daten-, Start- Stopp-Bits und der Parität für die serielle Kommunikation. Auswahl aus Dropdownliste: ▶ `8,1,1,N` ▶ `8,1,1,E` ▶ `8,1,2,E` ▶ `8,1,2,N` Default: `8,1,1,E`

7.3 Datalink


Settings
Outstation
Datalink
Application Layer
Events/Unsolicited
File Transfer
Binary Input
Double Bit Binary Input
Binary Output
Running Counter
Frozen Counter
Analog Input
Analog Output
Octet String
Datalink
Outstation DNP Address
1
Master DNP Address
100
Link status request interval
0
s
Data Link Confirmations
Data Link retries
0
Data Link timeout
2000
ms
Self address response
Validate master address
Channel
Master IP Address
127.0.0.1
Use specific master UDP port
Master UDP Port
20000
OK
Cancel
Help

Parameter	Beschreibung
Outstation DNP Address	DNP3-Adresse der Outstation. Default: `1`
Self address response	Antwortverhalten der Outstation auf ein `Request`. ▶ `aktiv:` die Outstation antwortet mit der eigenen DNP3 Adresse, wenn ein `Master Request` mit der der reservierten Adresse "0xFFFC" empfangen wird. ▶ `inaktiv:` Die Outstation antwortet nur auf `Requests` von der konfigurierten Master-Adresse. Default: `inaktiv` Hinweis: Es wird empfohlen diese Option nur vorübergehend zu aktivieren, um einen Master die Erkennung der Outstation zu ermöglichen.
Master DNP Address	Die DNP3 Adresse des Masters, der mit der Outstation kommunizieren soll. Default: `100` Hinweis: derzeit unterstützt das **Process Gateway** nur einen Master pro Instanz.
Validate master address	Validierung der Master-Adresse Hinweis: Diese Option ist immer aktiv. Die Outstation antwortet nur auf `Master requests`, die von der konfigurierte **Master DNP Address** stammen. Eine Ausnahme sind die Broadcast Addressen, wenn im Knoten **Outstation** in der Option **Channel** `TCP with UDP Broadcast` konfiguriert ist und im Knoten **Application Layer** die Option `Broadcasts allowed` aktiviert ist. Default: `aktiv`
Link status request interval	Intervall in Sekunden in der ein Datalink Layer `REQUEST_LINK_STATUS` von der Outstation an den Master gesendet wird, sofern kein weiterer Datenverkehr stattfindet. Dies dient bei Konfigurationen, bei denen keine zyklische Kommunikation mit dem Master stattfindet, zur Überwachung ob der Master noch verfügbar ist. Default: `0` Bei dem Wert 0 sendet die Outstation keinen Link Status

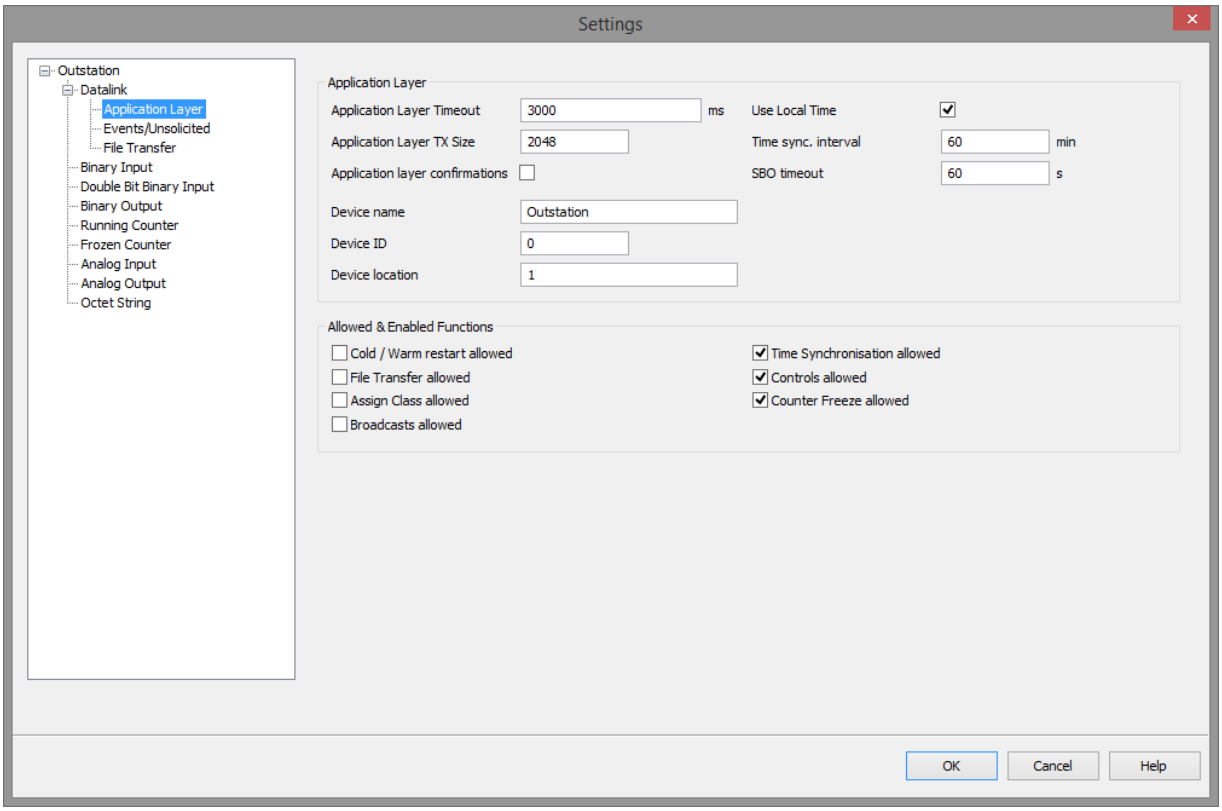
	Requests. Hinweis: Für TCP Verbindungen gibt es ein sogenanntes **TCP Keep Alive**. Dieses darf nicht mit dem `Link status Requests` verwechselt werden.
Data Link Confirmations	▶ `aktiv:` Die Outstation fordert für jedes Datalink Frame eine `Datalink Layer` Bestätigung des Masters an. ▶ `inaktiv:` Die Outstation fordert keine Bestätigung für `Datalink Layer Frames` an. Default: `inaktiv` Hinweis: Diese Option soll nur dann aktiviert werden, wenn es eine geringe Qualität der Verbindung voraussetzt. Für TCP/IP Verbindungen wird dringend empfohlen, diese Option nicht zu aktivieren.
Data Link retries	Anzahl der Wiederholungen für ein `Datalink Layer Frame` wenn eine Bestätigung vom Master angefordert wurde und diese Bestätigung nicht innerhalb der bei der Option **Datalink Layer Timeout** konfigurierten Zeit empfangen wurde. Default: `0`
Data Link Timeout	Timeout für den Empfang einer `Datalink Layer` Bestätigung des Masters auf ein von der Outstation angefordertes `Datalink Layer Request` für die eine `Datalink Layer Confirmation` angefordert wurde. Default: `2000`

CHANNEL

Parameter	Beschreibung
Master IP Address	IP Addresse des entfernten DNP3 Masters. Es werden nur Netzwerktelegramme von der hier konfigurierten IP Addresse akzeptiert. Ausnahme: `Requests` an der UDP Broadcast-Adresse wenn im Knoten **Outstation** die Option **Channel** mit `TCP/IP with UDP Broadcasts` konfiguriert ist und im Knoten **Application Layer** die Option **Broadcasts allowed** aktiv ist.

	Default: `127.0.0.1` Hinweis: Nicht aktiv, wenn bei der Option **Channel** `Serial` konfiguriert ist.
Use specific master UDP port	▶ `aktiv:` Die Outstation antwortet an den hier konfigurierten UDP-Port. ▶ `inaktiv:` Die Outstation antwortet an den UDP Port, von dem der `Master Request` gesendet wurde. Nur verfügbar wenn im Knoten **Outstation** die Option **Channel** mit `UDP` konfiguriert ist.
Master UDP Port	Dieser Port wird für die optionale `unsolicited null response` Nachricht verwendet. Alternative UDP Portnummer für die Antwort an den Master, wenn die Option **Use specific master UDP port** aktiviert ist. Default: `20000` Hinweis: Nur verfügbar wenn die im Knoten **Outstation** die Option **Channel** mit `UDP` konfiguriert ist.

7.3.1 Application Layer



Parameter	Beschreibung
Application Layer Timeout	Zeit die die Outstation auf eine Antwort vom Master wartet, wenn `Responses` eine Bestätigung verlangen. Default: `3000 ms`
Use Local Time	Verwendetes Zeitformat. Diese Option gilt sowohl für die Zeitsynchronisierung der Outstation als auch für die `Event Objects`, die einen Zeitstempel enthalten. ▶ `aktiv:` Das Zeitformat verwendet die Lokalzeit/Zeitzone des Rechners. Diese Option kann gesetzt werden wenn der Master die Uhrzeit ebenfalls als Lokalzeit betrachtet. ▶ `inaktiv:` Das Zeitformat ist UTC, entsprechend dem IEEE Std 1815tm Standard. Default: `aktiv`
Application Layer TX Size	Maximale Grösse eines `Application Layer response`. Der Wert kann entsprechend der Fähigkeiten des Masters und der Bandbreite der Verbindung angepasst werden Default: `2048`
Time sync. interval	Zeitraum, nachdem die Outstation erneut das Internal Indication Bit IIN1.4 (NEED_TIME) im `Response` setzt. Der Wert kann auf 0 gesetzt werden, wenn keine Zeitsynchronisierung durch den Master erforderlich ist. Beispiel: Der Rechner der Outstation synchronisiert seine Uhrzeit bereits von einer anderen Quelle. Default: `60 min`
Application layer confirmations	**Application Confirmation** für `Response` ▶ `aktiv:` Die Outstation fordert für jedes `Response` eine **Application Confirmation** von dem Master an. ▶ `inaktiv:` Die Outstation fordert nur für folgende `Responses` eine **Application Confirmation**

	vom Master an: - Response enthält events - unsolicited Responses - multi-fragment responses (Vom ersten bis zum vorletzten Fragment. Das letzte Fragment verlangt keine Confirmation) Default: `inaktiv` Hinweis: Aktivieren Sie diese Option nur in Ausnahmefälle, z.B. wenn die Qualität und Bandbreite der Verbindung dies erfordert (geringe Bandbreite).
SBO timeout	Zeit die auf ein **Operate Request** vom Master gewartet wird, nachdem ein **Select Request** vom Master empfangen wurde. Default: `60 s` Weitere Informationen zum **SBO** finden Sie im Kapitel **Select before Operate** (auf Seite 118).
Device name	Bezeichnung für die Outstation. Diese Option kann vom Master über das **Device Attribute** `User-assigned device name (g0v247)` gelesen und geschrieben werden. Default: `Outstation`
Device ID	Identifikation für die Outstation. Diese Option kann vom Master über das **Device Attribute** `User-assigned ID code/number (g0v246)` gelesen und geschrieben werden. Default: `0`
Device location	Standortbezeichnung für die Outstation. Diese Option kann von dem Master über das **Device Attribute** `User-assigned location name (g0v245)` gelesen und geschrieben werden. Default: `1`

ALLOWED & ENABLED FUNCTIONS

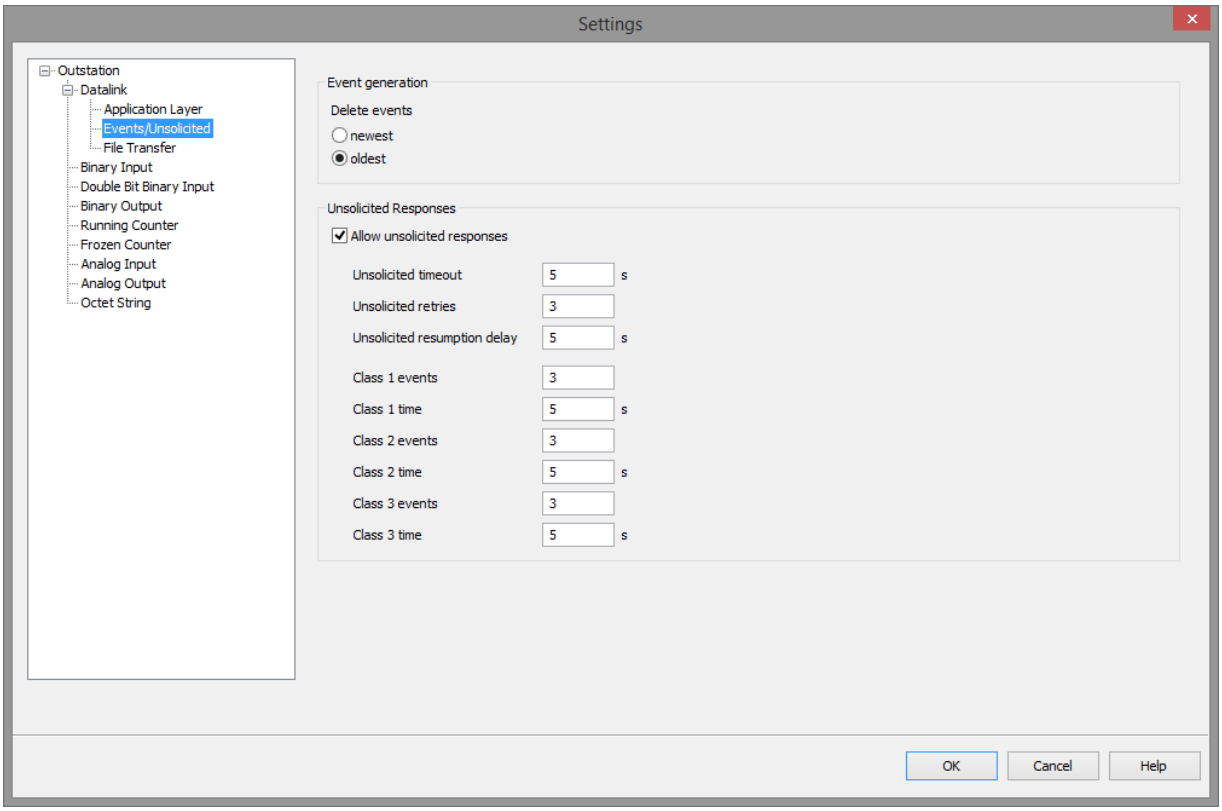
Funktionalitäten, die von der Outstation für die Verbindung zwischen Master und Outstation angeboten wird.

Ist eine Option nicht aktiv, antwortet die Outstation an den Master mit einem **Null Response** mit dem `Internal Indication Bit IIN2.0` (NO_FUNC_CODE_SUPPORT).

Ausnahme: auf `Broadcasts requests` wird generell nicht geantwortet.

- **Cold/Warm restart allowed**
  Default: `inaktiv`
- **File Transfer allowed**
  Default: `inaktiv`
- **Assign Class allowed**
  Default: `inaktiv`
- **Broadcasts allowed**
  Default: `inaktiv`
- **Time Syncrhonisation allowed**
  Default: `aktiv`
- **Controls allowed**
  Default: `aktiv`
- **Counter Freeze allowed**
  Default: `aktiv`

7.3.2 Events/Unsolicited



EVENT GENERATION

Parameter	Beschreibung
Delete events	Welche Events werden bei einem Überlauf des Eventpuffers der Outstation gelöscht: ▶ `newest:` neueste (=jüngste) Events. ▶ `oldest:` älteste Events Default: `newest`

UNSOLICITED RESPONSES

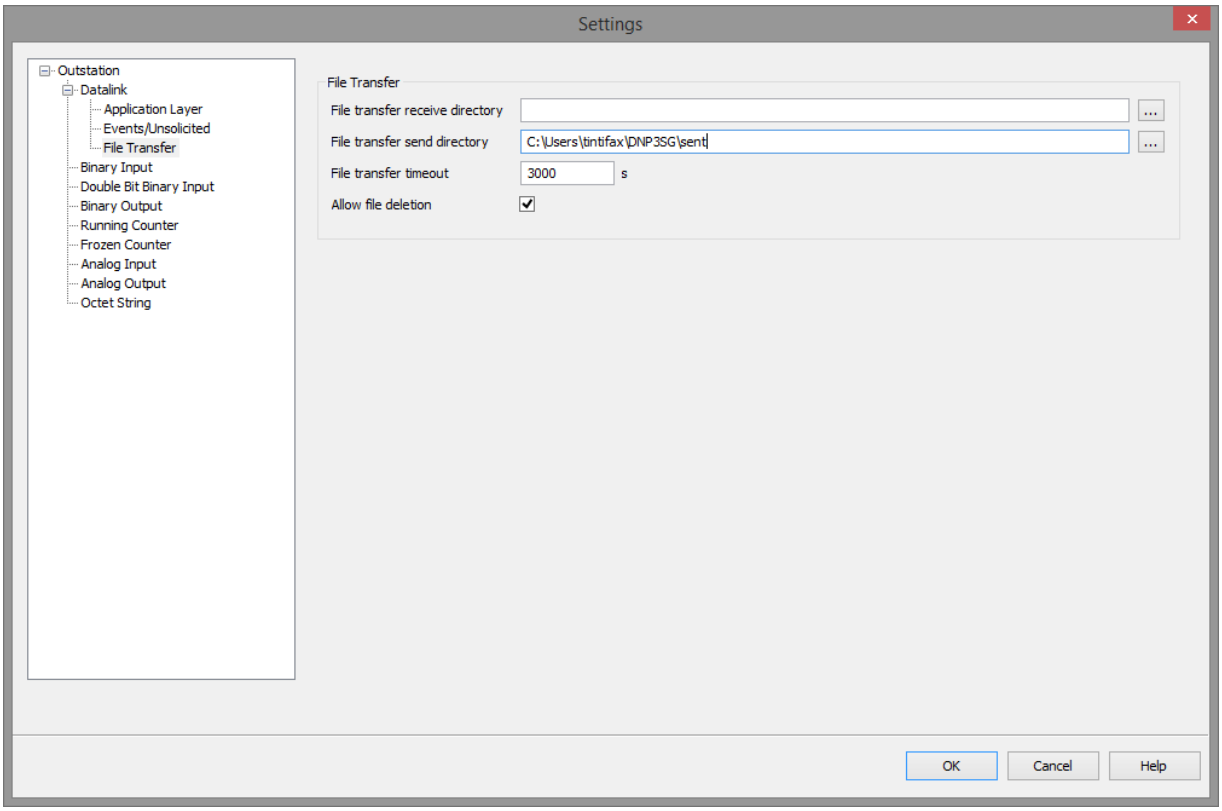
Die Optionen dieses Bereichs sind nur dann verfügbar, wenn die Option **Allow unsolicited responses** aktiviert ist.

Parameter	Beschreibung
Allow unsolicited responses	Checkbox zur Aktivierung der Verwendung der **Unsollicited responses**. ▶ `aktiv:` **Unsolicited responses** sind erlaubt und können vom Master aktiviert werden. ▶ `inaktiv:` **Unsolicited responses** sind nicht möglich und können vom Master nicht aktiviert werden. Achtung: Es werden keine **unsolicited responses** an den Master gesendet. Auch dann nicht, wenn diese vom Master aktiviert wurden. Default: `inaktiv` Hinweis: Stimmen Sie die Konfiguration dieser Option mit der Konfiguration des Masters ab.

Unsolicited timeout	Zeit die auf eine `Application Confirmation` vom Master für ein vorangegangenes `unsolicited Response` von der Outstation gewartet wird. Eingabebereich: `1 bis 60 s` Default: `5 s`
Unsolicited retries	Anzahl der Wiederholungen für ein `Unsolicited Response` (gesendet von der Outstation), nachdem die **Unsolicited timeout** Zeit abgelaufen ist, ohne dass eine `Application Confirmation` vom Master empfangen wurde Eingabebereich: `0 – 65535` (`0` = unendlich viele Versuche) Default: `3` Beispiel: Wird ein `Unsolicited Response` an den Master gesendet, soll der Master ein `Application Confirmation` senden. Bleibt diese Bestätigung durch den Master aus, wird so oft eine Wiederholung gesendet, wie in **Number of retries** projektiert. Anschließend wird wieder ein neues `Unsolicited Response` erzeugt.
Unsolicited resumption delay	Anzahl der Sekunden die die Outstation wartet, bis erneut mit dem Senden von `Unsolicited Responses` fortgefahren wird, wenn nach Ablauf der **Unsolicited timeout** Zeit und dem Versenden der Wiederholungen, keine `Application Confirmation` vom Master empfangen wurde. Default: `15 s`
Class 1 events	Maximale Anzahl der Events für die Eventklasse welche gepuffert werden, bevor ein `Unsolicited Response` an den Master gesendet wird. Eingabebereich: `0 – 255` Default: `3`
Class 1 time	Maximale Verzögerung der Eventklasse welche gewartet wird, bis nach dem Auftreten eines Events ein `Unsolicited Response` an den Master gesendet wird. Eingabebereich: `0 – 65535 Sekunden` (`0` = keine Verzögerung, d.h. bei jedem Event) Default: `5` Hinweis: Werden kontinuierlich neue Events erstellt, kann es dazu kommen, dass keine `Unsolicited`

	Response an den Master gesendet wird, wenn der Wert grösser 0 ist.
Class 2 events	Maximale Anzahl der Events für die Eventklasse welche gepuffert werden, bevor ein Unsolicited Response an den Master gesendet wird. Eingabebereich: 0 – 255 Default: 3
Class 2 time	Maximale Verzögerung der Eventklasse welche gewartet wird, bis nach dem Auftreten eines Events ein Unsolicited Response an den Master gesendet wird. Eingabebereich: 0 – 65535 Sekunden (0 = keine Verzögerung, d.h. bei jedem Event) Default: 5 Hinweis: Werden kontinuierlich neue Events erstellt, kann es dazu kommen, dass keine Unsolicited Response an den Master gesendet wird, wenn der Wert grösser 0 ist.
Class 3 events	Maximale Anzahl der Events für die Eventklasse welche gepuffert werden, bevor ein Unsolicited Response an den Master gesendet wird. Eingabebereich: 0 – 255 Default: 3
Class 3 time	Maximale Verzögerung der Eventklasse welche gewartet wird, bis nach dem Auftreten eines Events ein Unsolicited Response an den Master gesendet wird. Eingabebereich: 0 – 65535 Sekunden (0 = keine Verzögerung, d.h. bei jedem Event) Default: 5 Hinweis: Werden kontinuierlich neue Events erstellt, kann es dazu kommen, dass keine Unsolicited Response an den Master gesendet wird, wenn der Wert grösser 0 ist.

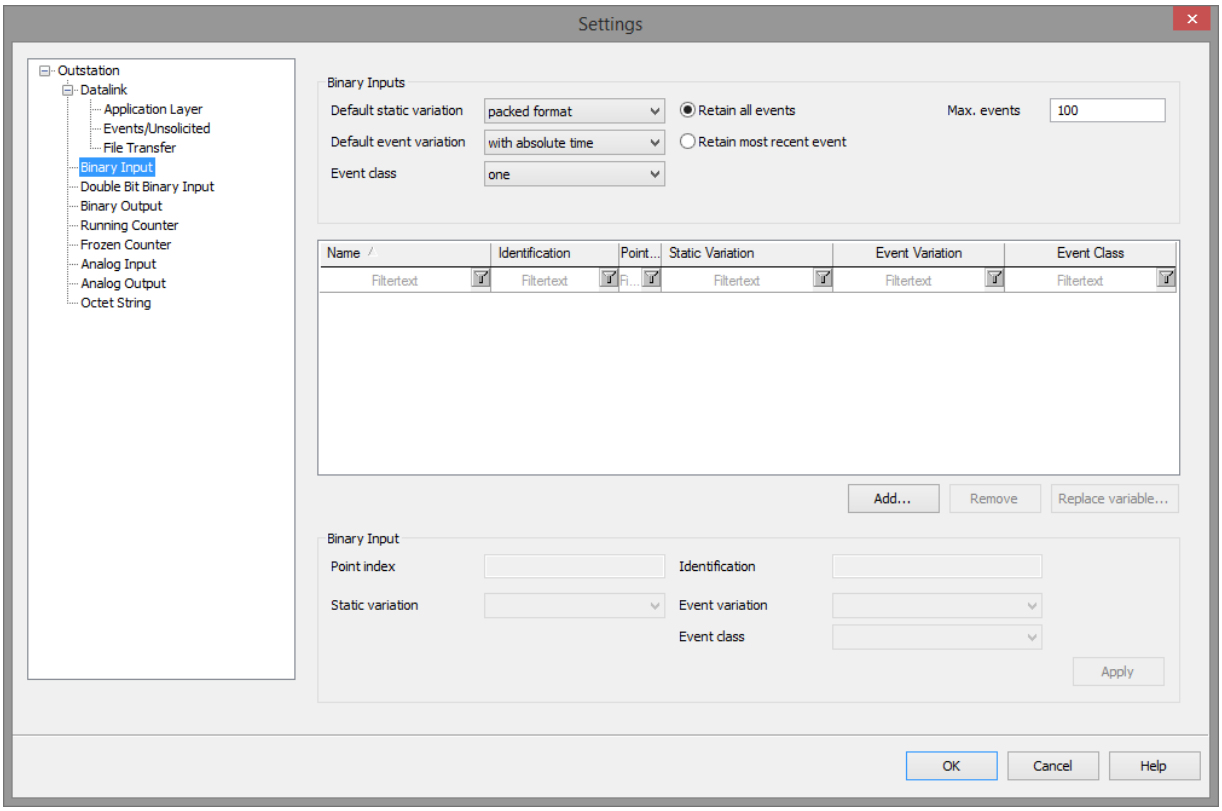
7.3.3 File Transfer



In diesem Dialog konfigurieren Sie die Parameter für den Dateitransfer von und zur Outstation. Ein Master kann auf Unterverzeichnise im Ablageort zugreifen jedoch nicht auf Überliegenden Verzeichnisse (\..\..\)

Parameter	Beschreibung
File transfer receive directory	Ablageort, für die vom Master an die Outstation gesendeten Dateien. Klick auf **...** öffnet den Dialog zur Auswahl eines Ablageortes. Hinweis: Wird der Ablageort unterschiedlich zur Option **File Transfer send directory** konfiguriert, kann der Master jene Dateien, die er selber schreibt weder lesen noch löschen.
File transfer send directory	Ablageort für Dateien, die vom Master gelesen werden können. Klick auf **...** öffnet den Dialog zur Auswahl eines Ablageortes.
File transfer timeout	Zeit in Sekunden bis die Outstation das `File Handle` schließt, sofern dieser `File Handle` nicht bereits vom Master geschlossen wurde. Default: `3000 s` Hinweis: Die Outstation unterstützt nur ein einziges File Handle.
Allow file deletion	Löschung durch den Master ▶ `aktiv:` Dateien dürfen durch den Master gelöscht werden. ▶ `inaktiv:` Der Master darf keine Dateien löschen. Default: `aktiv`

7.4 Binary Input



In diesem Knoten konfigurieren Sie die **DNP3 Binary Input points**.

BINARY INPUTS

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 1 - Variation 1 ▸ `with flags` Object Group 1 - Variation 2 Default: `packed format`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain all events`

Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variation	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `without time` Object Group 2 - Variation 1 ▸ `with absolute time` Object Group 2 - Variation 2 ▸ `with relative time` Object Group 2 - Variation 3 Default: `with absolute time`
Default event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet,

	erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ one Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ two Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ three Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden

	keine events mehr erzeugt. Default: `one`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
Name	Variablenname. Dem Namen wird der Projektname vorangestellt: Projektname#Variablenname
Identification	Kennung der Variable.
Point Index	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
Static Variation	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
Event Variation	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
Event Class	Aktuelle Konfiguration der event class für den point in dieser Object Group.
Control Model	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
Command Routing	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: BOOL, USINT

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

BINARY INPUT

In diesem Bereich konfigurieren Sie die Parameter für eine ausgewählte Variable der Liste. Ein Bearbeiten über Mehrfachauswahl ist nicht möglich.

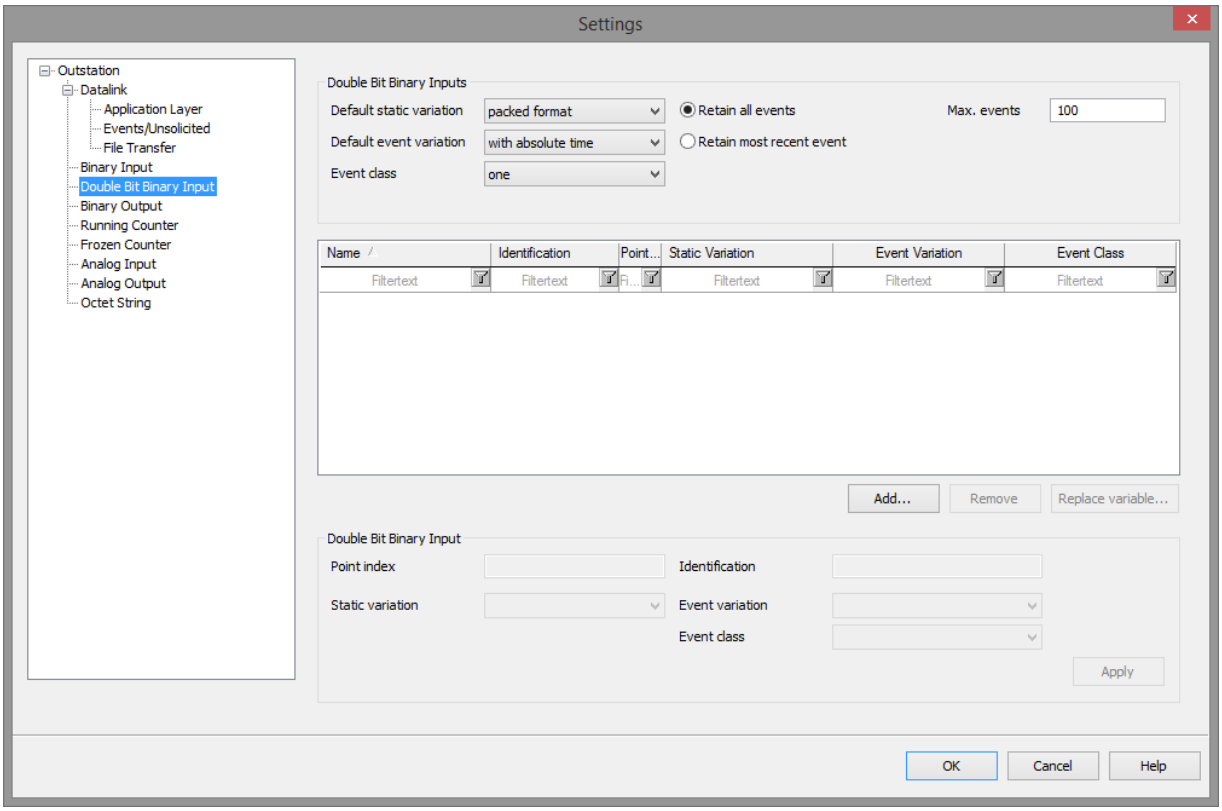
Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 1 - Variation 1 ▸ `with flags` Object Group 1 - Variation 2 Default: `default`

Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `without time` Object Group 2 - Variation 1 ▸ `with absolute time` Object Group 2 - Variation 2 ▸ `with relative time` Object Group 2 - Variation 3 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.5 Double Bit Binary Input



In diesem Knoten konfigurieren Sie die **DNP3 Double bit binary Input points**.

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 3 - Variation 1 ▸ `with flags` Object Group 3 - Variation 2 Default: `packed format`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain all events`
Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variation	Diese Option bestimmt die default event variation

	für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▶ `without time` Object Group 4 - Variation 1 ▶ `with absolute time` Object Group 4 - Variation 2 ▶ `with relative time` Object Group 4 - Variation 3 Default: `with absolute time`

Event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ `one` Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `two` Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen

	über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ Default: `one`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
`Name`	Variablenname. Dem Namen wird der Projektname vorangestellt: `Projektname#Variablenname`
`Identification`	Kennung der Variable.
`Point Index`	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
`Static Variation`	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
`Event Variation`	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
`Event Class`	Aktuelle Konfiguration der event class für den point in dieser Object Group.
`Control Model`	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
`Command Routing`	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: `BOOL, USINT`

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

DOUBLE BIT BINARY INPUT

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

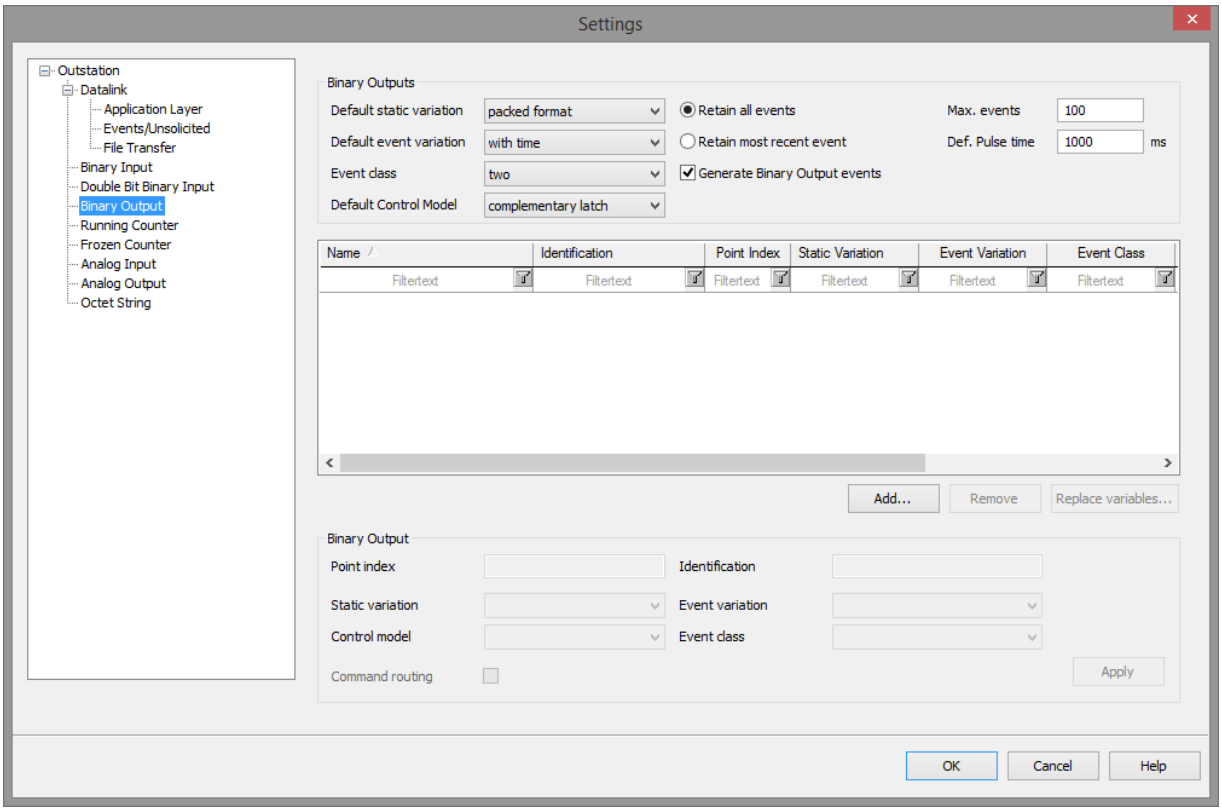
Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 3 - Variation 1 ▸ `with flags` Object Group 3 - Variation 2 Default: `default`

Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `without time` Object Group 4 - Variation 1 ▸ `with absolute time` Object Group 4 - Variation 2 ▸ with relative time Object Group 4 - Variation 3 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.6 Binary Output



In diesem Knoten konfigurieren Sie die **DNP3 Binary Output points**.

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 10 - Variation 1 ▸ `with flags` Object Group 10 - Variation 2 Default: `packed format`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain all events`

Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variation	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▶ `without time` Object Group 11 - Variation 1 ▶ `with time` Object Group 11 - Variation 2 Default: `with time`
Event Class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für

	die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ `one` Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `two` Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `two`

Default Pulse time	Diese Option bestimmt die **Default Pulse ON time**. Die bei dem Binary Output point verknüpfte Variable wird für die Dauer der hier definierte Zeit, auf dem Wert "1", "3", "4", "65" oder "129" gesetzt (abhängig vom Datentyp der Variable für den Binary Output) Im Anschluss wird die Variable wieder auf den Wert 0 gesetzt, wenn der Master ein Pulse ON control command mit dem Pulse On Wert `"0" (default)` sendet. Dies gilt für **Control Model** mit der Konfiguration `"Activation"` oder `"complementary two output"`. Sendet der Master mit dem Pulse On control command einen anderen Wert für Pulse On als `"0" (default)` wird die Zeit aus dem Request vom Master verwendet sofern dieser Zeit nicht grösser oder kleiner als den fünffachen Wert der **Default Pulse time** ist. Ist der Wert ausserhalb, verwendet die Outstation die hier konfigurierte **Default Pulse time**. Hinweis: Für Control Model complementary latch findet die **Default pulse time** keine Verwendung. Default: `1000 ms`
Generate Binary Output events	Diese Option bestimmt, ob bei Wertänderungen für Variablen, die für Binary Outputs konfiguriert sind, auch events erzeugt werden sollen. Nicht alle master unterstützen events für Binary Outputs. Default: `aktiv`
Default Control Model	Diese Option bestimmt das default **Control Model** für neu hinzugefügte Variablen. Hinweis: Bestimmte Kombinationen aus Master control requests und Datentyp (BOOL oder USINT) der Variable sind nicht möglich. Auch in Abhängigkeit davon, ob **Command Routing** aktiv oder nicht aktiv ist für eine ausgewählte Variable. Auswahl aus Dropdownliste: ▸ `activation`

	▸ `complementary latch` ▸ `complementary two output` Default: `complementary latch` Hinweis: Weitere Informationen finden Sie in der Übersichtstabelle im Kapitel "**LATCH ON und LATCH_OFF** (auf Seite 121)"

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
Name	Variablenname. Dem Namen wird der Projektname vorangestellt: Projektname#Variablenname
Identification	Kennung der Variable.
Point Index	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
Static Variation	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
Event Variation	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
Event Class	Aktuelle Konfiguration der event class für den point in dieser Object Group.
Control Model	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
Command Routing	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: UDINT, UINT

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

BINARY OUTPUT

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `packed format` Object Group 10 - Variation 1 ▸ `with flags` Object Group 10 - Variation 2 Default: `default`

Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `without time` Object Group 11 - Variation 1 ▸ `with time` Object Group 11 - Variation 2 Default: `default`
Control model	Diese Option bestimmt das Control Model für die ausgewählte Variable. Hinweis: Bestimmte Kombinationen aus Master control requests und Datentyp (BOOL oder USINT) der Variable sind nicht möglich. Auch in Abhängigkeit davon, ob **Command Routing** aktiv oder nicht aktiv ist für eine ausgewählte Variable. Auswahl aus Dropdownliste: ▸ `activation` Object Group n - Variation n ▸ `complementary latch` Object Group n - Variation n ▸ `complementary two output` Object Group n - Variation n Default: `default` Weitere Informationen dazu finden Sie im Kapitel **"Kommando-Bearbeitung - Command Routing"** (auf Seite 118).

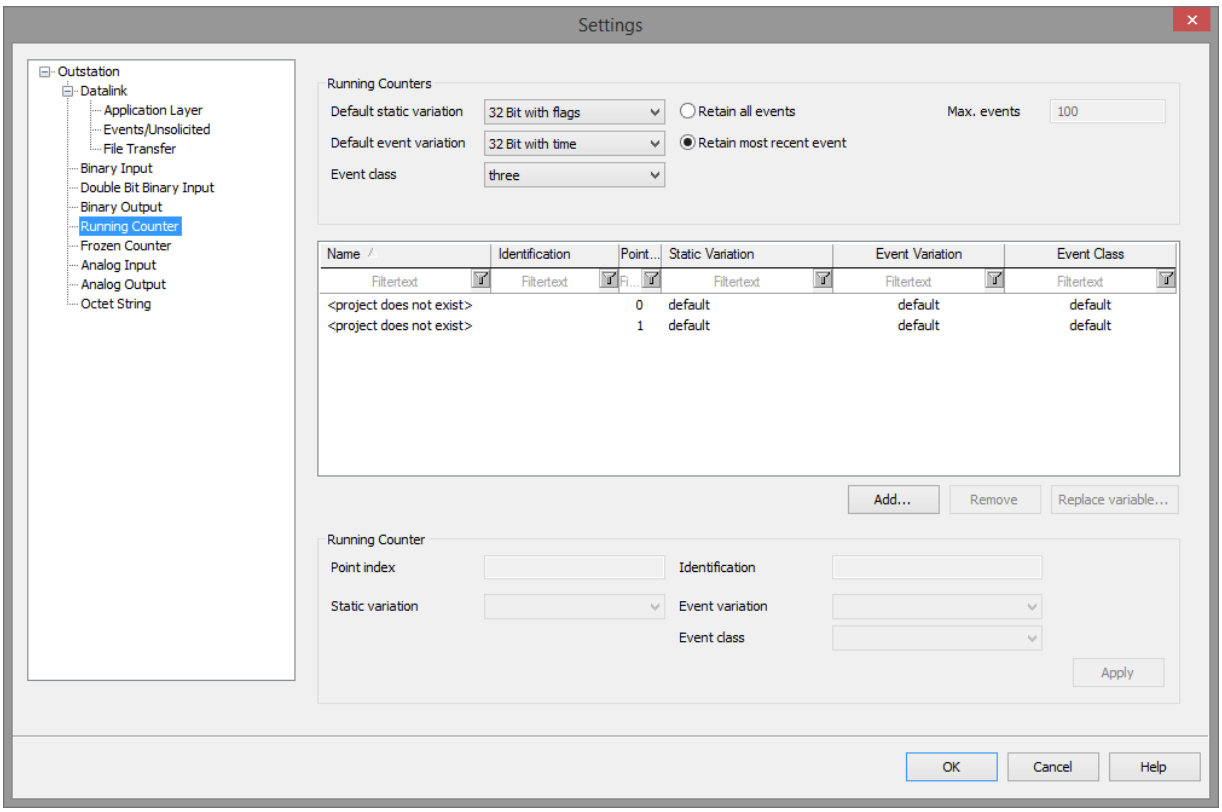
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Command routing	Diese Option bestimmt für die ausgewählte Variable das Verhalten vom Binary Output point bei select und operate requests vom master. ▶ `Aktiv:` select requests vom master und operate requests vom master werden an die **Befehlsgabe** in der zenon Runtime weitergeroutet. Erst nach erhaltener Rückmeldung auf das select oder operate von der SPS (an die **Befehlsgabe** in der Runtime) wird ein entsprechender `positive` oder `negative select response` oder `operate response` durch die Outstation an den Master gesendet. Direct operate requests werden von der Outstation sofort positiv beantwortet wenn die verknüpfte Variable in der Runtime geschrieben wurde. ▶ `Inaktiv:` select requests werden immer sofort von der Outstation positiv beantwortet, wenn eine Variable für den Binary Output point konfiguriert ist und die Variable in der Runtime vorhanden ist. operate requests und direct operate requests werden von der Outstation sofort positiv beantwortet wenn die verknüpfte Variable in der Runtime geschrieben wurde. Default: `aktiv` Weitere Details zu **Command routing** finden Sie im Kapitel "**Select before Operate** (auf Seite 118)".

Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.7 Running Counter



In diesem Knoten konfigurieren Sie die **DNP3 Running counter points**.

RUNNING COUNTER

Parameter	Beschreibung
Default static variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 20 - Variation 1 ▸ `16 Bit with flags` Object Group 20 - Variation 2 ▸ `32 Bit without flags` Object Group 20 - Variation 5 ▸ `16 Bit without flags` Object Group 20 - Variation 6 Default: `32 Bit with flags`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain most recent event`
Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events`

	konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variation	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 22 - Variation 1 ▸ `16 Bit with flags` Object Group 22 - Variation 2 ▸ `32 Bit with time` Object Group 22 - Variation 5 ▸ `16 Bit with time` Object Group 22 - Variation 6 Default: `32 Bit with time`
Event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die

	jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ `one` Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `two` Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `three`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
`Name`	Variablenname. Dem Namen wird der Projektname vorangestellt: `Projektname#Variablenname`
`Identification`	Kennung der Variable.
`Point Index`	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
`Static Variation`	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
`Event Variation`	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
`Event Class`	Aktuelle Konfiguration der event class für den point in dieser Object Group.
`Control Model`	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
`Command Routing`	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: `UDINT, UINT`

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

RUNNING COUNTER

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

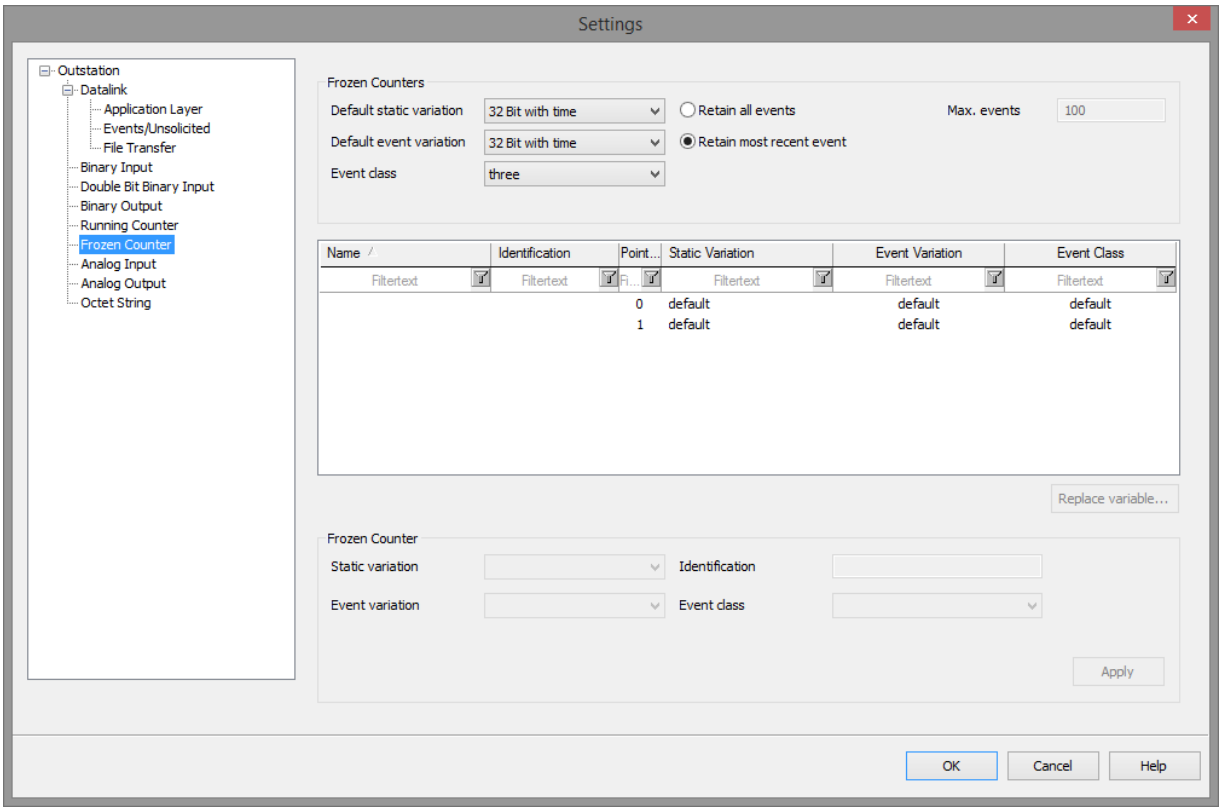
Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 20 - Variation 1 ▸ `16 Bit with flags` Object Group 20 - Variation 2 ▸ `32 Bit without flags` Object Group 20 - Variation 5 ▸ `16 Bit without flags` Object Group 20 - Variation 6 Default: `default`
Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object

	Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 22 - Variation 1 ▸ `16 Bit with flags` Object Group 22 - Variation 2 ▸ `32 Bit with time` Object Group 22 - Variation 5 ▸ `16 Bit with time` Object Group 22 - Variation 6 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.8 Frozen Counter



In diesem Knoten konfigurieren Sie die **DNP3 Frozen counter points**.

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 21 - Variation 1 ▸ `16 Bit with flags` Object Group 21 - Variation 2 ▸ `32 Bit with time` Object Group 21 - Variation 5 ▸ `16 Bit with time` Object Group 21 - Variation 6 ▸ `32 Bit without flags` Object Group 21 - Variation 9 ▸ `16 Bit without flags` Object Group 21 - Variation 10 Default: `32 Bit with time`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain most recent event`
Max events	Maximale Anzahl an Events welche im Eventpuffer für

	diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variations	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 23 - Variation 1 ▸ `16 Bit with flags` Object Group 23 - Variation 2 ▸ `32 Bit with time` Object Group 23 - Variation 5 ▸ `16 Bit with time` Object Group 23 - Variation 6 Default: `32 Bit with time`
Event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für

	die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ `one` Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `two` Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `three`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
`Name`	Variablenname. Dem Namen wird der Projektname vorangestellt: `Projektname#Variablenname`
`Identification`	Kennung der Variable.
`Point Index`	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
`Static Variation`	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
`Event Variation`	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
`Event Class`	Aktuelle Konfiguration der event class für den point in dieser Object Group.
`Control Model`	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
`Command Routing`	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: `UDINT, UINT`

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

FROZEN COUNTER

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

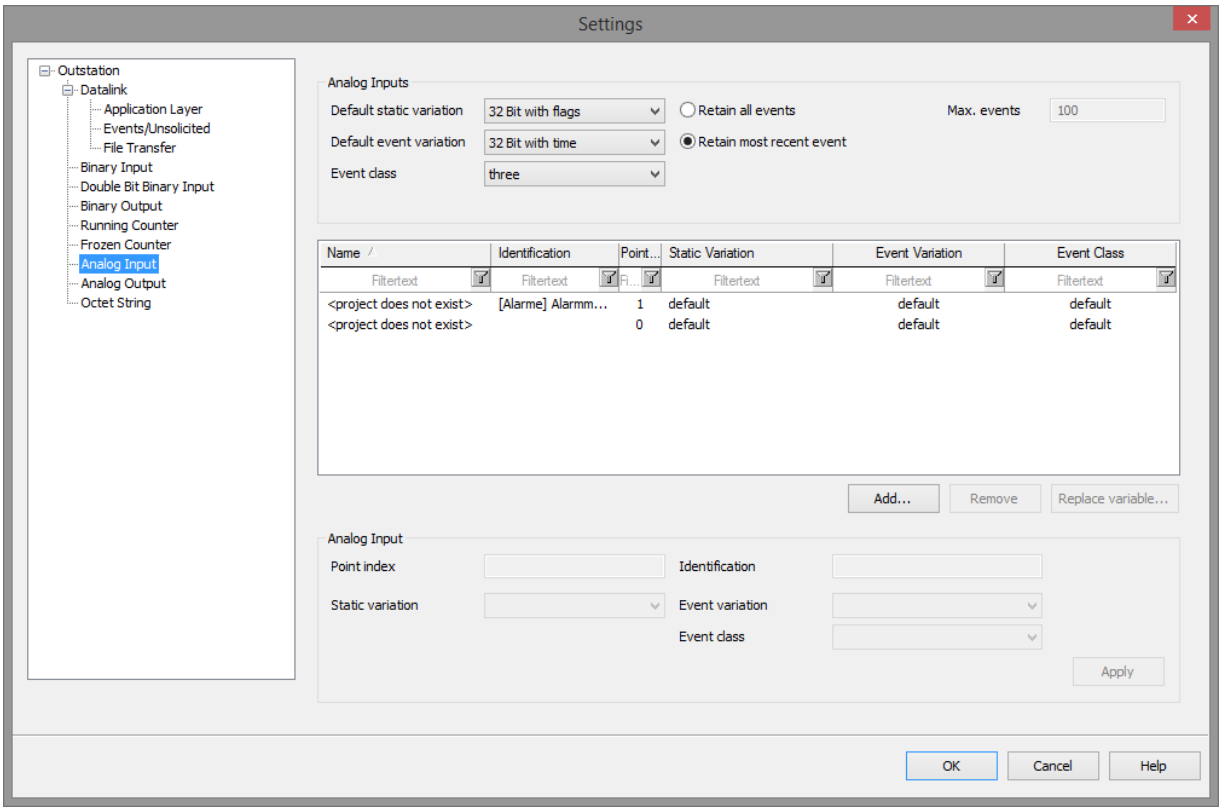
Parameter	Beschreibung
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 21 - Variation 1 ▸ `16 Bit with flags` Object Group 21 - Variation 2 ▸ `32 Bit with time` Object Group 21 - Variation 5 ▸ `16 Bit with time` Object Group 21 - Variation 6 ▸ `32 Bit without flags` Object Group 21 - Variation 9 ▸ `16 Bit without flags` Object Group 21 - Variation 10 Default: `default`
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 23 - Variation 1 ▸ `16 Bit with flags` Object Group 23 - Variation 2 ▸ `w32 Bit with time`

	Object Group 23 - Variation 5 ▸ `16 Bit with time` Object Group 23 - Variation 6 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.9 Analog Input



In diesem Knoten konfigurieren Sie die **DNP3 Analog Input points**.

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 30 - Variation 1 ▸ `16 Bit with flags` Object Group 30 - Variation 2 ▸ `32 Bit without flags` Object Group 30 - Variation 3 ▸ `16 Bit without flags` Object Group 30 - Variation 4 ▸ `Real with flags (Single-precision with flag)` Object Group 30 - Variation 5 ▸ `LReal with flags (Double-precision with flag)` Object Group 30 - Variation 6 Default: `32 Bit with flags`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat.

	Default: `Retain most recent event`
Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variation	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit without time` Object Group 32 - Variation 1 ▸ `16 Bit without time` Object Group 32 - Variation 2 ▸ `32 Bit with time` Object Group 32 - Variation 3 ▸ `16 Bit with time` Object Group 32 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 32 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 32 - Variation 6 ▸ `Real with time (Single-precision with time)` Object Group 32 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 32 - Variation 8 Default: `32 Bit with time`
Event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende

	event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ none Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ zero Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ one Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ two Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden

	keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `three`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
Name	Variablenname. Dem Namen wird der Projektname vorangestellt: Projektname#Variablenname
Identification	Kennung der Variable.
Point Index	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
Static Variation	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
Event Variation	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
Event Class	Aktuelle Konfiguration der event class für den point in dieser Object Group.
Control Model	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
Command Routing	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: BOOL, Byte, DINT, DWORD, Date, Date_and_Time, INT, LINT, LREAL, LWORD, REAL, SINT, TIME, TOD, UDINT, UINT, ULINT, USINT, WORD

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

ANALOG INPUT

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flags` Object Group 30 - Variation 1 ▸ `16 Bit with flags` Object Group 30 - Variation 2 ▸ `32 Bit without flags` Object Group 30 - Variation 3 ▸ `16 Bit without flags` Object Group 30 - Variation 4 ▸ `Real with flags (Single-precision with flag)` Object Group 30 - Variation 5 ▸ `LReal with flags`

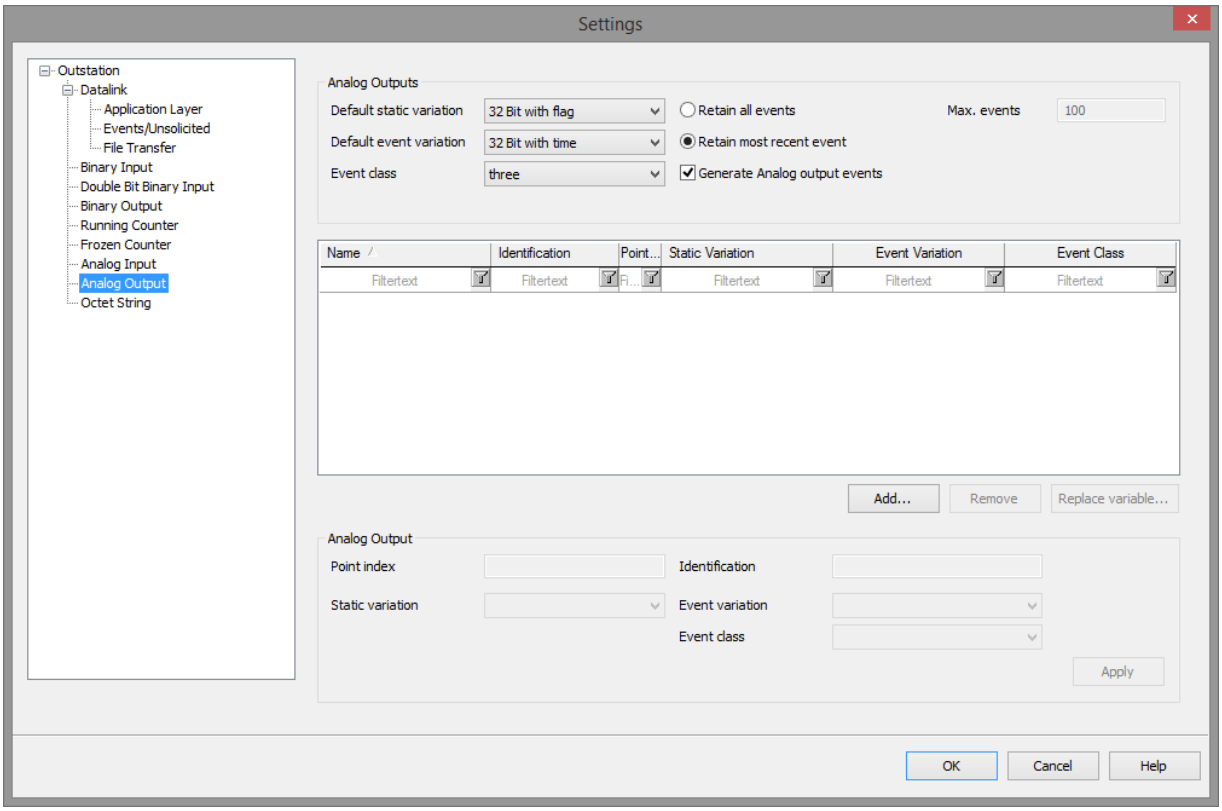
	(Double-precision with flag) Object Group 30 - Variation 6 Default: default

Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit without time` Object Group 32 - Variation 1 ▸ `16 Bit without time` Object Group 32 - Variation 2 ▸ `32 Bit with time` Object Group 32 - Variation 3 ▸ `16 Bit with time` Object Group 32 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 32 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 32 - Variation 6 ▸ `Real with time (Single-precision with time)` Object Group 32 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 32 - Variation 8 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.10 Analog Output



In diesem Knoten konfigurieren Sie die **DNP3 Analog Output points**.

Parameter	Beschreibung
Default static variation	Diese Option bestimmt die default static variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende static variation konfiguriert werden. Hinweis: Wählen sie eine static variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flag` Object Group 40 - Variation 1 ▸ `16 Bit with flag` Object Group 40 - Variation 2 ▸ `Real with flag (Single-precision with flag)` Object Group 40 - Variation 3 ▸ `LReal with flag (Double-precision with flag)` Object Group 40 - Variation 4 Default: `32 Bit with flag`
Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▸ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▸ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain most recent event`
Max events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein,

	wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.
Default event variations	Diese Option bestimmt die default event variation für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event variation konfiguriert werden. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit without time` Object Group 42 - Variation 1 ▸ `16 Bit without time` Object Group 42 - Variation 2 ▸ `32 Bit with time` Object Group 42 - Variation 3 ▸ `16 Bit with time` Object Group 42 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 42 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 42 - Variation 6 ▸ `Real with time (Double-precision with time)` Object Group 42 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 42 - Variation 8 Default: `32 Bit with time`
Event class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die

	Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ zero Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ one Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ two Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ three Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die

	jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `three`
Generate Analog output events	Diese Option bestimmt, ob bei Wertänderungen für Variablen, die für Binary Outputs konfiguriert sind, auch events erzeugt werden sollen. Nicht alle master unterstützen events für Binary Outputs. Default: `aktiv`

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
Name	Variablenname. Dem Namen wird der Projektname vorangestellt: Projektname#Variablenname
Identification	Kennung der Variable.
Point Index	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
Static Variation	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
Event Variation	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
Event Class	Aktuelle Konfiguration der event class für den point in dieser Object Group.
Control Model	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
Command Routing	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: BOOL, Byte, DINT, DWORD, Date, Date_and_Time, INT, LINT, LREAL, LWORD, REAL, SINT, TIME, TOD, UDINT, UINT, ULINT, USINT, WORD

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

ANALOG OUTPUT

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

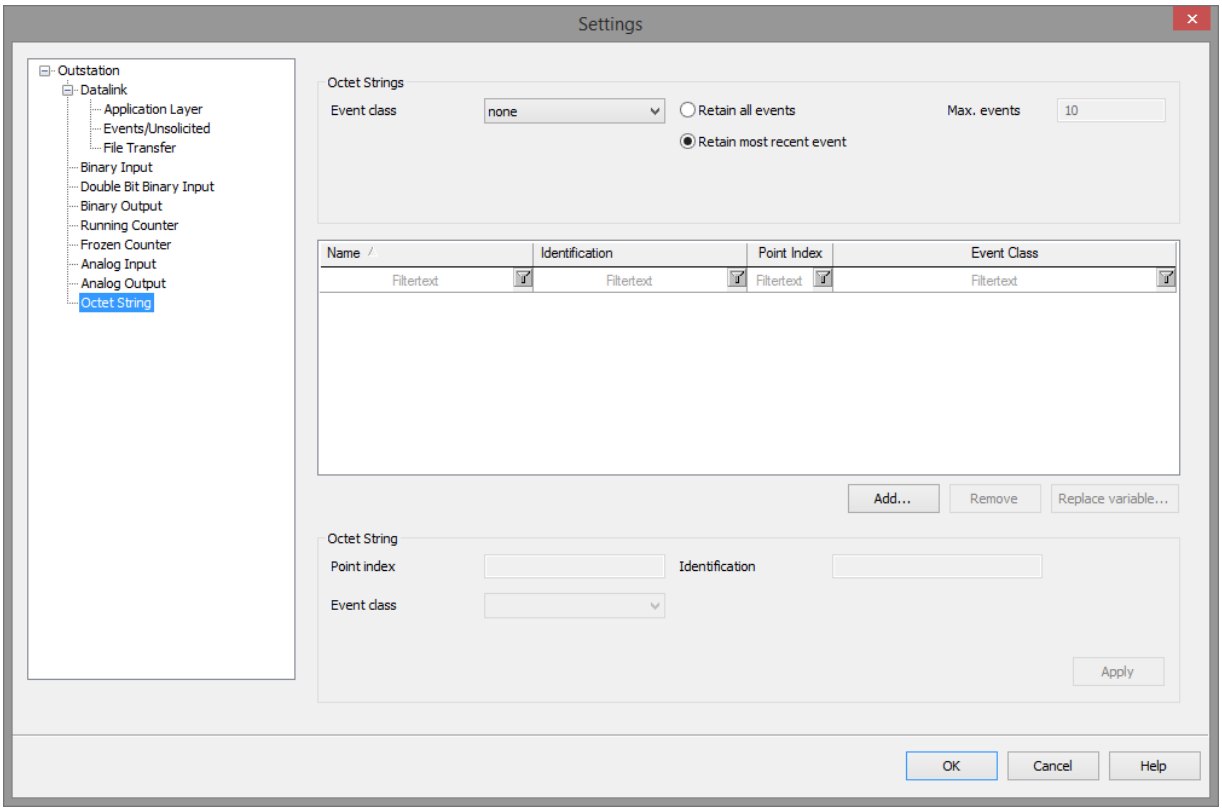
Parameter	Beschreibung
Pointer Index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Static variation	Diese Option bestimmt für die ausgewählte Variable die static variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, static variation verwendet. Hinweis: Wählen sie eine static variation welche vom master unterstützt wird. Nicht jeder master unterstützt alle static variations. Auswahl aus Dropdownliste: ▸ `32 Bit with flag` Object Group 40 - Variation 1 ▸ `16 Bit with flag` Object Group 40 - Variation 2 ▸ `Real with flag (Single-precision with flag)` Object Group 40 - Variation 3 ▸ `LReal with flag (Double-precision with flag)` Object Group 40 - Variation 4 Default: `default`

Event variation	Diese Option bestimmt für die ausgewählte Variable die event variation für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event variation verwendet. Hinweis: Wählen sie eine event variation, welche vom master unterstützt wird. Nicht jeder master unterstützt alle event variations. Auswahl aus Dropdownliste: ▸ `32 Bit without time` Object Group 42 - Variation 1 ▸ `16 Bit without time` Object Group 42 - Variation 2 ▸ `32 Bit with time` Object Group 42 - Variation 3 ▸ `16 Bit with time` Object Group 42 - Variation 4 ▸ `Real without time (Single-precision without time)` Object Group 42 - Variation 5 ▸ `LReal without time (Double-precision without time)` Object Group 42 - Variation 6 ▸ `Real with time (Double-precision with time)` Object Group 42 - Variation 7 ▸ `LReal with time (Double-precision with time)` Object Group 42 - Variation 8 Default: `default`
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.11 Octet String



In diesem Knoten konfigurieren Sie die **DNP3 Octet string points**.

Parameter	Beschreibung
Event Class	Diese Option bestimmt die default event class für diese Object Group, für neu hinzugefügte Variablen. Einzeln, pro Variable, kann auch eine abweichende event class konfiguriert werden. Auswahl aus Dropdownliste: ▸ `none` Variablen sind keiner event class zugeordnet und erzeugen keine events. In einer Antwort auf ein class 0 poll vom master sind keine static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen nur durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, nicht einer andere event class zuordnen. ▸ `zero` Variablen sind der event class 0 zugeordnet, erzeugen jedoch keine events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann die Variablen auch durch ein read request für die jeweilige static object group lesen. Ein Master kann die Variablen über ein assign class request, der event class 1, 2 oder 3 zuordnen, wodurch die Variablen ab diesem Zeitpunkt auch events erzeugen. ▸ `one` Variablen sind der event class 1 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 1 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 2 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `two` Variablen sind der event class 2 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 2 poll vom master sind eventuell

	vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 3 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. ▸ `three` Variablen sind der event class 3 zugeordnet, und erzeugen events. In einer Antwort auf ein class 0 poll vom master sind die static objects für die Variablen enthalten. Ein Master kann den aktuellen Wert der Variablen auch durch ein read request für die jeweilige static object group lesen. In einer Antwort auf ein class 3 poll vom master sind eventuell vorhandene event objects für die events der Variablen enthalten. Ein Master kann die Variablen über ein assign class request, der event class 0, 1 oder 2 zuordnen. Wenn die Variable durch den Master der event class 0 zugeordnet wird, werden keine events mehr erzeugt. Default: `none`

Event retention	Verhalten bei der Weitergabe von Wertänderungen. Auswahl aus Optionsfeld: ▶ `Retain all events` Für jede Änderung von Wert, Status oder Zeitstempel einer Variable, wird ein neues Event erstellt und im Eventpuffer gespeichert. Der Master bekommt alle Änderungen. ▶ `Retain most recent event` Der Eventpuffer enthält pro Variable jeweils nur ein Event für die letzte Änderung von Wert, Status oder Zeitstempel. Der Master bekommt nur die letzte Änderung und kann unter Umstände nicht unterscheiden ob oder wie oft sich ein Wert zwischendurch geändert hat. Default: `Retain most recent event`
Max. events	Maximale Anzahl an Events welche im Eventpuffer für diese Object Group, für alle definierten Variablen gespeichert wird. Wenn `Retain all events` konfiguriert ist, soll diese Anzahl mindestens so groß sein, wie die Anzahl der konfigurierten Variablen. Default: `100` Hinweis: Nur aktiv wenn `Retain all events` aktiviert ist.

In dieser Liste werden aus dem zenon Projekt übernommene Variablen mit ihrer aktuellen Konfiguration aufgelistet.

- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

Parameter	Beschreibung
Name	Variablenname. Dem Namen wird der Projektname vorangestellt: Projektname#Variablenname
Identification	Kennung der Variable.
Point Index	Aktuelle Konfiguration des DNP3 point index der Variable in dieser Object Group.
Static Variation	Aktuelle Konfiguration der static variation für den point in dieser Object Group.
Event Variation	Aktuelle Konfiguration der event variation für den point in dieser Object Group. Hinweis: Dieser Parameter ist im Knoten **Octet String** nicht verfügbar.
Event Class	Aktuelle Konfiguration der event class für den point in dieser Object Group.
Control Model	Aktuelle Konfiguration des **Control Model**. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.
Command Routing	Aktuelles Verhalten vom Binary Output point bei select und operate requests vom master. Hinweis: Dieser Parameter ist nur im Knoten **Binary Output** nicht verfügbar.

Unterstützte Datentypen: STRING, WSTRING

NAVIGATIONSLEISTE

Parameter	Beschreibung
Add...	Öffnet den Variablenauswahldialog (auf Seite 112) zur Übernahme von zenon Variablen in die Liste. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Remove	Entfernt ausgewählte Variablen aus der Liste. Vor dem Löschen der Variablen wird ein Bestätigungsdialog angezeigt. Nur aktiv, wenn eine oder mehrere Variablen in der Liste ausgewählt sind. Hinweis: Nicht aktiv im Knoten **Frozen Counter**.
Replace variable...	Öffnet den Variablenauswahldialog (auf Seite 112) um die in der Liste gewählte Variable mit einer anderen zenon Variable zu ersetzen. Nur aktiv, wenn exakt eine Variable in der Liste ausgewählt ist.

OCTET STRING

In diesem Bereich konfigurieren Sie die Parameter für ausgewählte Variable(n) der Liste.

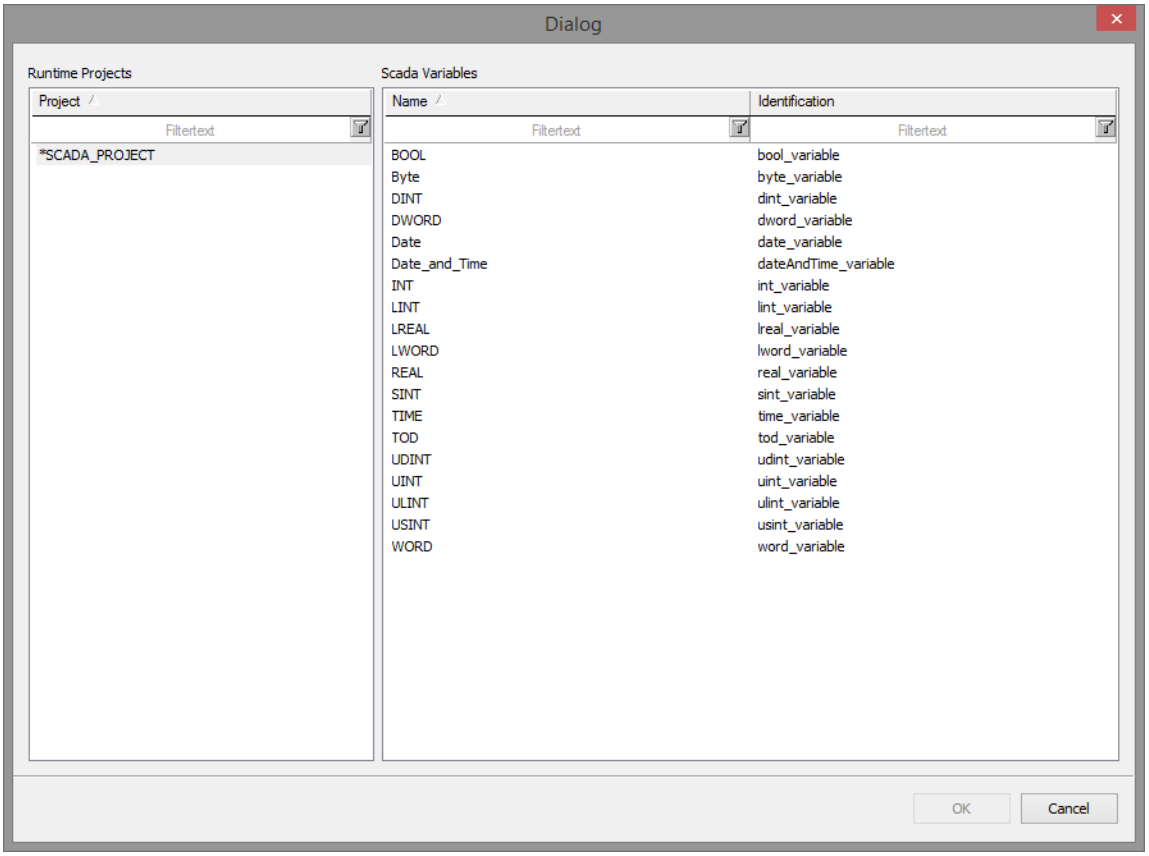
Parameter	Beschreibung
Point index	Diese Option bestimmt für die ausgewählte Variable den DNP3 point index in der Object Group. Werden Variablen hinzugefügt, wird für die neu hinzugefügten Variablen der point index automatisch gesetzt und beginnend mit dem höchsten bereits vorhandenen point index durchnummeriert. Ein point index darf nur einmal innerhalb einer Object Group vergeben werden. Die Eingabe wird validiert. Ist ein point index bereits vergeben, wird dies in einem Warndialog angezeigt. Hinweis: Es wird empfohlen, den point index beginnend mit 0 ohne Lücken zu vergeben. Die outstation unterstützt den maximalen point index. Wird ein hoher point index genutzt, soll sichergestellt werden, dass der master entsprechend auch qualifier codes mit 2 octets oder 4 octets unterstützt.
Identification	Die Kennung wird prinzipiell von der Variable aus der zenon Runtime übernommen. Mit dieser Option kann die Kennung zusätzlich lokal angepasst werden.
Event class	Diese Option bestimmt für die ausgewählte Variable die event class für den point in dieser Object Group. Mit dem Wert `default` wird die allgemeine, für diese Object Group definierte, event class verwendet. Auswahl: siehe Beschreibung für Option **Default event class**. Default: `default`
Apply	Übernimmt die Konfiguration für die ausgewählte(n) Variable(n).

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

7.12 Variablenauswahldialog

In diesem Dialog übernehmen Sie Variablen einer bestehenden zenon Projektierung in die **DNP3_SG** Konfiguration. Mehrfachauswahl ist möglich.



- Die Liste ist sortierbar
  - Klicken Sie für die Sortierung auf die Spaltenüberschrift.
  - Die Sortierreihenfolge wird mit einem Pfeilsymbol neben der Spaltenüberschrift visualisiert:
    `Pfeil nach oben`: aufsteigende Sortierung
    `Pfeil nach unten`: absteigende Sortierung
  - Ein nochmaliger Klick auf die Spaltenüberschrift kehrt die Sortierreihenfolge um.
- Die Liste ist filterbar
  Um die Liste zu filtern:
  - Tragen Sie den gewünschten Filterbegriff in das Eingabefeld unter der Überschrift ein. Die Default-Bezeichnung eines leeren Filters ist `Filtertext` (dargestellt in grauer Schrift).

RUNTIME PROJECTS

Parameter	Beschreibung
[Projektname]	Name der in der Runtime geladenen zenon Projekte. Das aktive Projekt ist mit einem * (Sternzeichen) gekennzeichnet

SCADA VARIABLES

Parameter	Beschreibung
[Variablenliste]	Liste der projektierten Variablen in den geladenen zenon Projekten. Die Liste bietet jene Datentypen zur Auswahl an, die für den jeweiligen Knoten unterstützt werden. ▸ Name: Name der Variable. Dies entspricht der Variableneigenschaft **Name** der aktuellen zenon Projektierung. ▸ Identification: Kennung der Variable. Dies entspricht der Variableneigenschaft **Kennung** der aktuellen zenon Projektierung.

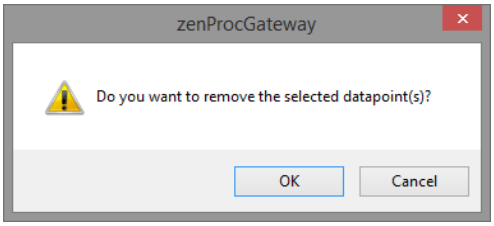
DIALOG BEENDEN

OK	Übernimmt Einstellungen und schließt den Dialog. Die ausgewählten Variablen werden in die **DNP3_SG** Konfiguration des aktuellen Knotens übernommen.
Cancel	Verwirft alle Änderungen und schließt den Dialog. Es werden keine Variablen in die DNP3_SG Konfiguration übernommen.

7.13 Warndialoge

VARIABLE (DATENPUNKT) ENTFERNEN

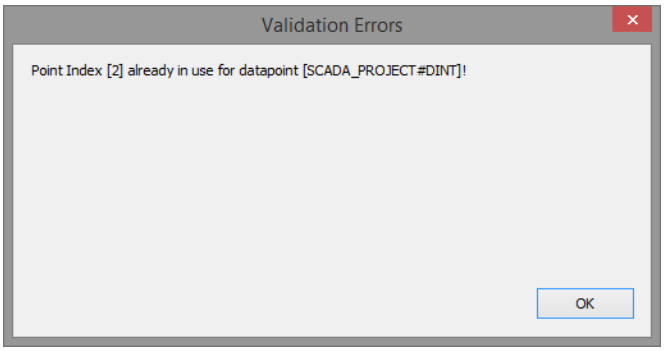
Vor dem Löschen einer Variable wird folgender Warndialog angezeigt:



Parameter	Beschreibung
OK	Löscht die ausgewählte(n) Variable(n) aus der Liste.
Cancel	Schließt den Dialog ohne die ausgewählte(n) Variable(n) zu löschen.

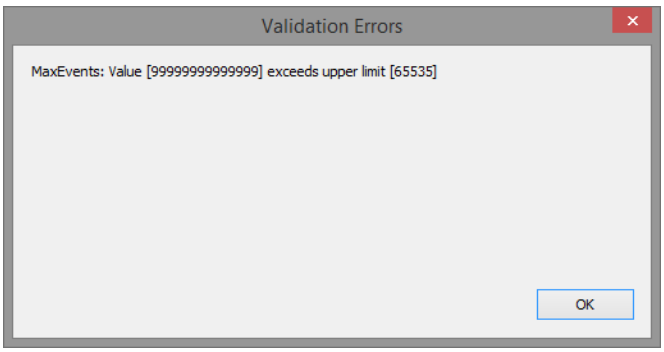
DATENPUNKT NICHT EINDEUTIG

Wird für eine Variable ein Point index konfiguriert, der bereits in der aktuellen Konfiguration vorhanden ist wird dies mit einem Warndialog visualisiert.



UNGÜLTIGE EINGABE

Wird für eine Option eine ungültige Eingabe konfiguriert wird dies in einem Warndialog angezeigt.



7.14 Statusmapping zenon - DNP3

Das Point-Mapping des zenon Status auf den DNP-Status wird durchgeführt, wenn ein Ereignis im Eventpuffer aufgezeichnet wird.

Das Mapping zwischen zenon und DNP 3 erfolgt nach folgender Aufstellung:

	Status DNP3							
Status zenon	**ONLINE** (s. Anm. 2)	**COMM_LOST** (s. Anm. 2)	**LOCAL_FORCED** (s. Anm. 2)	**OVER_RANGE** (s. Anm. 3)	**REFERENCEERR** (s. Anm. 3)	**DISCONTINUITY** (s. Anm. 3)	**ROLLOVER** (s. Anm. 3)	**CHATTERFILTER** (s. Anm. 3)
SPONT (s. Anm. 1)	1	0	0					
GA (s. Anm. 1)	1	0	0					
INVALID (s. Anm. 1)	0	1	0					
ALTVAL (s. Anm. 1)	0	0	1					
OFF (s. Anm. 1)	0	0	0					

N_UPDATE (s. Anm. 1)	0	0	0					
NT_870 (s. Anm. 1)	0	0	0					
SB_870 (s. Anm. 1)	1	0	1					
OV_870				1				
OR_870				1				

ANMERKUNGEN

- Anmerkung 1:

  In zenon muss mindestens einer der Status gesetzt sein. Ist dies nicht der Fall wird ein entsprechender LOG-Eintrag erstellt. Es werden keine zusätzlichen Flags verwendet.

- Anmerkung 2:

  Erzwungene Basis-Flags, diese Flags werden immer als Bit-Feld gesetzt.

- Anmerkung 3:

  Flags werden nur bei ONLINE oder LOCAL_FORCED gesetzt.

7.15 Select before Operate

`Analog Outputs` und `Binary Outputs` verfügen über die Funktionalität `SBO (Select before Operate)`. Dabei gilt:

- der DNP3-Master sendet ein `Select`-Kommando an die Outstation.
- bei Requests des Objekttyps `Binary Output` müssen folgende Anforderungen erfüllt sein, damit der Request positiv beantwortet wird:
  - die Variable existiert und ist angemeldet oder kann angemeldet werden.
  - der Datentyp der Variable ist `IEC_BOOL` oder `IEC_USINT`
  - der Wert des Kommandos ist valide:
    `NUL, PulseOn, PulseOff, LatchOn, LatchOff, PulseOn + Close, PulseOn + Trip`
- Anfragen (`Requests`) des Objekttyps `Analog Output` werden immer positiv beantwortet
- Das konfigurierte `Select`-Timeout wird automatisch aktiviert.
  Wird innerhalb dieses Timeouts kein `Operate`-Kommando für die jeweilige Variable empfangen, wird die Bearbeitung für die Variable automatisch beendet.
  Hinweis: Wird anstelle eines `Operate Request` für die jeweilige Variable ein anderer `Request` empfangen, stimmt die Sequenznummer nicht oder der `Operate Request` entspricht nicht dem `Select Request`. Die Bearbeitung wird automatisch beendet.
- Ein `Operate Request` ohne vorangegangenes `Select Request` wird mit einem Fehler quittiert.

> **Tipp**
>
> *Weitere allgemeine Informationen zu **Select before Operate** finden Sie im Handbuch Energy Edition in der **Befehlsgabe** im Kapitel Ausführung eines Befehls.*

7.16 Kommando-Bearbeitung - Command Routing

Grundlegende Voraussetzungen für die positive Ausführung des `Select`- oder `Operate`-Kommandos sind:

- der Variablentyp ist `Binary Output`
- die Variable existiert, ist angemeldet oder kann angemeldet werden und gültig (kein `I-Bit`).
- der Datentyp der Variable ist `IEC_REAL`, `IEC_BOOL` oder `IEC_USINT`.
- Der Kontrollcode ist gültig und kann in einen boolschen Wert gewandelt werden-
  Die folgenden Werte sind gültig (der entpsrechende boolsche Wert ist in Klammer aufgeführt):

- NUL (false)
- PulseOn (true)
- PulseOff (false)
- LatchOn (true)
- LatchOff (false)
- PulseOn + Close (true)
- PulseOn + Trip (false)

ABLAUF

Generell gilt: Command Routing wird nur für exportierte Variable vom Type Binary Output unterstützt, bei denen die Option **Command Routing** aktiviert ist.

- Wird ein Select-Kommando für ein Binary Output empfangen, bei dem die Option **Command Routing** deaktiviert ist, wird ein positiver Response an den Master zurückgesendet, sofern die grundlegende Voraussetzungen erfüllt sind.
- Ist beim Binary Output die Option **Command Routing** gesetzt, wird das Kommando COT_act mit dem Kontrollcode an die (bei **Binary Output**) konfigurierte Befehlsvariable in der Runtime gesendet.
  Hinweis: COT_act ist im Statuswert mit Hilfe der Bits COT_act, S_SE_BIT und S_SELECT codiert.
- Anschließend wird auf die Ausführung des Select-Kommandos durch die Auto-Remote Befehlsaktion gewartet, bevor der Response an den Master gesendet wird.
  Erst nachdem eine positive Antwort auf das Select-Kommando an den Master gesendet wird, startet der Select Timeout Timer.
- Die Bearbeitung des nachfolgenden Operate-Kommandos erfolgt gleich wie beim Select-Kommando.
  Zusätzlich wird bei einem Operate-Kommando geprüft, ob vorher ein Select–Kommando empfangen wurde, bevor das Kommando an die Befehlsgabe der Runtime weitergegeben wird.
- Ist beim Operate-Kommando die Option **Command Routing** nicht aktiv oder wurde zuvor kein Select-Kommando empfangen, wird ein Operate Request direkt beantwortet.
- Nach positiver Abarbeitung des Operate-Kommandos durch die Befehlsgabe in der Runtime, wird an den Master ein positiver Operate Response gesendet. Im Fehlerfall wird ein Operate Response mit Fehler an den Master gesendet.

STATUS DES COMMAND ROUTINGS

Folgende Statuscodes werden von der Outstation an den Master gesendet:

- **Status Code 0 - SUCCESS**
  Die Kommando-Bearbeitung war erfolgreich.

- **Status Code 1 - TIMEOUT**
  Während der Kommando-Bearbeitung ist ein Timeout aufgetreten.
  Die Befehlsgabe konnte die die Kommando-Bearbeitung nicht innerhalb von 2 Minuten erfolgreich abschließen.
- **Status Code 2 - NO_SELECT**
  Das `Select`-Kommando wurde von der Befehlsgabe mit einem ungültigen Status beendet.
  Ungültige Status sind:
  - `S_PN_BIT` (negative Bestätigung) erhalten
  - das `S_SE_BIT` ist nicht mehr gesetzt
  - keines der `COT`-Bits ist gesetzt
- **Status Code 6 - HARDWARE_ERROR**
  Dieser Fehler tritt unter folgenden Umständen auf:
  - die Variable ist nicht angemeldet oder kann nicht angemeldet werden
  - die Variable hat das **INVALID** Bit
  - der Wert kann nicht geschrieben werden oder der Befehl kann nicht abgesetzt werden.
- **Status Code 10 - AUTOMATION_INHIBIT**
  Das `Operate`-Kommando wurde von der Befehlsgabe mit einem ungültigen Status beendet.
  Ungültige Status sind:
  - `S_PN_BIT` (negative Bestätigung) erhalten
  - das `S_SE_BIT` ist nicht mehr gesetzt
  - keines der `COT`-Bits ist gesetzt
- **Status Code 12 - OUT_OF_RANGE**
  Der im `Request` enthaltene Kontrollcode ist ungültig.

7.17 LATCH_ON und LATCH_OFF

Der Command Control Code von der Steuerung wird über das **Process Gateway** auf USINT Variablen als 8-Bit Wert geschrieben, wenn **Command Routing** nicht aktiv ist.

Control Code		Control Mode	Variable linked	command routing	Acti
0x01	**Pulse On**	Activation	**BOOL**	disabled	set
0x01	**Pulse On**	Activation	**USINT**	disabled	set
0x01	**Pulse On**	Complementary latch	**n.a.**	disabled	ret
0x01	**Pulse On**	Complementary two output	**n.a.**	disabled	ret
0x01	**Pulse On**	Activation	**BOOL**	enabled	set
0x01	**Pulse On**	Activation	**USINT**	enabled	set
0x01	**Pulse On**	Complementary latch	**n.a.**	enabled	ret
0x01	**Pulse On**	Complementary two output	**n.a.**	enabled	ret
0x03	**Latch On**	Activation	**BOOL**	disabled	set
0x03	**Latch On**	Activation	**USINT**	disabled	set
0x03	**Latch On**	Complementary latch	**BOOL**	disabled	set
0x03	**Latch On**	Complementary latch	**USINT**	disabled	set
0x03	**Latch On**	Complementary two output	**BOOL**	disabled	ret
0x03	**Latch On**	Complementary two output	**USINT**	disabled	set
0x03	**Latch On**	Activation	**BOOL**	enabled	set
0x03	**Latch On**	Activation	**USINT**	enabled	set
0x03	**Latch On**	Complementary latch	**BOOL**	enabled	set
0x03	**Latch On**	Complementary latch	**USINT**	enabled	set
0x03	**Latch On**	Complementary two output	**BOOL**	enabled	set
0x03	**Latch On**	Complementary two output	**USINT**	enabled	set
0x04	**Latch Off**	Activation	**BOOL**	disabled	set
0x04	**Latch Off**	Activation	**USINT**	disabled	set
0x04	**Latch Off**	Complementary latch	**BOOL**	disabled	set
0x04	**Latch Off**	Complementary latch	**USINT**	disabled	set
0x04	**Latch Off**	Complementary two output	**BOOL**	disabled	ret
0x04	**Latch Off**	Complementary two output	**USINT**	disabled	set
0x04	**Latch Off**	Activation	**BOOL**	enabled	set
0x04	**Latch Off**	Activation	**USINT**	enabled	set
0x04	**Latch Off**	Complementary latch	**BOOL**	enabled	set
0x04	**Latch Off**	Complementary latch	**USINT**	enabled	set
0x04	**Latch Off**	Complementary two output	**BOOL**	enabled	set
0x04	**Latch Off**	Complementary two output	**USINT**	enabled	set

0x41	**Close**	Activation	**BOOL**	disabled	set
0x41	**Close**	Activation	**USINT**	disabled	set
0x41	**Close**	Complementary latch	**BOOL**	disabled	set
0x41	**Close**	Complementary latch	**USINT**	disabled	set
0x41	**Close**	Complementary two output	**BOOL**	disabled	ret
0x41	**Close**	Complementary two output	**USINT**	disabled	set
0x41	**Close**	Activation	**BOOL**	enabled	set
0x41	**Close**	Activation	**USINT**	enabled	set
0x41	**Close**	Complementary latch	**BOOL**	enabled	set
0x41	**Close**	Complementary latch	**USINT**	enabled	set
0x41	**Close**	Complementary two output	**BOOL**	enabled	set
0x41	**Close**	Complementary two output	**USINT**	enabled	set
0x81	**Trip**	Activation	**BOOL**	disabled	set
0x81	**Trip**	Activation	**USINT**	disabled	set
0x81	**Trip**	Complementary latch	**BOOL**	disabled	set
0x81	**Trip**	Complementary latch	**USINT**	disabled	set
0x81	**Trip**	Complementary two output	**BOOL**	disabled	ret
0x81	**Trip**	Complementary two output	**USINT**	disabled	set
0x81	**Trip**	Activation	**BOOL**	enabled	set
0x81	**Trip**	Activation	**USINT**	enabled	set
0x81	**Trip**	Complementary latch	**BOOL**	enabled	set
0x81	**Trip**	Complementary latch	**USINT**	enabled	set
0x81	**Trip**	Complementary two output	**BOOL**	enabled	set
0x81	**Trip**	Complementary two output	**USINT**	enabled	set

8. DNP3 Slave - AccessDNP3 veraltet

Ab zenon Version 8.00 wurde der **DNP3 Slave** (**AccessDNP3**) mit dem neuen **Process Gateway** Modul **DNP3_SG** ersetzt.

- Das veraltete Modul **AccessDNP3** steht für neue Konfigurationen nicht mehr zur Auswahl.
- Bestehende Konfigurationen mit dem bisherigen **AccessDNP3** Modul sind weiterhin lauffähig.

In eine zukünftige Version wird das **AccessDNP3** Modul nicht mehr im Lieferumfang enthalten sein. Das alte Modul wird vollständig durch das Modul **AccessDNP3_SG** ersetzt werden.

Info

Die Dokumentation für das veraltete Modul ist nicht mehr im Lieferumfang enthalten. Wenden Sie sich bitte mit der E-Mail-Adresse support@copadata.com an Ihren Supportpartner um diese Dokumentation anzufordern.

9. ICCP-TASE.2

Das Protokoll **IEC 60870-6 TASE.2 ICCP** kann über das zenon **Process Gateway** mit dem Modul **AccessICCP.dll** Modul genutzt werden. ICCP-TASE.2 unterstützt Conformance Blocks 1 and 2 als Client und Server. Per default verhält sich das **Process Gateway** beim Aufbau der Kommunikation wie ein Server.

Die Kommunikation in ICCP-Protokoll ist spontan. Ein ICCP-Server meldet die Wertänderungen zu seinem Kommunikationspartner, dem entfernten Client. Ein ICCP-Server kann jedoch nicht die Wertänderungen vom entfernten Client aktiv lesen. Um einen bidirektionalen Datenaustausch zu gewährleisten, sollen beide Kommunikationspartner über einen Client und einen Server verfügen.

MÖGLICHE SYSTEMKONFIGURATIONEN

- nur Server

  Kommunikation mit mehreren entfernten ICCP-Kommunikationspartner möglich.
  Dieser Fall trifft dann zu, wenn auf der Registerkarte **Client Variables** keine Variablen konfiguriert sind.

- Server und passiver Client

  Der Client wird die Kommunikation mit dem entfernten ICCP-Sever nicht selbst initialisieren.
  Die Kommunikation mit mehreren entfernten ICCP-Kommunikationspartner ist möglich.
  Dieser Fall trifft dann zu, wenn die Eigenschaft **Initalise communication** nicht aktiv ist.
  Hinweis: Registerkarte **General** im Konfigurationsdialog des **Process Gateway**.

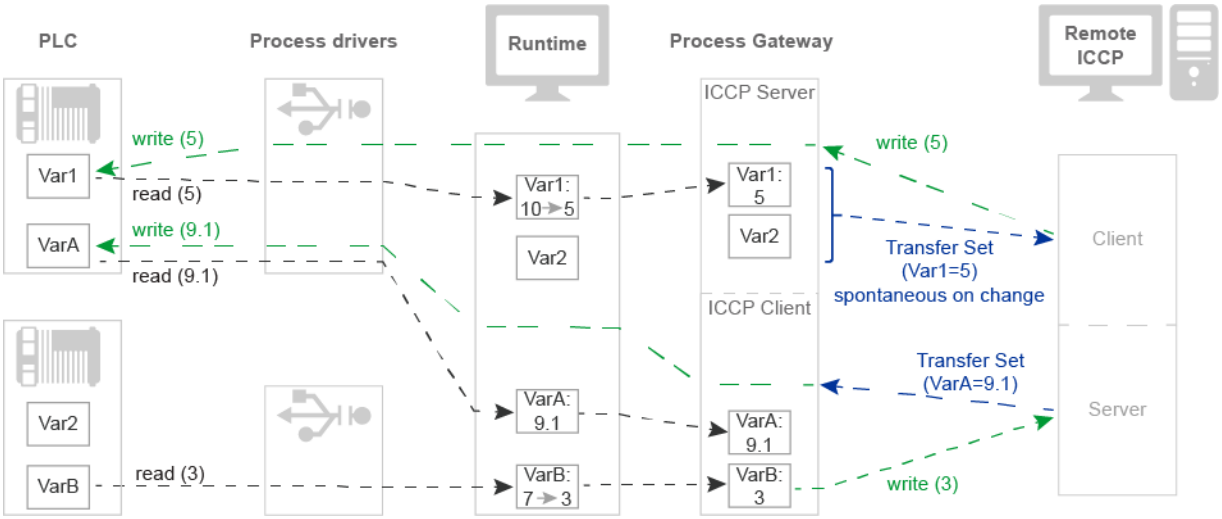
- Server und aktiver Client

  Der Client kann die Kommunikation mit entfernen ICCP-Server selbst initialisieren.
  Eine Kommunikation ist nur mit einem entfernten ICCP-Kommunikationspartner möglich.

KOMMUNIKATIONSAUFBAU

Nach Initialisierung der Kommunikation durch einen entfernten Client zum zenon ICCP-Server oder Initialisierung durch den eigenen Client, wird die Verifizierung der **bilateralen Tabellen-ID** durchgeführt. War die Verifizierung erfolgreich, registriert der zenon ICCP-Client die zu kommunizierenden Variablen am entfernten Server als DataSet und aktiviert die spontane Kommunikation des DSTransferSet (`MMS.InformationReport`).

Hinweis: Der ICCP-Client kommuniziert mit dem Server auch dann, wenn in der Konfiguration der Registerkarte **Client variables** (auf Seite 139) in der Liste **Available from ICCP** keine Variablen enthalten sind. In diesem Fall überprüft der Client nur die bilaterale Tabelle.

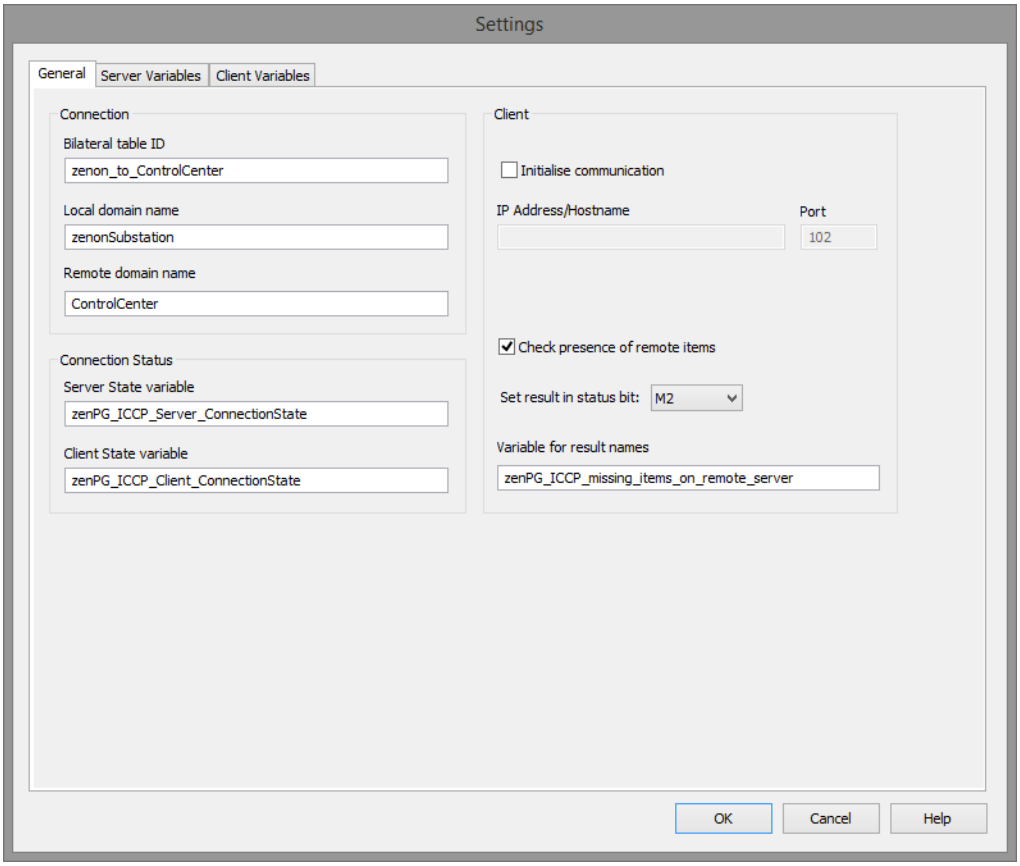


ICCP-DATENAUSTAUSCH

- ICCP-Server
  - Der ICCP-Server sendet Wertänderungen von Variablen spontan in sogenannten `DSTransferSet`s (`MMS.InformationReport`). Ändern sich die Werte von Variablen in der zenon Runtime, wird ein `DSTransferSet` zum entfernten Client gesendet.
  - Der ICCP-Server kann auch Schreibanforderungen vom entfernten Client (`MMS.write request`) empfangen. Diese werden an die zenon Runtime übergeben und die Runtime leitet diese Anforderung an entsprechende zenon Treiber weiter. Wird das Schreiben von der SPS bestätigt, ändert die zenon Variable ihren Wert.
- ICCP-Client
  - Bei Wertänderung einer Variable in der zenon Runtime sendet der ICCP-Client eine entsprechende Schreibanforderung (`MMS.write request`) zum entfernten Server.
  - Bei Empfang eines `DSTransferSet` (`MMS.InformationReport`) durch den ICCP-Client (vom entfernten Server) werden die empfangenen Werte an die zenon Runtime übergeben. Die Runtime leitet diese Anforderungen an die zenon Treiber weiter. Wird das Schreiben von der SPS bestätigt, ändern die zenon Variablen ihren Wert.

9.1 Modulkonfiguration

Die Konfiguration erfolgt mittels dreier Registerkarten. Sie kann erst mit Klick auf OK bestätigt und abgeschlossen werden, wenn alle drei Registerkarten konfiguriert wurden.



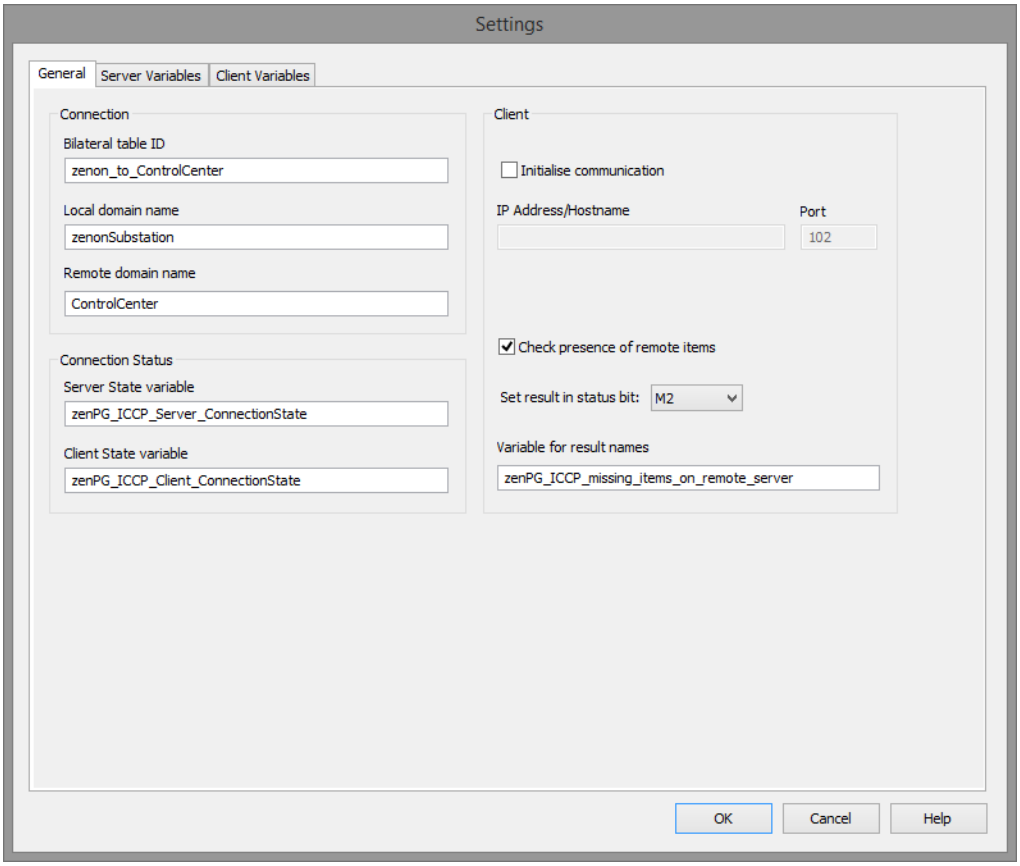
Parameter	Beschreibung
General (auf Seite 128)	Allgemeine Einstellungen.
Server variables (auf Seite 135)	Konfiguration der Server-Variablen. Auswahl der Variablen, die der ICCP-Server bereitstellt.
Client variables (auf Seite 139)	Konfiguration der Client-Variablen. Auswahl der Variablen, die vom ICCP-Client empfangen werden.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

9.1.1 General

Konfiguration der Verbindungseinstellungen.



CONNECTION

Die Einstellungen in der Sektion `Connection` sind beliebige Strings, die auf allen Geräten im System abgestimmt werden.

Die eindeutige bilaterale Tabellenkennzeichnung muss auf allen mit dem **ICCP Process Gateway** kommunizierenden Geräten gleich sein. Sie wird beim Verbindungsaufbau von Komunikationspartnern abgefragt. Bei fehlender Übereinstimmung wird die Verbindung getrennt.

Die Domänennamen geben die, auf MMS-Protokollebene, benutzten Referenzen der ICCP-Objekte an. Sie müssen bei den Kommunikationspartner spiegelverkehrt gleich eingestellt sein.

Diese Abgleichsregeln sind von ICCP-Standard vorgegeben.

Parameter	Beschreibung
Bilateral Table ID	Die bilaterale Tabellenkennzeichnung wird beim Verbindungsaufbau mit dem Kommunikationspartner abgeglichen. Der Zugriff auf die Daten ist nur für autorisierte entfernte Clients erlaubt. Der Abgleich erfolgt Domain-specific, der Kommunikationspartner muss in der Domäne mit dem Namen **Remote Domain Name** die bilaterale Tabelle mit der Bezeichnung **Bilateral Table ID** aufweisen. Hinweis: Die ID muss auf beiden Seiten der Kommunikation exakt gleichen konfiguriert sein. Hinweis: Fehlende oder fehlerhafte Eingaben werden als Fehlermeldung im Dialog visualisiert.
Local Domain Name	Name der lokalen ICCP-Domäne, die für Objekte des **Process Gateway** ICCP-Server verwendet wird. Hinweis: der Name muss bei den Kommunikationspartnern spiegelverkehrt eingestellt sein - der entfernte Client soll der Name in seinen Anforderungen an **ICCP Process Gateway** verwenden. Hinweis: Fehlende oder fehlerhafte Eingaben werden als Fehlermeldung im Dialog visualisiert.
Remote Domain Name	Name der Domäne am entfernten ICCP-Server. Der **Process Gateway** ICCP-Client verwendet Domain-specific Anforderungen mit diesem Name: ▸ im Abgleich der bilateralen Tabelle; ▸ für die Data Set und TranferSet an entfernten Server. Hinweis: ein Data Set wird Domain-specific angelegt, seine Elemente - die ICCP Datenpunkte - werden VMD-specific definiert. ▸ in Befehlen via **Device objects**. Hinweis: Fehlende oder fehlerhafte Eingaben werden als Fehlermeldung im Dialog visualisiert.
Fehlermeldungen	Anzeige fehlender Konfigurationsdaten.

CONNECTION STATUS

Verbindungsstatusvariablen ermöglichen zur Runtime die Anzeige des aktuellen Verbindungsstatus des **Process Gateways** zum entfernten ICCP-Kommunikationspartner. Im **Process Gateway** werden die zenon Variablennamen angegeben. Die Variablen sollen von numerischen Datentyp sein; Empfehlung: Internvariablen.

Weitere Informationen dazu finden Sie im Kapitel Statusvariablen (auf Seite 144).

Parameter	Beschreibung
Server state variable	Eingabefeld zur Eingabe eines Variablennamens für die Anzeige des aktuellen Verbindungsstatus des Server. Ist dieses Feld leer, können zur zenon Runtime keine Statusinformationen angezeigt werden. Default: `leer`
Client state variable	Eingabefeld zur Eingabe eines Variablennamens für die Anzeige des aktuellen Verbindungsstatus des Client. Ist dieses Feld leer, können zur zenon Runtime keine Statusinformationen angezeigt werden. Default: `leer`

CLIENT

Parameter	Beschreibung
Initialise communication	Checkbox zur Aktivierung des automatischen Verbindungsaufbaus zum entfernten ICCP-Server bei Start des Process Gateways. ▸ `aktiv:` Bei Start des Process Gateways versucht der Client automatisch eine Verbindung zum konfigurierten entfernten ICCP-Server aufzubauen. Schlägt der erste Verbindungsversuch fehl, wird nach 20 Sekunden erneut versucht eine Verbindung herzustellen. Während dieser Wartezeit kann ein Verbindungsversuch von einem entfernten ICCP-Kommunikationspartner angenommen werden, wenn dieser der konfigurierten IP-Adresse oder Hostnamen entspricht. ▸ `inaktiv:` Kein automatischer Verbindungsaufbau zum ICCP-Server bei Start des Process Gateways. Default: `inaktiv` Ist diese Option aktiviert, gilt folgende Einschränkung: ▸ Die Verbindung wird nur mit einem entfernten ICCP-Kommunikationspartner möglich. ▸ Verbindungsversuche von entfernten Clients mit unbekannter Adresse werden nicht angenommen.
IP Address/Hostname	Eingabefeld für IP-Adresse oder Hostname für die automatische Verbindung zum entfernten ICCP-Server.

	Hinweis: Nicht aktiv, wenn **Initialise communication** inaktiv ist. Hinweis: Fehlende oder fehlerhafte Eingaben werden als Fehlermeldung im Dialog visualisiert.
Port	Eingabefeld für Port-Nummer, die bei der Kommunikation zum entfernten ICCP-Server verwendet wird. Default: 102 Hinweis: Nicht aktiv, wenn **Initialise communication** inaktiv ist. Hinweis: Fehlende oder fehlerhafte Eingaben werden als Fehlermeldung im Dialog visualisiert.

Check presence of remote items	Checkbox für die Aktivierung der Überprüfung der Variablennamen der aktuellen Clientverbindung mit den Variablennamen des entfernten ICCP-Servers. Die Namen werden VMD-specific überprüft. ▸ `aktiv:` Aktiviert die Überprüfung und ermöglicht die Projektierung der Eigenschaften **Set result in status bit** und **Variable for result names**. Für die Verbindung liest der Client am entfernen Server ein DataSet ein, welches nur vorhandene Variablen beinhaltet. ▸ `inaktiv:` Es wird davon ausgegangen, dass die projektierten ICCP-Variablen am entfernten Server verfügbar sind. Sind eine oder mehrere projektierte ICCP-Variablen nicht am entfernten Server vorhanden, kann keine erfolgreiche Verbindung zum entfernten Server aufgebaut werden. Default: `inaktiv`
Set result in status bit	Statusbit für zenon Variablen des Clients, die nicht auf dem entfernten ICCP-Server vorhanden sind. Auswahl des entsprechenden Benutzerstatusbits aus Dropdownliste: ▸ `None` Für die Variable wird kein Benutzerstatusbit gesetzt, wenn die Variable am entfernten Server nicht vorhanden ist. Die Auswertung nicht vorhandener ICCP-Variablen kann mit **Variable for result names** erfolgen. ▸ `M1 bis M8` Für die Variable wird das konfigurierte Benutzerstatusbit gesetzt, wenn die Variable am entfernten Server nicht vorhanden ist. Default: `None` Dieses Benutzerstatusbit wird bei konfigurierten zenon Variablen gesetzt, wenn diese bei der Überprüfung auf dem entfernten Server nicht gefunden wurden. Hinweis: Nicht aktiv, wenn **Check presence of remote Items** inaktiv ist. Weitere Informationen zur Statusverarbeitung finden Sie im Handbuch Statusverarbeitung im Kapitel Benutzerstatus.
Variable for result names	Eingabefeld zur Projektierung eines Variablennamens für die Anzeige der Namen der fehlenden ICCP-Variablen.

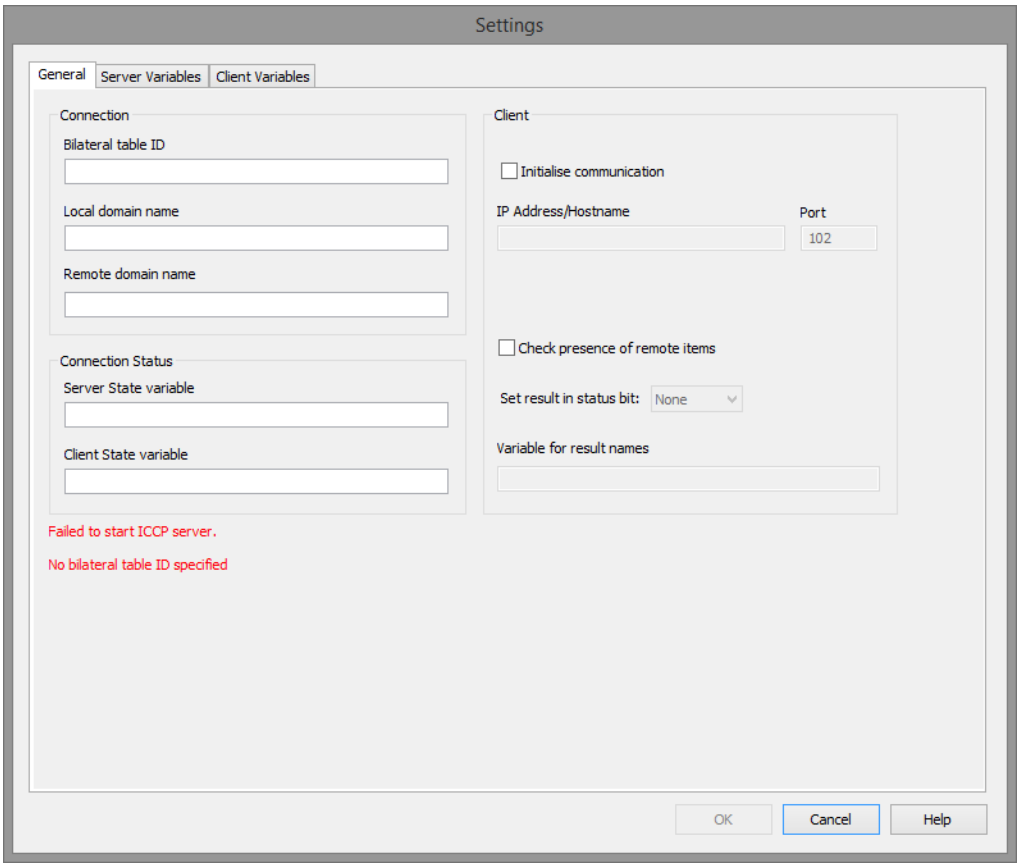
	In der Variable werden Variablennamen visualisiert, die am entfernten Server nicht vorhanden sind. Ist dieses Feld leer, erfolgt die Auswertung mit dem konfigurierten Benutzerstatusbit. Default: `leer` Hinweis: Nicht aktiv, wenn **Check presence of remote Items** inaktiv ist. Weitere Informationen über die Anzeige von nicht vorhandenen Variablen finden Sie im Kapitel Nicht vorhandene Variablen in Runtime anzeigen (auf Seite 145).

DIALOG BEENDEN

Parameter	Beschreibung
OK	Übernimmt Einstellungen aller Registerkarten, speichert die Konfiguration in der Datei **zenProcGateway.ini**, startet den ICCP Server und wartet auf eingehende Verbindungen.
Cancel	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Help	Öffnet die Online-Hilfe.

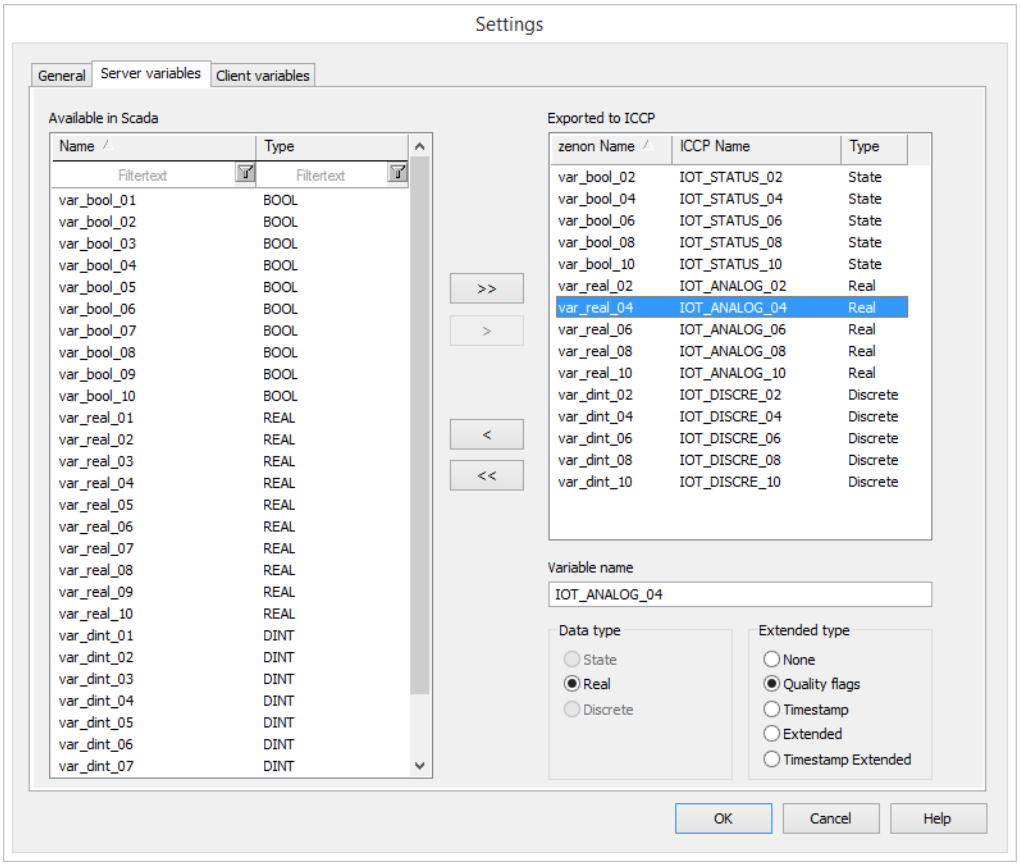
FEHLERMELDUNGEN - BEISPIEL

Fehlermeldungen werden in roter Schrift im Dialog angezeigt:



9.1.2 Server variables

Auswahl und Konfiguration der Variablen, die das zenon Process Gateway als ICCP -Server bereitstellt. Die Wertänderunen der Variablen werden spontan zum entfernten ICCP -Client gesendet.



Parameter	Beschreibung
SCADA variables	Liste mit allem in zenon Projekt verfügbaren numerischen Variablen. Die Liste ist sortier- und filterbar. Die Liste zeigt an: ▸ `Name`: Name in zenon ▸ `Type`: Datentyp der zenon Variable
Pfeiltasten	Via `Pfeil-Taste` werden ausgewählte Variablen zwischen den beiden Listen ausgetauscht. ▸ Taste **>>**: Kopiert alle Variablen von zenon nach ICCP. ▸ Taste **<<**: Entfernt alle Variablen aus Liste ICCP. ▸ Taste **>**: Kopiert ausgewählte Variablen von zenon nach ICCP ▸ Taste **<**: Entfernt ausgewählte Variablen aus Liste ICCP. Variablen können von der Liste für zenon auch über einen Doppelklick auf die Variable zur ICCP-Liste hinzugefügt werden.
Available for ICCP	Variablen aus der Liste werden vom Server für die entfernte ICCP-Client zur Verfügung gestellt. Die Liste zeigt an: ▸ `SCADA Variable`: Name in zenon ▸ `ICCP Name`: Name in ICCP Protokoll ▸ `Type`: ICCP-Type ICCP-Name und Typ können über die Optionen `ICCP Name`, `Data type` und `Extended type` geändert werden.
ICCP Name	Name - VMD-specific - des aus oberer Liste ausgewählten ICCP-Datenpunktes. Änderung des **ICCP Name** durch Eingabe des neuen Namens. Leer, wenn in der Liste **Available for ICCP** kein Eintrag ausgewählt ist. Hinweis: Der ICCP Name kann üblicherweise laut Standard beliebig benannt werden. Es ist jedoch darauf zu achten, dass beide Kommunikationspartner dieselben Namen verwenden.
Data type	ICCP Datentyp der aus oberer Liste ausgewählten Variable.

	Änderung des `Data type` durch Klick auf Optionsfeld: ▸ **State** nicht verfügbar, wenn in der Option **Object type** das Optionsfeld `Device object` aktiviert ist. ▸ **Command** Dieses Optionsfeld steht nur dann zur Verfügung, wenn in der Option **Object type** das Optionsfeld `Device object` aktiviert ist. ▸ **Real** ▸ **Discrete** Bei Übernahme der Variable aus der Liste **SCADA Variables** wird für die Liste **Available from ICCP** automatisch der `Type` vorgeschlagen. Hinweis: Standardmäßige Abbildung von zenon Variablen auf ICCP: ▸ `USINT/BOOL` auf **State**; Wertbereich 0..3, **DPI-Mapping**: • `0/False` - `Off` (`Tripped`), • `1/True` - `On` (`Closed`), • `2` - `Intermediate` (`Between`), • `3` - `Fault` (`Invalid`). ▸ `REAL` auf **Real**. ▸ `DINT/INT/SINT` auf **Discrete**. Abhängig von den von zenon zur Verfügung gestellten Variablen können Datentypen auch nicht zur Verfügung stehen (ausgegraut).
Extended type	ICCP-Zusatz der aus oberer Liste ausgewählten Variable. Änderung des **Extended type** durch Klick auf Optionsfeld: ▸ `None`: kein ICCP-Zusatz (nur Wert der Variable); IEC 60870-6-802 IndicationPoint Typ: Data_Discrete, Data_State, Data_Real. ▸ `Quality flags`: Wert mit Qualität; z.B.: Data_DiscreteQ. ▸ `Timestamp`: Wert + Qualität + Zeitstempel [s]; z.B.: Data_DiscreteQTimeTag. ▸ `Extended`: Wert + Qualität + Zeitstempel [s] + Änderungszähler; z.B.: Data_DiscreteExtended. ▸ `Timestamp Extended`: Wert + Qualität + erweiterter Zeitstempel [ms]; z.B.: Data_DiscreteQTimeTagExtended.

	Abhängig von den von zenon zur Verfügung gestellten Variablen können Datentypen auch nicht zur Verfügung stehen (ausgegraut).

Achtung

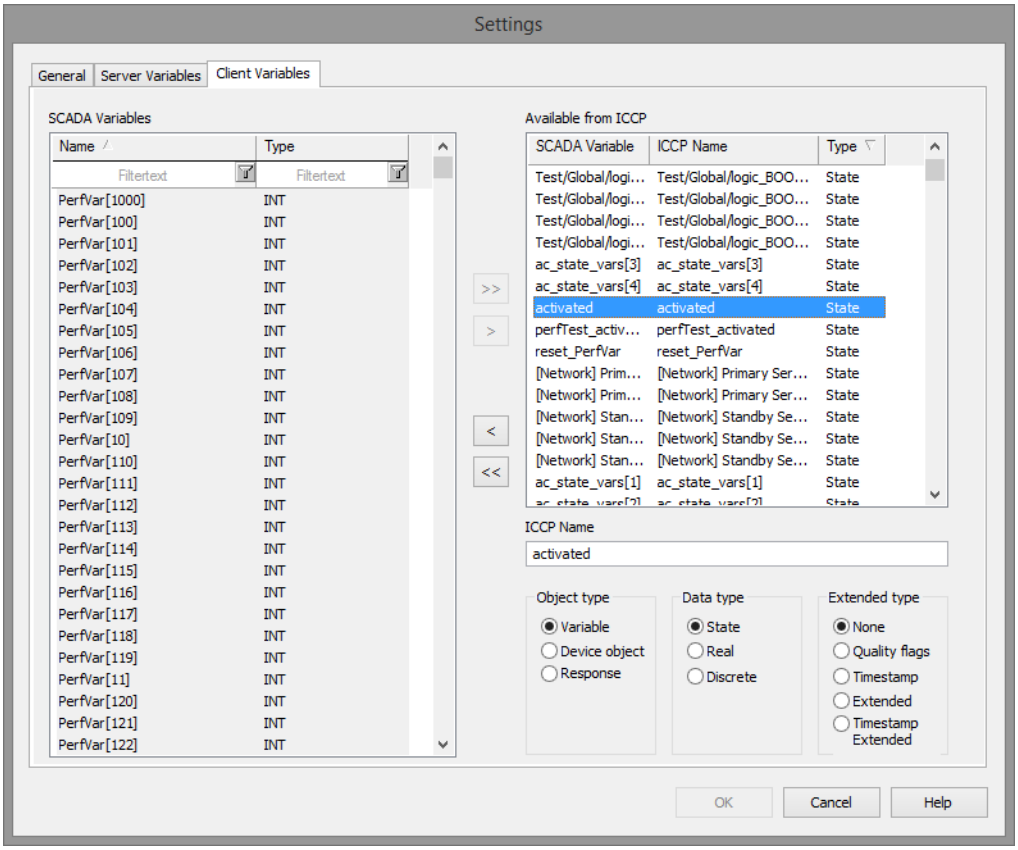
Data type und **Extended type** müssen für beide Kommunikationspartner korrekt zugeordnet sein. Fehlanpassungen führen dazu, dass die empfangenen Datensätze nicht vollständig interpretiert werden. Alle weiteren im Dataset vorhanden Informationen gehen verloren!

DIALOG BEENDEN

Parameter	Beschreibung
OK	Übernimmt Einstellungen aller Registerkarten, speichert die Konfiguration in der Datei **zenProcGateway.ini**, startet den ICCP Server und wartet auf eingehende Verbindungen.
Cancel	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Help	Öffnet die Online-Hilfe.

9.1.3 Client variables

Auswahl und Konfiguration der Variablen, die das zenon Process Gateway als ICCP-Client verwendet. Der Client überträgt auf die zenon Variablen die Wertänderungen, die er vom entfernten ICCP-Server empfängt.



Parameter	Beschreibung
SCADA variables	Liste mit allem in zenon Projekt verfügbaren numerischen Variablen. Die Liste ist sortier- und filterbar. Die Liste zeigt an: ▸ `Name`: Name in zenon ▸ `Type`: Datentyp der zenon Variable
Pfeiltasten	Via `Pfeil-Taste` werden ausgewählte Variablen zwischen den beiden Listen ausgetauscht. ▸ Taste **>>**: Kopiert alle Variablen von zenon nach ICCP. ▸ Taste **<<**: Entfernt alle Variablen aus Liste ICCP. ▸ Taste **>**: Kopiert ausgewählte Variablen von zenon nach ICCP ▸ Taste **<**: Entfernt ausgewählte Variablen aus Liste ICCP. Variablen können von der Liste für zenon auch über einen Doppelklick auf die Variable zur ICCP-Liste hinzugefügt werden.
Available from ICCP	Variablen aus der Liste werden vom entfernten ICCP-Server erwartet. Die Liste zeigt an: ▸ `SCADA Variable`: Name in zenon ▸ `ICCP Name`: Name in ICCP Protokoll ▸ `Type`: ICCP-Type ICCP-Name und Typ können über die Einstellungen **ICCP Name, Data type** und **Extended type** geändert werden.
ICCP Name	Name - VMD-specific - des aus oberer Liste ausgewählten ICCP-Datenpunktes. Änderung des **ICCP Name** durch Eingabe des neuen Namens. Leer, wenn in der Liste **Available for ICCP** kein Eintrag ausgewählt ist. Hinweis: Der ICCP Name kann üblicherweise laut Standard beliebig benannt werden. Es ist jedoch darauf zu achten, dass beide Kommunikationspartner dieselben Namen verwenden.

Parameter	Beschreibung
Object type	Typ der SCADA Variable für die Kommunikation zum ICCP Server. Mit dieser Object können Variablen als Kommando- oder Rückmeldevariable konfiguriert werden. Änderung des `Object type` durch Klick auf Optionsfeld: ▸ `Variable` Standardtyp einer Variable. Aktivieren Sie diese Option, wenn die zenon Variable keine Kommando- oder Rückmeldevariable ist. ▸ `Device object` Die Variable aus der Liste **SCADA Variables** wird bei aktivierter Option als Kommandovariable konfiguriert. Dabei sendet das Process Gateway bei Empfang eines Spontanwerts ein `Operate Request` an den ICCP Server. Durch Aktivierung dieses Optionsfeldes wechselt in der Option **Data type** das Optionsfeld `State` die Auswahlmöglichkeit in `Command`. ▸ `Response` Die Variable aus der Liste **SCADA Variables** wird bei aktivierter Option als Rückmeldevariable konfiguriert. Bei Empfang eines Spontanwerts wird die Antwort des Servers (`Indication`) via dem Befehl `operate` auf die Rückmeldevariable geschrieben. Für diesen Prozess sind keine weiteren Projektierungen notwendig. Default: `Variable`

Parameter	Beschreibung
Data type	ICCP Datentyp der aus oberer Liste ausgewählten Variable. Änderung des `Data type` durch Klick auf Optionsfeld: ▸ **State** nicht verfügbar, wenn in der Option **Object type** das Optionsfeld `Device object` aktiviert ist. ▸ **Command** Dieses Optionsfeld steht nur dann zur Verfügung, wenn in der Option **Object type** das Optionsfeld `Device object` aktiviert ist. ▸ **Real** ▸ **Discrete** Bei Übernahme der Variable aus der Liste **SCADA Variables** wird für die Liste **Available from ICCP** automatisch der `Type` vorgeschlagen. Hinweis: Standardmäßige Abbildung von zenon Variablen auf ICCP: ▸ `USINT/BOOL` auf **State**; Wertbereich 0..3, **DPI-Mapping**: • `0/False` - `Off` (`Tripped`), • `1/True` - `On` (`Closed`), • `2` - `Intermediate` (`Between`), • `3` - `Fault` (`Invalid`). ▸ `REAL` auf **Real**. ▸ `DINT/INT/SINT` auf **Discrete**. Abhängig von den von zenon zur Verfügung gestellten Variablen können Datentypen auch nicht zur Verfügung stehen (ausgegraut). Hinweis: Ausgeblendet, wenn bei der Option **Object Type** das Optionsfeld `Response` aktiviert ist.

Parameter	Beschreibung
Extended type	ICCP-Zusatz der aus oberer Liste ausgewählten Variable. Änderung des **Extended type** durch Klick auf Optionsfeld: ▸ `None`: kein ICCP-Zusatz (nur Wert der Variable); IEC 60870-6-802 IndicationPoint Typ: Data_Discrete, Data_State, Data_Real. ▸ `Quality flags`: Wert mit Qualität; z.B.: Data_DiscreteQ. ▸ `Timestamp`: Wert + Qualität + Zeitstempel [s]; z.B.: Data_DiscreteQTimeTag. ▸ `Extended`: Wert + Qualität + Zeitstempel [s] + Änderungszähler; z.B.: Data_DiscreteExtended. ▸ `Timestamp Extended`: Wert + Qualität + erweiterter Zeitstempel [ms]; z.B.: Data_DiscreteQTimeTagExtended. Abhängig von den von zenon zur Verfügung gestellten Variablen können Datentypen auch nicht zur Verfügung stehen (ausgegraut). Hinweis: Ausgeblendet, wenn bei der Option **Object Type** entweder das Obptionsfeld `Device object` oder `Response` aktiviert ist.

Achtung

Data type und **Extended type** müssen für beide Kommunikationspartner korrekt zugeordnet sein. Fehlanpassungen führen dazu, dass die empfangenen Datensätze nicht vollständig interpretiert werden. Alle weiteren im Dataset vorhanden Informationen gehen verloren!

DIALOG BEENDEN

Parameter	Beschreibung
OK	Übernimmt Einstellungen aller Registerkarten, speichert die Konfiguration in der Datei **zenProcGateway.ini**, startet den ICCP Server und wartet auf eingehende Verbindungen.
Cancel	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Help	Öffnet die Online-Hilfe.

9.2 Statusvariablen der Verbindung

Statusvariablen der Kommunikation werden für die Darstellung des Verbindungsstatus zwischen Process Gateway und entfernten ICCP-Kommunikatitonspartner verwendet. Für ICCP-Server sowie ICCP-Client wird jeweils eine eigene Variable projektiert. Das Process Gateway kommuniziert diese Status ausschließlich zur zenon Runtime. Eine Kommunikation von Werten dieser Variablen zu entfernten ICCP-Kommunikationspartner findet nicht statt.

PROJEKTIERUNG

Um den Verbindungsstatus in der Runtime anzuzeigen, führen Sie folgende Projektierungsschritte aus:

1. Erstellen Sie in Ihrem zenon Editor zwei numerische Variablen.
   Es wird empfohlen, diese für den Internen Treiber zu projektieren.
   - Die Variable darf nicht vom Datentyp STRING sein.
2. Konfiguration im Process Gateway:
   - Statusvariable für Serververbindung:
     Tragen Sie den Namen der Variable im Eingabefeld **Server state variable** in der Registerkarte **General** in der Process Gateway Modulkonfiguration ein.
   - Statusvariable für Clientverbindung:
     Tragen Sie den Namen der Variable im Eingabefeld **Client state variable** in der Registerkarte **General** in der Process Gateway Modulkonfiguration ein.

ZUGEWIESENE ZUSTÄNDE

Die Verbindungsstatus-Variablen haben folgende Werte:

Parameter	Beschreibung
0 IDLE	▸ Kommunikation inaktiv: unmittelbar nach Start des Process Gateways oder Nachladen des zenon Projekts. ▸ Verbindung wieder abgebaut: nach Beendigung des Process Gateways oder nachdem die Einstellungen im Process Gateway bearbeitet wurden und vor dem Aufbau der neuen Verbindung.
1 CONNECTING	Kommunikation via TCP/IP und MMS wird gerade aufgebaut. Dieser Status wird angezeigt, wenn ein eingehender CR (`Connection Request`) empfangen wurde oder wenn ein CR (`Connection Request`) erfolgreich gesendet wurde. Dies ist abhängig davon, welche Seite die TCP/IP Verbindung aufgebaut hat. Dieser Status ist für Server und Client gleich.
2 CONNECTED	ICCP-Kommunikation wurde erfolgreich aufgebaut. Dieser Zustand wird auf einer logischen Verbindung gesetzt, wenn erfolgreich ein Dataset (`DSTransferSet`) für die Übermittlung von Daten vereinbart wurde und der erste `MMS.write request` (logische Serververbindung) bzw. der erste `MMS.write response` (logische Clientverbindung) empfangen wurde. Wenn der ICCP-Client keine Daten vom entfernten Server anfordert, bleibt der Status der logischen Verbindung auf `CONNECTED`. Hinweis: Der Client fordert keine Daten an, wenn: ▸ Die Liste der Variablen **Available from ICCP** leer ist. ▸ Keine der Variablen der Liste **Available from ICCP** am entfernten Server verfügbar ist und die Überprüfung **Check presence of remote items** aktiviert wurde.
3 ERROR	Dieser Status wird angezeigt, wenn die TCP/IP Verbindung unterbrochen wird. Dieser Status ist für beide logische Verbindungen gleich.

9.3 Nicht vorhandene Variablen in Runtime anzeigen

Mit Hilfe dieser Projektierung können Sie in der zenon Runtime auswerten, welche der im ICCP-Client projektierten Variablen am entfernten Server nicht vorhanden sind.

Diese Auswertung kann mit zwei Projektierungen erfolgen:

1. Anzeige der fehlenden Variablennamen in einer zusätzlichen String-Variable.

2. Anzeige eines Benutzerstatusbits, bei nicht vorhandenen Variablen.

VORAUSSETZUNG

- Die Überprüfung der Variablennamen der aktuellen Clientverbindung mit dem Server muss aktiviert sein:
  Checkbox **Check presence of remote items** in der Registerkarte **General** in der Process Gateway Modulkonfiguration.

ANZEIGE DER FEHLENDEN VARIABLENNAMEN IN EINER STRING-VARIABLE

Um die fehlenden Variablen zur Runtime in einer Variable anzuzeigen:

1. Erstellen Sie im zenon Editor eine Variable vom Datentyp STRING.
   Es wird empfohlen, diese für den Internen Treiber zu projektieren.
2. Konfiguration im Process Gateway:
   Tragen Sie den Namen der Variable im Eingabefeld **Variable for result names** in der Registerkarte **General** in der Process Gateway Modulkonfiguration ein.

> **Info**
>
> Der Wert der Variable zur Runtime beinhaltet die Namen der nicht vorhandenen Variablen, getrennt durch einen Strichpunkt (;).

ANZEIGE EINES BENUTZERSTATUSBITS, BEI NICHT VORHANDENEN VARIABLEN

Um die zenon Variablen zur Runtime mit einem Benutzerstatusbit zu markieren, wenn sie nicht am entfernten ICCP-Server vorhanden sind:

- Wählen Sie das gewünschte Benutzerstatusbit aus der Dropdownliste der Eigenschaft **Set result in status bit** in der Registerkarte **General** in der Process Gateway Modulkonfiguration.

Hinweis: Dieses Benutzerstatusbit kann zur RUNTIME mit einer **Reaktionsmatrix** oder einem **Combi-Element** ausgewertet werden.

9.4 Direct control (non SBO) für ICCP control objects

- Eine zenon Variable kann als Kommandovariable mit einem `ICCP control object` verknüpft werden. Bei Empfang eines Spontanwerts für diese Kommandovariable wird ein `Direct`

`Control` ausgeführt. Dabei sendet das Process Gateway ein `Operate Request` an den ICCP Server.

- Eine zenon Variable kann als Rückmeldevariable mit einem `ICCP control object` verknüpft werden. Das Ergebnis der `Direct Control` Operation (= `indication`) wird auf die projektierte zenon Ergebnisvariable (via Sollwertsetzen) geschrieben.
- Jede empfangene `indication` wird auf die entsprechende ICCP Rückmeldevariable geschrieben.
  - Dabei erfolgt keine Prüfung ob ein Kommando ausständig ist.
  - Wird keine `indication` empfangen bleibt der Wert der Rückmeldevariable unverändert.
- `direct control` - unterstützte Telegrammtypen
  - `Operate (request)`
  - `Timeout (indication)`
  - `Local Reset (indication)`
  - `Success (indication)`
  - `Failure (indication)`
- Vor dem Senden eines neuen Kommandos wird die Rückmeldevariable auf einen festgelegten Wert gesetzt.
- Bei der Initialisierung des Gateways wird die Konfiguration validiert:
  - Sind pro `device object` mehrere Rückmeldevariablen konfiguriert, wird beim Starten des **Process Gateway**s eine Fehlermeldung erzeugt und nur die erste verwendet
  - Pro `device object` darf nur eine Kommandovariable konfiguriert sein.
  - Für jede Rückmeldevariable muss eine Kommandovariable vorhanden sein, sonst Fehlermeldung.

PROJEKTIERUNG IM PROCESS GATEWAY

- Öffnen Sie im **Process Gateway** die Registerkarte **Client Variables**
- Kommandovariable projektieren
- Rückmeldevariable projektieren

  Ein Response vom ICCP-Server wird einem **Device object** oder einer Variabe zugeordnet, indem er den gleichen ICCP Namen bekommt.

ERGEBNIS DER SCHREIBANFRAGE (WRITE REQUEST)

Ist zu einer Variable oder einem **Device Object** ein **Response Object** konfiguriert, so wird das Ergebnis des `write Requests` auf dieses Objekt geschrieben:

Parameter	Beschreibung
0	Write successful
1	Write pending - waiting for response
200	Write failed: OBJECT_INVALIDATED
201	Write failed: HW_FAULT
202	Write failed: TEMPORARILY_UNAVAILABLE
203	Write failed: OBJECT_ACCESS_DENIED
204	Write failed: OBJECT_UNDEFINED
205	Write failed: INVALID_ADDRESS
206	Write failed: TYPE_UNSUPPORTED
207	Write failed: TYPE_INCONSISTENT
208	Write failed: OBJECT_ATTRIBUTE_INCONSISTENT
209	Write failed: OBJECT_ACCESS_UNSUPPORTED
210	Write failed: OBJECT_NON_EXISTENT
211	Write failed: OBJECT_VALUE_INVALID

9.5 Konfigurationsdatei: INI-Einträge für ICCP-Tase.2

Die Konfigurationsdatei `zenProcGateway.ini` enthält folgende Einträge für ICCP-Tase.2:

[ICCP]

Eintrag	Beschreibung
SERVER_PORT=	Nummer des IP-Ports des Servers; der Port erlaubt die Verbindung von entfernten Clients. Der Port wird beim Start geöffnet und bleibt bestehen. Default: 102 Hinweis: Dieser INI-Eintrag kann nicht in der grafischen Benutzeroberfläche für das ICCP-TASE.2 Process Gateway konfiguriert werden.
MAX_CONNECTIONS=	Maximale Anzahl der Verbindungen von entfernten ICCP-Clients zum Server. Wird ignoriert wenn **CLIENT_AUTO_CONNECT**=1 konfiguriert wurde. Default: 0 (= uneingeschränkt) Hinweis: Dieser INI-Eintrag kann nicht in der grafischen Benutzeroberfläche für das ICCP-TASE.2 Process Gateway konfiguriert werden.
MAJOR_VERSION_NUMBER=	Hauptversionsnummer des ICCP Protokolls, z.B.: 1996, 2000. Default: 2000 Hinweis: Dieser INI-Eintrag kann nicht in der grafischen Benutzeroberfläche für das ICCP-TASE.2 Process Gateway konfiguriert werden.
MINOR_VERSION_NUMBER=	Unterversionsnummer des ICCP Protokolls. Default: 8 Hinweis: Dieser INI-Eintrag kann nicht in der grafischen Benutzeroberfläche für das ICCP-TASE.2 Process Gateway konfiguriert werden.
BILATERAL_TABLE_ID=	Bilaterale Tabellenkennzeichnung. Diese wird beim Verbindungsaufbau zwischen allen ICCP-Kommunikationspartnern abgeglichen. Hinweis: Die eindeutige bilaterale Tabellenkennzeichnung muss auf allen Geräten, die mit dem **ICCP-TASE.2 Process Gateway** kommunizieren, gleich sein.
LOCAL_DOMAIN_NAME=	Lokaler ICCP-Domänenname; Einstellung für Server, soll spiegelverkehrt zum Kommunikationspartner eingestellt sein.
REMOTE_DOMAIN_NAME=	Remote ICCP-Domänenname; Einstellung für Client, soll spiegelverkehrt zum Kommunikationspartner eingestellt sein.

CLIENT_AUTO_CONNECT=	Projektierung, ob bei Start des Process Gateways der Client automatisch den Verbindungsaufbau zum entfernten ICCP-Server initialisiert. 0: nicht initialisieren - nur warten 1: initialisieren
CLIENT_IP_NAME=	IP-Adresse oder Hostname des entfernten Server, wenn dieser Client die Kommunikation initialisieren soll.
CLIENT_PORT=	Port-Nummer des entfernten Server, wenn dieser Client die Kommunikation initialisieren soll.
CHECK_SERVER_Variables=	Überprüfung ob alle konfigurierten Variablen auf dem entfernten Server zur Verfügung stehen. 0: keine Überprüfung von projektierten Variablennamen. 1: Überprüfung von projektierte Variablennamen.
CHECK_SERVER_VARIABLES_MERKER_BIT=	Statusbit für Client-Variablen, die bei aktivierter Überprüfung der projektierten Variablen nicht am Server vorhanden sind. 0: kein Statusbit für fehlende Client-Variablen 1 bis 8: Benutzerstatusbit M1 bis M8
SERVER_STATE_VARIABLE=	Name der Variable für die Anzeige des Verbindungsstatus zwischen ICCP-Server und entferntem Client.
CLIENT_STATE_VARIABLE=	Name der Variable für die Anzeige des Verbindungsstatus zwischen ICCP-Client und entferntem Server.
NOT_AVAILABLE_SERVER_VARIABLES_VARIABLE=	Name der Variable für die Anzeige von Variablennamen (in der zenon Runtime), welche für die aktuelle Clientverbindung nicht am entfernten Server verfügbar sind.
CALLING_OSI_SESSION_SELECTOR= **CALLING_OSI_PRESENTATION_SELECTOR=** **CALLING_OSI_AE_QUALIFIER=** **CALLING_OSI_AP_TITLE=** **CALLED_OSI_SESSION_SELECTOR=** **CALLED_OSI_PRESENTATION_SELECTOR=** **CALLED_OSI_AE_QUALIFIER=** **CALLED_OSI_AP_TITLE=**	OSI-Kommunikationsparameter Hinweis: Dieser INI-Eintrag kann nicht in der grafischen Benutzeroberfläche für das ICCP-TASE.2 Process Gateway konfiguriert werden.

[VARIABLES]

Gruppe für konfigurierte Bezüge zwischen zenon Variablen und ICCP-Variablen. Gelistet werden:

- Parameter jeder ICCP-Variablen
- Anzahl der Variablen
- Namen der Variablen

Eintrag	Beschreibung
PARAM_0=	Parameter für ICCP-Name. Die Nummerierung beginnt mit `0`. Mit jedem weiteren Parameter wird der Zähler um 1 erhöht. Dieser Bezeichner entspricht dem Bezeichner des Variablennamens. **PARAM_0** entspricht **NAME_0**. Beispiel: **PARAM_0**=`1,0,0,ICCP_Name`
COUNT=	Zahl der konfigurierten Bezüge
NAME_0=	Name der zenon Variablen. Die Nummerierung beginnt mit `0`. Mit jedem weiteren Parameter wird der Zähler um 1 erhöht. Dieser Bezeichner entspricht dem Parameterbezeichner. **PARAM_0** entspricht **NAME_0**. Beispiel: **NAME_0**=`MyName`

9.6 LOG-Einträge

GENERAL

Parameter	Level	Hinweis
`Failed to get %s application object.`	**ERROR**	Starten Sie die zenon Runtime bevor Sie das Process Gateway starten.
`Failed to start ICCP server.`	**ERROR**	Korrigieren Sie die Einstellungen des Process Gateways. Konfigurieren Sie beim Starten von mehreren Process Gateways für jedes Process Gateway einen anderen Port.
▸ `Failed to interpret report for data set %s.%s. Bad message structure.` ▸ `Failed to interpret report for %s. Bad message structure.` ▸ `Failed to interpret write request for %s.%s. Bad message structure.` ▸ `Failed to interpret received data for %s.`	**ERROR**	Stimmen Sie die ICCP '**Data type**' und '**Extended type**' zwischen Process Gateway und entfernten ICCP-Kommunikationspartner ab.
`Failed to write '%s', error code: %u.`	**ERROR**	Der entfernte ICCP-Server antwortet die Schreibanforderung negativ, mit MMS Errorcode %u. Korrigieren Sie die Einstellungen am entfernten Server u/o schieben Sie im Process Gateway die Variable von Clientkonfiguration zum Servervariablen um.

AUTOMATISCHER VERBINDUNGSAUFBAU

Parameter	Level	Beschreibung
`Could not resolve host name %s. Errorcode: %d`	**ERROR**	Der voreingestellte Hostname oder die IP-Adresse konnten nicht aufgelöst werden. ▸ `%1` Hostname oder IP-Adresse der eingestellten Gegenstelle. ▸ `%2` Windows Socket Fehlercode

Autoconnection is not possible because no hostname (or IP address) is defined	**ERROR**	Der voreingestellte Hostname oder die IP-Adresse werden zur Adresse 0.0.0.0 aufgelöst und sind somit ungültig.
Maximum number of %1 connections reached. Declining new connection from %2.	**MSG**	Zurückweisung einer eingehenden Verbindung, da bereits die maximale Anzahl von möglichen Verbindungen erreicht ist: ▸ %1 maximale Anzahl von Verbindungen, die hergestellt werden können. ▸ %2 IPv4-Adresse der Gegenstelle.
Cannot verify selected host name %1. Declining new connection from %2.	**MSG**	Der konfigurierte Hostname kann nicht auf die IP Adresse aufgelöst werden. ▸ %1 konfigurierter Hostname ▸ %2 IPv4-Adresse der Gegenstelle
Only connection from %1 allowed. Declining new connection from %2.	**MSG**	Eine eingehende Verbindung wird zurückgewiesen, da die Quelladresse nicht der IP-Adresse des konfigurierten Hosts für den automatischen Verbindungsaufbau entspricht. ▸ %1 konfigurierte IP Adresse des eingestellten Hosts. ▸ %2 IPv4-Adresse der Gegenstelle.
Only one active connection allowed. Declining new connection from %2.	**MSG**	Eine eingehende Verbindung wird zurückgewiesen, da eine Verbindung bereits automatisch hergestellt wurde. ▸ %2 IPv4-Adresse der Gegenstelle.
Creating a socket connection to the ICCP server %s [%2:%3].	**DEBUG**	Eine TCP Socketverbindung zur konfigurierten Gegenstelle wird hergestellt. ▸ %1 Hostname oder IP-Adresse der eingestellten Gegenstelle. ▸ %2 IPv4-Adresse der Gegenstelle. ▸ %3 Portnummer der Gegenstelle.
Could not create the socket connection to the ICCP server %1. Errorcode: %2.	**DEBUG**	Eine TCP Socketverbindung zur eingestellten Gegenstelle konnte nicht innerhalb des Sockettimeouts hergestellt werden. ▸ %1

		Hostname oder IP-Adresse der eingestellten Gegenstelle. ▸ %2 Windows Socket Fehlercode.

STATUSVARIABLEN DER VERBINDUNG

Parameter	Level	Beschreibung
Server state variable with name '%s' does not exist or has an invalid type.	**ERROR**	Die konfigurierte Statusvariable zur Anzeige des Status der Serververbindung existiert im Projekt nicht oder hat einen ungültigen Datentyp (erforderlicher Datentyp: Numerisch). ▸ %s Projektierter Name der Statusvariable.
Client state variable with name '%s' does not exist or has an invalid type.	**ERROR**	Die konfigurierte Statusvariable zur Anzeige des Status der Clientverbindung existiert im Projekt nicht oder hat einen ungültigen Datentyp (erforderlicher Datentyp: Numerisch). ▸ %s Projektierter Name der Statusvariable.
set server state: %1(%2)	**MSG**	Neuer Status einer bereits bestehenden Serververbindung. ▸ %1 numerischer Wert des aktuellen Status der Serververbindung ▸ %2 aktueller Status der Serververbindung als Text (IDLE, CONNECTING, CONNECTED, ERROR)
set client state: %1(%2)	**MSG**	Neuer Status einer bereits bestehenden Clientverbindung. ▸ %1 numerischer Wert des aktuellen Status der Clientverbindung ▸ %2 aktueller Status der Clientverbindung als Text (IDLE, CONNECTING, CONNECTED, ERROR)

ANZEIGE NICHT VORHANDENEN VARIABLEN

Parameter	Level	Beschreibung
Unavailable server items variable with name '%1' does not exist	**ERROR**	Die konfigurierte Statusvariable zur Anzeige der Namen der nicht existierenden Variablen existiert

or has an invalid type.		im Projekt nicht oder hat einen ungültigen Datentyp (erforderlicher Datentyp: STRING). ▸ %1 Projektierter Name der Statusvariable.
Error configured client variable '%1' is not available on the remote device.	**ERROR**	Eine im Process Gateway für den Client projektierte Variable ist nicht am entfernten ICCP-Server verfügbar. ▸ %1 ICCP Name der Variable, die am entfernten Server nicht vorhanden ist

Parameter	Level	Beschreibung
Response variable [iccp name] already configured.	**ERROR**	Fehlermeldung bei Fehlprojektierung im Process Gateway. Bei dieser Fehlermeldung wurden einem ICCP Client mehr als eine SCADA Variable zugeordnet. Das erste Response Object wird übernommen, alle weiteren Response Objects werden verworfen.
No ICCP object found for configured response variable [iccp name]	**ERROR**	Fehlermeldung bei Fehlprojektierung im Process Gateway. Bei dieser Fehlermeldung wurde einem ICCP Client eine SCADA Variable für ein Response Object zugeordnet, allerdings fehlt eine Zuordnung für ein passendes Device object.

9.7 Interoperability

Das Process Gateway für ICCP-TASE.2 basiert auf dem **IEC 60870-6 TASE.2 ICCP** Protokoll.

Unterstützte Protokollversionen: 2000.8 und 1996.8.

ICCP-TASE.2 unterstützt Conformance Blocks 1 and 2 als Client und Server. Als Client wird Direct Control von Block 5 unterstützt.

Per default verhält sich das Process Gateway beim Aufbau der Kommunikation wie ein Server.

10. IEC870 Slave

SPRACHE

Das IEC870Sl Process Gateway und seine Dialoge zur Konfiguration stehen ausschließlich in Englisch zur Verfügung.

STRUKTUR

In dem IEC870 Slave Gateway können Sie mehrere virtuelle Steuerungen (**Devices**) konfigurieren. Je nach Protokoll sind diese virtuelle Geräte direkt einem Master zugeordnet (870-104) oder werden von einem Master abgefragt (870-101). Die **Devices** können mehrere Sektoren beinhalten. Diese enthalten IOs (Information Objects), welche die eigentlichen Variablen repräsentieren. Ein IO bezieht sich auf eine Variable des Prozessleitsystems.

BETRIEB VON MEHREREN IEC870 SLAVES

Das **IEC870Sl Process Gateway** kann sowohl auf einem Einzelplatz-Rechner wie auch im zenon **Netzwerk** (Server/Standby/Client) gestartet werden. Alle Process Gateways, die im Netzwerk gestartet sind, liefern das gleiche Prozessabbild, jedoch mit dem Umfang, wie er in der Konfigurationsdatei angegeben ist.

Netzwerkserver sind meist Rechner mit hoher Auslastung. In umfangreichen und aufwendigen Systemen kann es ratsam sein, die Gateways auf (dedizierten) Netzwerkclients zu betreiben. Dies hat einen weiteren Vorteil: bei Redundanzumschaltung verbindet sich ein Netzwerk Client automatisch zum jeweiligen prozessführenden Server und kann weiterhin die Daten an die übergeordnete Systeme liefern.

Mehre Instanzen des IEC870 Slaves können die gleiche, oder sehr ähnliche Konfiguration nutzen, die auf mehrere Dateien übertragen wurden, z.B. mit Export/Import (auf Seite 195).

Es ist auch möglich mehrere Instanzen des IEC870 Slaves auf einen Rechner zu starten. Voraussetzung: eigene Kommunikationsparameter für jede Instanz - keine Konflikte bei IP- oder COM-Ports.

REDUNDANZ DES IEC870 SLAVES

Wenn zwei Process Gateways mit der gleichen Konfiguration gestartet wurden dann sind sie für einen 870-Master wie ein redundanter Slave, also eine redundante Verbindung zu zenon System.

Zwei Process Gateways mit gleicher Konfiguration können auf zwei Rechnern im zenon **Netzwerk**, oder auf einen Rechner mit 2 Ethernet Karten, oder mit unterschiedlichen IP Ports gestartet werden.

In Melderichtung liefern beide IEC870 Slaves die gleichen Events und Messwerte, da die zenon Runtime exakt das gleiche Prozesabbild zur Verfügung stellt. In Befehlsrichtung kann über ein IO mit dem internen Typ T00 und IOA 2 (auf Seite 214) einer der zwei Slaves in `silent mode` versetzt werden. Somit werden die Befehle des 870-Masters nur über eine Verbindung an die Runtime weitergeleitet.

Die Redundanz auf Data Layer - nach den Prozeduren der Edition 2 des IEC 60870 Standards, wird nicht unterstützt.

IDENTIFIZIERUNG UND ADRESSIERUNG

Pro jeweilige gestartete Instanz des IEC870Sl Process Gateways:

- 870-101: Die virtuelle Steuerung (**Device**) wird durch die Link-Adresse des Slaves identifiziert.
- 870-104: Die virtuelle Steuerung wird durch die IP-Adresse des Masters identifiziert.
- Sektor: Wird durch **COA** (Common Object Address/Common Address of ASDU) adressiert.
- IO: Wird durch **Type Identification** und **IOA** adressiert.

In den folgenden Kapiteln lesen Sie, wie Sie den IEC870SI Gateway konfigurieren und wie die Kommunikation abläuft.

> **Info**
>
> *Bei Variablen, die im IEC870Sl Process Gateway als Befehlsvariable konfiguriert sind, wird bei aktiver Eigenschaft **Sollwert setzen über VBA** (Gruppe **Protokollierung in CEL**) jeder Befehl als **Sollwert Absetzen** über die API-Schnittstelle zu Runtime in der CEL protokolliert.*

10.1 Modulkonfiguration

Im IEC870 Slave Process Gateway stehen Ihnen viele Einstellungsmöglichkeiten zur Verfügung. Die Konfiguration des IEC870SI wird als XML-Datei (auf Seite 195) abgespeichert.

In den Konfigurationsdialogen sind die Listen generell sortierbar. Um die Sortierreihenfolge zu ändern, klicken Sie einfach auf den Spaltentitel.

PROTOKOLL 870-101 VS 104

Das Gateway unterstützt sowohl Protokoll IEC 60870-5-101 (kurz: 870-101, serielle Kommunikation) wie IEC 60870-5-104 (kurz: 870-104, TCP/IP Kommunikation). Die Einstellungen für die Protokolle -101 und -104 unterscheiden sich.

Im Hauptfenster **Settings** entscheiden Sie, welches Protokoll Sie nutzen möchten und legen Sie allgemeine Einstellungen fest:

- (Protocol) 870-101 (auf Seite 161)
  gibt die Einstellungen für das Protokoll 101 frei und sperrt andere, nicht relevante, Eingabemöglichkeiten.
- (Protocol) 870-104
  gibt die Einstellungen für das Protokoll 104 frei und sperrt andere, nicht relevante, Eingabemöglichkeiten.

DEVICES

In der Liste **Devices** des Hauptfensters werden virtuelle Geräte (Verbindungen) angelegt und gelöscht. Ein **Device** definiert wie das Process Gateway als IEC870 Slave für einen bestimmten 870-Master funktionieren soll. Die Konfiguration dieser **Devices** erfolgt in den Registerkarten. Es können mehre Geräte eingelegt werden - für mehre Master.

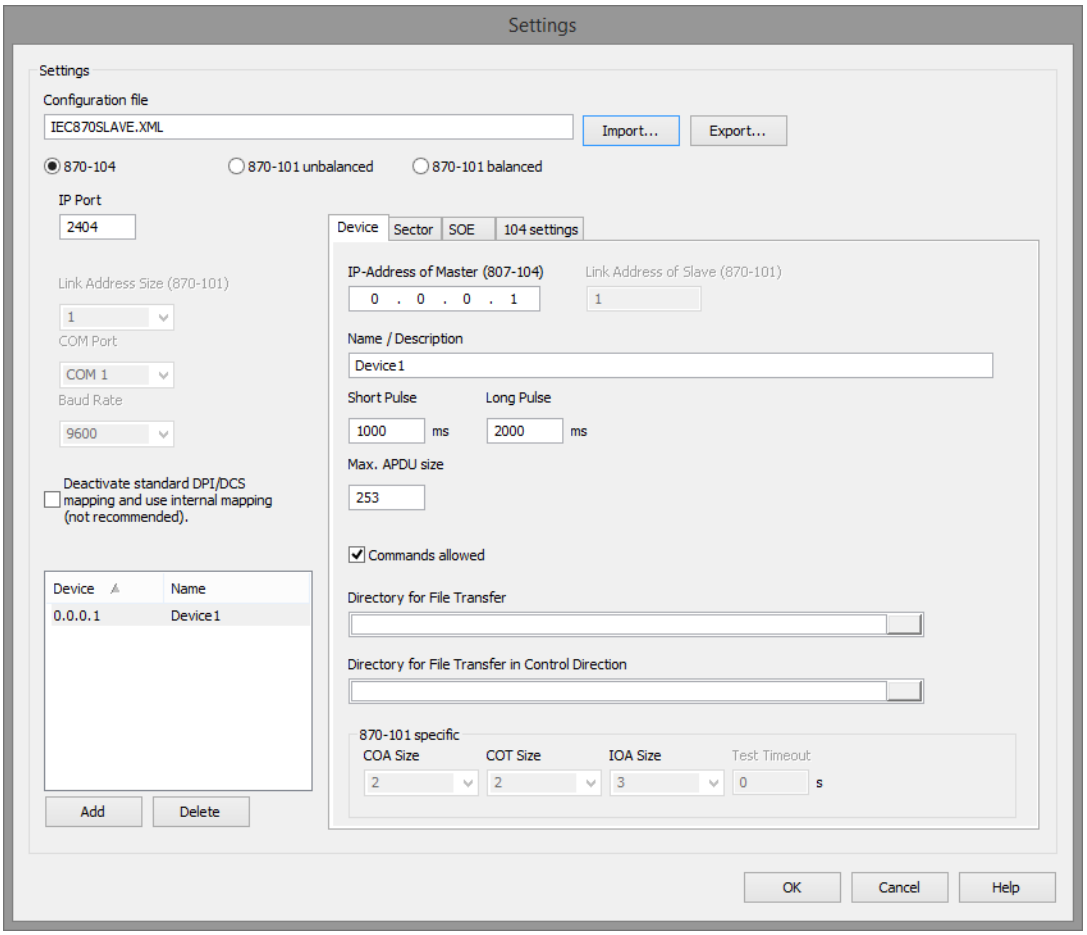
Folgende Registerkarten stehen im Process Gateway zur Verfügung:

- **Device** (auf Seite 166) - generelle Einstellungen sowie Master IP Adresse (-104) oder Link Adresse des Slaves (-101).
- **Sector** (auf Seite 170) - Einstellungen gültig für ein Common Address of ASDU (**COA**). Ein Device soll wenigstens ein Sektor beinhalten; und ein Sektor - wenigsten ein Information Object (**IO**):
  - **Information Objects** - die Liste der Information Object Address (**IOA**), deren Type Identifications (**TID**) und die Zuweisung zu zenon Variablen.
- **Sequences of Events (SOE (auf Seite 205))**
  SOE nach Prozeduren definiert in IEC 60870-5-101: 7.4.11.3 "Transmission of sequences of events" (in a section of a data file).
- **104 settings** (auf Seite 192)
  Data Layer Timeouts und APDU Zähler

Info

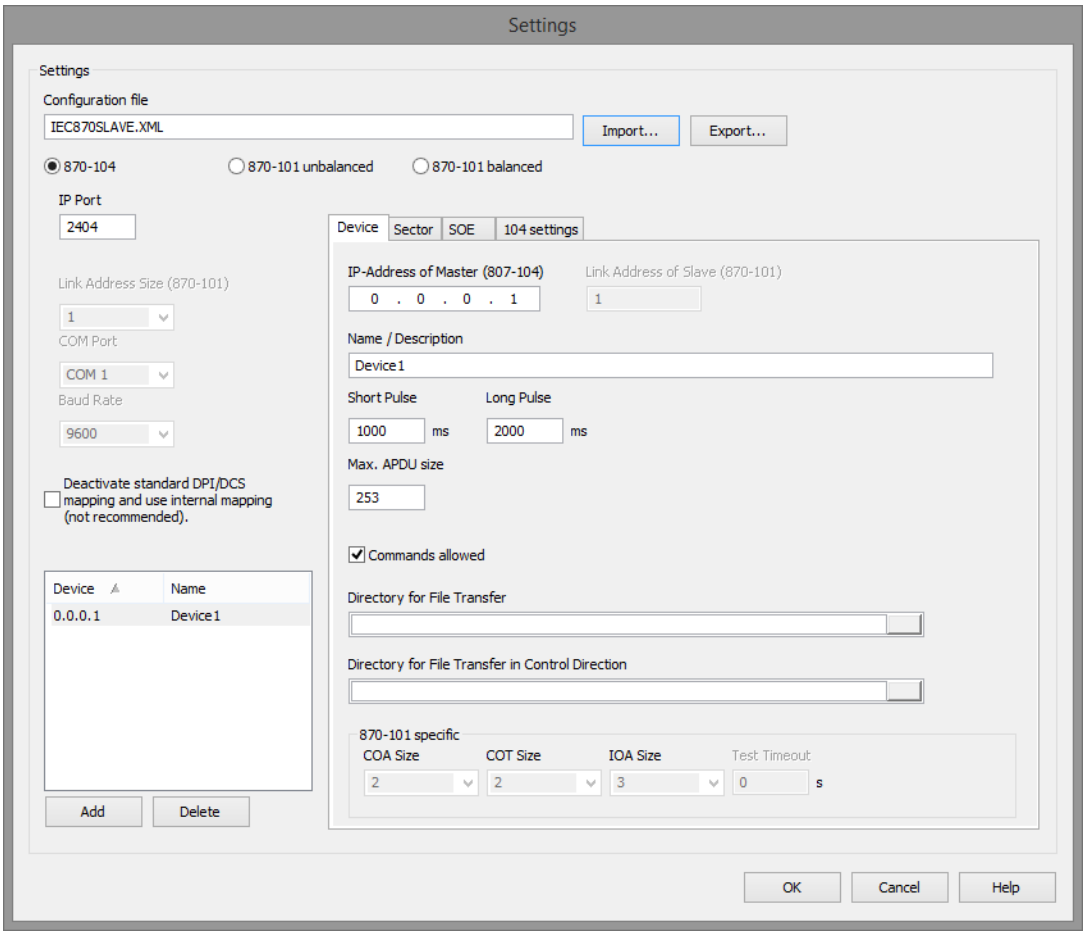
In diesem Abschnitt sind nur die allgemeinen Eigenschaften dokumentiert. Die protokollabhängigen Konfigurationen finden Sie im Abschnitt des jeweiligen Protokolles (101) oder in der Beschreibung der jeweiligen Geräteeinstellungen (101 und 104).

REGISTERKARTEN FÜR DIE KONFIGURATION DER DEVICES



Parameter	Beschreibung
Device	Registerkarte für die Konfiguration eines Gerätes (auf Seite 166) (Device). Ein virtuelles Gerät entspricht einer Schnittstelle zu einen 870-Master.
Sector	Registerkarte für die Konfiguration der Sektoren (auf Seite 170) des Gerätes. Ein Sektor eintspricht einer Common Object Address/Common Address of ASDU und wird mit **COA** adresiert. Hinweis: Ein Sektor kann mehrere **Information Objects** beinhalten, die mit zenon Variablen verknüpft, und mit **Type Identification** (**Type ID**) und **Information Object Address** (**IOA**) adressiert sind.
SOE	Registerkarte für die Konfiguration der Aufnahme und Übertragung der SOE Dateien (auf Seite 205) (Sequences of Events)
104 settings	Registerkarte für die Konfiguration der für Protokoll 870-104 spezifisches Einstellungen für Timeouts (auf Seite 192) und APDU Zähler.
OK	Übernimmt alle vorgenommenen Einstellungen. Diese werden sofort in die aktuelle Konfigurationsdatei (wie in **Configuration file** definiert) gesichert. Achtung: Wenn Sie keine Änderungen vorgenommen haben, beenden Sie der Dialog mit Button **Cancel**, nicht **OK**. Mit Klick auf **OK** werden alle bestehenden Verbindungen mit 870-Mastern beendet und die Kommunikation wird mit den neuen Einstellungen neu gestartet.
Cancel	Verwirft alle vorgenommen Änderungen und schlisst der Dialogfenster. Der Dialog wird geschlossen ohne die bereits bestehende Kommunikation zu 870-Mastern zu unterbrechen.
Help	Öffnet die Online-Hilfe.

10.1.1 Algemeine Einstellungen



Info

In diesem Abschnitt sind nur die allgemeinen Eigenschaften des IEC870 Slaves dokumentiert. Weitere Informationen finden Sie in der Beschreibung des virtuellen Gerätes (**Device**).

SETTINGS

Allgemeine Einstellungen:

Parameter	Beschreibung
Configuration file	Name der XML-Konfigurationsdatei mit der Konfiguration des IEC870 Slaves im Process Gateway. In diese werden alle aktiven Einstellungen gesichert und aus ihr werden sie auch wieder ausgelesen. Der eingegebene Name muss ein gültiger Dateiname sein, sonst können die Daten nicht gesichert werden. In diesem Fall erhalten Sie eine entsprechende Fehlermeldung. Eingegeben wird nur der Dateiname ohne Pfad. Die Dateiendung XML muss nicht explizit angegeben werden, sie wird beim Speichern automatisch ergänzt. Als Pfad wird immer das aktuelle INI-Verzeichnis verwendet. Ablageort: %CD_SYSTEM% Die INI Datei finden Sie im System-Verzeichnis des Prozessleitsystems als '<SCADA>ProcGateway.ini' - <SCADA> ist Platzhalter für den Namen des Prozessleitsystems. Der Dateiname wird automatisch in die INI-Datei des Process Gateways als Konfiguration eingetragen. Die Beschreibung des Struktur der XML-Datei finden Sie im Kapitel XML-Datei (auf Seite 195).
Import...	Lädt eine Projektierung (auf Seite 201) aus einer XML- oder .CSV-Datei. Die aktuellen Einstellungen werden durch jene in der Datei ersetzt. Die aktuelle Einstellung beim **Configuration file** bleibt erhalten. Klick auf Schaltfläche **...** öffnet den Dateiauswahldialog zur Auswahl einer Konfigurationsdatei. Hinweis: Damit können Sie die Konfigurationen von anderen Projekten übertragen oder die Konfigurationen mit größeren Mengen an Variablen importieren, die Sie mit anderen Applikationen vorbereitet und exportiert haben. Beim Importieren wird der Dateiinhalt überprüft. Im Fehlerfall wird ein entsprechender Warndialog angezeigt.
Export...	Exportiert aktuelle Projektierung (auf Seite 201) in eine XML- oder .CSV-Datei. Klick auf Schaltfläche **...** öffnet den Dateiauswahldialog zur Auswahl des Ablageorts für die Zieldatei. Achtung: Ein **Device**, das noch die beim Anlegen vergebene IP oder `Link Addresse` hat, wird nicht exportiert. Um es zu exportieren, konfigurieren Sie zuvor die Verbindung zum 870-Master im Konfigurationsfenster **Device** (auf Seite 166). Hinweis: ▸ Sind in der aktuellen Process Gateway Projektierung keine IOs vorhanden, wird keine Datei exportiert. ▸ Für 870-104 Verbindungen wird die projektierte `IP Addresse` exportiert. ▸ Für 870-101 wird die projektierte `Link Addresse` exportiert.

870-104	aktiviert Protokoll 104, schaltet alle für 104 spezifischen Einstellungen frei und sperrt Einstellungen für 101. In diesem Mode können mehrere Slaves mit mehreren Mastern über die TCP/IP Schnittstellen kommunizieren. Jedes TCP Socket definirt eine 1:1 Verbindung zwischen Slave und Master, somit ist es möglich dem Slave die Wertänderungen spontan übertragen. Der Verlust der Verbindung kann sowohl von Slave wie Master erkannt werden, dank im Hintergrund laufenden Test-Prozeduren (Austausch kleiner APDUs auf dem Link Layer).
870-101 unbalanced	aktiviert Protokoll 101 im `unbalanced Mode`, schaltet alle für 101 spezifischen Einstellungen frei und sperrt Einstellungen für 104. In diesem Mode können mehrere Slaves mit mehreren Mastern über die serielle Schnittstelle kommunizieren. Ein Slave darf einem Master die Daten nur an eine (zyklische) Link Layer Anforderung übertragen, da sonst die Konflikte auf dem Link Layer entstehen könnten. Es gibt keine Test-Prozeduren für Slave zu Überprüfung ob die Verbindung zum Master weiter besteht.
870-101 balanced	aktiviert Protokoll 101 im `balanced Mode`, schaltet alle für 101 spezifischen Einstellungen frei und sperrt Einstellungen für 104. In diesem Mode kann nur ein Slave mit nur einem Master über die serielle Schnittstelle kommunizieren. Es ermöglicht dem Slave die Wertänderungen spontan übertragen. Der Verlust der Verbindung kann sowohl von Slave wie Master erkannt werden, dank im Hintergrund laufenden Test-Prozeduren (Austausch kleiner APDUs auf dem Link Layer).
IP Port	IP-Port für die Kommunikation via IEC 60870-5-104 Protokoll. Der Port des Slaves auf den sich die Master verbinden sollen. Das Protokoll definiert diesen fix als 2404. Default: `2404` Gültige Portnummern: `1200` bis `65535` Hinweis: Gilt für alle Verbindungen für Protokoll 104. Inaktiv wenn als Protokoll `870-101 unbalanced` oder `870-101 balanced` gewählt ist. Es ist möglich mehrere Instanzen des Process Gateways gleichzeitig laufen zu lassen. Daher ist der Port einstellbar.
Link Address Size (870-101)	Größe der Link Layer Adresse, in Bytes, für die Kommunikation via IEC 60870-5-101 Protokoll. Auswahl aus Dropdownliste. Default: `1` Hinweis: Die gleiche Größe gilt für alle Verbindungen mit 870-101 Mastern. Inaktiv wenn als Protokoll 870-104 gewählt ist.
COM Port	Serielle Schnittstelle des Computers mit zenon Runtime. Auswahl aus Dropdownliste Default: `1` Hinweis: Gilt für alle Verbindungen mit 870-101 Mastern (`balanced` und `unbalanced`). Inaktiv wenn als Protokoll 870-104 gewählt ist.

Baud Rate	Datenübertragungsrate. Auswahl aus Dropdownliste Default: `9600` Hinweis: Gilt für alle Verbindungen mit 870-101 Mastern (`balanced` und `unbalanced`). Inaktiv wenn als Protokoll 870-104 gewählt ist.
Deaktivate standard DPI/DCS mapping and use internal mapping	Deaktiviert die Standardkommunikation via DPI/DCS und verwendet in der Kommunikation internes Mapping. Bei Aktivierung dieser Option wird das normkonforme Verhalten außer Kraft gesetzt. Default: `Inaktiv` Achtung: Die Aktivierung dieser Option wird nicht empfohlen.

10.1.2 Devices Liste

Die virtuellen Geräte im IEC870 Slave - **Devices** - werden im linken Bereich des Konfigurationsdialogs angelegt und gelöscht.
Die Details pro Device werden im rechten Bereich des Dialogs mit den entsprechenden Registerkarten konfiguriert.

Um die Konfiguration ein bestehendes **Device** zu ändern, wählen Sie das gewünschte **Device** in der Liste. Die dazugehörigen Konfigurationen können dann im rechten Bereich angepasst werden.

Parameter	Beschreibung
Device	Identifizierung des Devices: IP Adresse des Masters (104) oder Link Adresse des Slaves (101). Default: ???? (nicht definiert) Achtung: Bevor Sie es verwenden oder exportieren können, muss das Gerät konfiguriert werden.
Name	Name des Devices/Verbindung - beliebiges Text zur leichteren Identifizierung.
Add	Fügt ein neues Gerät mit Standardeinstellungen hinzu. Dieses erhält standardmäßig die ungültige ID "????" und ungültige IP/Link-Adressen. Achtung: Bevor Sie es verwenden oder exportieren können, muss das Gerät konfiguriert werden.
Delete	Löscht das ausgewählte Gerät. Wurde kein Gerät gewählt, ist die Schaltfläche inaktiv. Achtung: Der ausgewählte Eintrag wird sofort ohne vorherige Sicherheitsabfrage gelöscht.

Die Neuanlage oder das Löschen eines Devices wird in der linken Seite des Dialoges durchgeführt..

Klick auf die Schaltfläche **Add** fügt ein Device hinzu, welches in den Registerkarten **Device**, **Sector**, **SOE** und **104 settings** konfiguriert wird. Es können mehrere Devices angelegt werden. Ein einfacher Klick auf das jeweilige Device in der Liste öffnet die dazugehörige Konfiguration.

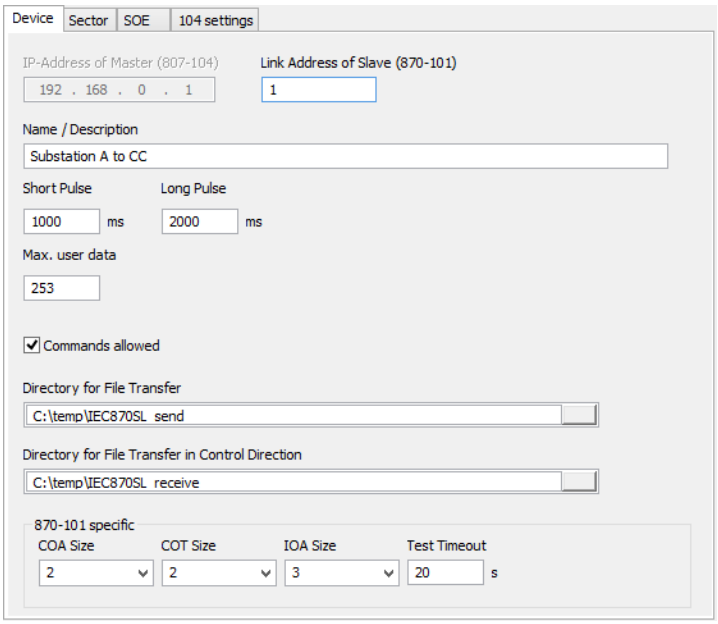
VALIDIERUNG

Konfigurationen werden bei Klick auf die Schaltfläche **OK** validiert. Im Fehlerfall wird ein entsprechender Warndialog angezeigt. Die Überprüfung berücksichtigt dabei die gewählte Verbindung (870-104 oder 870-101). Wird die Konfiguration aus einer Datei übernommen, wird der Dateiinhalt bei der Validierung ebenfalls berücksichtigt.

Auch bei Start des IEC870 Process Gateway findet zuerst eine Überprüfung der bestehenden Konfiguration statt.

10.1.3 Device - Konfiguration

In der Registerkarte **Device** des Hauptfensters legen Sie die Einstellungen des virtuellen Gerätes fest. Je nach gewähltem Protokoll - 870-104 oder 101 - sind bestimmte Bereiche ausgegraut. Eine genaue Definition welche Eingabefelder für welches Protokoll verfügbar sind findet sich in der untenstehenden Auflistung.



Parameter	Protokoll	Beschreibung
IP-Address of Master (807-104)	104	IP-Adresse des Masters, dem erlaubt ist, eine Verbindung zum Slave aufzubauen. Über diese IP Adresse wird das **Device** erkannt, mit welchem der Master kommuniziert. Alle IP-Adressen ausgenommen 255.255.255.255 sind erlaubt. Die konfigurierte IP-Adresse wird in der Liste der `Devices` in der Spalte `Device` angezeigt. Default: `255.255.255.255` Hinweis: Wird das Protokoll umgestellt, geht der aktuelle Wert verloren.
Link Adress of Slave (870-101)	101	Die Link Layer Adresse des Slaves. Über die im TF 1.2 Frame enthaltene Link-Adresse legt der Master fest, für welches **Device** das Telegramm gilt. Über Link-Adressen kann ein Master mit mehreren Devices kommunizieren. Der erlaubte Bereich hängt von der Einstellung **Link Address Size** in den Haupteinstellungen ab. ▸ **Link address size** = 0 gilt nur für balanced Mode. Kein Bereich erlaubt. Die Verbindung ist eindeutig (1 Slave : 1 Master) ▸ **Link address size** = 1 erlaubter Bereich: 0 bis 254 ▸ **Link address size** = 2 erlaubter Bereich: 0 bis 65534 Default: `-1 (ungültig)` Hinweis: Eine Projektierung dieser Option ist nicht notwendig, wenn `870-101 balanced` ausgewählt ist. Wird das Protokoll umgestellt, geht der aktuelle Wert verloren.
Name/Description	101 104	Name des Gerätes zur leichteren Identifizierung, ein beliebiges Text. Der projektierte Name wird in der Liste der **Devices** in der Spalte `Name` angezeigt.
Short pulse	101 104	Definiert die Länge des Pulses für den `Qualifier of Command` (auf Seite 224) in der Ausprägung 'short pulse duration' (QOC = 1) in Millisekunden. Default: `1000 ms` Gültige Werte: `0 bis 4294967295`

Long pulse	101 104	Definiert die Länge des Pulses für den `Qualifier of Command` (auf Seite 224) in der Ausprägung 'long pulse duration' (QOC = 2) in Millisekunden. Default: `2000 ms` Gültige Werte: `0 bis 4294967295`.
Max APDU size	104	Maximale Länge der APDU Daten wie in im Standard IEC 60870-5-104 definiert. Default: `253` Gültige Werte: `zwischen 25 und 253`.
Max. user data	101	Maximale Länge der User Data wie für TF1.2 in IEC60870-5-2 Abschnitt 3.2 definiert. Default: `253` Gültige Werte: `zwischen (19 + LAF-Size + COA-Size + COT-Size + IOA-Size) und (255)`.
Commands allowed	101 104	Diese Option erlaubt die Befehle von Master dauerhaft zu unterbinden. Ist die Einstellung deaktiviert, werden die Befehle nicht akzeptiert. In der LOG-Datei werden dafür entsprechende Einträge erstellt. Default: `aktiv` Hinweis: über eine Variable in zenon (T00, IOA 2) (auf Seite 214) können die Befehle dynamisch deaktiviert oder aktiviert werden.
Directory for File Transfer	101 104	Definiert das Verzeichnis, in dem die Dateien für den Verzeichnis- und Dateitransfer abgelegt sein müssen. Klick auf die Schaltfläche **...** öffnet den Dateiauswahldialog zur Auswahl eines Quellverzeichnis. Hinweis: Die Dateien werden an den Master versendet.
Direcotry for File Transfer in Control Direction	101 104	Definiert das Verzeichnis, in dem die Dateien für den Verzeichnis- und Dateitransfer abgelegt werden. Klick auf die Schaltfläche **...** öffnet den Dateiauswahldialog zur Auswahl eines Zielverzeichnis. Hinweis: Diese Dateien werden vom Master empfangen. Alle Dateien mit der NOF 255 (Die temporäre NOF für Reverse File Transfer) werden bei der Verzeichnisabfrage ignoriert.

870-101 SPECIFIC

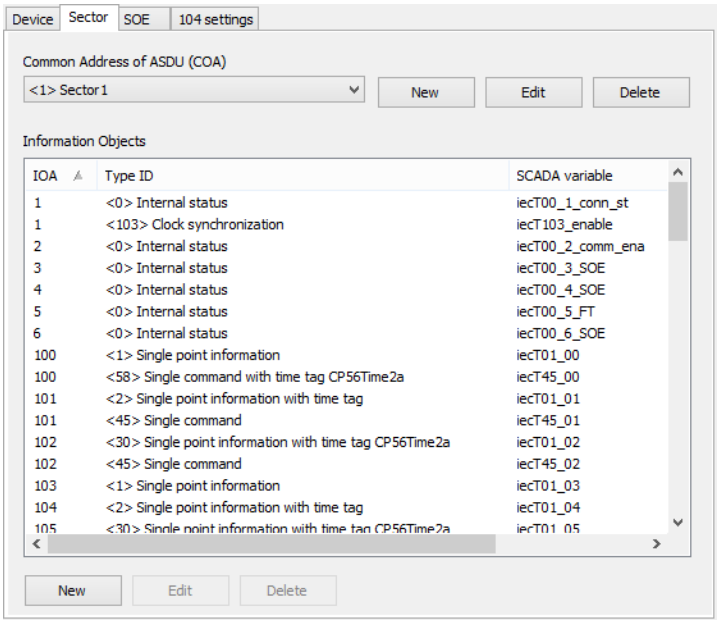
Parameter	Protokoll	Beschreibung

COA size	101	Definiert die Länge der COA (Common Object Address/Common Address of ASDU). Auswahl der Adressgröße aus Dropdownliste: ▸ 1 octet ▸ 2 octets ▸ Hinweis: Ist bei Link layer die Verbindungsart **60870-5-104 (TCP/IP)** gewählt, wird laut Norm Wert `2 octets` erwartet. Default: 2
COT size	101	Definiert die Länge der COT (Cause of Transmission). Auswahl der Adressgröße aus Dropdownliste. Gültig: ▸ 1 octet ▸ 2 octets ▸ Hinweis: Ist bei Link layer die Verbindungsart **60870-5-104 (TCP/IP)** gewählt, wird laut Norm Wert `2 octets` erwartet. Default: 2
IOA size	101	Definiert die Länge der IOA (Information Object Address). Auswahl der Adressgröße aus Dropdownliste. Gültig: ▸ 1 octet ▸ 2 octets ▸ 3 octets ▸ Hinweis: Ist bei Link layer die Verbindungsart **60870-5-104 (TCP/IP)** gewählt, wird laut Norm Wert `3 octets` erwartet. Default: 3
Test timeout	101	Timeoutzeit für 870-101 Verbindungen. Eingabe in Sekunden ▸ In `101 unbalanced`: Gibt die Zeit in Sekunden an, in der der Master eine Abfrage an den Slave ausführen muss. Wenn innerhalb dieser Zeit keine Abfrage von dem Master antrifft, beendet der Slave die Verbindung. ▸ In `101 balanced`: Gibt die Zeit in Sekunden an, in der der Slave die Test-Prozeduren auf dem Link Layer ausführen wird. Die Test-Prozedur ist von IEC60870-5-101 Standard für `balanced` obligatorisch vorgegeben. Damit kann der Slave der Verbindungsverlust erkennen

		während keine spontane Datenübertragung stattfindet. Default: `20` Hinweis: im `unbalanced Mode` wird bei einem projektierten Wert `0` die Überwachung deaktiviert. Im `balanced Mode` muss ein Wert größer 0 projektiert sein. Bei Inaktitvität (keine Daten zum Senden) wird nach der hier konfigurierten Zeit ein Testframe geschickt. Dieses muss vom Master bestätigt werden. Bei Ausbleiben einer Antwort wird der letzte Frame (Testframe oder spontane Daten) wiederholt. Wird auch darauf keine Antwort empfangen, startet das Process Gateway die Kommunikation mit einem `Request Status of Link` neu.

10.1.4 Sector - Konfiguration

Die Registerkarte **Sector** listet alle im **Device** konfigurierten Sektoren auf. Wenn Sie einen Sektor aus der Dropdownliste auswählen, werden dessen `Information Objects` (auf Seite 176) (IOs) angezeigt.



COMMON ADDRESS OF ASDU (COA)

In oberem Bereich projektieren Sie Sektoren pro Gerät (**Device**). Dargestellt wird die **COA** und der Name des Sektors.

Projektierte Sektoren werden in einer Dropdownliste angezeigt. Per Default wird immer der erste Sektor der Dropdownliste dargestellt. Für diesen Sektor projektierte `Information Objects` werden im darunterliegenden Bereich angezeigt.

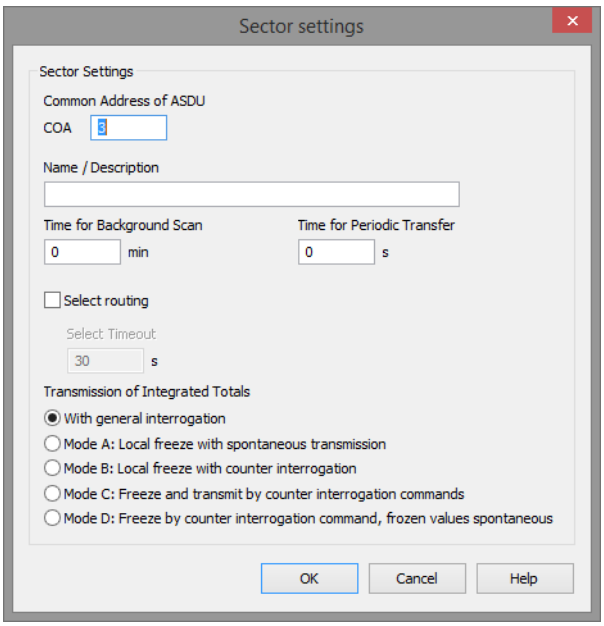
Parameter	Beschreibung
New	Öffnet den Dialog zur Erstellung eines neuen Sektors (auf Seite 172).
Edit	Aktiviert den Dialog zur Bearbeitung des ausgewählten Sektors. Bereits projektierte Sektoren werden aus der Dropdownliste ausgewählt. Ein Doppelklick auf den Sektor in der Liste ruft ebenfalls den Editor auf. Wurde kein Sektor ausgewählt, ist die Schaltfläche inaktiv.
Delete	Löscht den ausgewählten Sektor. Wurde kein Sektor ausgewählt, ist die Schaltfläche inaktiv. Achtung: der ausgewählte Sektor wird ohne Rückfrage sofort gelöscht.

INFORMATION OBJECTS

In unterem Bereich werden `Information Objects (IOs)` für Sektoren verwaltet. Dargestellt werden die `IOs` mit **COA** des gewählten Sektors.

Parameter	Beschreibung
`Liste der projektierten Information Objects pro COA`	Liste der projektierten IOAs für den ausgewählten COA. ▸ `IOA` konfigurierte Adresse der `Information Objects`. ▸ `Type ID` konfigurierte **Type Identification** des `Information Objects`. ▸ `SCADA variable` verknüpfte SCADA Variable des `Information Objects`. Die Liste kann mit einem Klick auf die Spaltenüberschrift sortiert werden. Die Spaltenbreiten sind individuell anpassbar.
New	Aktiviert den Dialog für die Erstellung neuer IOs (auf Seite 176). Mit 'OK' wird dieser Folgedialog beendet und ein neues IO erstellt. Wurde kein Sektor ausgewählt, ist die Schaltfläche inaktiv.
Edit	Aktiviert den Dialog zur Bearbeitung des ausgewählten IOs. Ein Doppelklick auf das IO in der Liste ruft ebenfalls den Editor auf. Wurde kein IO ausgewählt, ist die Schaltfläche inaktiv.
Delete	Löscht das ausgewählte Information Object. Wurde kein IO ausgewählt, ist die Schaltfläche inaktiv.

Sector Settings



Parameter	Beschreibung
COA	Die Common Address of ASDU/Common Object Address (IEC 60870-5-101 7.2.4), unter welcher der Sektor angesprochen wird. Diese Nummer muss pro **Device** eindeutig sein (1..254).
Name/Description	Beliebiger Name. Er erleichtert die Identifizierung des Sektors.
Time for Background scan [min]	Der "Background scan" Zyklus nach Anforderungen des Standards IEC 60870-5-101 Kapitel 7.4.13 "Background scan". Die Werte werden mit der COT = 2 (background scan/COT_back) verschickt. Die Zeitangabe erfolgt in Minuten. Im eingestellten Zeitraster erfolgt für alle IOs mit aktivierter Einstellung **Background scan** eine zyklische Wertübertragung. Eine zyklische Wertübertragung findet nur dann statt wenn innerhalb der eingestellten Zeit noch kein Spontanwert übertragen wurde (kein COT = 3). Bei Wert "0" ist die zyklische Wertübertragung für diesen Sektor deaktiviert. Default: 0
Time for periodic transfer [s]	Unterstützung der zyklischen Datenübertragung entsprechend des IEC 60870-5-5 6.3. Die Werte werden mit der COT = 1 (periodic, cyclic / COT_per) verschickt. Die Zeitangabe erfolgt in Sekunden. Die zyklische Wertübertragung erfolgt für alle IOs mit aktivierter Einstellung **Periodic data transfer**. Diese IOs werden nur zyklisch und nicht spontan übertragen. Bei Wert "0" ist die zyklische Wertübertragung für diesen Sektor deaktiviert. Default: 0 Hinweis: die schnell flackende Messwerte können die spontane Übertragung unnötig belasten. Wir empfehlen jedoch - anstatt Periodic data transfer - die **Hysterese** für jeweiliges IO zu konfigurieren.
Select routing	Wenn aktiv, ist bei Befehlsausführung für ein `Execute` ein vorangegangenes `Select` nötig. Die Schaltbefehle vom Master werden über die **Befehlsgabe** - über die `auto/remote` Aktionen - weitergeleitet - an den jeweiligen Energy Treiber in der zenon Runtime. Hinweis: Weitere Informationen dazu finden Sie im Kapitel **Select routing** (auf Seite 222).
Select Timeout [s]	Timeout für `Select before Execute` in Sekunden. Innerhalb der projektierten Zeit muss der `Select`-Ablauf beendet sein und ein `Execute` angestoßen werden. Kommt von zenon Runtime keine Bestätigung des `Select` oder wurde das `Select` bestätigt jedoch sendet der Master kein `Execute` - bricht das Process Gateway den Ablauf ab.

	Hinweis: Inaktiv wenn **Select routing** nicht aktiv. Default: `30 s` Eingabebereich: 1 bis 65535 Hinweis: Weitere Informationen dazu finden Sie im Kapitel **Select routing** (auf Seite 222).

TRANSMITION OF INTEGRATED TOTALS

Auswahl des Übertragungsmodus (siehe IEC 60870-5-101 7.4.8) für die Integrated Totals - die IOs von Type-ID:

- 15 (`M_IT_NA_1`)
- 16 (`M_IT_TA_1`)
- 37 (`M_IT_TB_1`)

Es wird kein Reset unterstützt; ein FRZ mit Reset erzeugt eine Warnung im LOG.

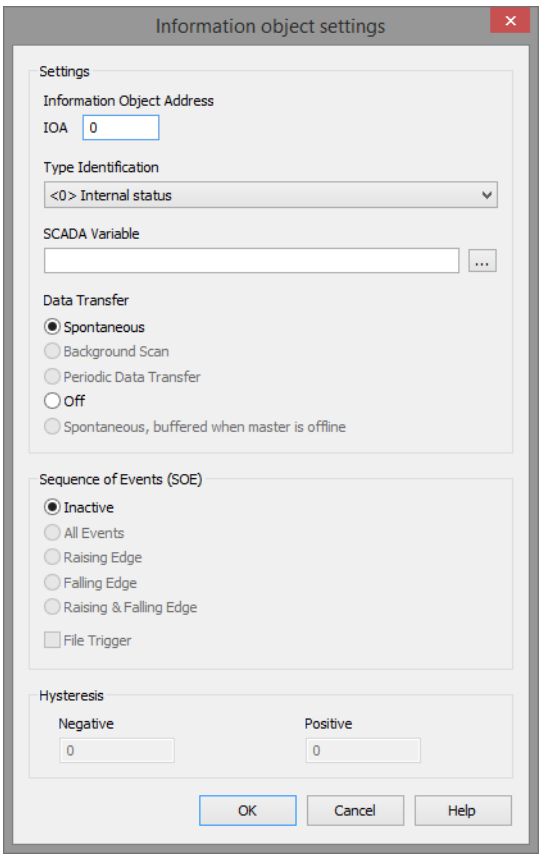
Auswahl aus Optionsliste.
Default: `With general interrogation`

Bei `Mode A` bis `Mode D` werden die IOs sofort nach dem Verbinden des Masters von zenon angefordert. Wertänderungen der `Integrated Totals IOs` werden in einem eigenen Puffer verwaltet. Dieser kann den größeren Wert puffern von (`Anzahl der ITs IO`) * 4, oder 2048 Wertänderungen.

Parameter	Beschreibung
	Integrated Totals werden bei der Generalabfrage (`C_IC_NA_1`) mit übertragen.

Mode A: Local freeze with spontaneous transmission	Integrated Totals werden bei Wertänderung übertragen.
Mode B: Local freeze with counter interrogation	Integrated Totals werden nur bei einer Counter Interrogation Command (`C_CI_NA_1`) übertragen. Dazu wird nach Aktivierung der Abfrage maximal 5 Sekunden gewartet bis alle ITs mit Werten versorgt sind. ITs ohne Wert - die zenon Variable ohne ein Wert - werden mit dem Qualifier `invalid` übertragen. Die Wertänderungen werden bis zum Abschluss der Abfrage gepuffert (freeze active). Alle ITs übertragen somit den Wert von dem Zeitpunkt, an dem das letzte IT mit einem Wert versorgt wurde.
Mode C: freeze and transmit by counter interrogation Transmition of integrated totals commands	Gleiches Verhalten wie für **Mode B**.
Mode D: freeze by counter interrogation command, frozen values spontaneous	Übertragung wie bei Mode C, nach Abschluss der Abfrage wird Mode A aktiviert.
OK	Mit Betätigung der Schaltfläche **OK** wird überprüft, ob die COA beim Gerät frei ist. Wird sie bereits verwendet, erhalten Sie eine Fehlermeldung. Ist sie frei, wird die Einstellung übernommen und der Dialog geschlossen.
Cancel	Verwirft alle Einträge und beendet den Dialog.
Help	Öffnet die Online-Hilfe.

Information Objects



SETTINGS

Parameter	Beschreibung
Information Object Address	Definiert die Adresse (**IOA**), unter der das IO angesprochen wird (IEC 60870-5-101 7.2.5). Default: 1 Die IOA muss in je monitoring und controling direction pro Sektor eindeutig sein. Beispiel: ▸ gültige Adressierung: zwei IOs, beide mit gleicher IOA, ein T01 (single-point - M_SP_NA_1) und ein T45 (single command - C_SC_NA_1). Die Adressierung ist empfehlenswert wenn dieser single command genau die single-point information kontrolliert. ▸ ungültige Adressierung: zwei IOs, beide mit gleicher IOA, ein T01 und ein T15 (integrated totals - M_IT_NA_1). Diese Adressierung ist unzulässig weil beide Type-IDs der monitoring direction gehören.
Type identification	Dropdownliste mit den unterstützten Type identifications (IEC 60870-5-101 7.2.1). Default: `<0> Internal status` Diese Einstellung wird auch Type-ID genannt. Sie bestimmt auch, welche Variable des Prozessleitsystems aufgrund ihrer Datenart dem IO zugeordnet werden darf. Hinweis: Genauere Informationen zur Type-ID finden Sie auch in den Kapiteln Zuordnung der Datentypen (auf Seite 183) und Interoperability (auf Seite 237).
SCADA variable	Definiert die dem IO zugeordnete Variable aus dem Prozessleitsystem. Syntax: <Projektname>#<Variablenname> definiert den Projektbezug. Ist für eine Variable kein Projektbezug angegeben, wird davon ausgegangen, dass diese Variable im Startprojekt vorhanden ist. Bei der Eingabe wird geprüft ob die angegebene Variable im entsprechenden Projekt vorhanden ist und eine mit der Type identification kompatible Datenart hat. Die Schaltfläche '**...**' aktiviert den Dialog zur Variablenauswahl. Details dazu lesen Sie im Abschnitt 'Variable selection (auf Seite 182)'. (SCADA ist Platzhalter für den Namen des Prozessleitsystems.) Unterstützte Datentypen: `BOOL; BYTE; WORD; DWORD; LWORD; SINT; INT; DINT; LINT; USINT; UINT; UDINT; ULINT; REAL; LREAL; long double; reelle Zahl IEEE`

DATA TRANSFER

Bestimmt die Art der Datenübertragung für den ausgewählten IO. Auswahl aus Optionenliste.
Default: `Spontaneous`

Parameter	Beschreibung
Spontaneous	Datenübertragung der IOs erfolgt spontan - wenn der Wert der verknüpfter SCADA Variable sich ändert.
Background Scan	IOs werden entsprechend des IEC 60870 Standards als Background Scan übertragen. Datenpunkt der innerhalb der im Sektor eingestellten **Time for Background Scan** keinen neuen Spontanwert erhält, wird automatisch durch das Gateway nach Ablauf der Zeit verschickt; mit COT = 2. Die Wertübertragung findet nur dann statt wenn innerhalb der eingestellten Zeit noch kein Spontanwert (COT = 3) für dieses IO übertragen wurde. Einstellung ist inaktiv wenn der Background Scan beim Sektor gesperrt ist (`Time=0`) oder wenn die verwendete Type-ID laut Protokoll für ein Background Scan ungeeignet ist. Erlaubte Type-IDs: ▸ `M_SP_NA_1` und `M_SP_TA_1`, `M_SP_TB_1` - als `T01`, ▸ `M_DP_NA_1` und `M_DP_TA_1`, `M_DP_TB_1` - als `T03`, ▸ `M_ST_NA_1` und `M_ST_TA_1`, `M_ST_TB_1` - als `T05`, ▸ `M_BO_NA_1` und `M_BO_TA_1`, `M_BO_TB_1` - als `T07`, ▸ `M_ME_NA_1` und `M_ME_TA_1`, `M_ME_TD_1` - als `T09`, ▸ `M_ME_NB_1` und `M_ME_TB_1`, `M_ME_TE_1` - als `T11`, ▸ `M_ME_NC_1` und `M_ME_TC_1`, `M_ME_TF_1` - als `T13` Hinweis: Die Type-IDs mit Zeitstempel (zB. `M_ME_TA_1`) werden dabei auf den entsprechenden TID ohne Zeitstempel (zB. `M_ME_NA_1`) beim Versenden des Backgroundscans gemappt. Für alle anderen Type-ID ist die Einstellung nicht verfügbar.
Periodic Data Transfer	Periodischer Datentransfer entsprechend des Standards IEC 60870-5-5 Abschnitt 6.3. IOs werden zyklisch und nicht spontan übertragen; mit COT = 1. Einstellung ist inaktiv wenn die Periodische Übertragung beim Sektor gesperrt ist (**Time for periodic transfer**=0) oder wenn die verwendete Type-ID die Periodische Übertragung nicht unterstützt. Nur die Type-ID=9, 11 und 13 unterstützen eine Periodische Übertragung - `M_ME_NA_1`, `M_ME_NB_1`, `M_ME_NC_1` - "measured values" ohne Zeitstempel. Für alle anderen Type-ID ist die Einstellung nicht verfügbar. Hinweis: die schnell flackende Messwerte können die spontane Übertragung unnötig belasten. Wir empfehlen jedoch - anstatt Periodic Data Transfer - die **Hysterese** für jeweiliges IO zu konfigurieren.

Off	Wertänderungen werden nicht übertragen. Für das IO wird bei einer General Interrogation (GI) kein Wert geliefert.
Spontaneous, buffered when master is offline	Bei einem Verbindundsausfall werden die Daten so lange gepuffert, bis der Master sich erneut verbindet. Alle gepufferten Daten werden mit dem Status COT_spont (3) gesendet. Der letzte gepufferte Wert wird nochmals mit dem Status COT_inrogen (20) gesendet. Die Pufferung kann für die Type-IDs <30..37> aktiviert werden: `M_SP_TB_1, M_DP_TB_1, M_ST_TB_1, M_BO_TB_1, M_ME_TD_1, M_ME_TE_1, M_ME_TF_1 und M_IT_TB_1` Hinweis: Als Events sind nur die Type-IDs erlaubt die ein Zeitstempel mit Datum haben ("time tag CP56Time2a"). Für alle anderen Type-IDs ist die Einstellung nicht verfügbar.

SEQUENCES OF EVENTS (SOE)

Diese Einstellungen definieren das IO spezifische Verhalten für SOE. Die Abspeicherung der SOE Dateien und deren Übertragung (siehe IEC Norm 60870-5-101 7.4.11.3.1) kann nur für die Type-IDs <30..37> aktiviert werden:

`M_SP_TB_1`, `M_DP_TB_1`, `M_ST_TB_1`, `M_BO_TB_1`, `M_ME_TD_1`, `M_ME_TE_1`, `M_ME_TF_1` und `M_IT_TB_1`.

Default: `Inactive`

Hinweis: Als Events sind nur die Type-IDs erlaubt die ein Zeitstempel mit Datum haben ("time tag CP56Time2a").

Parameter	Beschreibung
Inactive	Datenpunkt wird für SOE Aufzeichnung nicht verwendet. Wird automatisch für alle Type-IDs verwendet welche kein SOE unterstützen.
All events	Jede Wert- oder Statusänderung löst das Schreiben in die geöffnete Datei aus. Einstellung ist gesperrt wenn Type-ID nicht für die SOE Funktionalität vorgesehen ist.
Raising Edge	Wenn der Wert von 0 auf <> 0 wechselt, löst dies das Schreiben in die geöffnete Datei aus. Einstellung ist gesperrt wenn Type-ID nicht für die SOE Funtionalität vorgesehen ist.
Falling Edge	Wenn der Wert von <> 0 auf 0 wechselt, löst dies das Schreiben in die geöffnete Datei aus. Einstellung ist gesperrt wenn Type-ID nicht für die SOE Funtionalität vorgesehen ist.
Raising + Falling Edge	Ein Wertwechsel von 0 auf <> 0 oder von <> 0 auf 0, löst das Schreiben in die geöffnete Datei aus. Einstellung ist gesperrt wenn Type-ID nicht für die SOE Funtionalität vorgesehen ist.
File trigger	Wenn Aufgrund der vorher beschriebenen Einstellungen ein Wert geschrieben werden soll, wird bei aktiviertem File trigger eine neue Datei erzeugt und geöffnet, sofern diese noch nicht geöffnet ist. Das Triggerereignis selbst wird ebenfalls in die SOE Datei geschrieben. Einstellung ist gesperrt wenn Type-ID nicht für die SOE Funtionalität vorgesehen ist. Default: `nicht aktiv`

HYSTERESIS

Wertänderungen, welche innerhalb der Hysterese sind, werden ignoriert - nicht an Master übertragen.

Hysterese gilt nur für die erlaubten **TID** ("measured values") und nur wenn diese für spontane Datenübertragung konfiguriert wurden.

Eine Übertragung wird ausgelöst:

- Wenn `Type Identification` die Hysterese zulässt.
  Zugelassen sind die `Type Identifications` `<9> bis <14>` sowie `<34> bis <36>`.
- Wenn **Data Transfer** = `spontaneuous`
- Bei einer Wertänderung, welche >= der Hysterese (entsprechend der Richtung) handelt.

Und ungeachtet der Hysterese:

- Änderung der Quality (Statusbits der Variable) löst immer die Übertragung aus;
- Auf ein GI (`C_IC_NA_1`) und ein Read (`C_RD_NA_1`) Befehl wird immer der aktuelle Wert gesendet.

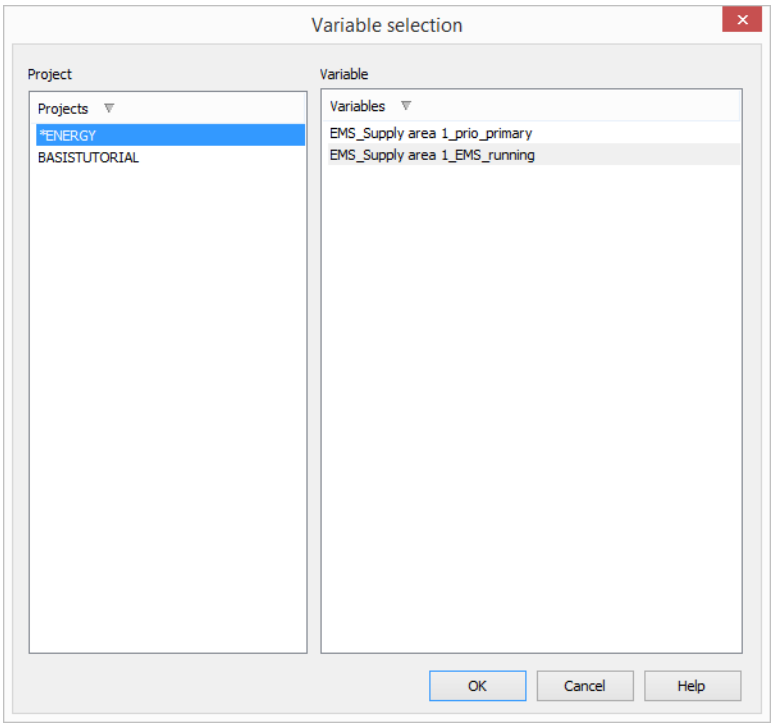
Parameter	Beschreibung
Negative	Negative Hysterese; ausgehend vom **Messbereich** der Variable. Default: `0` Hinweis: nicht aktiv wenn Type Identification nicht `<9>` bis `<14>` oder `<34>` bis `<36>` ist. Dieses Feld ist auch dann inaktiv, wenn der Wert **Data Transfer** nicht `Spontaneous` ist. Fehlerhafte und sinnlose Eingaben (z.B. Texteingaben oder falsche Hysteresewerte) werden ignoriert und müssen vom Projektanten korrigiert werden.
Positive	Positive Hysterese; ausgehend vom **Messbereich** der Variable. Default: `0` Hinweis: nicht aktiv wenn Type Identification nicht `<9>` bis `<14>` oder `<34>` bis `<36>` ist. Dieses Feld ist auch dann inaktiv, wenn der Wert **Data Transfer** nicht `Spontaneous` ist. Fehlerhafte und sinnlose Eingaben (z.B. Texteingaben oder falsche Hysteresewerte) werden ignoriert und müssen vom Projektanten korrigiert werden.
OK	Mit Betätigung der Schaltfläche OK werden die Einstellungen überprüft. Sind diese zulässig, werden sie übernommen und der Dialog geschlossen.
Cancel	Verwirft alle Einträge und beendet den Dialog.
Help	Öffnet die Online-Hilfe.

Info

*Die Kommunikation für **Spontaneous** Datatransfer, **Background Scan** und **Periodic Data Transfer** beginnt erst nach der ersten empfangenen Generalabfrage (`C_IC_NA_1`). Für Background scan und Periodic data transfer der Zyklusstart der Übertragung wird pro IO berechnet - ab dem Zeitpunkt der Antwort auf die Generalabfrage.*

Variablenauswahldialog

Dieser Dialog listet die verfügbaren Projekte und deren Variablen auf.



Parameter	Beschreibung
Projekte	Liste der in der zenon Runtime gestarteten Projekte. Ein Stern (*) kennzeichnet das Defaultprojekt.
Variablen	Die Liste der Variablen ist gefiltert. Angezeigt werden nur Variablen, die mit der beim IO gewählten **Type identification** verwendbar sind. Die Auswahl eines Projektes aktualisiert die Liste der Variablen.
OK	Übernimmt die ausgewählte Variable. Alternativ kann die Variable auch per Doppelklick übernommen werden. Beim Aufschalten werden das aktuell eingestellte Projekt und die Variable ausgewählt. Die gleiche Variable kann mehrfach verwendet werden: bei mehreren IOs beim gleichen Sektor oder anderen Sektoren, beim gleichen Gerät oder einem anderen. Wird bei geöffnetem Dialog in der Runtime nachgeladen, dann werden alle Listen geleert und nach Abschluss des Nachladens wieder gefüllt.
Cancel	Verwirft alle Einträge und beendet den Dialog.
Help	Öffnet die Online-Hilfe.

Zuordnung der Datentypen

In folgender Tabelle werden die zenon Datentypen den IEC 60870 Datentypen gegenübergestellt.

Process Gateway		zenon	
IEC 60870-5-101 ASDU	**Type Identification**	**Datentyp**	**Kommentar**
M_SP_NA_1	1	BOOL	
M_SP_TA_1	2	BOOL	
M_SP_TB_1	30	BOOL	
M_DP_NA_1	3	USINT	0..3, mit DPI/DCS Mapping
M_DP_TA_1	4	USINT	0..3, mit DPI/DCS Mapping
M_DP_TB_1	31	USINT	0..3, mit DPI/DCS Mapping
M_ST_NA_1	5	USINT	Entspricht gesamten VTI (IEC60870-5-101 7.2.6.5); damit ist das höhste Bit in USINT - das Transient Bit.
M_ST_TA_1	6	USINT	Entspricht gesamten VTI (IEC60870-5-101 7.2.6.5); damit ist das höhste Bit in USINT - das Transient Bit.
M_ST_TB_1	32	USINT	Entspricht gesamten VTI (IEC60870-5-101 7.2.6.5); damit ist das höhste Bit in USINT - das Transient Bit.
M_BO_NA_1	7	UDINT	
M_BO_TA_1	8	UDINT	
M_BO_TB_1	33	UDINT	
M_ME_NA_1	9	REAL	der Wertebereich laut IEC 60870 Standard ist <-1 .. $1-2^{-15}$> mit einer Genauigkeit auf 4 Nachkommastellen. Das entspricht in zenon Variablen mit einem Wertebereich -1.0 bis 0.9999. Außerhalb des Bereiches für normalized values wird in ASDU der Wert des IOs angepasst und das OV Qualitäts-Bit gesetzt.
M_ME_TA_1	10	REAL	der Wertebereich laut IEC 60870 Standard ist <-1 .. $1-2^{-15}$> mit einer Genauigkeit auf 4 Nachkommastellen. Das entspricht in zenon Variablen mit einem Wertebereich -1.0 bis 0.9999. Außerhalb des Bereiches für normalized values wird in ASDU der Wert des IOs angepasst und das OV Qualitäts-Bit gesetzt.
M_ME_TD_1	34	REAL	der Wertebereich laut IEC 60870 Standard ist <-1 .. $1-2^{-15}$> mit einer Genauigkeit auf 4 Nachkommastellen. Das entspricht in zenon Variablen mit einem Wertebereich -1.0 bis 0.9999.

			Außerhalb des Bereiches für normalized values wird in ASDU der Wert des IOs angepasst und das OV Qualitäts-Bit gesetzt.
M_ME_NB_1	11	INT	
M_ME_TB_1	12	INT	
M_ME_TE_1	35	INT	
M_ME_NC_1	13	REAL	
M_ME_TC_1	14	REAL	
M_ME_TF_1	36	REAL	
M_IT_NA_1	15	DINT	
M_IT_TA_1	16	DINT	
M_IT_TB_1	37	DINT	
M_EI_NA_1	70	USINT	eine Wertänderung von 126 auf 127 veranlasst, dass End of initialization (auf Seite 212) an den Master gesendet wird
C_SC_NA_1	45	BOOL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SC_TA_1	58	BOOL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_DC_NA_1	46	USINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx. 0 oder 1, mit DPI/DCS Mapping
C_DC_TA_1	59	USINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx. 0 oder 1, mit DPI/DCS Mapping
C_RC_NA_1	47	USINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx. 0 oder 1, mit DPI/DCS Mapping
C_RC_TA_1	60	USINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx. 0 oder 1, mit DPI/DCS Mapping

C_SE_NA_1	48	REAL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SE_TA_1	61	REAL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SE_NB_1	49	INT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SE_TB_1	62	INT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SE_NC_1	50	REAL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_SE_TC_1	63	REAL	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_BO_NA_1	51	UDINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_BO_TA_1	64	UDINT	die Variable spiegelt den Wert des Befehls der vom 870-Master empfangen wurde und die Statusbits der Variable spiegeln den Ablauf des Befehls. Beispiel: COTx.
C_IC_NA_1	100	-	Der Befehl wird im Hintergrund automatisch ausgeführt. Alle vom IEC 60870 Standard vorgesehene ASDUs werden an den 870-Master gesendet
C_CI_NA_1	101	-	Der Befehl wird im Hintergrund automatisch ausgeführt. Alle vom IEC 60870 Standard vorgesehene ASDUs werden an den 870-Master gesendet
C_RD_NA_1	102	-	Der Befehl wird im Hintergrund automatisch ausgeführt. Alle vom IEC 60870 Standard vorgesehene ASDUs werden an den 870-Master gesendet
C_CS_NA_1	103	BOOL	Enable-Variable der Zeitsynchronisierung
C_TS_NA_1	104	-	Der Befehl wird im Hintergrund automatisch ausgeführt. Alle vom IEC 60870 Standard vorgesehene ASDUs werden an den 870-Master gesendet
C_TS_TA_1	107	-	Der Befehl wird im Hintergrund automatisch ausgeführt. Alle vom IEC 60870 Standard vorgesehene ASDUs werden an den 870-Master gesendet
C_RP_NA_1	105	USINT	die Variable spiegelt der Wert QRP des Befehls der vom 870-Master empfangen wurde.

F_DR_TA_1	126	USINT	die Variable der Auslösung der spontanen Verzeichnisübertragung (auf Seite 226)

WERTANPASSUNG VON IOS

Es ist möglich einem IO (mit einer bestimmter Type ID) mit eine Variable zu verknüpfen, deren Wertbereich den Wertbereich der Type ID des IO überschreitet.

Beispiel: Ein IO mit `Type ID = 1 - single-point information` wurde mit einer UDINT Variable `*/stVal[ST]` vom IEC850 Treiber verknüpft.

In diesem Fall wird das Process Gateway die Werte für IO anpassen und in ASDU zusätzlich die Qualitäts-Bits setzen, so wie es im IEC 60870 Standard vorgesehen ist.

ASDU Type ID	zenon Variablenwert	Wert und Quality in ASDU
M_SP single-point	< 0 > 1	0 + IV 1 + IV
M_DP double-point	< 0 > 3	11b + IV 11b + IV
M_ST step position	< -64 > 255	x40 + OV xFF + OV
M_BO bitstring 32-bits	< 0 > xFFFFFFFF	0 + OV xFFFF FFFF + OV
M_ME_xA normalized	< -1.0 >= 1.0	x8000 + OV x7FFF + OV
M_ME_xB scaled	< -2^{15} > $2^{15}-1$	x8000 + OV x7FFF + OV
M_ME_xC short floating	< -MAX Float > MAX Float	-MAX Float + OV MAX Float + OV
M_IT integrated totals	< -2^{31} > $2^{32}-1$	x8000 0000 + IV x7FFF FFFF + IV

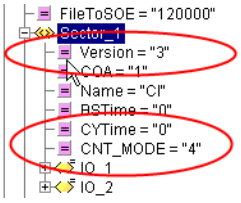
Info

Wenn die zenon Variable gerade das Statusbit `INVALID` oder das Statusbit `OV_870` gesetzt hat, wird der Wert des IOs ebenfalls mit `Quality IV` oder `Quality OV` gesendet.

Weitere Informationen dazu finden Sie im Kapitel "Mapping von Statusbits in Melderichtung (auf Seite 219)".

Sector

XML FELDER



CNT_MODE	0
ToSbe	45000
SBE	1

(This tag has no subtags.)

Parameter	Beschreibung
Version	Ab Version 3 wird das Feld CNT_MODE erwartet.
CNT_MODE	Projektierter Modus der Integrated Totals IOs.
Sektor	Beim Sektor wurden die neuen Items ToSBe und ob SBE eingefügt: ▸ ToSBe: Timeout [ms] für Select before execute: 1000 - 65535000 ▸ SBE: 0 => nicht erforderlich 1 => nötig für execute

PROJEKTIERTER MODUS

Parameter	Beschreibung
0	Mit Generalabfrage. (Default)
1	Mode A
2	Mode B
3	Mode C
4	Mode D

FEHLERCODES

Code	Beschreibung
13	Advise for integrated totals fail.
14	Queue overrun. Integrated totals values lost.
16	Advise für Befehlsvariable fehlgeschlagen
17	Liste der Wertänderungen für Befehlsvariablen mit SBE zu lange. Wertänderung ist verloren. Minimal werden 2048 Wertänderungen gepuffert oder die vierfache Anzahl der aktiven Select before Execute (SBE) - je nachdem, welcher Wert größer ist.
572	Counter interrogation. Unsupported QCC.
573	Counter interrogation. Unexpected IOA.
574	Counter interrogation. Unsupported COT.
575	Buffer for integrated totals interrogation data to small.

WARNUNGEN

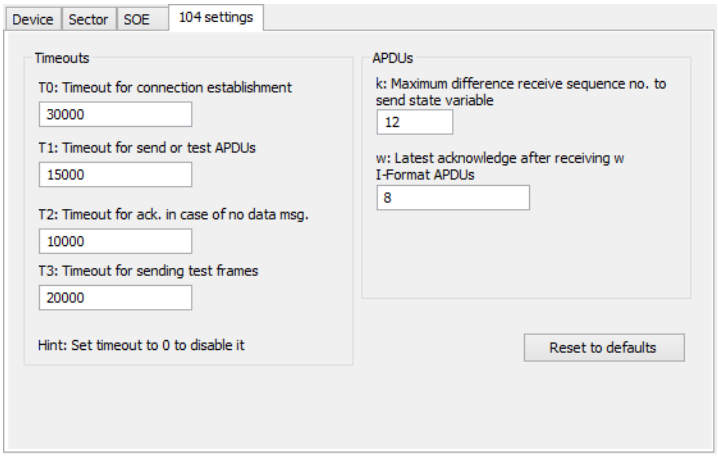
Code	Beschreibung
1011	QCC FRZ is only partly supported.
1012	Counter interrogation already active.
1014	Activation comfirmation bereits bestätigt.

FEHLERMELDUNGEN

Meldung	Beschreibung
Unsupported qualifier of counter interrogation. <command>	Unbekannter QCC. RQT wird nur mit 5 unterstützt. Für `C_CI_NA_1` (`Counter interrogation`) gilt, dass `RQT != 5` gleich wie `RQT = 5` behandelt wird.
Counter interrogation processed without reset. < command>	Warnung. Reset wird nicht unterstützt.
Can't advise IT data point. Device:(<device>)<device> Sector:(<COA>)<Sektorname> adv:0x<Advise-ID>	Datenpunkte für Integrated Totals konnten nicht angefordert werden.
IT Values lost, queue full! Device:(<device>)<device> Sector:(<COA>)<Sektorname> <Wertänderung>	Der Puffer für die Wertänderungen der IT IOs ist voll. Die Wertänderung wurde verloren.

10.1.5 104 settings

In der Registerkarte **104 settings** stellen Sie die Timeout-Zeiten des Link Layers und die Anzahl an APDUs ein. Die Einstellungen sollen für Slave und Master gleich sein. Überschreitet ein Master ein Timeout, oder die Anzahl der zu bestätigen Frames, schließt der Slave die Verbindung.



Hinweis: Die Eingabefelder dieser Registerkarte stehen nur dann zur Verfügung, wenn als Protokoll 870-104 ausgewählt wurde.

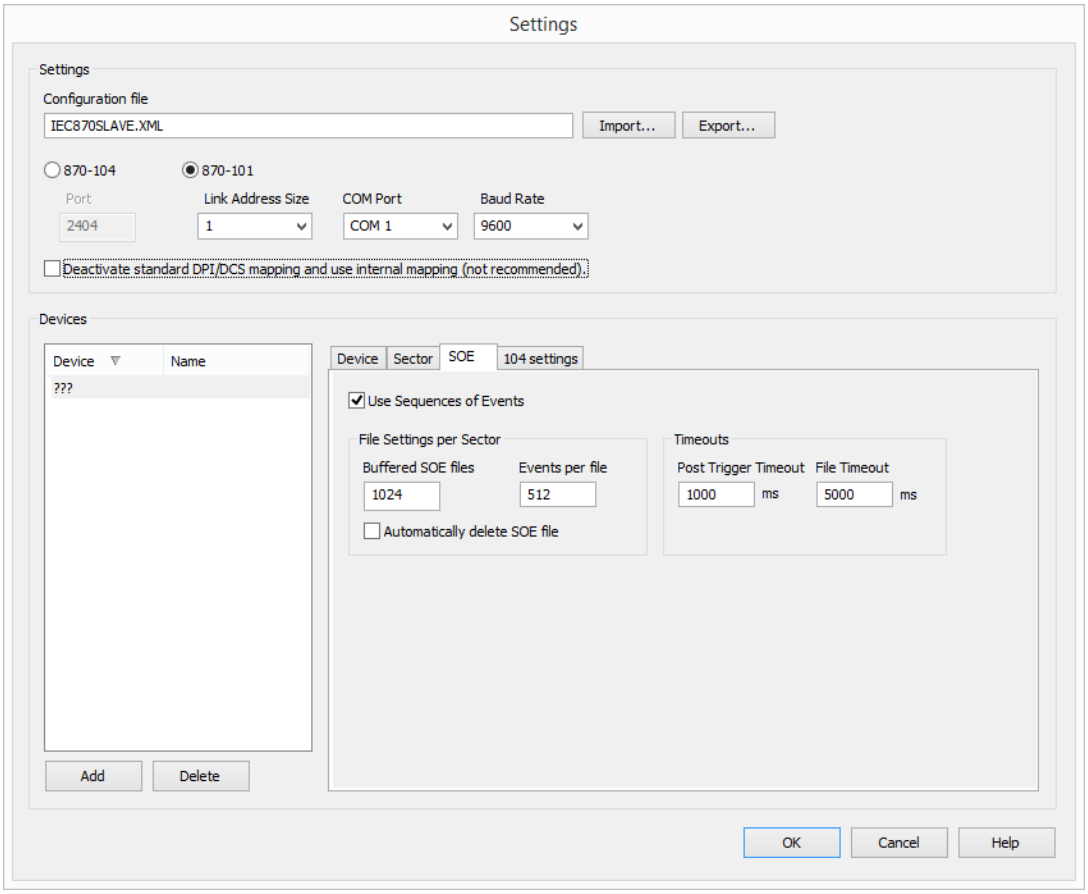
Parameter	Beschreibung
Timeouts	Timeout Einstellungen der Verbindung
T0: time-out of connection establishment	Timeout für den Verbindungsaufbau. Wertbereich 0 - 4294967295 Default: 30000 In der Zeit soll auch wenigstens ein U-Frame vom Master erfolgen.
T1: Time-out send or test APDUs	Timeout für Frame-Bestätigung durch den Master. Wertbereich: 0 - 4294967295 Default: 15000
T2: Time-out for ack. in case of no data msg	Timeout, innerhalb dessen eine Bestätigung durch den Master erfolgen soll, wenn kein Datenaustausch zu Stande kommt. Wertbereich: 0 - 4294967295 Default: 10000
T3: Time-out for sending test frames	Zeit, nach der ein U-Frame zum Master gesendet wird, wenn keine Daten zu übertragen sind. Wertbereich: 0 - 4294967295 Default: 20000
APDUs	*Einstellungen für den ISO 7816-Standard: Application Protocol Data Unit (APDU).*
k: Maximum difference receive sequence nr to send state	Anzahl der maximal durch den Master noch nicht bestätigten I-Frames. Wertbereich: 0 - 4294967295 Default: 12
w: Latest acknowledge after receiving w I-Format APDUs	Anzahl der erhaltenen I-Frames, nachdem eine Bestätigung gesendet wird. Wertbereich: 0 - 4294967295 Default: 8

Info

*Die Schaltfläche **Reset to default** setzt alle Eingabe auf die vom IEC 60870-5-104 Standard vorgegebenen Werte zurück.*

10.1.6 DPI/DCS Mapping

Es kann konfiguriert werden ob DPI/DCS (`double-point information`/`double command state`) konform zu IEC 60870 Standard oder in der zenon internen Darstellung übertragen wird. Default ist die Übertragung zwischen dem Process Gateway und einem 870-Master entsprechend der Norm.



Einstellung	Funktionalität
Deactivate standard	eine Einstellung für Kompatibilität beim Konvertieren von Projekten aus sehr frühen zenon Versionen oder Verwendung mit zenon Treibern für die das **Mapping** von **Double Point Values** ebenso deaktiviert wurde. Default: `Inaktiv`. `Inaktiv`: Werte für DPI/DCS (und RCS) werden wie folgt gemappt, was auch garantiert das korrekte Verhalten für den Modul **Befehlsgabe** sowie für die Kommunikation mit einen 870-Master: ▸ zenon Wert `0` <-> DPI/DCS/RCS Wert `1` (= `OFF/LOWER`) ▸ zenon Wert `1` <-> DPI/DCS/RCS Wert `2` (= `ON/HIGHER`) ▸ zenon Wert `2` -> DPI Wert `0` (`intermediate`) ▸ zenon Wert `3` -> DPI Wert `3` (`failure/error`) DPI sind die "double-point information" IOs; DCS sind die "double command" IOs; RCS sind die "regulating step command"; Mapping ab zenon Version 8.00 `Aktiv`: Mapping wird deaktiviert. Werte für DPI/DCS/RCS entsprechen direkt den Werten im zenon. Wird nicht empfohlen. Hinweis: Aus Kompatibilitätsgründen diese Einstellung wird automatisch aktiviert, wenn eine XML-Datei mit Wert `1` im Feld `Version` in der Konfiguration des Process Gateways eingelesen wird.

10.1.7 Konfiguration via XML-Datei

Die Konfiguration des IEC870SI wird als XML-Datei abgelegt. Beim Start des Gateways wird die aktive XML-Datei eingelesen. Die aktive Konfiguration wird in der INI-Datei (auf Seite 19) vermerkt, zum Beispiel:

```
[IEC 870]
SETTINGS=IEC870SLAVE.XML
```

Der Name ist frei wählbar, der Pfad entspricht dem der INI-Datei.

Hinweis: Unterschiedliche Konfigurationen können Sie über die Schaltflächen **'Export'** und **'Import'** im Hauptfenster exportieren und wieder einlesen.

AUFBAU DER XML-DATEI

Die Felder in der XML Datei entsprechen Einstellungen, welche in den jeweiligen Konfigurationsdialogen zugelassen sind.

Die Werte und Struktur der XML Datei werden beim Lesen verifiziert. Fehlerhafte Einträge führen zu einer Fehlermeldung, die auch im LOG - **Diagnosis Viewer** - protokolliert wird.

> **Achtung**
>
> *XML-Einträge sind case sensitive. Achten Sie auf Groß- und Kleinschreibung.*

Name	Eigenschaft
process_gateway_IEC870Slave	Der Wurzelknoten muss diesen Namen haben, sonst wird der Ladevorgang mit einem Fehler beendet.
Version	Beinhaltet die aktuelle Version der Attribute und Knoten.
Is101active	Data Layer Auswahl: ▸ `0: 870-104` ▸ `1: 870-101 unbalanced` ▸ `2: 870-101 balanced`
DeActDPIStd	DPI /DCS konforme (IEC60870 konform) Datenübertragung oder Leitsystem internes Format

Device_x	Knoten für die Geräte-Konfiguration. Beim Speichern wird X durch eine laufende Nummer beginnend mit 1 ersetzt. Hinweis: Beim Import ist keine bestimmte Knotenbenennung erforderlich. Für jeden Knoten wird ein **Device** erzeugt.
Version	ab 6 kennzeichnet eine XML Datei mit den Feldern für SOE Unterstützung
LinkAdr	Im Protokoll 101 wir hier die `Link Address` gesetzt. Diese muss für alle Geräte eindeutig sein. Für das Protokoll 104 hat der Eintrag keine Bedeutung.
IpAdr	Im Protokoll 104 wird hier die IP-Adresse des Masters eingetragen. Diese muss bei allen Geräten eindeutig sein. Für das Protokoll 101 hat der Eintrag keine Bedeutung.
EnableSOE	▸ 0 = inaktiv ▸ 1 = SOE aktiv
MaxBufferedFilesSOE	die maximale Anzahl der gepufferten SOE Dateien.
AutodeleteFilesSOE	▸ 0 = kein Autodelete ▸ 1 = Autodelete wenn die maximale Anzahl der SOE Dateien erreicht wurde
MaxEventsPerFileSOE	Maximale Anzahl der ASDU pro SOE Datei
PostTrgToSOE	**Post Trigger Timeout** in [ms]
FileToSOE	**File Timeout** in [ms]

Sector_x	Knoten für die Konfiguration eins Sektors. Beim Speichern wird X durch die **COA** ersetzt. Hinweis: Beim Import ist keine bestimmte Knotenbenennung erforderlich. Für jeden Knoten wird ein Sektor erzeugt.
Version	ab 4 kennzeichnet eine XML Datei mit den Feldern für **Select Routing** unterstüzung.
COA	`Common Address of ASDU` (`Common Object Address`) Muss pro Sektor beim Device eindeutig sein.
BSTime	Periode für `Background Scan`. Einheit: ms. 0 deaktiviert den Background Scan. Default: 0
CYTime	Periode für zyklischen Datentransfer Einheit: ms. 0 deaktiviert den zyklischen Transfer. Default: 0 Verfügbar ab Version 2

IO_x	Knoten für die Konfiguration eines `Information Object`. Beim Speichern wird X durch eine laufende Nummer beginnend mit 1 ersetzt. Hinweis: Beim Import ist keine bestimmte Knotenbenennung erforderlich. Für jeden Knoten wird ein `IO` erzeugt.
Version	▸ ab 4 kennzeichnet eine XML Datei mit den Feldern für SOE Unterstützung; ▸ ab 5 kennzeichnet eine XML Datei mit den Feldern für Unterstützung der Hysterese.
TypeId	`Type Identification` Es wird beim Laden geprüft, ob die **Type ID** mit dem Datentyp der angegebenen Variablen möglich ist.
ScadaVar	Name der zenon Variable, Syntax: [ScadaProjektName#]ScadaVariablenName. Variablen ohne Projektbezug werden im Defaultprojekt gesucht (INI Datei - Sektion [GENERAL]: PROJECT=). Beim Laden der XML-Datei wird verifiziert ob die angegebene Variable im Projekt existiert.
IOA	`Information Object Address` In Kombination mit der **TypeId** muss diese pro Sektor und Richtung (controlling/monitoring) eindeutig sein.
KindOfTransfer	Gibt die Art der Datenübertragung für `IO` in monitoring direction an. ▸ 0 Spontan (`COT`=3) ▸ 1 `Background scan` (`COT`=2) ▸ 2 Periodisch/Zyklisch (`COT`=1) ▸ 3 Aus; keine Übertragung an 870-Master ▸ 4 Spontaner Transfer mit Pufferung bei Kommunikationsausfall Default: 0 Verfügbar ab Version 3
TrgSOE	▸ 0 = inaktiv ▸ 1 = das `IO` fungiert als Trigger für SOE
EvDetectionSOE	Bestimmt wann eine Änderung am Wert oder Statusbits des `IO` ein schreiben in die SOE Datei auslösen kann. ▸ 0 SOE deaktiviert ▸ 1 Alle Änderungen ▸ 2 Steigende Flanke ▸ 3 Fallende Flanke ▸ 4 Steigende und fallende Flanke

HysteresisNegative	Wert der Negativen Hysterese.
HysteresisPositive	Wert der Positiven Hysterese.

Beispiel Konfiguration für das Protokoll 870-101:

```
process_gateway_IEC870Slave
  Version = "1"
  Is101active = "1"
  ServerPort = "2404"
  ComPort = "4"
  Baud = "9600"
  LAFSize = "1"
  Device_1
    Version = "5"
    Name = "device1"
    CommandsActivated = "1"
    LinkAdr = "5"
    IpAdr = "0.0.0.5"
    MaxDataLen = "255"
    ShortPulseDuration = "10000"
    LongPulseDuration = "10000"
    COASize = "2"
    COTSize = "2"
    IOASize = "3"
    FTDir = ""
    Sector_7
      Version = "1"
      COA = "7"
      Name = "sector 7"
      IO_1
      IO_2
      IO_3
        Version = "2"
        TypeId = "1"
        ScadaVar = "SUBPRJ#Bool"
        IOA = "4"
      IO_4
      IO_5
    Sector_2
  Device_2
```

Beispiel Konfiguration für das Protokoll 870-104:

```
process_gateway_IEC870Slave
  Version = "1"
  Is101active = "0"
  ServerPort = "2404"
  ComPort = "4"
  Baud = "9600"
  LAFSize = "2"
  Device_1
  Device_2
    Version = "5"
    Name = "device1"
    CommandsActivated = "1"
    LinkAdr = "3232235636"
    IpAdr = "192.168.0.116"
    MaxDataLen = "253"
    ShortPulseDuration = "10000"
    LongPulseDuration = "10000"
    COASize = "2"
    COTSize = "2"
    IOASize = "3"
    FTDir = "D:\Projects\rq1575\Dev1\"
    Sector_7
      Version = "1"
      COA = "7"
      Name = "any sector"
      IO_1
      IO_2
      IO_3
      IO_4
      IO_5
      IO_6
      IO_7
        Version = "2"
        TypeId = "1"
        ScadaVar = "SUBPRJ#Bool"
        IOA = "4"
      IO_8
      IO_9
    Sector_2
```

.CSV Export und Import

Konfigurationen des Process Gateways können in eine Textdatei exportiert oder aus einer Textdatei ins Process Gateway importiert werden.

Die erste Zeile in der CSV-Datei muss zwingend ein Header mit den Namen der Eigenschaften sein. Die einzelnen Eigenschaften müssen zwingend mit einem TAB getrennt sein.

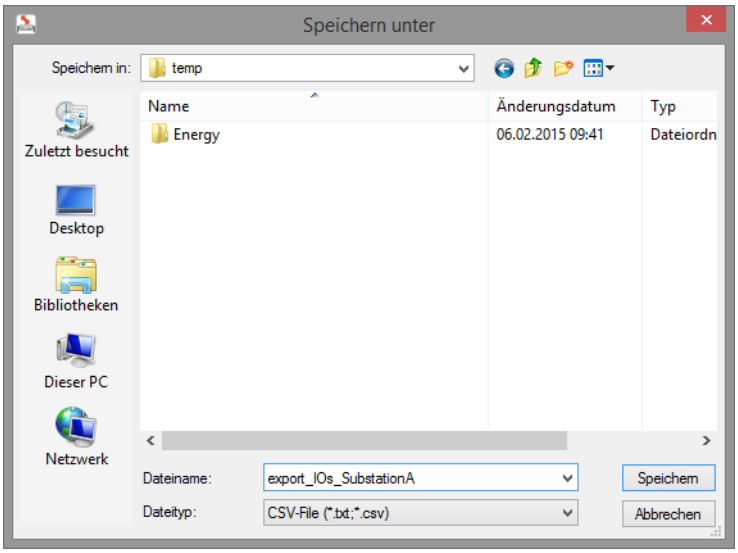
Pro Zeile der CSV-Datei werden folgende Informationen des IOs gespeichert:

- LinkAdr oder IpAdr des Devices
- COA des Sektors
- alle Einstellungen des IOs

Beim Import werden typischerweise neue IOs im Process Gateway angelegt.
Zusätzlich kann die Eigenschaft `KindOfTransfer` beim Import dazu benutzt werden, bereits bestehende IOs der Process Gateway Konfiguration zu aktualisieren oder zu löschen.

EXPORT

Führen Sie folgende Schritte aus um eine Process Gateway Konfiguration zu exportieren:



1. Klicken Sie die Schaltfläche **Export...** im Konfigurationsdialog.
   Der Dateiauswahldialog wird geöffnet.
2. Wählen Sie in der Dropdownliste **Dateityp** den Eintrag `CSV-File`.
3. Benennen Sie die Exportdatei im Eingabefeld **Dateiname**
4. Klicken Sie auf **Speichern**.
   Die Datei wird exportiert und am gewählten Ablageort gespeichert.

IMPORT

Die Konfiguration des Process Gateway kann aus einer .CSV-Datei erstellt werden oder mit einer .CSV-Datei erweitert werden. Die .CSV-Datei kann von einem Export aus einem anderen Process Gateway stammen oder mit einem externen Programm (z.B. MS Excel) erstellt werden.

Führen Sie folgende Schritte aus um eine .CSV-Datei in die Konfiguration des Process Gateway zu importieren:

1. Klicken Sie die Schaltfläche **Import...** im Konfigurationsdialog.
   Der Dateiauswahldialog wird geöffnet.
2. Wählen Sie in der Dropdownliste **Dateityp** den Eintrag `CSV-File`.
3. Wählen Sie die Datei.
4. Klicken Sie auf Öffnen.
   Die Datei wird geladen und die Konfiguration des Process Gateways wird aktualisiert.

REGELN FÜR IMPORT

Für den Import und die Anpassung gelten folgende Regeln:

1. `LinkAdr` oder `IpAdr`:

- der Import einer `LinkAdr` ist nur dann erlaubt, wenn das Process Gateway bereits für 870-101 konfiguriert ist.
- der Import einer `IpAdr` ist nur dann erlaubt, wenn das Process Gateway bereits für 870-104 konfiguriert ist.

  Im Falle eines Konflikts zwischen Protokoll und Adressierung gilt:

  - Ist im Process Gateway noch kein Device angelegt, passt der Importprozess die Protokollenstellung entsprechend an und legt im Process Gateway ein Device an. Dieses Device wird mit Defaulteinstellungen konfiguriert.
  - Ist im Process Gateway bereits ein Device (für ein anderes Protokoll) vorhanden, wird der Import mit einer Fehlermeldung abgebrochen.

1. Sektor

- Wurde im Device noch kein Sektor für den zu importierenden COA der Importdatei konfiguriert, erstellt der Importprozess einen neuen Sektor.
- Anschließend erfolgt der Import des IOs in diesem Sektor.

1. IO
   Die Spalte `KindOfTransfer` bestimmt den Bearbeitungsschritt des IOs:

- 0...4 - normaler Wertbereich für `KindOfTransfer`
  erstellt neuen IO
- 100...104
  Löscht aus dem Sektor das IO welches genau `IOA` und `TypeID` hat
- 300...304
  löscht aus dem Sektor das IO welches mit `ScadaVar` verknüpft ist.
- 200...204
  aktualisiert das IO welches genau IOA hat und dessen `TypeID` im passenden Bereich liegt.
  Bereiche gelten:

  - 1...37
    Monitoring Direction
  - 45...126
    Controlling Direction
  - 0
    für interne TypeIDs
- 400...404
  aktualisiert IO welcher mit `ScadaVar` verknüpft ist.

Nach diesem Bearbeitungsschritt wird anschließend `KindOfTransfer` auf den normalen Wertebereich umgerechnet.

Beispiel

Im Process Gateway wurde ein **IO** mit `COA=1`, `IOA=1` und `TypeID=1` für spontane Übertragung konfiguriert sowie mit der Variable mit dem Namen `Var1` verknüpft.

Die Importdatei enhtält für die Variable `Var1` eine Zeile mit `COA=1`, `IOA=100`, `TypeID=30` und `KindOfTransfer=400`. Dadurch wird für die Variable `Var1` der `TypeID` des **IO** von `1` auf `30` geändert und `IOA` wird von `1` auf `100` geändert.

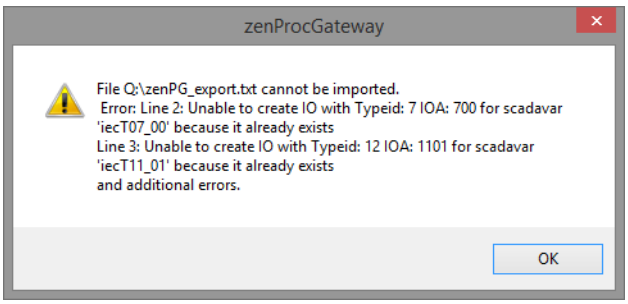
Die Übertragung - **Data Transfer** - bleibt spontan.

FEHLERBEHANDLUNG

Wird während des Importvorganges ein Fehler erkannt, wird ein Eintrag in der LOG-Datei erstellt. Zusätzlich wird im Process Gateway am Ende des Imports ein entsprechender Warndialog angezeigt.

Mögliche Fehlerursachen:

- Ein IO ist noch nicht in einem bereits konfigurierten Sektor vorhanden, jedoch in der Datei mit `KindOfTransfer` zur Aktualisierung vorgesehen.
- Der Import erlaubt es nicht, mehr als ein IO mit bestimmten `IOA` im jeweiligen `TypeID` Bereich anzulegen.
- Auch das Anlegen von IOs für Variablen, die nicht im Projekt vorhanden sind oder einen nicht numerischen Datentyp haben wird als Fehler betrachtet.



Der Header gilt als Zeile mit Nummer 1.

10.2 Sequence of Events (SOE)

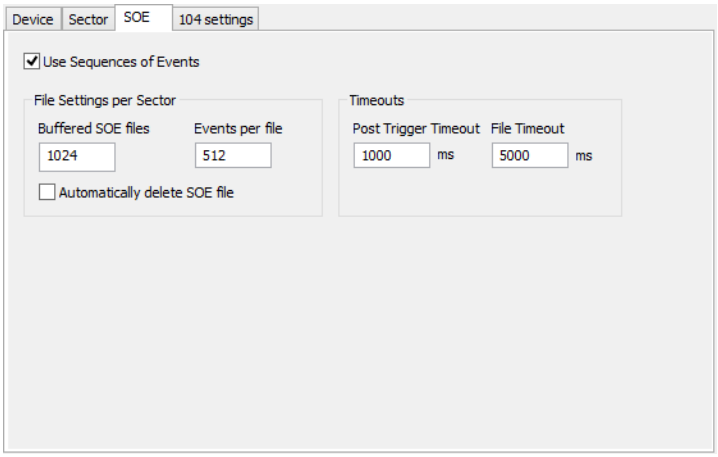
Das **Process Gateway IEC870 Slave** unterstützt die automatische Aufnahme und die Übertragung der SOE Dateien - nach Prozeduren definiert in IEC 60870-5-101, Abschnitt 7.4.11.3 "Transmission of sequences of events" (in a section of a data file).

Beim Sektor muss mindestens ein IO die Option **File Trigger** aktiviert haben, sonst werden keine SOE Dateien erzeugt.

Hinweis: Die Option **File Trigger** finden Sie in der Registerkarte **Sector** - **Information Objects** (auf Seite 176), im Bereich **Sequence of Events (SOE)**.

Die SOE Dateien werden im Ordner für den Dateitransfer abgelegt. Es muss somit in **Device** ein Ordner für den Dateitransfer konfiguriert sein (Option **Directory for File Transfer**).

EINSTELLUNGEN IN DER REGISTERKARTE SOE



Parameter	Beschreibung
Use sequences of events	Wenn aktiv dann ist SOE für das ausgewählte Device aktiviert. Bei `inaktiv` werden alle SOE Einstellungen an den IOs ignoriert. Default: `inaktiv` Hinweis: Der IEC870 Slave speichert in SOE Dateien nur events der IOs für die, in der Registerkarte **Sector** - **Information Objects** (auf Seite 176), das SOE aktiviert wurde.

FILE SETTINGS PER SECTOR

Parameter	Beschreibung
Buffered SOE Files	Wird die eingestellte Anzahl der SOE Dateien im Ordner für den Sektor erreicht, wird eine spontane Verzeichnisübertragung (auf Seite 226) (Typ ID `F_DT_TA_1`) ausgelöst. Bei der doppelten Anzahl von Dateien im

	Ordner wird begonnen die älteste Datei zu löschen (Ringpuffer Funktionalität). Wertbereich `1 bis 1024`. Default: `1024` Achtung: Es wird ein Verzeichnis - jedoch kein Dateien-Transfer automatisch ausgelöst. Die SOE-Dateien muss der 870-Master anfordern. Hinweis: der aktuelle Status des SOE kann über die Variablen, die mit dem Internen Typ T00 (auf Seite 214) verknüpft sind, ausgewertet werden.
Events per File	Anzahl der Wertänderungen welche in eine Datei geschrieben werden bevor diese geschlossen wird. Wertbereich: `1 bis 4096`. Default: `512`
Automatically delete SOE file	Wenn aktiv dann wird auf Reaktion für Typ ID `F_AF_NA_1` (Bestätigung für Datei) für eine SOE Datei diese automatisch gelöscht.

TIMEOUTS

Parameter	Beschreibung
Post trigger timeout	Wenn innerhalb dieses Zeitraums kein neuer ASDU in die Datei geschrieben wird, wird diese geschlossen. Mit jedem geschriebenen ASDU beginnt der Zeitraum von neuem. Wertbereich: `100ms bis 120000ms` Default: `1000ms`
File timeout	Maximale Zeit welche die Datei geöffnet bleibt (unabhängig von der Anzahl der Events). Wertbereich: `100ms bis 120000ms` Default: `5000ms`

ERSTER WERT

Der Datenaustausch zwischen IEC870Sl und der zenon Runtime für SOE beginnt sofort wenn das Gateway fertig initialisiert ist. Der erste Wert welcher zur Initialisierung kommt, löst kein Event aus und wird somit auch nicht in die Datei eingetragen. Das Gleiche gilt wenn die Runtime nach dem Nachladen wieder bereit für einen Datenaustausch ist.

Der Datenaustausch der Daten in Monitoring direction beginnt erst wenn ein Master sich verbunden hat und eine General Interrogation (GI) (auf Seite 210) auslöst.

FILE TRANSFER

Es wird im Zuge der SOE Funktionalität zu transparent (1) auch event (3) als `NOF` unterstützt.

Es wird kein automatischer File transfer bei überschreiten der maximalen Anzahl zu puffernden SOE Dateien ausgeführt.

Das Gateway unterstützt im aktuellen Ausbau spontane Verzeichnisübertragung, jedoch keine durch den Slave aktivierte Dateienübertragung. Die SOE-Dateien muss der 870-Master anfordern.

Hinweis: der aktuelle Status des SOE kann über die Variablen, die mit Interner Typ T00 (auf Seite 214) verknüpft sind, ausgewertet werden.

DIRECTORY TRANSFER

Entsprechend dem IEC 60870 Standard wird bei den folgenden Gründen eine automatische spontane Verzeichnisübertragung (auf Seite 226) bei existierenden Dateien für SOE aktiviert:

- 24 Stunden nach der letzten automatischen Verzeichnisübertragung.
- wenn sich der Master verbindet.
- wenn die maximale Anzahl der zu puffernden Dateien überschritten wurde.

10.2.1 Warnungen

Warnmeldung	Bedeutung
`Sequences of events for device <Devicename> sector <Sectorname> are activated but no trigger defined`	Das Device hat SOE aktiviert, beim Sektor existiert ein IO mit aktiven SOE aber bei keinem IO wurde der Trigger aktiviert. Ohne Trigger wird keine Datei erzeugt. Mindestens ein IO beim Sektor muss den Trigger aktiviert haben.
`SOE value change without trigger and file is closed. Value lost! Device:(<Device ID>)<Device-Name> Sector:(<COA>)<Sektorname> <Wertinformation>`	Eine Wertänderung wurde nicht geschrieben da die SOE Datei nicht geöffnet ist. Code:1010

10.2.2 Fehlermeldungen

Fehlermeldung	Bedeutung
Sequence of events active but no directory for the file transfer defined!	SOE Dateien werden im Ordner für den Dateitransfer abgelegt. Es muss somit ein Ordner für den Dateitransfer vorhanden sein.
SOE Values lost, queue full! Device:(<Device ID>)<Device-Name> Sector:(<COA>)<Sektorname> <Wertinformation>	Der Datenaustausch zu Stack wird für IO's welche SOE aktiviert haben über eine Queue ausgeführt. Diese kann den größeren Wert von 2048 oder 4 * Anzahl der SOE IO's beim Sektor Einträge aufnehmen. Wenn diese Anzahl überschritten ist, wird diese Fehlermeldung erzeugt und die Wertänderung ist verloren. Errorcode: 6
Can't create subdirectory for file transfer. Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> system error:<Systemfehlercode>	Für SOE wird automatisch der Unterordner für den Sektor im Ordner für den Dateitransfer erzeugt. Dieser Fehler zeigt an, dass dieser Ordner nicht erzeugt werden konnte. SOE steht dann für diesen Sektor NICHT zur Verfügung. Errorcode: 10
Systemfehlercode	Beinhaltet den vom Betriebssystem zurückgegebenen Fehlercode.
Can't advise SOE data point. Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> adv:0x<Advise-ID>	Datenpunktanforderung für SOE IO's ist fehlgeschlagen. SOE steht für diesen Sektor NICHT zur Verfügung. Errorcode: 11
Write asdu to file <Dateiname> fail! Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname> <Value> system error: <Systemfehlercode>	Schreiben des ASDU in SOE Datei fehlgeschlagen. Errorcode: 8
Illegal file name <Dateiname> detected! Device:(<Device ID>)<Devicename> Sector:(<COA>)<Sektorname>	Dateinamen für den Dateitransfer haben ein definiertes Format <IOA>.<NOF>. Die Datei hält sich nicht an diese Konvention. Errorcode: 12
Auto delete for file failed! System error: <Systemfehlercode> <Dateitransferinformation>	Errorcode: 9
Dateitransferinformation	Enthält die Information welche den Dateitransfer identifiziert.

10.3 Kommunikation

In diesem Kapitel lesen Sie Details zur Kommunikation im **Process Gateway**:

> **Info**
>
> Wird vom **Process Gateway** ein **ASDU** mit unbekannter `Type-ID` empfangen, wird dies mit `COT 44` negativ an den 870-Master beantwortet.

10.3.1 Start

Beim Start des Process Gateways lädt dieses die INI-Datei. Das **Process Gateway** versucht dann, seine aktive Konfiguration aus der - in der INI-Datei eingetragenen - XML-Datei zu lesen. Ist dies nicht möglich, wird eine Fehlermeldung angezeigt und der Konfigurationsdialog geöffnet.

Wenn der Konfigurationsdialog nicht mit Klick auf den Button **OK** verlassen wird, beendet sich das **Process Gateway**.

> **Achtung**
>
> *Wird das **Process Gateway** unsichtbar gestartet, wird kein Konfigurationsdialog aufgeschaltet. Bei einer ungültigen oder fehlenden Konfiguration wird das Process Gateway automatisch beendet.*

Nachdem IEC870 Slave seine Konfiguration aus XML-Datei erfolgreich eingelesen hat, erwartet er einen Verbindungsaufbau von einen Master. Der IEC 60870 Standard definiert dass eine Session zwischen Master und Slave immer mit einer Generalabfrage (GI - `C_IC_NA_1`) beginnen muss. Deswegen ist der Empfang der GI von dem Master eine Voraussetzung für die spontane Kommunikation. Erst die Generalabfrage löst im IEC870 Slave den Verbindungsaufbau zur zenon Runtime aus und die Anforderung der aktuellen Werten der Variablen für IOs.

Hinweis: öffnet ein Master die Verbindung auf dem Link Layer, sendet jedoch keine Generalabfrage, können Sie versuchen die GI mit Senden eines ASDU T70 - End of Initialization (auf Seite 212) - provozieren.

10.3.2 Generalabfrage

Die Generalabfrage (GI) löst im IEC870Slave den Verbindungsaufbau zur zenon Runtime aus. Der Empfang der GI - `C_IC_NA_1` - von einem Master ist die Voraussetzung für die spontane Kommunikation, `Background scan` oder `Periodic data transfer`. Solange IEC870 Slave die GI noch nicht empfangen hat sendet er keine Werte zu Master.

> **Achtung**
>
> Gemäß IEC60870 Standard muss ein Master die Kommunikation mit einer Generalanfrage beginnen. Deswegen schließt der Slave die Verbindung, wenn der Master in der vorgesehenen Zeit `(T0)` keine GI gesendet hat.
>
> Hinweis: öffnet ein Master die Verbindung auf dem Link Layer, sendet jedoch keine Generalabfrage, können Sie versuchen die GI mit Senden eines ASDU T70 - End of Initialization (auf Seite 212) - provozieren.

Nach dem Empfang des GI wird die Verbindung zur zenon Runtime sektorweise (**COA**) aufgebaut. Kann die Verbindung zur Runtime nicht aufgebaut werden, erfolgt eine negative Bestätigung der Generalabfrage (`COT_actcon + PN`). Die für den Sektor bereits aktive Generalabfrage wird beendet und eine Warnung protokolliert.

Eine GI für im Slave nicht existierenden Sektor (unbekannte **COA**) wird negativ bestätigt (**COT**=`46 + PN`) und die Verbindung zum Master abgebrochen. Dies ist ein Indikator für einen Konfigurationsfehler des Systems.

Andere Konfigurationsfehler des Systems werden toleranter behandelt. Der Empfang von ungültigen Werten wird in der LOG-Datei protokolliert. Für diesen Fall gilt:

- `COT != 6` wird mit `COT 45` und negativ beantwortet. Die Verbindung bleibt bestehen.
- `QOI != 20` wird positiv bestätigt und löst trotzdem `Station interrogation` aus. Dies entspricht dem Verhalten gleich wie bei `QOI = 20`.
- `IOA !=0` wird gleich wie `IOA=0` behandelt.

Für Generalabfragen unterstützt der Slave die globale Adresse **COA**=`0xFF(FF)`. Weitere Informationen dazu finden Sie im Kapitel Broadcasts (auf Seite 229). Damit kann ein Master mehrere Sektoren mit nur eine GI abfragen. Dies ist eine vom Standard vorgesehene Alternative, die ein Master verwenden kann. Sonst muß ein Master die Sektoren nacheinander (einer nach dem anderen) abfragen.

Wenn der Master eine weitere GI zusendet, während die aktuelle noch nicht abgeschlossen ist, wird die laufende GI abgebrochen und die neue begonnen. Überlappende GIs werden als Warnung protokolliert. Dies ist ein Hinweis auf ein nicht Standard-konformes Verhalten eines Masters.

Wertänderungen für IOs, deren Wert bereits im Rahmen der aktuellen GI dem Master gemeldet wurde, werden gepuffert solange die GI - wegen andere IOs im selben Sektor (**COA**) - noch läuft. Beinhaltet ein Sektor viele IOs deren Werte sich oft ändern und ist die Verbindung zum Master vergleichsweise

langsam kann der Puffer schneller wachsen als der Master die Werte abnimmt. Dies kann passieren wenn z.B. die **Baud Rate** klein ist, die **Max.user data** eingeschränkt wurde oder der Master langsam reagiert. In solchen Fällen kann das Verteilen der IOs auf mehr Sektoren ratsam sein.

> **Achtung**
>
> Wird der interne Puffer mit Wertänderungen überfüllt bevor die GI abgeschlossen sein könnte, wird die aktuelle Verbindung zum Master abgebrochen.
>
> Weitere Informationen dazu finden Sie im Kapitel Übertragung der Wertänderungen (auf Seite 213).

Während ein Projekt in der Runtime nachgeladen wird (auf Seite 213), kann die Verbindung zur Runtime nicht hergestellt werden. Die Generalabfrage schlägt dann fehl.

Die GI und das Puffern der Wertänderungen sind unabhängig für unterschiedlichen **Devices** (für unterschiedlichen Master). Ein Master kann die laufende Generalabfrage durch ASDU<100> mit `COT_deact(8)` deaktivieren.

Wenn die Verbindung zur Runtime aufgebaut wurde, die Runtime aber keine Werte für manche IOs liefert, dann kann der Slave die GI zwar bestätigen jedoch nicht abschließen (kein `COT_actterm`). Dies gilt für jene IOs, die aufgrund der aktuellen Generalabfrage (= GI) an den Master gemeldet werden sollen. Sehr oft liegt der Fehler, dass keine Werte geliefert werden an einem Konfigurationsfehler. Beispiel: eine schreibgeschützte (= write-only) zenon Variable (ein Output oder Befehl) ist im Slave mit einem IO mit dem **Type ID** einer Meldung (`T01-T37`) verknüpft.

Der Slave protokolliert die Information über **IOA** und **COA** Adressen der betroffenen IOs bis er die benötigten Werte erkennt. Diese Protokollierung wird zyklisch durchgeführt. Diese Information wird mit Level `DEEPDEBUG` in der LOG-Datei protokolliert und kann mit dem `Diagnosis Viewer` ausgelesen werden.

Text: "Information object <IOA> missing value. Device: <IP> Sector: <COA>".

10.3.3 Verbindungsende und Resets

Der Reset im Slave wird ausgelöst, wenn sich ein Master verbindet:

- 870-101: Bis der Funktionscode für den Reset (0) empfangen wird, werden alle Kommandos abgelehnt. Der Code löst den Reset im LinkLayer und Applikationslayer aus.
- 870-104: Durch den Abbau der Verbindung kann ein Master die Unterbrechung sofort erkennen und erneut mit dem Initialisierungsablauf beginnen.

BEENDEN DER KOMMUNIKATION

Generell wird die Kommunikation beendet, wenn:

- das Process Gateway beendet wird
- Änderungen in der Konfiguration des Slaves aktiviert werden
- die Runtime ein Nachladen ausführt
- der Master die Verbindung aufgebaut hat, dann jedoch keine Generalanfrage gesendet wurde
- die Liste der noch nicht gesendeten Wertänderungen beim Sektor übergelaufen ist
- ein nicht unterstütztes Kommando empfangen wird (Kommando wird negativ beantwortet und die Verbindung aus Sicherheitsgründen beendet)
- ein Kommando mit nicht unterstützter Adresse empfangen wird (Kommando wird negativ beantwortet und die Verbindung aus Sicherheitsgründen beendet)

'Reset process command' - T105 - Anforderung

Der Slave kann über Erhalt eines `C_RP_NA_1` Kommandos die zenon Runtime informieren. Dafür:

- legen Sie einen IO vom Typ `T105` mit der **IOA** 0 an

Wenn ein Master das Kommando `C_RP_NA_1` versendet, wird der Wert der Variable zuerst auf 0 und dann auf den Wert von `QRP` gesetzt.

Der Slave bestätigt dem Master der Erhalt des Kommandos und ändert der Wert der entsprechenden `T105` Variable für die Runtime aber sonst hat im Slave das Kommando keine weitere Auswirkung. Das optionale Senden des 'End of initialization' (`T70`) wird nicht automatisch ausgeführt.

'End of Initialization' - T70 - Senden

Vom Prozessleitsystem kann veranlasst werden, dass ein `ASDU<70>` - `M_EI_NA_1` (End of initialization) - an den Master gesendet wird. Dafür:

- legen Sie einen IO vom Typ `T70` mit **IOA** 0 an.
- und verknüpfen mit einer Internen Variable mit Datentyp `UINT`

Die Übertragung wird ausgelöst, wenn der aktuelle Wert der Variablen `126` ist und dieser auf `127` geändert wird. Als COI (Cause of initialization) wird 2 (remote reset) verwendet.

Hinweis: manche Master senden die Generalabfrage (GI) erst wenn sie ein End of initialization empfangen haben.

PROJEKTIERUNGSTIPP

In Systemen wo der 870-Master ein End of initialization erwartet um die Generalabfrage zu senden, können Sie den Wert der Variable, die mit IO mit `T70` und **IOA** `0` verknüpft wurde zwischen `126` und `127` ändern, wenn die andere Variable mit `T00` und **IOA** `1` (Master verbunden) (auf Seite 214) den Wert ändert. Erstellen Sie dafür zwei **Sollwert setzen** Funktionen für `T70` Variable und verknüpfen diese Funktionen in den **Grenzwerten** der `T00 IOA 1` Variable:

- bei Wert 0 der T00 IOA 1 Variable - setzen 126 auf T70 Variable;
- und bei Wert 1 - setzen 127.

10.3.4 Übertragung der Wertänderungen

Der IEC870Slave erhält vom zenon Runtime sämtliche Wertänderungen der angeforderten Variablen. Da dadurch die Liste der noch nicht zum Master übertragenen Prozessinformationen sehr groß werden kann, ist die Liste der ausstehenden Wertänderungen begrenzt.

Limit: wenigstens 2048 Werte oder 4 x Anzahl der IOs pro Sektor.

Wenn ein Master die Werte langsamer empfängt als sie im Prozessleitsystem entstehen, kann die Liste dieses Maximum überschreiten. Denn der Slave kann dann nicht mehr die noch nicht übertragenen Wertänderungen puffern. Deswegen beendet der Slave die Verbindung zum Master. Somit wird der Master aufgefordert, die nächste Verbindung mit einer Generalabfrage anzufangen, damit er sich aktuelle, nur letzte Variablenwerte holt.

Achtung

Immer wenn Datenverlust möglich ist (verlorene Wertänderungen), wird ein Reset des Link Layers erzwungen .

Der Master muss in der Lage sein, eine Unterbrechung der Kommunikation zu erkennen, um mit einer neuen Generalabfrage reagieren zu können. Daher werden keine Processkommandos ausgeführt, solange der Master keinen Reset des Links veranlasst hat.

10.3.5 Nachladen der Runtime

Wird eines der Projekte in der zenon Runtime nachgeladen, dann wird die Verbindung zu allen Projekten abgebaut und die Verbindungen zu Master abgebaut. Für jede Gerät (`Device`) im IEC870 Slave wird ein Reset des Link Layers ausgelöst. Danach gelten die Regeln für der Start der Kommunikation, z.B. es wird eine Generalabfrage erwartet.

Achtung: Solange das Nachladen nicht abgeschlossen ist, kann keine Verbindung zur Runtime aufgebaut werden. Damit ist auch nicht möglich die Generalabfrage positiv abzuschließen.

Hinweis: Nach dem Nachladen der Runtime können Sie auch - mit der Variable verknüpft zu `T00 IOA 8` - auch Nachladen der XML Konfiguration des Process Gateway auslösen.

10.3.6 Datenprioritäten

Der Treiber unterstützt class 1 und class 2 Daten.

Die spontane Kommunikation, **Background scan** und **Periodic data transfer** gehören zur class 1.

Sämtlicher Dateitransfer wird als class 2 abgearbeitet. Damit ist sichergestellt, dass Wertänderungen auch während eines Filetransfers an den Master übertragen werden.

10.3.7 Interner Typ T00 (Statusvariablen)

Der interne Typ `T00` dient zu Steuern und Überwachen des Verhaltens des IEC870 Slave Process Gateways. Die T00 Variablen werden nur zwischen dem Slave und dem Prozessleitsystem kommuniziert, nicht zu 870-Master.

Die **IOs** mit `T00` sollen mit numerischen (`UINT`, `DWORD` etc.) zenon Variablen verlinkt werden, z.B. vom **Intern-Treiber**.

IOA 1 - MASTER VERBUNDEN

Die Variable liefert die Statusinformation ob bereits eine Link Layer Verbindung zu dem 870-Master besteht. In **Device**, bei beliebigen Sektor (**COA**), legen Sie einen **IO** vom Typ `T00` mit der `IOA 1` an. Der Wert wechselt bei der Verbindung von 0 auf 1.

Hinweis: Öffnet ein Master die Verbindung auf dem Link Layer, ohne dass eine Generalabfrage (auf Seite 210) gesendet wird, so wird das **Process Gateway** nach dem Ablauf von **T0** Timeout die Verbindung schließen. In solchen Fällen können Sie versuchen, die GI von Master zu provozieren, z.B. durch Senden eines ASDU T70 - End of Initialization (auf Seite 212).

IOA 2 - BEFEHLE ABWEISEN ODER IGNORIEREN

Die Variable bestimmt wie der IEC870 Slave umgehen soll mit Befehlen, die der 870-Master zusendet. Es kann pro Sektor (**COA**) entschieden werden, ob Befehle in control direction ausgeführt werden. Wenn kein **IO** projektiert ist, sind die Befehle freigegeben.

Hinweis: Die Befehle des Masters können auch über die Eigenschaft **Commands active** (auf Seite 166) generell abgewiesen werden.

Die zenon Variable wird zu **IO** mit `T00` und `IOA 2` zugeordnet. Sie können die selbe Variable in allen Sektoren (**COA**) verknüpfen. Die Variable darf keinen Status `INVALID` haben.

Die Bedeutung der Variable:

Wert	Beschreibung
0	Disabled Die ASDUs mit Befehlen werden abgelehnt. ▸ Für abgelehnte Befehle erhält ein Master als Bestätigung `COT_actcon (7) + PN` (negativ). ▸ Abgelehnte Befehle erzeugen einen Warneintrag in der LOG-Datei. ▸ die zenon Runtime wird über Befehle nicht informiert. Beispiel: Anhängig dem Modus local/remote einer Substation mit Process Gateway zum entfernten 870-Master. Der entfernte 870-Master kann über ein **IO** in monitoring direction (z.B. mit `T30`) über den aktuellen Modus informiert werden, und während die Substation sich in lokalem Modus befindet, sind die remote Befehle unerlaubt.
1	Enabled Befehle sind freigegeben.
2	Silent mode Die ASDUs mit Befehlen werden ignoriert. ▸ Keine Reaktion auf die ASDUs mit Befehlen; ohne Warnung im Log. ▸ die zenon Runtime wird über Befehle nicht informiert. Beispiel: Redundante Verbindung zu einen 870-Master von zwei Process Gateways, die mit der gleichen Konfiguration gestartet wurden: auf zwei Rechner in zenon **Netzwerk**, oder auf einen Rechner mit 2 Ethernet Karten, oder mit unterschiedlichen IP Ports. Ein Befehl dieses Masters soll nur über einer der Process Gateways an die zenon Runtime weitergeleitet werden.
3....	reserviert für zukünftige Versionen

Info

Zugehörige Warnmeldung:

`Commands for sector deactivated, execution prohibited.` *<Kommandotext>*

Die Bedeutung: Ein Befehl in Control-Richtung wurde empfangen, aber Aufgrund eines fehlenden ***enable*** *abgelehnt.*

IOA 3 - SOE: FILES OVERFLOW

Statusvariable wird auf 1 gesetzt wenn die Anzahl der SOE Dateien im Transferverzeichnis die Anzahl der maximal gepufferten Dateien überschreitet.

IOA 4 - SOE: CURRENT FILE STATUS

Variable liefert Statusinformationen über die aktuell od. zuletzt bearbeitete SOE Datei:

0 - Initialisierung. Solange keine SOE Datei geschrieben wurde.

1 - SOE Datei ist geöffnet

2 - Datei wurde geschlossen

3 - Zu viele SOE Dateien im Verzeichnis

4 - Fehler beim Schreiben der SOE Datei

5 - Fehler beim Erzeugen der SOE Datei

IOA 5 - STATUS OF FILE TRANSFER

Variable liefert Statusinformation über den Dateitransfer in `Monitoring Direction` ("`Standard Direction`" - vom Master zum Slave).

0 - Kein Transfer aktiv

1 - Sektion wird ausgewählt

2 - Sektion wird angefordert

3 - Auf Sektionbestätigung wartend

4 - Auf Dateitransferbestätigung wartend

255 - Fehler beim Transfer

IOA 6 - SOE: ANZAHL DER DATEIEN

Variable liefert die Anzahl der im Transferverzeichnis vorhandenen SOE Dateien.

IOA 7 - STATUS OF FILE TRANSFER IN REVERSE DIRECTION

Variable liefert Statusinformation über den Dateitransfer in `Control Direction` ("`Reverse Direction`" - vom Slave zum Master).

Die Information wird in ein `DWORD` wie folgt integriert: [IOA des übertragenen Files] * 256 + [Statusinformation].

Mögliche Statusinformation:

1 - Transfer läuft

2 - Transfer läuft - der Transfer, welcher zuvor noch nicht abgeschlossene wurde, wird durch den aktuell begonnenen Transfer abgebrochen.
Für den abgebrochenen Transfer wird ebenfalls der Status 3 geschrieben. Da jedoch der neue Status sofort anschließend geschrieben wird, kann im Falle von zenon nicht garantiert werden, dass dieser in der RT sichtbar wird.

3 - Transfer wurde aufgrund Fehler abgebrochen.

4 - Transfer wurde abgeschlossen und die Datei ist verfügbar auf dem Laufwerk.

IOA 8 - NACHLADEN DER XML KONFIGURATION

Im IEC870 Slave Process Gateway kann mit dem `IOA 8` mit Type ID `<0> Internal Status` eine zenon Variable verknüpft werden.

Hat diese verknüpfte Variable in der zenon Runtime den Wert `1` wird die Konfiguration für das Process Gateway neu eingelesen. Nach erfolgreichem Einlesen der neuen Konfiguration wird die Kommunikation im IEC870 Slave neu gestartet und der Wert der Variable wird auf `0` zurückgesetzt. Schlägt dieser Prozess fehl arbeitet das Process Gateway mit der bei Programmstart geladenen Konfiguration weiter.

Neustart und Nachladen der XML Konfiguration

Ab zenon 8.00 können der Neustart für das Process Gateway - IEC870 Slave via einer verknüpften Variable ausgelöst werden. Bei diesem Neustart wird die XML-Datei (auf Seite 195) mit der Konfiguration erneut eingelesen.

ALLGEMEIN

Im IEC870 Slave Process Gateway kann mit dem `IOA 8` mit Type ID `<0> Internal Status` eine zenon Variable verknüpft werden.

Hat diese verknüpfte Variable in der zenon Runtime den Wert `1` wird die Konfiguration für das Process Gateway neu eingelesen. Nach erfolgreichem Einlesen der neuen Konfiguration wird die Kommunikation im IEC870 Slave neu gestartet und der Wert der Variable wird auf `0` zurückgesetzt. Schlägt dieser Prozess fehl arbeitet das Process Gateway mit der bei Programmstart geladenen Konfiguration weiter.

PROJEKTIERUNG

Verknüpfen Sie ein **Information Object** mit einer internen zenon Variable für Type Identification `8 - <0> Internal Status`.

Führen Sie dazu im Process Gateway folgende Schritte durch:

1. Wählen Sie die Registerkarte **Sector** im Konfigurationsdialog **Settings**.
2. Wählen Sie aus der Liste **Devices** ein konfiguriertes `Device`.
3. Wählen Sie aus der Liste **Sectors** einen konfigurierten `Sector`.
4. Erstellen Sie ein neues `Information Object`.
   a) Wählen Sie dazu ein bereits konfiguriertes Objekt mit `IOA 8` und klicken Sie auf die Schaltfläche **Edit** oder klicken Sie auf die Schaltfläche **New** um ein neues `Information Object` zu erstellen.
   Der Dialog **Information object setting** (auf Seite 176) wird geöffnet.
   b) Konfigurieren Sie die `IOA` mit Wert `8` in der Eigenschaft **Information Object Address**.
   c) Wählen Sie in der Dropdownliste **Type Identification** den Eintrag `<0> Internal status`.
   Hinweis: Dieser Eintrag ist per Default bereits ausgewählt.
   d) Klicken Sie auf die Schaltfläche ... um eine zenon Variable zu verknüpfen.
   Der Dialog **Variable selection** zur Auswahl einer Variable wird geöffnet.
   e) Wählen sie im Bereich Variable eine Variable aus der Liste.
   f) Bestätigen Sie die Konfiguration durch Klick auf die Schaltfläche **OK**.
   g) Klicken Sie die Schaltfläche OK um den IO-Settings-Dialog zu beenden.

> **Info**
>
> *Diese Projektierung ist in jedem Sektor möglich. Diese Projektierung gilt für das gesamte Process Gateway, nicht für einzelne Sektoren.*
>
> *Projektierungstipp:*
>
> *Projektieren Sie die Verknüpfung nur in einem Sektor oder verknüpfen Sie in unterschiedlichen Sektor immer dieselbe zenon Variable.*

VERHALTEN IN DER ZENON RUNTIME

Bei Start der Runtime:

- Die aktuellen Werte der projektierten Variablen werden gelesen.

- Der Wert der verknüpften Variable für `IOA 8 - <0> Internal Status` wird immer auf `0` zurückgesetzt.
  Hinweis: Die XML-Datei mit der Konfiguration wird nur ein Mal gelesen, auch wenn die Variable bei Kommunikationsstart den Wert `1` hatte.

Bei laufender Runtime:

- Bei Wertänderung der verknüpften Variable auf `1`:
  a) Werden alle laufende Verbindungen des Process Gateways geschlossen.
  b) Das Process Gateway liest die XML-Konfigurationsdatei neu.
  c) Wird der Wert der `IOA 8`-Variable auf `0` gesetzt.
  Hinweis: ist ein **Information Object** mit `IOA 8` nicht mehr in der neuen Konfiguration vorhanden, bleibt der Wert für diese Variable auf `1`.
  - Schlägt das Einlesen der neuen Konfiguration fehl, erfolgt die Kommunikation weiter mit den alten (= bisher bestehenden) Konfigurationsdaten. Die Variable für `IOA 8 - <0> Internal Status` wird nicht zurückgesetzt.

PROJEKTIERUNGSTIPP

Mit einer Variable für `IOA 8 - <0> Internal Status` haben Sie in der zenon Runtime folgende Anwendungsmöglichkeiten:

- Überwachung des korrekten Starts des Process Gateways:
  Hat die interne IOA 8-Variable einen initialen Wert größer `1` (z.B. `99`) kann diese Variable dazu verwendet werden, den korrekten Start des Process Gateways zu überprüfen. Bleibt dieser Initialwert unverändert, so wurde das Process Gateway nicht korrekt gestartet.
  Mögliche Fehlerursache: die konfigurierten Variablen existieren nicht mehr oder der IP-Port ist belegt.
- Nachladen der Konfiguration für das Process Gateway via Runtime
  Nach dem Nachladen des Projekts in der zenon Runtime wird die XML-Konfigurationsdatei des Process Gateways nicht automatisch neu gelesen.
  Wurden Variablen im zenon Editor geändert oder neu angelegt, welche auch im Process Gateway projektiert sind, können Sie nach dem Nachladen des Projekts in der zenon Runtime auch das Einlesen einer neuen XML-Konfigurationsdatei im Process Gateway auslösen. Setzen Sie dafür den Wert 1 der IOA 8 Variable in der Runtime auf 1.

10.3.8 Mapping von Statusbits in Melderichtung

Der IEC870Slave unterstützt das Versenden von Qualitätsbits für die Information Objects. Die Qualitätsbits des IO werden - in ASDUs in 'montior direction' - gesetzt anhand bestimmten Statusbits der verknüpften zenon Variable.

Die Qualitätsbits werden den Statusbits der Variablen wie folgt zugeordnet:

Statusbit in zenon	IEC60870 quality descriptor bits	Anmerkungen
INVALID	IV (invalid) NT (not topical)	Der Slave leitet in Melderichtung das INVALID Statusbit als zwei Qualitätsbits weiter. Hinweis: das Qualitätsbit IV kann auch wegen Wertanpassung des IOs gesetzt werden.
ALT_VAL	SB (substituted)	Bis zenon Version 7.10 gleich gemapped wie OFF Bit.
OFF	IV (invalid)	
N_UPDATE (Nicht aktualisiert in zenon Netzwerk)	NT (not topical)	Beispiel: Das zenon **Process Gateway** läuft auf einem Netzwerkclient. Während einer Redundanzumschaltung der Netzwerkserver werden die Variablen vorübergehend mit N_UPDATE gekennzeichnet. Die Werte der IOs werden zum 870-Master mit NT Qualitätsbit gesendet. Nach Abschluss der Redundanzumschaltung verbindet sich der Netzwerkclient automatisch zum aktuellen Netzwerkserver und die Werte sind wieder spontan. Die Werte der IOs werden ohne NT Qualitäts-Bit zum 870-Master gesendet.
BL_870	BL (blocked)	
SB_870	SB (substituted)	wie ALT_VAL
NT_870	NT (not topical)	
OV_870	OV (overflow)	OV_870 Statusbits, welche für die Variablen in zenon bereits gesetzt wurden, werden zum Master gesendet. Für normalized values gilt zusätzlich: der Wertebereich laut IEC 60870 Standard ist <-1 .. $1-2^{-15}$> mit einer Genauigkeit auf 4 Nachkommastellen. Das entspricht in zenon Variablen mit einem Wertebereich -1.0 bis 0.9999. Außerhalb des Bereiches für normalized values wird in ASDU der Wert des IOs angepasst und das OV Qualitäts-Bit gesetzt.

		Hinweis: das Qualitätsbit OV kann auch wegen Wertanpassung eines IO mit anderen Type-IDs gesetzt werden.
T_INVAL (Zeit ungültig)	Nur in "Binary Time 2a" (CP24Time2a/CP56Time2a) BS[24] - IV (invalid)	Der Slave leitet das T_INVAL Statusbit im Zeitstempel der IOs nur mit "time tag" weiter.

Info

Das Process Gateway kann die Werte mit IV oder OV Quality kennzeichnen, wenn der Wert der zenon Variable den Wertbereich des ASDU-Typ überschreitet.

Weitere Informationen dazu finden Sie im Kapitel Zuordnung der Datentypen (auf Seite 183) im Abschnitt "Wertanpassung von IOs".

10.3.9 Befehle

Ein emphangener Befehl - T45..T64 - wird als Wert bei der zenon Variablen gesetzt. In Sektoren ohne Select routing wird an Master gleich eine, immer positive, Betätigung gesendet - COT_actcon (7) und COT_actterm (10).

In solchen Sektoren wird auch ein Select gleich automatisch positiv bestätigt - COT_actcon + SE und die zenon Variable ändert dabei weder Wert noch Statusbits.

Hinweis: Bei Variablen, die im IEC870Sl Process Gateway als Befehlsvariable konfiguriert sind, wird bei aktiver Eigenschaft **SW setzen über VBA** (Gruppe **Protokollierung in CEL**) jeder Befehl als **Sollwert Absetzen** über die API-Schnittstelle in der **CEL** protokolliert.

Select routing

Das Select routing kann nur zusammen mit einer Projektierung des zenon Moduls **Befehlsgabe** verwendet werden und steht nur für die Befehlsgabeaktion **Auto/Remote Befehl** zur Verfügung.

Das Process-Gateway führt auf einer Befehlsvariable nur dann ein Select Routing durch, wenn die Variable in zenon mit aktiviertem **Select Before Operate** projektiert ist.

Ist bei den zenon Variablen die Eigenschaft **Select Before Operate** inaktiv verhält sich das **Process Gateway** gleich wie für Sektoren ohne **Select routing** und beantwortet ein empfangenes Select immer positiv.

ÜBERSICHT ZENON VARIABLENPROJEKTIERUNG - PROCESS GATEWAY

zenon Variable **Eigenschaft** Select Before Operate	Process Gateway - **Sektor-Option:** Select Routing	Verhalten im Process Gateway
aktiv	aktiv	Ein Select ist erforderlich. Ein Execute ohne Select wird abgelehnt. Ein Select wird an die **Befehlsgabe** weitergeleitet.
aktiv	inaktiv	Ein Select wird automatisch positiv beantwortet. Die **Befehlsgabe** wird in der Befehlsausführung nicht berücksichtigt. Ein Execute wird direkt an den Process-Treiber weitergegeben.
inaktiv	aktiv	

ABLAUF

Wenn für den angesprochenen Sektor die Option Select routing aktiviert wurde und für die zenon Befehlsvariable **Select Before Operate** aktiviert ist, dann wird bei Erhalt eines Select gewartet bis über die Befehlsvariable von zenon der Status SE_870 mit COT_actcon (7) und eventuell mit N_CONF (entspricht PN) zurückgemeldet wird.

Erst dies löst eine Bestätigung am Master (z.B. ein Control Center) aus. So lange eine Selektierung via Select läuft wird jede weitere für dies IO abgelehnt.

TIMEOUT, CANCEL UND FEHLENDES SELECT

Für die verschiedenen Antwortmöglichkeiten auf ein Select routing gelten folgende Regeln:

- Ist auf einem Sektor Select routing konfiguriert, so wird bei jedem Select ein Select Timeout pro IO gestartet.
- Der Slave überprüft, ob das Timeout abgelaufen ist und setzt dann die Variable auf SE, COT_actterm (10) und PN.
- Wird vom Master ein Cancel - COT_deact (8) - für ein laufendes Select empfangen, wird dies mit COT_deactcon (9) bestätigt und das Select wird gelöscht.
- Wenn ein Execute oder Cancel ohne vorheriges Select empfangen wird, so wird mit PN (also negativ) geantwortet.

Pulsbildung - 'Qualifier of Command'

Für empfangene Befehle der IEC870 Slave unterstützt die Pulsbildung. Das Qualifier of Command (`QoC`) wird für alle im IEC60870 Standard vorgesehene Befehle (`T45`, `T46`, `T47` und `T58`, `T59`, `T60`) unterstützt.

Zum Beispiel ein empfangener Einzelbefehl - ein ASDU mit Typ `T45` - mit `QoC` wirkt sich auf eine zenon Variable wie folgt aus:

- **QoC 0**: Undefiniert entsprechend IEC-Standard. Verhält sich im Slave wie QoC 3 (persistent output).
- **QoC 1**: Short pulse. Die zenon Variable nimmt den Wert 'EIN' an und wechselt nach der in den Process Gateway Einstellungen (auf Seite 166) definierten Zeit auf 'AUS'.
- **QoC 2**: Long pulse. Die zenon Variable nimmt den Wert 'EIN' an und wechselt nach der in den Process Gateway Einstellungen (auf Seite 166) definierten Zeit auf 'AUS'.
- **QoC 3**: Persistent output. Die zenon Variable nimmt den Wert 'EIN' (oder 'AUS') an und behält den Wert bei. Dieser muss für die Entgegennahme weiterer Befehle zurückgesetzt werden.

Sobald die Kommunikation beendet wird, wird auch für alle noch aktiven Puls-Variablen der Puls beendet, also ein 'AUS' geschrieben. Wird ein Puls für eine noch laufende Pulsbildung erneut veranlasst, wird die laufende Pulsbildung beendet. Die Variable erhält den Wert 'AUS'.

Hinweis: Für SCS (single command state) ist der Wert für 'AUS' immer 0. Für DCS (double command state) gilt: Ein 'AUS' hat auf Protokollebene den Wert 00b; dann in zenon entweder den Wert 2 oder 0, abhängig von den Einstellungen für **DPI/DCS mapping** (auf Seite 194).

Hinweis: In Sektoren mit `Select routing` kann Pulsbildung nicht durchgeführt werden da 'AUS' nach Select im Konflikt mit Execute steht.

10.3.10 Read - T102 - Anforderung

Der IEC870 Slave antwortet die Read command - `C_RD_NA_1` - Anforderungen des Masters automatisch. Dazu sind keine zusätzlichen Projektierungsschritte notwendig.

Es können maximal 2048 Read Anforderungen gleichzeitig aktiv sein. Sobald diese Grenze erreicht wird, werden weitere Read Anforderungen abgelehnt.

Die Beantwortung der Read Anforderungen muss nicht projektiert werden und läuft automatisch im Hintergrund ab. Dabei gilt:

- Für jedes IO mit der angeforderten `IOA` wird ein Read zu zenon ausgeführt und mit von Runtime gelieferten Wert an den Master beantwortet.
- Es können auch mehrere Anforderungen für ein IO gleichzeitig aktiv sein.

- Read Anforderungen werden auch dann abgearbeitet, wenn die Generalabfrage noch nicht beendet ist.

10.3.11 Zeit-Synchronisierung - T103 - Anforderung

Damit die Zeitsynchronisierung - beim Erhalt von `C_CS_NA_1` - ausgeführt wird, muss in `Device` bei einem beliebigen `Sektor` ein IO von Typ `T103` zu einer Prozessvariable verbunden werden. Das erste IO welches den Typ `T103` verwendet wird als Enable verwendet.

Diese Variable muss einen Wert <>0 haben und keinen der folgenden Statusbits gesetzt:

- INVALID (Bit 18)
- OFF (Bit 20)
- ALT_VAL (Bit 27)
- OV_870 (Bit 47)
- BL_870 (Bit 44)

Wenn die `Enable`-Variable nicht konfiguriert wurde oder der Wert für diese Variable fehlt, wird die Verbindung abgebaut, wenn ein `C_CS_NA_1` erhalten wird, weil es deutet darauf dass das System unvollständig konfiguriert wurde.

Das nicht ausführen des Kommandos Aufgrund des fehlenden Enables (Wert 0 oder Statusbits) oder Rechten in Betriebssystem führt zu keinem Verbindungsabbau, das Kommando wird nur negativ Bestätigt - `COT_actcon + PN`.

Hinweis: um die Uhr des Rechners umzustellen benötigt die Applikation des Process Gateways die Administrator Rechte in Windows.

10.3.12 Test - T104 und T107 - Anforderungen

Der IEC870 Slave beantwortet an den Master automatisch die beiden Testkommandos `C_TS_NA_1` (`ASDU<104>`) und `C_TS_TA_1` (`ASDU<107>`). Dazu sind keine zusätzlichen Projektierungsschritte notwendig.

Info

Weitere Informationen zu unterstützten Befehlen finden Sie im Kapitel Interoperability (auf Seite 237).

10.3.13 Dateitransfer

Das **Process Gateway IEC870 Slave** unterstützt die Übertragung der Dateien in monitoring und controlling direction. Dafür konfigurieren Sie in **Device** die Ordner-Eigenschaft **Directory for File Transfer** und **Directory for File Transfer in Control Direction**.

Die Prozeduren des Dateitransfers mit dem 870-Master werden im Hintergrund automatisch abgearbeitet. Die verwendeten ASDUs werden als class 2 gesendet. Damit ist sichergestellt, dass Wertänderungen auch während eines laufenden Dateitransfers an den Master übertragen werden.

Für die ASDUs des Dateitransfers gilt:

- im AFQ UI4 werden die Werte 1-4 unterstützt
- im SCQ UI4 werden die Werte 0–6 unterstützt;
  0 und 1 sind beide 'select file'

Wird bei einem bereits aktiven Dateitransfer ein neuer Transfer angestoßen, dann wird der laufende Transfer abgebrochen und der neue Transfer wird ausgeführt.

Eine verschlüsselte Kommunikation (auf Seite 227) des **IEC870 Slave Process Gateways** wird in der `zenProcGateway.INI`-Datei projektiert.

Hinweis: Der aktuelle Status des Dateitransfers kann über die Variablen, die mit dem Internen Typ T00 (auf Seite 214) verknüpft sind, ausgewertet werden.

Ablage und Benennung

Ablage und Benennung werden nach folgendem Schema definiert:

<Konfiguriertes Verzeichnis>\<COA>\<IOA>.<NOF>

Als NOF (Name of file) wird nur 1 = 'transparent file' unterstützt.
Die maximal übertragbare Dateigröße ist auf 16.711.680 Bytes begrenzt.

BEISPIEL

Ordner in der Modulkonfiguration (auf Seite 166): `C:\TEMP\IEC870`

Datei `1100.1`: für IO von Sektor **COA=151** und **IOA 1100**

Ablageort: `C:\TEMP\IEC870\151\1100.1`

Spontane Verzeichnisübertragung

Die spontane Übertragung der Dateien im Ordner für den Dateitransfer kann vom Prozessleitsystem ausgelöst werden. Dazu:

- legen Sie einen IO vom Typ F_DR_TA_1 (T126) mit der IOA '0' an

Wert	Bedeutung
Wechsel von 1 auf 2	Übertragung wird gestartet
3	Übertragung läuft
4	Übertragung wurde beendet
5	Fehler bei der Übertragung

Verschachtelte Aktivierung wird verhindert.

TLS-verschlüsselte Kommunikation

Eine verschlüsselte Kommunikation des **IEC870 Slave Process Gateways** wird in der `zenProcGateway.INI`-Datei konfiguriert.

Parameter	Beschreibung
TLS_ACTIVE=	Art der abgesicherten Kommunikation. ▶ `0: deaktiviert` Keine verschlüsselte Kommunikation. ▶ `1: aktiviert` Kommunikation erfolgt verschlüsselt, auf Basis der folgenden Parameter. Default: `0`
TLS_CERTIFICATE_STORE_PATH=	Absoluter Ablageort der Zertifikate. Dieses Basisverzeichnis muss die beiden Unterverzeichnisse **CA** und **PRIVATE** enthalten. ▶ **CA** für vertrauenswürdige Zertifikate und ▶ **PRIVATE** für eigene Zertifikate
TLS_CERTIFICATE_FILE=	Name des TLS-Zertifikates. Der Default-Ablageort wird mit dem Eintrag für den `[Ablageort der Zertifikate]` (**TLS_CERTIFICATE_STORE_PATH=)** projektiert. Das TLS-Zeritifikat muss dort im Ordner **PRIVATE** vorhanden sein.
TLS_CIPHER_LIST=	Liste der unterstützten Verschlüsselungsverfahren. Die Liste enthält die Kürzel im openSSL-Format.
TLS_PEER_CERTIFICATE_SUBJECT=	Verwendete Zertifikats-Identifizierung des Clients. Das Client-Zertifikat muss das hier konfigurierte **SUBJECT** enthalten. Bei Nicht-Übereinstimmung wird die Verbindung abgebrochen.
TLS_RENEGOTIATION_TIMEOUT=	Maximale Dauer einer verschlüsselten Verbindung, bevor diese erneuert wird. Nach Ablauf der projektierten Zeit wird die Verschlüsselung neu initialisiert. Dabei werden Zertifikate und Schlüssel neu ausgetauscht. Zeitangabe in Sekunden. Default: `86400`
TLS_RENEGOTIATION_MAX_BYTES=	Datenmenge, für die eine verschlüsselte Verbindung benutzt wird, bevor diese erneuert wird. Nach Übertragung der projektierten Datenmenge wird die Verschlüsselung neu initialisiert. Dabei werden Zertifikate und Schlüssel neu ausgetauscht. Angabe in Bytes. Default: `1048576`

TLS_RESUMPTION_TIMEOUT=	Maximale Dauer eines benutzten Schlüssels, bevor er erneuert wird. Es erfolgt nur eine Erneuerung des Schlüssels. Das Zertifikat wird dabei nicht aktualisiert. Nach Ablauf der projektierten Zeit wird das Zertifikat erneuert. Zeitangabe in Sekunden. Default: `43200`
TLS_RESUMPTION_MAX_BYTES=	Datenmenge, für die ein Schlüssel benutzt wird, bevor er erneuert wird. Nach Übertragung der projektierten Datenmenge wird die Verschlüsselung erneuert. Das Zertifikat wird dabei nicht aktualisiert. Angabe in Bytes. Default: `10485760`
TLS_CRL_CHECK_INTERVAL=21600	Zeitintervall für die Prüfung, ob das aktuell verwendete Zertifikat in der **REVOCATION LIST** enthalten ist. Ist das Zertifikat in der **REVOCATION LIST** enthalten, ist die verschlüsselte Verbindung nicht mehr sicher. Die Verbindung wird abgebrochen. Zeitangaben in Sekunden. Default: `21600`

10.4 Broadcasts

Global Address 0xFF(FF) im `Common Address of ASDU` (`COA`) wird für folgende `Typ IDs` unterstützt:

- `C_IC_NA_1` - Generalabfrage - ASDU<100>
- `C_CS_NA_1` - Zeitsynchronisierung - ASDU<103>

Dazu sind keine zusätzlichen Projektierungsschritte notwendig.

Achtung

Die Broadcast Address 0xFF(FF) - "all Outstations" - in Data Link Layer in der 60870-4-101 Kommunikation wird nicht unterstützt.

10.5 Fehleranalyse

Sollte es zu Kommunikationsproblemen kommen, bietet dieses Kapitel Hilfe, um den Fehler zu finden.

10.5.1 Analysetool

Process Gateway verhält sich wie ein zenon Treiber und schreibt Meldungen in eine gemeinsame Log-Datei. Um sie korrekt und übersichtlich anzuzeigen, benutzen Sie das Programm Diagnosis Viewer (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter Start/Alle Programme/zenon/Tools 8.00 -> Diagviewer. Generell gilt:

zenon Treiber protokollieren alle Fehler in LOG-Dateien. LOG-Dateien sind Textdateien mit einer speziellen Struktur. Der Standardordner für die LOG-Dateien ist der Ordner **LOG** unterhalb des Ordners **ProgramData**, zum Beispiel:

%ProgramData%\COPA-DATA\LOG.

Achtung: Mit den Standardeinstellungen zeichnet ein Treiber nur Fehlerinformationen auf. Mit dem Diagnosis Viewer kann bei den meisten Treibern die Diagnose-Ebene auf „Debug" und „Deep Debug" erweitert werden. Damit protokolliert der Treiber auch alle anderen wesentlichen Aufgaben und Ereignisse.

Im Diagnosis Viewer kann man auch:

- neu erstellte Einträge in Echtzeit mitverfolgen
- die Aufzeichnungseinstellungen anpassen
- den Ordner, in dem die LOG-Dateien gespeichert werden, ändern

Hinweise:

1. Der Diagnosis Viewer zeigt alle Einträge in UTC (Koordinierter Weltzeit) an und nicht in der lokalen Zeit.
2. Der Diagnosis Viewer zeigt in seiner Standardeinstellung nicht alle Spalten einer LOG-Datei an. Um mehr Spalten anzuzeigen, aktivieren Sie die Eigenschaft **Add all columns with entry** im Kontextmenü der Spaltentitel.
3. Bei Verwendung von reinem **Error-Logging** befindet sich eine Problembeschreibung in der Spalte **Error text**. In anderen Diagnose-Ebenen befindet sich diese Beschreibung in der Spalte **General text**.
4. Viele Treiber zeichnen bei Kommunikationsprobleme auch Fehlernummern auf, die die SPS ihnen zuweist. Diese werden in **Error text** und/oder **Error code** und/oder **Driver error parameter(1** und **2)** angezeigt. Hinweise zur Bedeutung der Fehlercodes erhalten Sie in der Treiberdokumentation und der Protokoll/SPS-Beschreibung.
5. Stellen Sie am Ende Ihrer Tests den Diagnose-Level von **Debug** oder **Deep Debug** wieder zurück. Bei **Debug** und **Deep Debug** fallen beim Protokollieren sehr viele Daten an, die auf der Festplatte gespeichert werden und die Leistung Ihres Systems beeinflussen können. Diese werden auch nach dem Schließen des Diagnosis Viewers weiter aufgezeichnet.

Achtung

Unter Windows CE werden aus Ressourcegründen Fehler standardmäßig nicht protokolliert.

Hinweis: Process Gateway steht unter Windows CE nicht zur Verfügung.

Info

Weitere Informationen zum Diagnosis Viewer finden Sie im Kapitel Diagnosis Viewer (main.chm::/12464.htm).

Logging

Der Treiber unterstützt ein umfangreiches Logging für:

- Clientname
- zenProcGateway_IEC870Slave
- Module

CLIENTNAME

Wie der Name des Clients im Diagnosis Viewer angezeigt wird, hängt von der Konfiguration ab. Daher ändert sich auch der Name, wenn der TCP-Port, der COM-Port oder das Protokoll geändert wird. In diesem Falle können sich auch die Einstellungen für das Logging ändern.

Wird der Process Gateway mehrfach mit den gleichen Einstellungen gestartet, wird auch der gleiche Clientname verwendet. Die Fehlermeldungen der beiden Instanzen lassen sich durch die Prozess-ID unterscheiden.

Beispiel

zenProcGateway_IEC870Slave

- 104 Protokoll mit Standardport 2404
- 101 wenn COM1 ausgewählt.
- Wurde die Konfiguration noch nicht gelesen, sind Fehlermeldungen beim Startup immer unter diesen Client sichtbar.

zenProcGateway_IEC870Slave:COM4

- 101 mit COM4.
  Der Name enthält den verwendeten COM-Port.

zenProcGateway_IEC870Slave:Port:12345

- 104 mit Port 12345.
  Der Name enthält den Port für die Masteranbindung.

MODULE

Fehlermeldungen werden für alle Module angezeigt:

- IEC870 Slave
- OSI2
- OSI7
- SCADA-Runtime (ScadaRT)

IEC870 SLAVE

Betrifft die Anbindung zum Process Gateway und dessen Konfiguration.

Informationen zu:

- Fehler bei der Projektierung

OSI2

Betrifft die Protokollebene. Hier ist das Framing mit den entsprechenden Funktionen implementiert.

Informationen zu:

- Empfangene Daten als Byte Dump
- Gesendete Daten als Byte Dump
- Function codes für 101
- Fehler auf dieser Ebene, z. B. im Framing
- APCI für 104 (870-5-104 5)

Levels

- Msg: Frame Interpretierung
- Debug:
  Empfangene und gesendete Daten
  Frame Auswertung.
  Neue Master Verbindung.
- Deep Debug: Jede Abfrage, ob Zeichen vorhanden. Produziert sehr viele Loggingeinträge!

OSI7

Betrifft die Applikationsebene.
Hier werden die Kommandos interpretiert und ausgeführt.

Informationen zu:

- Protokollierung der 'process and system information'
- Fehler in den ASDUs

Levels:

- Warning: Rekursive interrogation
- Msg: Process information, System information
- Debug: Bestätigte I-Frames
- Deep Debug: Interrogation Durchläufe. Produziert sehr viele Loggingeinträge!

SCADA-RUNTIME (SCADART)

Meldungen zur Anbindung an die Runtime:

Informationen zu:

- angeforderte und abgemeldete Variablen
- aktualisierte Variablen
- Deaktivierung des Projekts

Levels

- Debug:
  geänderte Werte
  Projekt inaktiv

LOG Einträge

Der IEC870 Slave verwendet in Einträgen in die LOG-Datei folgende Parameter:

Parameter	Beschreibung
deviceIp:	IP-Adresse vom Device
devicename:	Benennung vom Device
COA:	Care of Address -> Lt. Norm
SecName:	Sektorbenennung
TypeyId:	Kommandonummer
TypeIdAsTxt:	Kommando als Text
IOA:	Lt. Norm
AdviseID:	ID über welche Wertänderungen angefordert und zugeordnet werden
Wert:	Übertragener Wert
TimeStamp:	Zeitpunkt der Wertänderung. Vom Treiber od. von command
QUALDESC:	lt. Norm
SBEValue:	COT, select und pn gemappt auf eindeutige enums: 0: Nicht select relevant 1: actcon+pn+select 2: actcon+select 3: deactcon+pn+select, 4: deactcon+select 5: actterm+pn+select, 6: actterm+select,
Confirmed:	1 wenn Select bereits bestätigt
COT:	Cause of Transmission lt. Norm
CotTxt:	COT als Text
Qualifier:	lt. Norm und Commando
Originator:	Lt. Norm

QU:	Qualifier of command QU 7.2.6.26
DCS:	7.2.6.16 Double command (IEV 371-03-03)
ControlCommand:	Je nach control-Kommando

BEISPIEL FÜR 45:

```
<TypeId> <TypeIdAsTxt> Device:(<deviceip>)<devicename> Sector:(<IOA>)< SecName >
ioa:<IOA> cot:(<COT>)<CotTxt> ori:<Originator> sel:<QOC S/E> qu:<QU> dcs:<DCS>
ValueChange cmd(<TypeyId>)<<TypeId>> <TypeIdAsTxt> ioa:<IOA> id:<AdviseID> value:<Wert>
time:<TimeStamp> state:< QUALDESC > sir:<SBEValue>
```

LOG Eintrag	Module/Level	Beschreibung
SBE IED processing. Device:(<deviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange> SBE command: advise:<AdviseID> conf:<confirmed> <ControlCommand>	OSI7/Debug	Jede Wertänderung welche für SBE bearbeitet wird erzeugt diese LogMeldung
Wrn:1014 SBE activation already confirmed! Device:(<DeviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange>	OSI7/Warnung	Erneute Bestätigung für Select erhalten und ignoriert
Error:16 Advise for SBE data point fail. SBE command: advise:<AdviseId> conf:<confirmed> <ControlCommand>	Scada/error	Variable konnte nicht angefordert werden.
Error:17 SOE value lost, queue full! Device:(<DeviceIp>)<devicename> Sector:(<COA>)<SecName> <ValueChange>	Scada/Error	Wertänderung (ValueChange) für SOE verloren da Queue voll.

10.5.2 Fehlercodes

Folgende Fehlercodes können im Diagnosis Viewer angezeigt werden:

Fehlercode	Bedeutung
570	Wertänderung für unbekanntes zyklisches IO.
571	Transferpuffer zu klein um ein zyklisches IO zu übertragen.
576	Verbindungsverlust. (Master does not respond.)

FEHLERMELDUNGEN FÜR FILE TRANSFER IN REVERSE DIRECTION

Fehlernummer	Error Level	Beschreibung
20	Fehlermeldung	Erzeugen der temporären Datei für den file transfer in reverse direction ist fehlgeschlagen.
1017	Warnung	Dateitransfer in reverse direction ist bereits aktiv, der Laufende wird beendet und ein neuer begonnen.
21	Fehlermeldung	Schreibfehler der Segmentdaten aufgetreten
22	Fehlermeldung	Länge der Daten überschreitet die erwartete Filelänge
23	Fehlermeldung	Prüfsumme der übertragenen Segmente für Sektor falsch.
24	Fehlermeldung	Prüfsumme für File falsch.
25	Fehlermeldung	Für Sektor übertragene Segmentdaten überschreiten die am Beginn definierte Datenlänge
26	Fehlermeldung	Sektor Datenübertragung wird als fertig gemeldet jedoch stimmt die Anzahl der übertragenen Segmentdaten nicht mit der angegeben am Beginn überein.
27	Fehlermeldung	File Daten wurden als fertig gemeldet jedoch stimmt die angegebene Filelänge nicht überein.
581	Fehlermeldung	Unerwartete IOA bei SG
582	Fehlermeldung	Unerwartete NOF bei SG
583	Fehlermeldung	Unerwartete NOS bei SG
584	Fehlermeldung	Unerwartete NOS bei LS
585	Fehlermeldung	Unerwartete NOS bei LS
586	Fehlermeldung	Unerwartete NOS bei LS
587	Fehlermeldung	FR mit nicht unterstützter NOF. Wir unterstützen nur 1 == transparent

10.5.3 Checkliste

870-104

870-101

1. Ist der COM Port von einer anderen Anwendung belegt, bzw. stimmen seine Einstellungen?

Allgemein

1.
2. Ist das Gerät (die SPS), mit dem versucht wird eine Kommunikation herzustellen, an das Stromversorgungsnetz angeschlossen?
3. Ist das Verbindungskabel zwischen Steuerung (SPS) und PC bzw. IPC korrekt angeschlossen?
4. Wurde die „Error – Textdatei“ analysiert (Welche Fehler sind aufgetreten)?
5. Zur weiteren Fehleranalyse senden Sie eine Projektsicherung, die .INI- und .XML-Datein des IEC870 Slaves sowie die LOG Datei des **Diagnosis Viewer** an Ihren zuständigen Support.

10.6 Interoperability

This companion standard presents sets of parameters and alternatives from which subsets must be selected to implement particular telecontrol systems. Certain parameter values, such as the choice of 'structured' or'unstructured' fields of the informationobject address of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per syste m. Other parameters, such as the listed set of different process information incommand and in monitor direction allow the specification of the complete set orsubsets, as appropriate for given applications. This clause summarizes theparameters of the previous clauses to facilitate a suitable selection for aspecific application. If a system is composed of equipment stemming from different manufacturers, it is necessary that all partners agree on the selected parameters.

The interoperability list is defined as in IEC 60870-5-101 and extended with parameters used in this standard. The text descriptions of parameters which are not applicable to this companion standard are strike-through (corresponding check box is marked black).

NOTE In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should bemarked in the white boxes as follows:

[ ] Function or ASDU is not used
[X] Function or ASDU is used as standardized (default)

[R] Function or ASDU is used in reverse mode
[B] Function or ASDU is used in standard and reverse mode

The possible selection (blank, X , R , or B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

1. SYSTEM OR DEVICE

(system-specific parameter, indicate definition of a system or a device by marking one of the following with ' X ')

[ ] System definition
[ ] Controlling station definition (Master)
[X] Controlled station definition (Slave)

2. NETWORK CONFIGURATION: 101 ONLY

(network-specific parameter, all configurations that are used are to be marked ' X ')

[X] Point-to-point	[X] Multipoint
[X] Multiple point to point	[ ] Multipoint-star

3. PHYSICAL LAYER: 101 ONLY

(network-specific parameter, all interfaces and data rates that are used are to be marked ' X ')

TRANSMISSION SPEED (CONTROL DIRECTION)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange Circuit X.24/X.27
[ ] 100 bit/s	[X] 2400 bit/s	[X] 2400 bit/s
[ ] 200 bit/s	[X] 4800 bit/s	[X] 4800 bit/s
[X] 300 bit/s	[X] 9600 bit/s	[X] 9600 bit/s
[X] 600 bit/s	[X] 19200 bit/s	[X] 19200 bit/s
[X] 1200 bit/s	[X] 38400 bit/s	[X] 38400 bit/s
	[X] 56000 bit/s	[X] 56000 bit/s
	[X] 57600 bit/s	[X] 57600 bit/s

	[X] 115200 bit/s	[X] 115200 bit/s
	[X] 128000 bit/s	[X] 128000 bit/s
	[X] 256000 bit/s	[X] 256000 bit/s

TRANSMISSION SPEED (MONITOR DIRECTION)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s	Balanced interchange Circuit X.24/X.27
[ ] 100 bit/s	[X] 2400 bit/s	[X] 2400 bit/s
[ ] 200 bit/s	[X] 4800 bit/s	[X] 4800 bit/s
[X] 300 bit/s	[X] 9600 bit/s	[X] 9600 bit/s
[X] 600 bit/s	[X] 19200 bit/s	[X] 19200 bit/s
[X] 1200 bit/s	[X] 38400 bit/s	[X] 38400 bit/s
	[X] 56000 bit/s	[X] 56000 bit/s
	[X] 57600 bit/s	[X] 57600 bit/s
	[X] 115200 bit/s	[X] 115200 bit/s
	[X] 128000 bit/s	[X] 128000 bit/s
	[X] 256000 bit/s	[X] 256000 bit/s

4. LINK LAYER: 101 ONLY

(network-specific parameter, all options that are used are to be marked ' X '. Specify the maximum frame length. If a non-standard assignment of class 2 messages is implemented for unbalanced transmission, indicate the Type ID and COT of all messages assigned to class 2.)

Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.

Link transmission	Frame length [octets]	Address field of the link
[X] Balanced transmission	[255] Maximum length L (both directions)*	[O] not present (balanced transmission only) optional
[X] Unbalanced transmission		[X] One octet
		[X] Two octets
		[X] Structured
		[X] Unstructured

*may be reduced by the system

Link Address 0xFF(FF) - broadcast to all outstations - is not supported.

The structure of Link Address is not interpreted.

When using an unbalanced link layer, the following ASDU types are returned in class 2 messages (low priority) with the indicated causes of transmission:

[ ] The standard assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission
9, 11, 13, 21	<1>

[X] A special assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission
120 - 126	All (as specified in the standard)

Note: (In response to a class 2 poll, a controlled station may respond with class 1 data when there is no class 2 data available).

5. APPLICATION LAYER

TRANSMISSION MODE FOR APPLICATION DATA

Mode 1 (Least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

COMMON ADDRESS OF ASDU

(system-specific parameter, all configurations that are used are to be marked ' X ')

ASDU adress	
101 only [X] One octet	[X] Two octets

The ASDU address 0xFF(FF) - global address - is supported only for C_IC_NA_1 and C_CS_NA_1

INFORMATION OBJECT ADDRESS

(system-specific parameter, all configurations that are used are to be marked ' X ')

Object adress	
101 only [X] One octet	[X] Structured
101 only [X] Two octets	[X] Unstructured
[X] Three octets	

CAUSE OF TRANSMISSION

(system-specific parameter, all configurations that are used are to be marked ' X ')

Cause of transmission	
101 only [X] One octet	[X] Two octets (with originator address) Originator address is set to zero if not used.

LENGTH OF APDU: 104 ONLY

(system-specific parameter, specify the maximum length of the APDU per system)
The maximum length of APDU for both directions is 253. The maximum length may be reduced by the system.

[253] Maximum length of APDU per system

SELECTION OF STANDARD ASDUS

PROCESS INFORMATION IN MONITOR DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[B]	<1>: = Single-point information	M_SP_NA_1
[B]	<2>: = Single-point information with time TAG	M_SP_TA_1
[B]	<3>: = Double-point information	M_DP_NA_1
[B]	<4>: = Double-point information with time TAG	M_DP_TA_1
[B]	<5>: = Step position information	M_ST_NA_1
[B]	<6>: = Step position information with time TAG	M_ST_TA_1
[B]	<7>: = Bitstring of 32 bit	M_BO_NA_1
[B]	<8>: = Bitstring of 32 bit with time TAG	M_BO_TA_1
[B]	<9>: = Measured value, normalized value	M_ME_NA_1
[B]	<10>: = Measured value, normalized value with time TAG	M_ME_TA_1
[B]	<11>: = Measured value, scaled value	M_ME_NB_1
[B]	<12>: = Measured value, scaled value with time TAG	M_ME_TB_1
[B]	<13>: = Measured value, short floating point value	M_ME_NC_1
[B]	<14>: = Measured value, short floating point value with time TAG	M_ME_TC_1
[B]	<15>: = Integrated totals	M_IT_NA_1
[B]	<16>: = Integrated totals with time TAG	M_IT_TA_1
[ ]	<17>: = Event of protection equipment with time TAG	M_EP_TA_1
[ ]	<18>: = Packed start events of protection equipment with time TAG	M_EP_TB_1
[ ]	<19>: = Packed output circuit information of protection equipment with time TAG	M_EP_TC_1
[ ]	<20>: = Packed single-point information with status change detection	M_SP_NA_1
[ ]	<21>: = Measured value, normalized value without quality descriptor	M_ME_ND_1
[B]	<30>: = Single-point information with time TAG CP56Time2a	M_SP_TB_1
[B]	<31>: = Double-point information with time TAG CP56Time2a	M_DP_TB_1
[B]	<32>: = Step position information with time TAG CP56Time2a	M_ST_TB_1
[B]	<33>: = Bitstring of 32 bit with time TAG CP56Time2a	M_BO_TB_1
[B]	<34>: = Measured value, normalized value with time TAG CP56Time2a	M_ME_TD_1
[B]	<35>: = Measured value, scaled value with time TAG CP56Time2a	M_ME_TE_1
[B]	<36>: = Measured value, short floating point value with time TAG CP56Time2a	M_ME_TF_1
[B]	<37>: = Integrated totals with time TAG CP56Time2a	M_IT_TB_1
[ ]	<38>: = Event of protection equipment with time TAG CP56Time2a	M_EP_TD_1

[ ]	<39>: = Packed start events of protection equipment with time TAG CP56Time2a	M_EP_TE_1
[ ]	<40>:= Packed output circuit information of protection equipment with time TAG CP56Time2a	M_EP_TF_1

Either the ASDUs of the set <2>, <4>, <6>, <8>, <10>, <12>, <14>, <16>, <17>, <18>, <19> or of the set <30> – <40> are used.

PROCESS INFORMATION IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID 'X' if it is only used in the standard direction, 'R' if only used in the reverse direction, and 'B' if used in both directions).

Mark	Parameter	Type
[X]	<45>: = Single command	C_SC_NA_1
[X]	<46>: = Double command	C_DC_NA_1
[X]	<47>: = Regulating step command	C_RC_NA_1
[X]	<48>: = Set point command, normalized value	C_SE_NA_1
[X]	<49>: = Set point command, scaled value	C_SE_NB_1
[X]	<50>: = Set point command, short floating point value	C_SE_NC_1
[X]	<51>: = Bitstring of 32 bit	C_BO_NA_1
[X]	<58>: = Single command with time TAG CP56Time2a	C_SC_TA_1
[X]	<59>: = Double command with time TAG CP56Time2a	C_DC_TA_1
[X]	<60>: = Regulating step command with time TAG CP56Time2a	C_RC_TA_1
[X]	<61>: = Set point command, normalized value with time TAG CP56Time2a	C_SE_TA_1
[X]	<62>: = Set point command, scaled value with time TAG CP56Time2a	C_SE_TB_1
[X]	<63>: = Set point command, short floating point value with time TAG CP56Time2a	C_SE_TC_1
[X]	<64>:= Bitstring of 32 bit with time TAG CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45> – <51> or of the set <58> – <64> are used.

SYSTEM INFORMATION IN MONITOR DIRECTION

(station-specific parameter, mark 'X' if used)

Mark	Parameter	Type
[X]	<70> : = End of initialization	M_EI_NA_1

SYSTEM INFORMATION IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[X]	<100>: = Interrogation command*	C_IC_NA_1
[X]	<101>: = Counter interrogation command	C_CI_NA_1
[X]	<102>: = Read command	C_RD_NA_1
[X]	<103>: = Clock synchronization command (option see 7.6)*	C_CS_NA_1
[X]	<104>: = Test command	C_TS_NA_1
[X]	<105>: = Reset process command	C_RP_NA_1
[ ]	<106>: = Delay acquisition command	C_CD_NA_1
[X]	<107>: = Test command with time TAG CP56Time2a	C_TS_TA_1

*also with global address - with Common Address of ASDU = 0xFF(FF)

PARAMETER IN CONTROL DIRECTION

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[ ]	<110>: = Parameter of measured value, normalized value	P_ME_NA_1
[ ]	<111>: = Parameter of measured value, scaled value	P_ME_NB_1
[ ]	<112>: = Parameter of measured value, short floating point value	P_ME_NC_1
[ ]	<113>: = Parameter activation	P_AC_NA_1

FILE TRANSFER

(station-specific parameter, mark each Type ID ' X ' if it is only used in the standard direction, ' R ' if only used in the reverse direction, and ' B ' if used in both directions).

Mark	Parameter	Type
[B]	<120>: = File ready	F_FR_NA_1
[B]	<121>: = Section ready	F_SR_NA_1
[B]	<122>: = Call directory, select file, call file, call section	F_SC_NA_1
[B]	<123>: = Last section, last segment	F_LS_NA_1
[B]	<124>: = Ack file, ack section	F_AF_NA_1
[B]	<125>: = Segment	F_SG_NA_1
[X]	<126>: = Directory {blank or X, only available in monitor (standard) direction}	F_DR_TA_1

TYPE IDENTIFIER AND CAUSE OF TRANSMISSION ASSIGNMENTS

(station-specific parameters)

Shaded boxes: option not required.
Blank: functions or ASDU not used.

Mark Type Identification/Cause of transmission combinations:
' X ' if only used in the standard direction;
' R ' if only used in the reverse direction;
' B ' if used in both directions.

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		B	B		X									X					
<2>	M_SP_TA_1			B		X														
<3>	M_DP_NA_1		B	B		X									X					
<4>	M_DP_TA_1			B		X														
<5>	M_ST_NA_1		B	B		X									X					
<6>	M_ST_TA_1			B		X														
<7>	M_BO_NA_1		B	B		X									X					
<8>	M_BO_TA_1			B		X														
<9>	M_ME_NA_1	B	B	B		X									X					
<10>	M_ME_TA_1			B		X														
<11>	M_ME_NB_1	B	B	B		X									X					
<12>	M_ME_TB_1			B		X														
<13>	M_ME_NC_1	B	B	B		X									X					
<14>	M_ME_TC_1			B		X														
<15>	M_IT_NA_1			B		X									X[1]	X				
<16>	M_IT_TA_1			B		X										X				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			B		X														
<31>	M_DP_TB_1			B		X														
<32>	M_ST_TB_1			B		X														
<33>	M_BO_TB_1			B		X														
<34>	M_ME_TD_1			B		X														
<35>	M_ME_TE_1			B		X														
<36>	M_ME_TF_1			B		X														
<37>	M_IT_TB_1			B		X										X				

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X[2]	X[2]	X								X	X
<46>	C_DC_NA_1						X	X	X[2]	X[2]	X								X	X
<47>	C_RC_NA_1						X	X	X[2]	X[2]	X								X	X
<48>	C_SE_NA_1						X	X	X[2]	X[2]	X								X	X
<49>	C_SE_NB_1						X	X	X[2]	X[2]	X								X	X
<50>	C_SE_NC_1						X	X	X[2]	X[2]	X								X	X
<51>	C_BO_NA_1						X	X	X[2]	X[2]	X								X	X
<58>	C_SC_TA_1						X	X	X[2]	X[2]	X								X	X
<59>	C_DC_TA_1						X	X	X[2]	X[2]	X								X	X
<60>	C_RC_TA_1						X	X	X[2]	X[2]	X								X	X
<61>	C_SE_TA_1						X	X	X[2]	X[2]	X								X	X
<62>	C_SE_TB_1						X	X	X[2]	X[2]	X								X	X
<63>	C_SE_TC_1						X	X	X[2]	X[2]	X								X	X
<64>	C_BO_TA_1						X	X	X[2]	X[2]	X								X	X
<70>	M_EI_NA_1*				X															
<100>	C_IC_NA_1						X	X	X	X	X							X	X	
<101>	C_CI_NA_1						X	X			X							X	X	
<102>	C_RD_NA_1					X												X	X	X
<103>	C_CS_NA_1						X	X										X	X	
<104>	C_TS_NA_1						X	X												
<105>	C_RP_NA_1						X	X										X	X	
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X												
<110>	P_ME_NA_1																			
<111>	P_ME_NB_1																			
<112>	P_ME_NC_1																			

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1													X					X	
<121>	F_SR_NA_1													X					X	
<122>	F_SC_NA_1					X								X				X	X	
<123>	F_LS_NA_1													X					X	
<124>	F_AF_NA_1													X				X	X	
<125>	F_SG_NA_1													X					X	
<126>	F_DR_TA_1*			X		X														

* Blank or X only

[1] Optional.

[2] Slave confirms deactivations of write command (TI <45> ... <64>) but by deactivated '**Select routing**' the corresponding activation is already transferred to execution; COT_actcon and COT_actterm were already sent.

COT	Cause of Transmission	
<0>	not used	
<1>	periodic, cyclic	per/cyc
<2>	background scan	back
<3>	spontaneous	spont
<4>	initialized	init
<5>	request or requested	req
<6>	activation	act
<7>	activation confirmation	actcon
<8>	deactivation	deact
<9>	deactivation confirmation	deactcon
<10>	activation termination	actterm
<11>	return information caused by a remote command	retrem
<12>	return information caused by a local command	retloc
<13>	file transfer	file
<14…19>	reserved	
<20>	interrogated by station interrogation	inrogen
<21…36>	interrogated by interrogation of the group 1..16	inro1..16
<37>	requested by general counter request	reqcogen
<38…41>	requested by counter interrogation of the group 1 … 4	reqco1..4
<42, 43>	reserved	
<44>	unknown type identification	
<45>	unknown cause of transmission	
<46>	unknown common address of ASDU	
<47>	unknown information object address	
<48, 63>	for special use (private range)	

6. BASIC APPLICATION FUNCTIONS

STATION INITIALIZATION

(station-specific parameter, mark 'X' if function is used)

[ ] Remote initialization

CYCLIC DATA TRANSMISSION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions)

[X] Cyclic data transmission

READ PROCEDURE

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions)

[X] Read procedure

SPONTANEOUS TRANSMISSION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions)

[B] Spontaneous transmission

DOUBLE TRANSMISSION OF INFORMATION OBJECTS WITH CAUSE OF TRANSMISSION SPONTANEOUS

(station-specific parameter, mark each information type 'X ' where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project-specific list.

[ ] Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1
[ ] Double-point information M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1
[ ] Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1
[ ] Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1 (if defined for a specific project)
[ ] Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1
[ ] Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1
[ ] Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

STATION INTERROGATION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] global		
[ ] group 1	[ ] group 7	[ ] group 13
[ ] group 2	[ ] group 8	[ ] group 14
[ ] group 3	[ ] group 9	[ ] group 15
[ ] group 4	[ ] group 10	[ ] group 16
[ ] group 5	[ ] group 11	Information object addresses assigned to each group must be shown in a separate table.
[ ] group 6	[ ] group 12	

CLOCK SYNCHRONIZATION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Clock synchronization

[ ] Day of week used
[ ] RES1, GEN (time tag substituted/not substituted) used
[B] SU-bit (summertime) used

optional, see 7.6

COMMAND TRANSMISSION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Direct command transmission
[X] Direct set point command transmission
[X] Select and execute command
[X] Select and execute set point command
[X] C_SE ACTTERM used

[X] No additional definition
[X] Short-pulse duration (duration determined by a system parameter in the outstation)
[X] Long-pulse duration (duration determined by a system parameter in the outstation)
[X] Persistent output

[setting] Supervision of maximum delay in command direction of commands and set point commands
[no limit] Maximum allowable delay of commands and set point commands

TRANSMISSION OF INTEGRATED TOTALS

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Mode A: Local freeze with spontaneous transmission
[X] Mode B: Local freeze with counter interrogation
[X] Mode C: Freeze and transmit by counter-interrogation commands
[X] Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

[X] Counter read
[X] Counter freeze without reset
[ ] Counter freeze with reset
[ ] Counter reset

[X] General request counter
[ ] Request counter group 1
[ ] Request counter group 2
[ ] Request counter group 3
[ ] Request counter group 4

PARAMETER LOADING

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[ ] Threshold value
[ ] Smoothing factor
[ ] Low limit for transmission of measured values
[ ] High limit for transmission of measured values

PARAMETER ACTIVATION

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[ ] Act/deact of persistent cyclic or periodic transmission of the addressed object

TEST PROCEDURE

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Test procedure

FILE TRANSFER

(station-specific parameter, mark 'X' if function is used).
File transfer in monitor direction

[X*] Transparent file
[ ] Transmission of disturbance data of protection equipment
[X] Transmission of sequences of events
[ ] Transmission of sequences of recorded analogue values

* a data can be transparently transported by the system but not generated or evaluated. Maximum file size is 16711680 bytes.

File transfer in control direction

[X] Transparent file

BACKGROUND SCAN

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[X] Background scan

ACQUISITION OF TRANSMISSION DELAY

(station-specific parameter, mark 'X ' if function is only used in the standard direction, 'R ' if only used in the reverse direction, and 'B ' if used in both directions).

[ ] Acquisition of transmission delay

DEFINITION OF TIME OUTS: 104 ONLY

Parameter	Default value	Remarks	Selected value
t0	30 s	Time-out of connection establishment	setting
t1	15 s	Time-out of send or test APDUs	setting
t2	10 s	Time-out for acknowledges in case of no data messages; t2 < t1	setting
t3	20 s	Time-out for sending test frames in case of a long idle state; t3 > t1	setting

Recommended range for timeouts t0 - t2 : 1s to 255s, accuracy 1s

Recommended range for timeout t3 : 0s to 48hrs, accuracy 1s
Long timeouts for t3 may be needed in special cases where satellite links or dialup connections are used (e.g. to establish connection and collect values only once per day or week). For dialup connections it may be necessary to give up the connection supervision completely. This is achievable by setting the timeout t3 to zero.

MAXIMUM NUMBER OF OUTSTANDING I FORMAT APDUS K AND LATEST ACKNOWLEDGE APDUS (W): 104 ONLY

Parameter	Default value	Remarks	Selected value
k	12 APDUs	Maximum difference receive sequence number to send state variable	setting
w	8 APDUs	Latest acknowledge after receiving w I format APDUs	setting

Recommended range of values k: 1 to 32767 APDUs, accuracy 1 APDU

Recommended range of values w: 1 to 32767 APDUs, accuracy 1 APDU (Recommendation: w should not exceed two-thirds of k)

PORTNUMBER: 104 ONLY

Parameter	Default value	Remarks
Portnumber	2404	setting

REDUNDANT CONNECTIONS

[ ] Number N of redundancy group connections used

RFC 2200 SUITE

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

[ ] Ethernet 802.3
[ ] Serial X.21 interface
[ ] Other selection from RFC 2200:

List of valid documents from RFC 2200
1. .....
2. .....
3. .....
4. .....
5. .....
6. .....
7. etc.

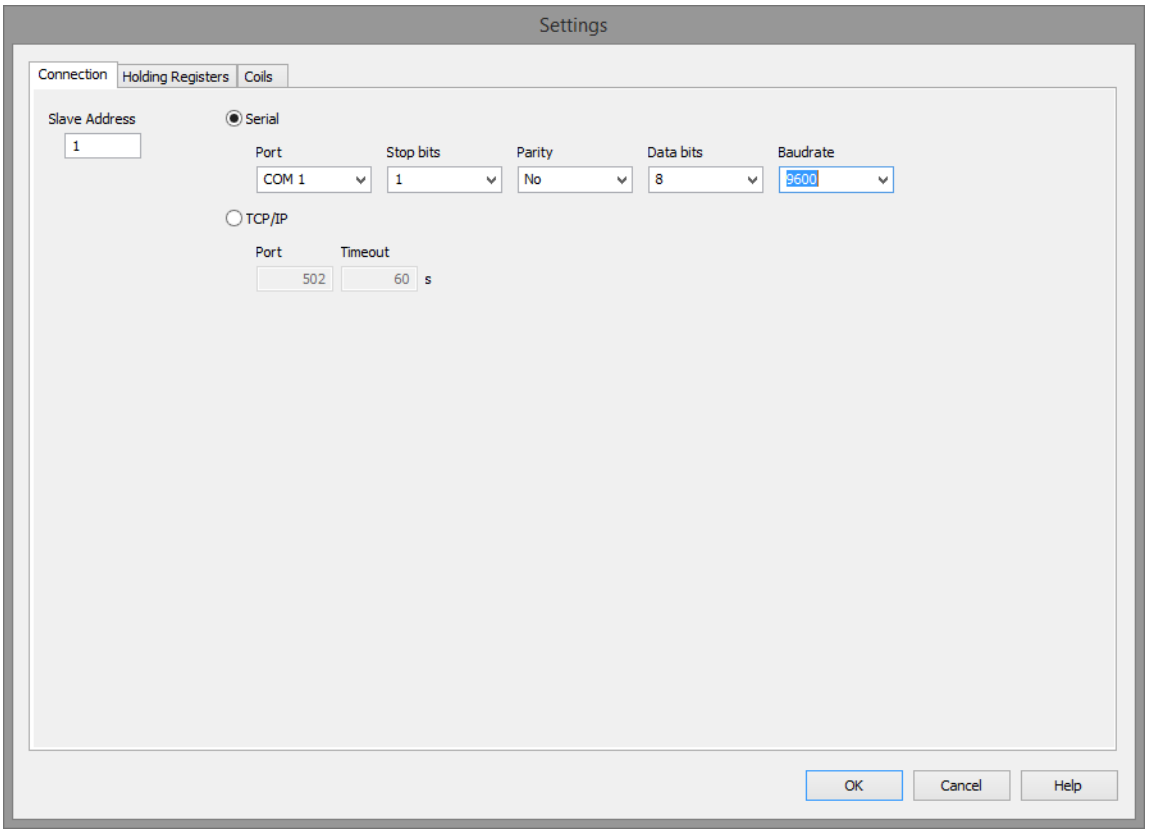
11. MODBUS Slave

Das **Process Gateway** für **MODBUS Slave** unterstützt:

- Function Code 1:
  Read Coils
- Function Code 3:
  Read Multiple Registers
- Function Code 5:
  Write Single Coil
- Function Code 6:
  Write single Register
- Function Code 15:
  Write Multiple Coils
- Function Code 16:
  Write Multiple Registers

11.1 Modulkonfiguration

Klick auf die Schaltfläche Settings im Startdialog für das Process Gateway öffnet den Dialog zur Auswahl der Variablen. Die ausgewählten Variablen werden in zenon oder im MODBUS-System laufend synchronisiert.

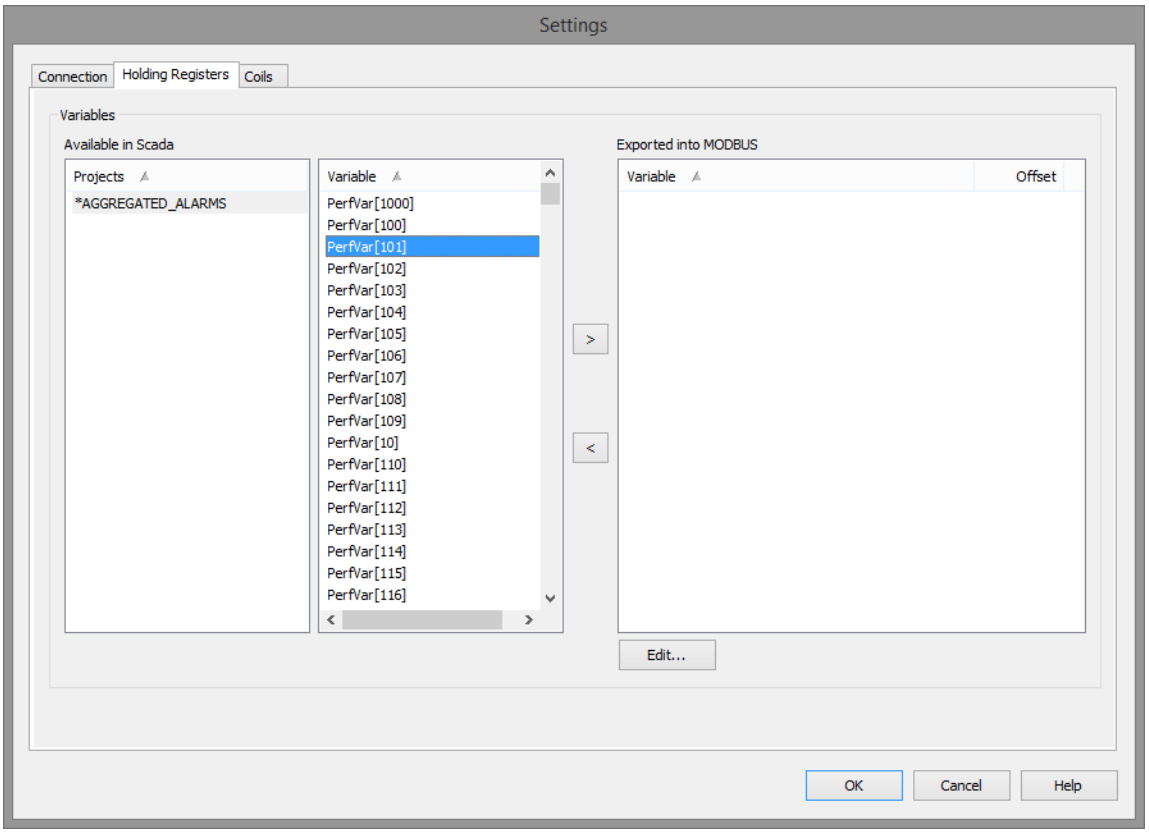


MODBUS CONNECTION

MODBUS Connection	Einstellungen der MODBUS-Verbindung.
Slave Adress **Unit Identifier**	Modbus Hardware-Adresse für die MODBUS-Kommunikation. ▶ **Slave Adress** bei serieller Kommunikation. ▶ **Unit Identifier** bei TCP/IP-Kommunikation. Default: 1
Serial	Aktiv: Serielle Verbindung wird benutzt. ▸ **Port**: Serielle Schnittstelle, über die auf das MODBUS-System zugegriffen wird. Auswahl aus Dropdownliste. ▸ **Stop bits**: Anzahl der Stop-Bits. Auswahl aus Dropdownliste. Default: 1 ▸ **Parity:** Paritätseinstellung. Auswahl aus Dropdownliste: No (Default) Odd Even ▸ **Datenbit**: Anzahl der Daten-Bits. Auswahl aus Dropdownliste. Default: 8 ▸ **Baudrate**: Baudrate der seriellen Schnittstelle. Auswahl aus Dropdownliste. Default: 9600
TCP/IP	Kommunikation zum Master per TCP/IP-Protokoll. ▶ Aktiv: TCP/IP-Verbindung mit MODBUS TCP wird benutzt. ▶ inaktiv: serielle Verbindung mit MODBUS RTU ▸ Port: Port-Adresse für die Verbindung. ▸ Timeout: Wartezeit.
OK	Übernimmt Einstellungen und schließt den Dialog.
Cancel	Verwirft alle Änderungen und schließt den Dialog.

11.2 Holding Registers

Klick auf die Schaltfläche Settings im Startdialog für das Process Gateway öffnet den Dialog zur Auswahl der Variablen. Die ausgewählten Variablen werden in zenon oder im MODBUS-System laufend synchronisiert.



VARIABLES

Auswahl der Variablen, die nach MODBUS exportiert und mit zenon synchronisiert werden.

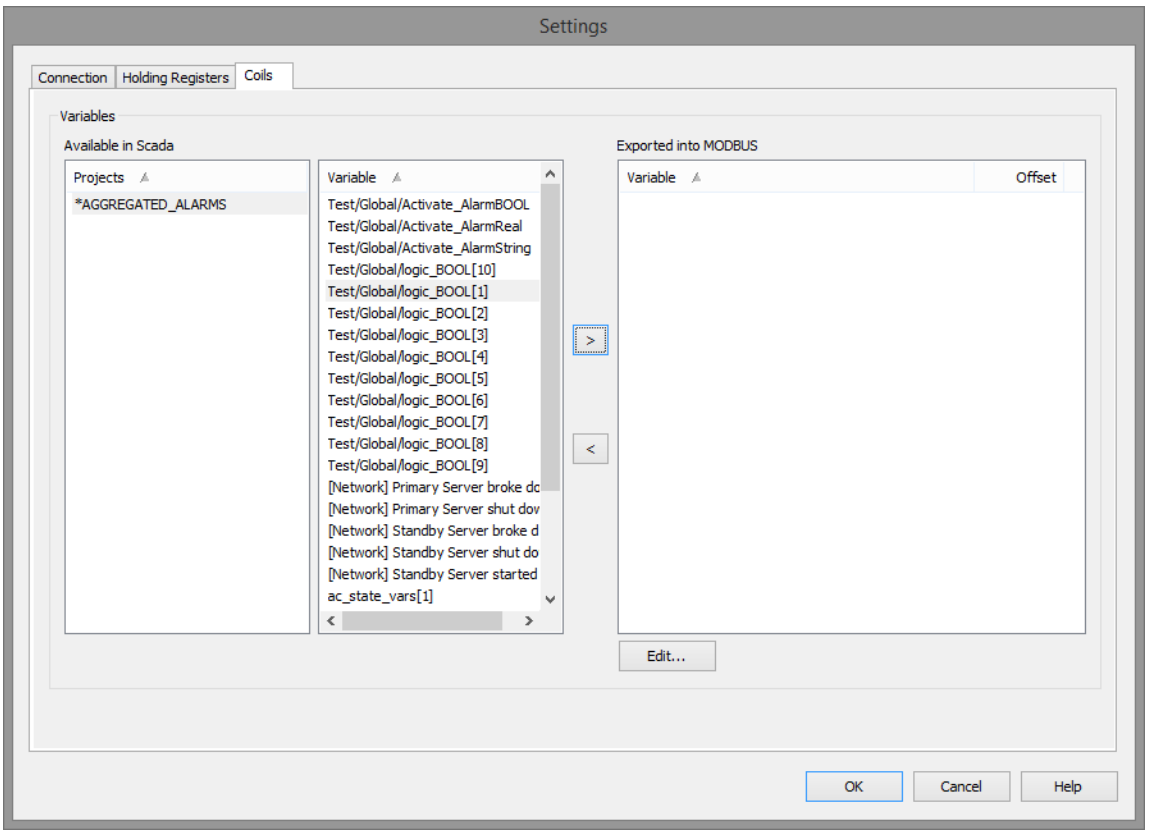
Parameter	Beschreibung
Available in Scada	Zeigt alle in zenon verfügbaren Variablen an Bei Mehrprojektverwaltung können Variablen aus allen aktiven Projekten ausgewählt werden. ▸ **Projects**: zeigt alle verfügbaren Projekte an, das Standardprojekt wird mit * gekennzeichnet ▸ **Variables**: bietet alle Variablen des gewählten Projekts zur Übernahme an
Exported into MODBUS	Listet alle nach MODBUS exportierten Variablen und deren Offset auf. Liste ist durch Klick auf Überschriften sortierbar. ▸ `Variables:` Liste aller SCADA-Variablen welche zum MODBUS exportiert werden. Variablen, die nicht aus dem Standardprojekt stammen, erhalten als Präfix den Projektnamen getrennt durch ein **#** vor dem Variablennamen. ▸ `Offset:` Offset der Variable (MODBUS Holding Register Offset)
Pfeil-Tasten	Zuordnung der Variablen zu MODBUS: ▸ Klick auf die Taste **>** fügt Variable aus der Liste **Available in Scada** zur Liste **Exported into MODBUS** hinzu. Es wird ein Dialog zur Zuordnung der Adresse geöffnet (siehe Abschnitt **Dialog Zuordnung**). ▸ Klick auf die Taste **<** entfernt Variable aus der Liste `Exported into MODBUS` und fügt sie wieder in die Liste `Available in Scada` ein.
Edit...	▸ Öffnet den Dialog für die Adressierung (auf Seite 262) der ausgewählten zenon Variable zu einem Offset in MODBUS.

Hinweise:

- Namensänderung Projekt und/oder Variable im Editor:
  Werden ein Projektname oder ein Variablenname im Editor geändert, muss diese Änderung in der Konfigurationsdatei (auf Seite 263) nachgezogen werden, damit die Namen in der INI-Datei und im Präfix der **Exported into MODBUS** angepasst werden.
- Änderung Standardprojekt:
  Wird in einer Mehrprojektverwaltung das Standardprojekt geändert, dann müssen die Variablen angepasst werden.
  Variablen des Standardprojekts dürfen kein Präfix haben. Alle anderen Variablen benötigen ein Präfix **Projektname#**.
- Write Holding Register:

- Maximum **block size**: 123

11.3 Coils



VARIABLES

Auswahl der Variablen, die nach MODBUS exportiert und mit zenon synchronisiert werden.

Parameter	Beschreibung
Available in Scada	Zeigt alle in zenon verfügbaren Variablen an Bei Mehrprojektverwaltung können Variablen aus allen aktiven Projekten ausgewählt werden. ▸ **Projects**: zeigt alle verfügbaren Projekte an, das Standardprojekt wird mit * gekennzeichnet ▸ **Variables**: bietet alle Variablen des gewählten Projekts zur Übernahme an
Exported into MODBUS	Listet alle nach MODBUS exportierten Variablen vom Datentyp `BOOL` und deren Offset auf. Liste ist durch Klick auf Überschriften sortierbar. ▸ `Variables:` Liste aller SCADA-Variablen welche zum MODBUS exportiert werden. Variablen, die nicht aus dem Standardprojekt stammen, erhalten als Präfix den Projektnamen getrennt durch ein **#** vor dem Variablennamen. ▸ `Offset:` Offset der Variable (MODBUS Holding Register Offset)
Pfeil-Tasten	Zuordnung der Variablen zu MODBUS: ▸ Klick auf die Taste **>** fügt Variable aus der Liste **Available in Scada** zur Liste **Exported into MODBUS** hinzu. Es wird ein Dialog zur Zuordnung der Adresse geöffnet (siehe Abschnitt **Dialog Zuordnung**). ▸ Klick auf die Taste **<** entfernt Variable aus der Liste `Exported into MODBUS` und fügt sie wieder in die Liste `Available in Scada` ein.
Edit	Öffnet den Dialog für die Adressierung (auf Seite 262) der ausgewählten zenon Variable zu einem Offset in MODBUS.

Hinweise:

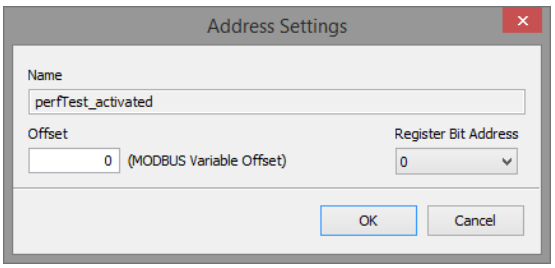
- Namensänderung Projekt und/oder Variable im Editor:
  Werden ein Projektname oder ein Variablenname im Editor geändert, muss diese Änderung in der Konfigurationsdatei (auf Seite 263) nachgezogen werden, damit die Namen in der INI-Datei und im Präfix der **Exported into MODBUS** angepasst werden.
- Änderung Standardprojekt:
  Wird in einer Mehrprojektverwaltung das Standardprojekt geändert, dann müssen die Variablen angepasst werden.
  Variablen des Standardprojekts dürfen kein Präfix haben. Alle anderen Variablen benötigen ein Präfix **Projektname#**.
- Write Holding Register:

- Maximum **block size**: `123`

11.4 Address Settings

DIALOG ZUORDNUNG

Wird eine Variable durch Klick auf die Schaltfläche **>** in die Liste `Exported into MODBUS` verschoben, erscheint der Dialog zur Adressierung der zenon Variable zu einem Offset in MODBUS:



Parameter	Beschreibung
Name	Name der Variablen. Hinweis: Der Name der Variable ist ausgegraut, da er nicht mehr änderbar ist.
Offset	Offset: MODBUS Holding Register Offset. Offset der Variable in MODBUS Achtung: Per Default ist immer 0 als Offset eingetragen.
Register Bit Address	▸ Auswahl aus Dropdownliste: `Bool`: Bitnummer im Holding-Register Default: `0` ▸ `USINT`: `Low Byte` oder `High Byte` im Holding-Register. Default: `High Byte` Hinweis: nicht verfügbar für die Zuordnung von **Coils**.
OK	Übernimmt Einstellungen und schließt den Dialog.
Cancel	Verwirft alle Änderungen und schließt den Dialog.

11.5 Hardware-Adresse des MODBUS-Systems

Das MODBUS-System vom **Process Gateway** hat eine einstellbare MODBUS-Hardware-Adresse. Diese wird im Konfigurationsdialog (auf Seite 256) über die Eigenschaft **Slave Adress eingestellt.** Als Standard wird die Adresse `1` vorgegeben.

11.6 Konfigurationsdatei: spezifische Einträge für AccessMODBUS

Die Konfigurationsdatei `zenProcGateway.ini` enthält spezifische Einträge für MODBUS. Die Datei muss im Systemordner liegen.

[MODBUS]

Eintrag	Beschreibung
COMPORT=	Serielle Schnittstelle (COM1 = 0, COM2 = 1,...). Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
BAUD=	Baudrate der seriellen Schnittstelle. Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
BYTESIZE=	Anzahl Datenbits der seriellen Schnittstelle
PARITY=	Paritätseinstellungen der seriellen Schnittstelle: 0=No 1=Odd 2=Even Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
STOPBITS=	Anzahl Stoppbits der seriellen Schnittstelle 0=1 1=1.5 2=2 Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
TIMEOUT=	Timeoutintervall bei serieller Kommunikation in Millisekunden
HWADDRESS=	Hardwareadresse des MODBUS-Systems. Default: 1 Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
REFRESHRATE=	Updatezeit in Millisekunden
SERIELL=	Serielle oder TCP/IP Kommunikation: 1 = seriell 0 = TCP/IP Kann über Konfigurationsdialog (auf Seite 256) eingestellt werden.
PORT=	TCP/IP-Port
TCPTIMEOUT=	TCP/IP-Timeout Intervall in Sekunden

[MODBUS VARIABLES]

Variablen, die im AccessMODBUS ausgetauscht werden sollen.

Hinweis: Dieser Eintrag ersetzt ab zenon 7.11 den ursprünglichen Eintrag **[VARIABLES]**. Damit wird verhindert, dass für Variablen aus dem Standardprojekt ein überflüssiger Austausch über die Option **Logic zu SCADA Verbindung** durchgeführt wird.

Eintrag	Beschreibung
COUNT	Anzahl der zu exportierenden Variablen
OFFSET_n	Modbus-Adresse, die der entsprechenden Nummer (**n**) zugeordnet ist. Nummerierung startet mit 0.
NAME_n	Name der Variable, die der entsprechenden Nummer (**n**) zugeordnet ist. Nummerierung startet mit 0. Beispiele: ▶ OFFSET_0=0 NAME_0=EMS_Supply area 1_forecast_final_consumption Variable aus Startprojekt ▶ OFFSET_1=2 NAME_1=BASISTUTORIAL#Temperaturfühler Variable aus Projekt "BASISTUTORIAL"

▶

12. MS Azure

Das **Process Gateway** baut mit dem Kommunikationsmodul **AccessAzure.dll** eine zyklische Verbindung zu MS Azure auf. Vom Process Gateway unterstützte Dienste in MS Azure:

- Servicebus Queue
- Event Hub
- IoT Hub

Zusätzlich kann das Process Gateway Daten vom IoT Hub via `Cloud2Device-Message` empfangen.

Daten aus der **Queue** können dann mit dem **AzureDrv** Treiber von MS Azure geholt und in Prozesse von zenon integriert werden. Inhalte des **IoT Hubs** können nur dann vom Treiber ausgelesen werden, wenn diese in die **Servicebus Queue** überführt werden. Dies kann mit Standarddiensten der MS Azure Cloud realisiert werden.

Daten aus einem **Event Hub** dienen für Drittanwendungen und können von zenon nicht mehr eingelesen werden.

KOMMUNIKATION DES TREIBERS

Zum Auslesen der Werte aus der Servicebus Queue baut der **AzureDrv** Treiber eine Verbindung zur Servicebus Queue mit dem konfigurierten Namen auf und entnimmt ihr alle enthaltenen Nachrichten. Diese Nachrichten werden entpackt und die darin enthaltenen Onlinewerte den Variablen zugewiesen. Schlüssel dafür ist die **Symbolische Adresse**.
Alle beim ersten erfolgreichen Verbindungsaufbau bereits in der Queue befindlichen Nachrichten werden geladen und verworfen. Es werden immer nur die aktuellen Werte angezeigt.

Das heißt:
Jede Treiberinstanz auf jedem Rechner besitzt ihre eigene **Queue** als Eingangskanal.

Beispiel

Die Servicebus Queue für den Rechner **MYSERVER1** und dem konfigurierten Präfix **onlinedata** heißt in MS Azure: **onlinedata_myserver1**.

Es werden numerische `(DOUBLE)` und alphanumerische `(STRING)` Werte unterstützt. Der Zeitstempel und die System-Statusbits werden in die Zielvariable übernommen.

Allgemeine Informationen zu MS Azure finden Sie auch im Handbuch **MS Azure**.

12.1 Modulkonfiguration

Konfigurationsdialog für MS Azure Anbindung via Process Gateway.

VARIABLES

In der Gruppe **Variables** konfigurieren sie jene Variablen, deren Werte von zenon in einen MS Azure Service Bus gespeichert werden.

Parameter	Beschreibung
Available in Scada	Zeigt alle in zenon verfügbaren Variablen an Bei Mehrprojektverwaltung können Variablen aus allen aktiven Projekten ausgewählt werden.
`Projects`	Liste aller verfügbaren Projekte. Das Standardprojekt ist mit einem `*` gekennzeichnet.
`Variables`	Liste aller Variablen des gewählten Projekts. Liste ist sortierbar, Mehrfachauswahl ist möglich. Tipp: Doppelklick auf Variable verschiebt diese.
Schaltfläche >	Ausgewählte Variablen aus Liste Variables werden in die Liste **Exported into MS Azure** verschoben.
Schaltfläche <	Ausgewählte Variablen werden aus der Liste **Exported into MS Azure** entfernt.
Exported into MS Azure	Liste der Variablen, die via Process Gateway in den MS Azure Service Bus geschrieben werden. Diese werden mit Name (**Points**) und Kommunikationsrichtung (**Direction**) angezeigt. Doppelklick auf einen Eintrag schaltet die Kommunikationsrichtung um. Der Schlüssel für die Werte in MS Azure ist immer `PROJECTNAME#VARIABLENAME`. **Points:** Der Name besteht aus: ▸ Projektname ▸ **#** (als Trennzeichen) ▸ Variablenname. **Direction:** Zeigt die Kommunikationsrichtung an: ▸ `read only` ▸ `write only` ▸ `read/write` Default: `read only` Änderung der Richtung durch Doppelklick auf Variable. Hinweis: Nur verknüpfte Variablen können beschrieben werden.

MS AZURE CONNECTIONS

In der Gruppe **MS Azure Connections** sind alle Zielverbindungen in MS Azure anzugeben, in welche die aktuellen Werte der gewählten Variablen eingefügt werden.

Die aktuellen Variablenwerte werden dabei gleichzeitig in alle erstellten Verbindungen eingefügt.

Parameter	Beschreibung
Liste MS Azure Verbindungen	Listet alle konfigurierten Verbindungen zu MS Azure auf. Jede Verbindung besteht aus dem Verbindungsnamen (MS Azure Connection) und dem Service Bus Queue Name. ▶ Neuanlage einer Verbindung mit der Schaltfläche **New...** ▶ Mit der Schaltfläche **Edit...** kann eine gewählte Verbindung geändert werden.
MS Azure Connection	**MS Azure** Verbindungsadresse.
Service Bus Queue Name	Name der Warteschlange im MS Azure Service-Bus.
New...	Öffnet Dialog zur Konfiguration der **MS Azure** Verbindung.
Edit...	Öffnet bestehende Verbindungen zur Konfiguration der **MS Azure** Verbindung.
Delete	Löscht die ausgewählte **MS Azure** Verbindung aus der Liste.

MS AZURE SETTINGS

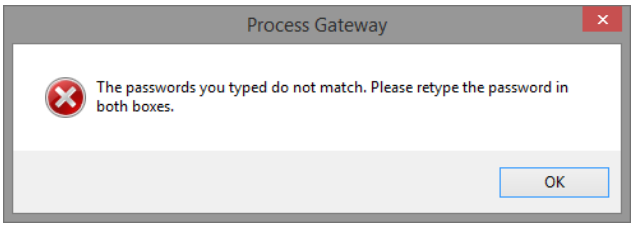
Parameter	Beschreibung
Integrity period	Zeitintervall, in dem die aktuellen Werte der gewählten Variablen als Abbild in die **MS Azure Service Bus** Warteschlange geschrieben werden. Findet während dieser Zeitspanne eine Wertänderung einer Variable statt, wird diese Wertänderung sofort an die **MS Azure** Warteschlange übermittelt. Default: 5 s
HTTP Proxy Domain	Adresse des Proxy-Servers im Netzwerk.
HTTP Proxy User	Eingabefeld des Benutzernamens für die Anmeldung am Proxy-Server.
HTTP Proxy Password	Eingabefeld des Passworts für die Anmeldung am Proxy-Server. Hinweis: Die Eingabe wird - auch während der Eingabe - mit Punkten dargestellt.
Confirm HTTP Proxy Password	Eingabefeld für die Bestätigung des Passworts für die Anmeldung am Proxy-Server. Hinweis: Die Eingabe wird - auch während der Eingabe - mit Punkten dargestellt.

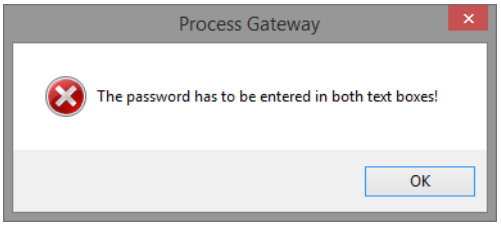
NAVIGATION

Parameter	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Cancel	Verwirft alle Änderungen und schließt den Dialog.

DIALOGE BEI FEHLEINGABEN

Konfigurationen des Proxy-Servers werden validiert. Bei einer Fehlprojektierung wird ein entsprechender Warndialog angezeigt.



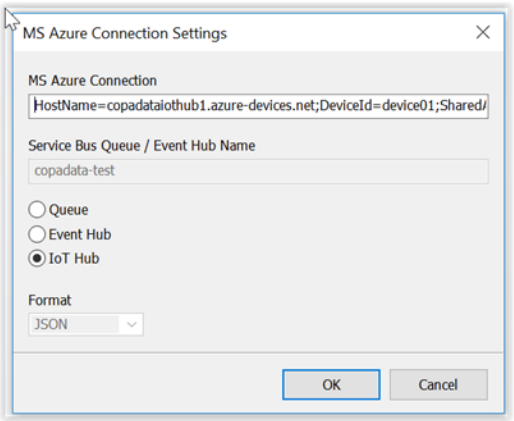


Achtung

Damit die Proxy-Einstellungen funktionieren, muss auf dem Rechner .NET 4.5 installiert sein.

12.1.1 MS Azure Connection Settings

Konfigurationsdialog der Verbindung zu **MS Azure**:



Parameter	Beschreibung
MS Azure Connection	**MS Azure** Verbindungsadresse. Hinweis: Diese Adresse können Sie im Azure-Verwaltungsportal unter **Manage Connection Strings** des gewünschten **Servicebus Namespace** auslesen und kopieren.
Service Bus Queue / Event Hub Name	Name der **MS Azure Service Bus** Warteschlange oder des **Event Hub**. Auswahl über Radiobuttons. Hinweis: Es sind nur Zeichen erlaubt, die im Namen einer MS Azure Service Bus Queue oder in einem Event Hub vorkommen dürfen. Verwenden Sie einfache, kurze und aussagekräftige Namen. Vermeiden Sie Sonderzeichen, sprachspezifische Buchstaben und Leerzeichen. **Service Bus Queue** Der Name der **Service Bus Queue** setzt sich zusammen aus: ▸ einem frei konfigurierbaren Präfix ▸ einem Unterstrich (_) ▸ dem NETBIOS-Rechnernamen (ohne Domänensuffix) in Kleinschreibung Existiert der Name im Namespace noch nicht, wird eine Queue mit diesem Namen erstellt. Voraussetzung: Die entsprechenden Rechte sind vorhanden. **Event Hub** ▸ **Nachrichtenformat**: Auswahl in Dropdownliste der Option **Format**. ▸ **Event Hub Name**: Name des Event Hubs im MS Azure Service Bus Der Event Hub muss mit diesem Namen im Namespace bereits angelegt sein. Hinweis: Ausgegraut, wenn die Option **IoT Hub** als Kommunikationsart zu MS Azure aktiviert ist.
Queue	Kommunikation zu MS Azure via `Service Bus Queue`. ▸ `Aktiv`: Die Verbindung wird über den Namen der **Service Bus Queue** hergestellt.
Event Hub	▸ Kommunikation zu MS Azure via `Event Hub`. ▸ `Aktiv`: Die Verbindung wird über den Namen des **Event Hub** hergestellt. Auswahl des Nachrichtenformats über Option **Format**.

Parameter	Beschreibung
IoT Hub	▸ Kommunikation zu MS Azure via `IoT Hub`. ▸ `Aktiv`: Versenden von Istwerten und Empfang von Nachrichten via `IoT Hub`. Eintrag des `Device-Connection Strings` im Eingabefeld **MS Azure Connection**.
Format	Nachrichtenformat für Verbindung über **Event Hub**. Auswahl aus Dropdownliste: ▸ `XML` ▸ `JSON` ▸ `BOND` (compact binary)
OK	Übernimmt Einstellungen und schließt den Dialog.
Cancel	Verwirft alle Änderungen und schließt den Dialog.

Info

Der Service Bus Queue Name ist frei konfigurierbar.
Bei der ersten Kommunikation zu MS Azure wird diese Warteschlange automatisch in MS Azure angelegt.

Legen Sie für jeden Rechner, der Daten von MS Azure abruft, über den **AzureDrv** Treiber eine eigene MS Azure Connection an.

Kommunikation via IoT Hub

Für die Kommunikation via IoT Hub gilt:

- Alle Inhalte werden durch das **Process Gateway** in dem korrekten Format übermittelt.
- Istwerte werden als Nachricht an den **IoT Hub** gesendet.
- Vom **IoT Hub** empfangene Nachrichten werden dekodiert und als Sollwert auf die enthaltene Variable abgesetzt.

IOT HUB NACHRICHT

Eine Nachricht via `IoT Hub` enthält folgende Inhalte:

Parameter	Datentyp	lesen (receive)	schreiben (send)	Beschreibung
project	STRING	X	X	zenon Projektname
variable	STRING	X	X	zenon Variablenname
isstring	NUMBER	X	X	Typ der zenon Variable ▸ 0 = Variablenwert ist numerisch ▸ 1 = Variablenwert ist ein String
numvalue	NUMBER	X	X	Numerischer Wert der zenon Variable. Hinweis: nur für numerische Variablen wenn `isstring=0`. Bei String-Variablen ist der Wert für `numvalue 0`.
strvalue	STRING	X	X	Stringwert Hinweis: nur für String-Variablen wenn `isstring=1`. Bei numerischen Variablen ist der Wert für `strvalue` leer.
time	ISO timestamp	X	X	Zeitstempel der zenon Variable.
status	NUMBER	X	X	Status der zenon Variable.
isintegrity	NUMBER	--	X	Auslöser für die Datenübertragung. ▸ 0 = Nachricht aufgrund von Wertänderung an MS Azure gesendet. ▸ 1 = Nachricht wurde zyklisch gesendet. Hinweis: Das zyklische Senden entspricht der projektierten Option **Integrity period** in der Modulkonfiguration (auf Seite 266).
sequence	NUMBER	--	X	Laufende Nummer für gesendete Nachrichten. Dies wird durch das Process Gateway automatisch durchgeführt und wird bei jeder Nachricht erhöht.
msgtime	ISO timestamp	--	X	Zeitstempel der Nachricht.

Legende:

X: wird unterstützt

--: wird nicht unterstützt

Die Nachrichten in Empfangsrichtung (`receive`) sind identisch mit jenen in Senderichtung (send). Die Parameter `isintegrity`, `sequence` und `msgtime` werden beim Empfangen (`receive`) nicht benötigt. Alle Inhalte werden durch das **Process Gateway** in dem korrekten Format übermittelt. Alle Inhalte werden durch das **Process Gateway** in dem korrekten Format übermittelt.

Beispiel

Mögliches Format einer Nachricht, die vom Process Gateway gesendet wird:

```
{"isintegrity":1,"isstring":1,"msgtime":"2017-05-03T08:51:23.26
0Z","numvalue":0,"project":"AZURE_TEST","sequence":1,"status":1
078067200,"strvalue":"sunshine","time":"2017-04-11T05:28:33.542
Z","variable":"weather"}
```

12.2 Nachrichtenformate

SERVICE BUS QUEUE

Die Nachrichten in der Queue müssen als Nutzdaten eine serialisierte .NET Klasse `OnlineValueMessage` besitzen.

FORMAT:

```
public class OnlineValueMessage
{
  public string strVarProject { get; set; }
  public string strVarName { get; set; }
  public long nVarID { get; set; }
  public double fValue { get; set; }
  public string strValue { get; set; }
  public bool bValueIsString { get; set; }
  public long nTime_s { get; set; }
  public short nTime_milli { get; set; }
  public long nStatus { get; set; }
}
```

EVENT HUB

Bei Verwendung des Event Hub als Verbindung, kann zwischen `XML`, `JSON` und `BOND` (compact binary) als Nachrichtenformat gewählt werden. Partition-Key ist der zenon Variablenname.

FORMAT BOND

```
namespace AzureServiceBusShared

struct ArchiveMessage
{
  0: string strArvProject;
  1: string strArvName;
  2: string strVarProject;
  3: string strVarName;
  4: int64 nVarID;
  5: int16 nCalc;
  6: double fValue;
  7: string strValue;
  8: bool bValueIsString;
  9: int64 nTime_s;
  10: int16 nTime_milli;
  11: int64 nStatus;
}

struct OnlineValueMessage
{
  0: string strVarProject;
  1: string strVarName;
  2: int64 nVarID;
  3: double fValue;
  4: string strValue;
  5: bool bValueIsString;
  6: int64 nTime_s;
  7: int16 nTime_milli;
  8: int64 nStatus;
}
```

12.3 Fehlermeldungen

Fehlermeldung in der LOG-Datei:

Fehlermeldung	Debug Level	Beschreibung
Project %s inactive fail!	**Error**	Der Project-Inactive Event konnte vom Gateway nicht in der vorgegebenen Zeit bearbeitet werden.
Project %s active fail!	**Error**	Der Project-Active Event konnte vom Gateway nicht in der vorgegebenen Zeit bearbeitet werden.
Value change queue full: Value change for id:%u of project:%s lost!	**Error**	Die Queue für die Wertänderungen aus der Runtime ist voll. Zusätzliche Werte werden verworfen. Vermutlich ist die Verbindung zu Azure zu langsam, um alle Wertänderungen weiter zu geben.
Adding of %d Online Rows failed for '%s' Queue '%s	**Error**	Das Einfügen von Wertänderungen in die Service Bus Queue ist fehlgeschlagen.
Adding of %d Online Rows succeeded for '%s' Queue '%s'	**Deep Debug**	Das Einfügen von Wertänderungen in die Service Bus Queue war erfolgreich.
Starting to add Online Rows failed for '%s' Queue '%s'	**Error**	Der Verbindungsaufbau zur Service Bus Queue ist fehlgeschlagen.
Creating Azure Wrapper failed	**Error**	Die ManagedAzureWrapper.dll konnte nicht geladen werden.

IOT

Parameter	Beschreibung	Beschreibung
Sending value to '%s' failed	**Error**	Das Sollwert setzen ist fehlgeschlagen
IoTHubMessage_GetByteArray() failed	**Error**	Der SDK API-Aufruf ist fehlgeschlagen.
Received message from IoT Hub	**Deep Debug**	Nachricht vom IoT-Hub empfangen.
Exception when deserializing payload: %s	**Error**	Empangene Nachricht ist im falschen Format
IoTHubClient_CreateFromConnectionString() failed	**Error**	Der SDK API-Aufruf ist fehlgeschlagen.
IoTHubClient_SetMessageCallback() failed	**Error**	Der SDK API-Aufruf ist fehlgeschlagen.
IoTHubMessage_CreateFromByteArray() failed	**Error**	Der SDK API-Aufruf ist fehlgeschlagen.
Sent message to IoT Hub	**Deep Debug**	Die Nachricht wurde gesendet.

12.4 Konfigurationsdatei: spezifische Einträge für MS Azure

Die Konfigurationsdatei `zenProcGateway.ini` enthält folgende Einträge für MS Azure.

[AZURE]

Eintrag	Beschreibung
INTEGRITYPERIOD=	Intervall des Schreibzyklus in Sekunden.
PROXY_DOMAIN=	Adresse des Proxy-Servers.
PROXY_USER=	Benutzername für die Anmeldung am Proxy-Server.
PROXY_PWD=	Passwort für die Anmeldung am Proxy-Server. Hinweis: Das Passwort wird auch in der .INI-Datei verschlüsselt dargestellt.

Hinweis: Damit die Proxy-Einstellungen funktionieren, muss auf dem Rechner .NET 4.5 installiert sein.

[VARIABLES]

Eintrag	Beschreibung
Name_n=	Name der Variable für das Process Gateway. Format: `QUELLPROJEKT#QUELLVARIABLE`. Nummerierung (n): `n` steht dabei für die laufende Nummer der Verbindungskonfiguration. Diese Nummerierung beginnt mit 0. Beispiel: ▸ 1 Variable ergibt `n = 0` ▸ 3 Variablen ergeben für den dritten Eintrag `n = 2`
Count=	Anzahl der Variablen. Hinweis: Hier beginnt die Nummerierung mit `1`. Acht Verbindungen ergeben `Count=8`.

[CONNECTIONS]

Eintrag	Beschreibung
CONN_n=	Name der MS Azure Verbindung. Nummerierung (n): `n` steht dabei für die laufende Nummer der Verbindungskonfiguration. Diese Nummerierung beginnt mit 0. Beispiel: ▸ 1 Verbindung ergibt `n = 0` ▸ 3 Verbindungen ergeben für den dritten Eintrag `n = 2`
QUEUE_n=	Name der MS Azure Warteschlange. Für die Nummerierung (_n) gelten dieselben Regeln wie für **CONN**
COUNT=	Anzahl der konfigurierten MS Azure Verbindungen. Hinweis: Hier beginnt die Nummerierung mit `1`. Zwei Verbindungen ergeben `Count=2`.

13. OPC UA Server

Der **OPC UA Server** stellt die im Projekt verwendeten Variablen allen standardkonformen **OPC UA Client** Tools zur Verfügung.

> **Info**
>
> *Der **OPC UA Server** kann für alle zenon Versionen ab 5.50 eingesetzt werden.*

Der Vorläufer des **OPC UA Servers** ist der OPC DA Server. Dieser ist nicht Bestandteil des **Process Gateways**.

Der **OPC UA Treiber** und alle Dialoge stehen nur in englischer Sprache zur Verfügung.

MEHRPROJEKTFÄHIGKEIT

Der **OPC UA Server** ist ab zenon 7.11 mehrprojektfähig. Es können Variablen aus dem Runtime-Projekt und allen seinen Unterprojekten gewählt werden. Dabei wird der Objektname aus dem Variablennamen und dem Projektnamen zusammengesetzt. Konfigurationen für **OPC UA Clients**, die vor zenon 7.11 erstellt wurden, sind damit nicht kompatibel. Diese müssen beim Einsatz von zenon 7.11 oder höher angepasst werden.

Ab der Version 7.60 ändern sich die `NodeIDs`, wenn Variablen nicht explizit in der Konfiguration definiert sind. Der Datentyp der `NodeID` ist `String`. Die `NodeID` setzt sich zusammen aus der Project GUID und der Variablen ID. Wenn Variablen explizit konfiguriert sind, hat diese Änderung keine Auswirkung.

Beispiel

NodeID: "6d5ea886-1b04-4d9c-aa35-2b5664c500cd-38d"; dabei gilt: "6d5ea886-1b04-4d9c-aa35-2b5664c500cd" ist die **Projekt ID** und "38d" entspricht der Variablen ID 909.

Ab der Version 7.60 ist bei neuen Konfigurationen der `NameSpace Index` per Default "2", anstelle "80". In der Benutzeroberfläche ist der `NameSpace Index` nicht länger konfigurierbar. Für bestehende Konfigurationen hat diese Änderung keine Auswirkung.

Über den Eintrag "**configuration_version=1**" in der INI-Datei kann das frühere Verhalten erneut aktiviert werden. Dies kann z.B. für einen `NameSpace Index` mit 80 oder für einen konfigurierbaren `NameSpace Index` sinnvoll sein.

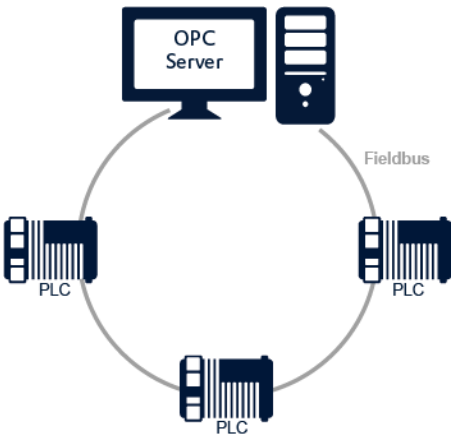
13.1 Vergleich OPC Server und OPC UA Server

Die **OPC UA Server** Technik hat den OPC Server abgelöst. In den meisten Einsatzgebieten empfiehlt sich die Verwendung des **OPC UA Servers.**

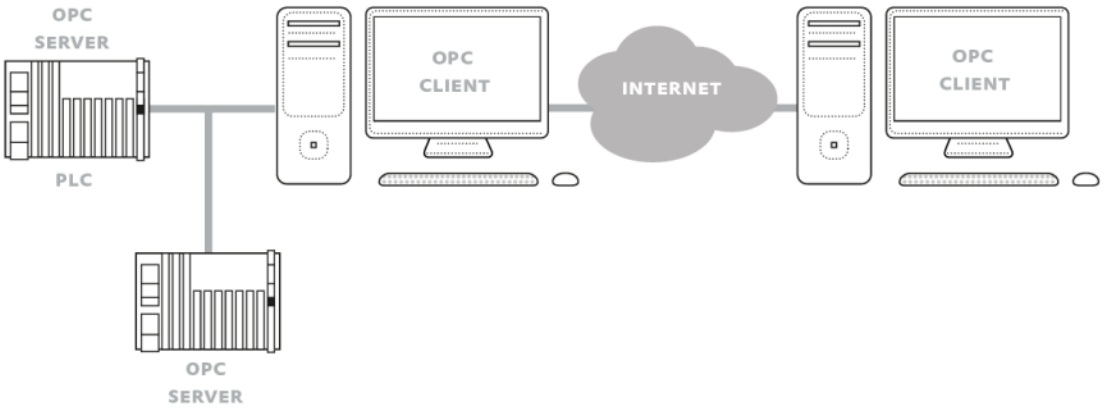
OPC SERVER

- OPC Task Force seit 1995
- einheitliche Schnittstelle in der Automatisierungstechnik
- baut auf Microsofts COM/DCOM Technologie auf
- Seit 1996 OPC Foundation

- mehrere Spezifikationen für unterschiedliche Anwendungsfälle



OPC UA SERVER



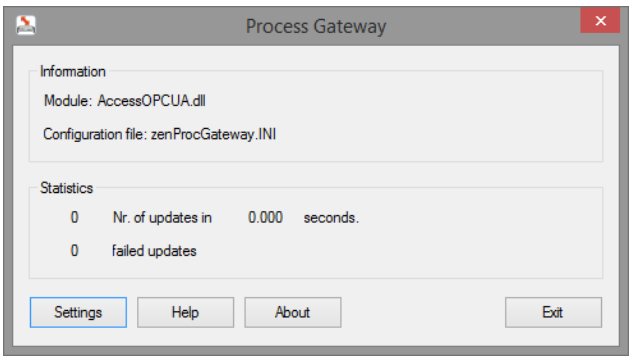
VERGLEICH OPC SERVER MIT OPC UA SERVER

Parameter	OPC Server	OPC UA Server
Datenmodell	eingeschränkt	für alle Anwendungsgebiete
Implementierung	herstellerabhängig	herstellerunabhängig
Interoperabilität	--	**X**
Konfiguration	aufwändig	einfach
Mehrere Instanzen	--	**X**
Netzwerk-Einsatz	--	**X**
Netzwerk-Technologie	--	OPC UA TCP binary Protocol
Performance	niedrig	hoch
Plattformen	nur Windows	unabhängig
Ressourcenbedarf	hoch	niedriger
Serviceorientierte Architektur	--	**X**
Sicherheit	veraltet	nach aktuellen Standards
Verbindungssicherheit	--	**X**
Windows CE	--	**X**

Legende:

- **X**: wird unterstützt
- **--**: wird nicht unterstützt

13.2 Modulkonfiguration



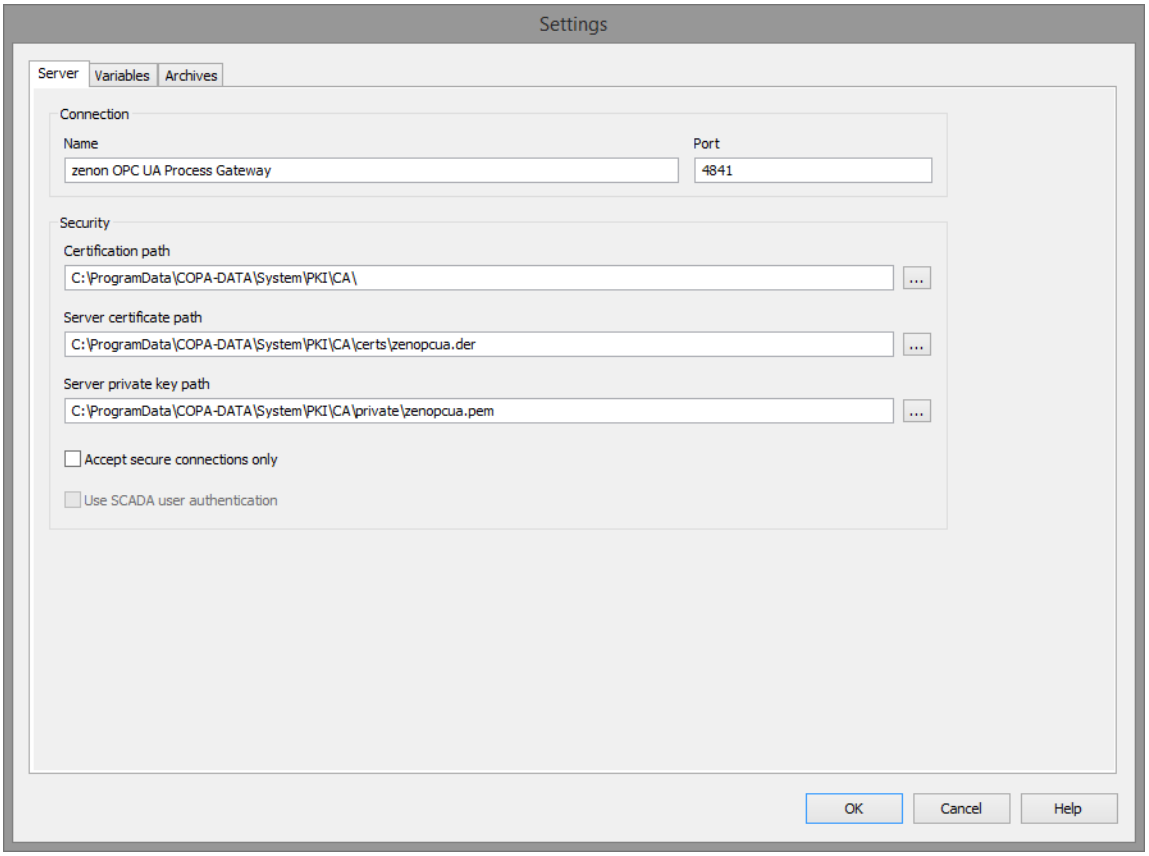
Um Server und Variablen für OPC UA zu konfigurieren:

1. Klicken Sie im **Process Gateway** Statistik-Dialog die Schaltfläche **Configuration**. Der Dialog für die Konfiguration des **OPC UA Moduls** wird geöffnet.

2. Im Modul-Konfigurationsdialog konfigurieren Sie:
   - Server (auf Seite 284)
     Verbindungsparameter zum **OPC UA Server**
   - Variablen (auf Seite 291)
     Variablenzuordnung

13.2.1 Server

In der Registerkarte **Server** konfigurieren Sie Verbindungsparameter zu einem OPC UA Server.



CONNECTION

Kommunikationsparameter zum OPC UA Server.

Parameter	Beschreibung
Name	Name des Servers, wie er am Client angezeigt wird. Default: `zenon OPC UA Process Gateway`
Port	TCP Port, über den der Client die Verbindung zum OPC UA Server herstellt. Default: `4841`

SECURITY

Einstellungen und Zuordnung von Zertifikaten für die gesicherte und verschlüsselte Kommunikation.

Parameter	Beschreibung
Certificates path	Pfad für den Stammordner mit den Zertifikatsdateien und den jeweiligen Unterordnern. Klick auf die Schaltfläche **...** öffnet den Datei-Auswahldialog. Default: `%ProgramData%\PKI\CA\` Hinweis: Wird dieser Stammordner geändert, müssen auch die Pfade für **Server certificate path** und **Server private key path** angepasst werden.
Server certificate path	Datei mit öffentlichem Schlüssel des Servers. Dieser öffentliche Schlüssel wird vom Client zum Verschlüsseln der Nachrichten verwendet. Klick auf die Schaltfläche **...** öffnet den Datei-Auswahldialog. Im Dialog werden Dateien mit folgendem Dateityp zur Auswahl angezeigt: ▸ Format: `DER` ▸ Endung: `.der` Default: `%ProgramData%\PKI\CA\certs\zenopcua.der` Hinweis: Dieser Pfad muss ein Unterordner des projektierten **Certificates path** sein. Im Fehlerfall wird eine Warnmeldung visualisiert und die Schaltfläche **OK** ist ausgegraut.
Server private key path	Datei mit privatem Schlüssel des Servers. Dieser private Schlüssel wird vom Server zum Entschlüsseln der Nachrichten des Clients verwendet. Klick auf die Schaltfläche **...** öffnet den Datei-Auswahldialog. Im Dialog werden Dateien mit folgendem Dateityp zur Auswahl angezeigt: ▸ Format: `PEM` ▸ kein Passwort Default: `%ProgramData%\PKI\CA\private\zenopcua.pem` Hinweis: Dieser Pfad muss ein Unterordner des projektierten **Certificates path** sein. Im Fehlerfall wird eine Warnmeldung visualisiert und die Schaltfläche **OK** ist ausgegraut.
Accept secure connections only	Checkbox zur Aktivierung, dass nur eine sichere Verbindung für den Verbindungsaufbau verwendet werden kann. ▸ `Aktiv`: Ein Verbindungsaufbau wird nur mit einer sicheren Verbindung akzeptiert. Die Absicherung erfolgt jeweils mit Signatur oder mit Signatur und Verschlüsselung.

	▸ `Inaktiv`: Ein Verbindungsaufbau wird auch mit einer unsicheren Verbindung akzeptiert. Als sichere Verbindung gelten: ▸ `Basic128Rsa15`: RSA15 als Key Wrap-Algorithmus und 128-bit Basic als Algorithmus für die Nachrichtenverschlüsselung ▸ `Basic256`: 256-bit Basic als Algorithmus für die Nachrichtenverschlüsselung Default: `inaktiv` Hinweis: Bei Verbindungen mit Signatur sind die Pakete gegen Manipulationen gesichert, der Paketinhalt wird jedoch nicht verschlüsselt.
Trust all client certificates	Checkbox zur Auswahl, ob alle Client-Zertifikate oder nur am Server bekannte Client-Zertifikate für den Verbindungsaufbau akzeptiert werden. ▸ `Aktiv`: Alle Client-Zertifikate werden akzeptiert, die Verbindung wird aufgebaut. ▸ `Inaktiv`: Alle Clients, die sich mit dem Server verbinden wollen, müssen ein gültiges Zertifikat übermitteln. Dieses Zertifikat wird zusätzlich darauf überprüft, ob es sich in der Liste der bekannten Client-Zertifikate (**Certificate trust list path**) befindet. Default: `aktiv` Hinweis: Ein **OPC UA Client** ohne sichere Verbindung sendet kein Zertifikat. Für einen **OPC UA Client** mit sicherer Verbindung muss das Zertifikat immer im Ordner der **Trust List** abgelegt sein.
Use SCADA user authentication	Checkbox für Konfiguration der Verbindung. Ist im zenon Projekt keine Benutzerverwaltung projektiert, ist diese Checkbox ausgegraut. ▸ `Aktiv`: Clients können sich am Server nur mit gültiger Benutzerkennung und Passwort anmelden. Hinweis: Dazu muss im zenon Editor die Benutzerverwaltung aktiviert und projektiert sein. ▸ `Inaktiv`: Anonymes Login ist möglich. Achtung: Benutzername und Passwort werden im **OPC UA** Telegramm nicht verschlüsselt übertragen. Verwenden Sie bei Bedarf eigene Benutzer im Projekt ohne zugeordnete Berechtigungsebenen.

DIALOG BEENDEN

OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog. Achtung: Alle aktiven Verbindungen werden beendet und der Server wird mit den neuen Einstellungen neu gestartet. Dabei können Verbindungsfehler auftreten.
Cancel	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Help	Öffnet die Online-Hilfe.

FEHLERMELDUNGEN

Fehlermeldungen werden in roter Schrift im Dialog angezeigt. Dabei werden nicht alle Fehler gemeinsam dargestellt. Angezeigt wird jeweils der erste Fehler der Konfiguration. So kann es dazu kommen, dass nach Korrektur einer fehlerhaften Konfiguration eine neue Fehlermeldung angezeigt wird.

Zertifikate

*Alle Zertifikate müssen dem Format **X509** entsprechen.*

CERTIFICATE STORE

*Der **OPC UA Server** im **Process Gateway** verwendet ein sogenanntes `OpenSSL Directory certificate store`.*

SERVER CERTIFICATE

Das beim ersten Start optional erstellte Serverzertifikat ist ein selbstsigniertes Zertifikat. Das Zertifikat enthält den öffentlichen Schlüssel. Zusätzlich wird ein passender privater RSA-Schlüssel (1024 Bit) generiert.

*Obwohl das **OPC UA Gateway** grundsätzlich mit dem selbst erstellten Zertifikat lauffähig ist, wird empfohlen für den Produktivbetrieb ein den Anforderungen entsprechenden Zertifikat zu erstellen. Hierzu kann das selbst erstellte Zertifikat als Vorlage dienen für Anwendungen die eine Erstellung der Zertifikate nach Benutzerkriterien ermöglichen.*

Achtung

Der Dateiname `zenopcua` wird sowohl vom **OPC UA Server** als auch vom **OPCUA32 Client Treiber** verwendet. Wenn eine verschlüsselte Kommunikation zwischen dem **OPCUA32 Client Treiber** und **OPC UA Server** im **Process Gateway** auf dem gleichen Rechner für Tests konfiguriert wird, wird am besten das Zertifikat für den **OPCUA32 Client Treiber** umbenannt, bevor das Zertifikat in die **Trust List** vom **OPC UA Server** kopiert wird.

*Als URI im Feld **SubjectAltName** enthält das Zertifikat den `Full Qualitfied Domainname` (FQN) des Rechners, die Default-Portnummer `4841` vom **OPC UA Server** und einen DNS Eintrag mit dem Rechner-Namen. Die URI des **OPCUA Servers** wird dynamisch zusammengestellt und ändert sich, z. B. wenn die Portnummer geändert wird. In diesem Fall wird das Serverzertifikat nicht geändert. Ein Client, der eine strikte Überprüfung des Serverzertifikats durchführt, kann dann das Serverzertifikat verweigern. Auch kann es notwendig sein, ein eigenes Zertifikat zu erstellen, welches für den `SubjectAltname` neben den DNS Eintrag auch einen Eintrag für die IP Addresse enthält.*

CERTIFICATE TRUST LIST/CERTIFICATE TRUST CHAIN

*Alle Zertifikate mit der Endung `.der` werden vom **OPC UA Server** geprüft und als vertrauenswürdiges Zertifikat eingestuft. Alternativ zu einem selbstsignierten Application Instance **Zertifikat** eines **OPC UA Clients** kann hier auch das öffentliche Root-Zertifikat der Certificate Authority abgelegt werden, das zum Signieren eines Application Instance **Zertifikats** eines **OPC UA Clients** verwendet wurde. Das Application Instance **Zertifikat** des Clients braucht in diesem Fall nicht am Server vorhanden sein.*

CERTIFICATE REVOCATION LIST

Wenn ein **OPC UA Client** ein Application Instance **Zertifikat** verwendet, das von einer Certificate Authority signiert wurde, wird vom Server zusätzlich zum Zertifikat auch die Certificate Revocation List geprüft. Der **OPC UA Server** prüft sämtliche Dateien mit der Endung **.crl** im Unterordner **Crl**.
Achtung: Die **.crl** Datei muss im `PEM Format` abgelegt sein. Pro Certificate Authority darf jeweils nur eine einzige **.crl** Datei vorhanden sein.

Die Verbindung wird abgelehnt, wenn:

- das Client Zertifikat in der **Revocationlist** enthalten ist (BadCertificateRevoked)
- die CRL nicht mehr aktuell ist (BadCertificateTimeInvalid)

Ist für ein Zertifikat, das von einer Certificate Authority signiert wurde, keine **Certificate Revocation List** vorhanden, wird angenommen dass es keine solche Revocation List gibt. In diesem Fall wird das Zertifikat als vertrauenswürdig betrachtet.

Zertifikate neu erstellen

Bei Bedarf können beim Start des OPC UA Servers neue Zertifikate generiert werden.
Dazu:

1. Entfernen Sie die vorhandenen Zertifikate.
   Dazu haben Sie zwei Möglichkeiten:
   a) Löschen Sie den Ordner `%programdata%\COPA-DATA\system\PKI`.
   Dabei werden alle **Trusted certificates** und **Certificate revocation lists** entfernt.
   Oder:
   b) Löschen Sie folgende Dateien, um alle anderen Zertifikate und Listen zu erhalten:
   zenopcua.der im Ordner `%programdata%\COPA-DATA\system\PKI\CA\certs`
   zenopcua.pem im Ordner `%programdata%\COPA-DATA\system\PKI\CA\private`
   zenopcua.crl im Ordner `%programdata%\COPA-DATA\system\PKI\CA\crl`
2. Öffnen Sie die Konfigurationsdatei **zenprocgateway.ini**.
3. Navigieren Sie zum Abschnitt **[OPCUA]**.
4. Entfernen Sie folgende Einträge:
   - **CERTIFICATES_PATH=**
   - **SERVER_CERTIFICATE_LOCATION=**
   - **SERVER_PRIVATE_KEY_LOCATION=**
5. Starten Sie das OPC UA **Process Gateway**.
6. Bestätigen Sie die Nachrichtenbox mit Klick auf die Schaltfläche **Ja**, um neue Zertifikate zu erstellen.

> **Achtung**
>
> *Wenn Server-Zertifikate erneuert werden, muss jeder OPC UA Client, der ein gültiges Server Zertifikat benötigt, das erneuerte Zertifikat erhalten.*

NEUERSTELLUNG DES SERVER ZERTIFIKATS

Folgende Situationen können es notwendig machen das Server-Zertifikat neu zu erstellen:

- mehrere Instanzen vom Process Gateway auf dem gleichen System mit unterschiedliche Port Nummer
- eine Änderung von der Port Nummer
- eine Änderung vom Rechnernamen
- das Zertifikat ist abgelaufen

13.2.2 Variables

Variablen für das **OPC UA Process Gateway** können automatisch oder individuell ausgewählt werden.

Es können Variablen aus dem laufenden Projekt und seinen Unterprojekten ausgewählt werden. Der Name am Server setzt sich aus dem Variablennamen und dem Projektnamen zusammen.

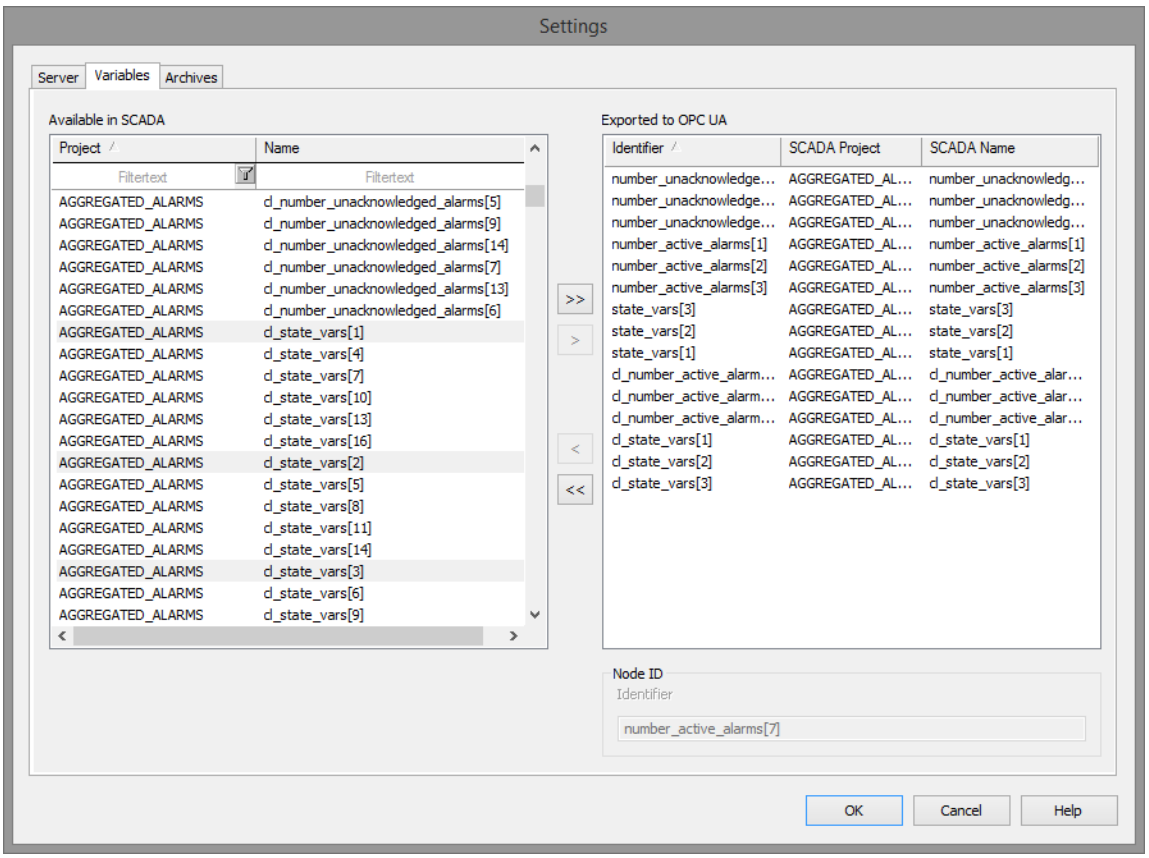
> **Achtung**
>
> Variablen, für die im zenon Editor die Eigenschaft **Sollwert setzen** nicht aktiviert wurde, können von einem **OPC UA Client** nicht beschrieben werden.

VARIABLEN DEFINIEREN

Werden in der Konfiguration für den **OPC UA Server** keine Variablen definiert, werden alle Variablen aus den verfügbaren zenon Projekten zum `OPC UA Address Space` hinzugefügt. Damit sind alle Variablen (auch Systemtreibervariablen) für die Clients sichtbar.

Bei der manuellen Konfiguration (mindestens eine konfigurierte Variable), werden die vom Benutzer definierten Werte verwendet. Standardmäßig wird der **Namespace Index 2** verwendet und ein **String Identifier** mit dem Variablennamen eingetragen. Der **Namespace Index** ist nicht konfigurierbar.

In der Registerkarte **Variables** konfigurieren Sie die Variablen welche als Node im OPC UA Datenmodell zur Verfügung gestellt werden sollen.



Parameter	Beschreibung
Available in SCADA	Liste der in zenon Runtime projektierten und verfügbaren Variablen. ▶ `Project` zenon Projectname ▶ `Name` zenon Variablenname Entspricht der Variablen-Eigenschaft **Name** im zenon Editor. Die Liste ist sortier- und filterbar: ▶ Sortierung durch Klick auf Spaltenüberschrift. Ein nochmaliger Klick ändert die Sortierrichtung. ▶ Filterung durch Eingabe des Filtertexts in Eingabefeld. Als Platzhalter sind die Zeichen * oder ? möglich.
Pfeil-Tasten	Zuordnung der zenon Variablen zu OPC UA: ▸ Klick auf die Schaltfläche **>** fügt ausgewählte Variable aus der Liste **Available in SCADA** zur Liste **Exported to OPC UA** hinzu. ▸ Klick auf die Taste **<** entfernt ausgewählte Variable aus der Liste **Exported to OPC UA** und fügt sie wieder in die Liste **Available in SCADA** ein. ▸ Klick auf die Schaltfläche >> fügt alle Variablen der Liste **Available in SCADA** zur Liste **Exported to OPC UA**. ▸ Klick auf die Schaltfläche << entfernt alle Variablen aus der Liste **Exported to OPC UA**. Mehrfachauswahl ist mit Mausklick + CTRL-Taste möglich.
Exported to OPC UA	Liste der exportierten Variablen. Die Variablen können sortiert werden nach: ▸ Identifier ▸ SCADA Project ▸ SCADA Name
Node Id	Das Eingabefeld für **Node ID** ist nur dann verfügbar, wenn ein Eintrag in der Liste **Exported to OPC UA** ausgewählt ist.

NODE ID

Parameter	Beschreibung
Identifier	Definiert den `Node Identifier` für die ausgewählte OPC UA Variable. Der **Node Identifier** Typ wird anhand der Eingabe automatisch definiert: ▸ Bei einer numerischen Eingabe wird automatisch ein numerischer `Identifier` erzeugt. ▸ Bei Eingabe einer GUID, wird automatisch ein `GUID Identifier` erzeugt, andernfalls wird der `Identifier` als String behandelt. GUID-Form: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx) Das Feld darf nicht leer sein. Der **Node Identifier** muss im gesamten Addressraum eindeutig sein.

VARIABLENEXPORT NACH OPC UA

Um Variablen nach OPC UA zu exportieren:

1. Wählen Sie in **Available in SCADA** die gewünschten Variablen per Mausklick aus. Mehrfachauwahl ist möglich.
2. klicken Sie auf den Pfeil Richtung **Exported to OPC UA**

   Um alle Variablen zu exportieren, klicken Sie auf den Doppelpfeil

Als **Browsename** und **Displayname** wird der zenon Variablenname verwendet.

Info

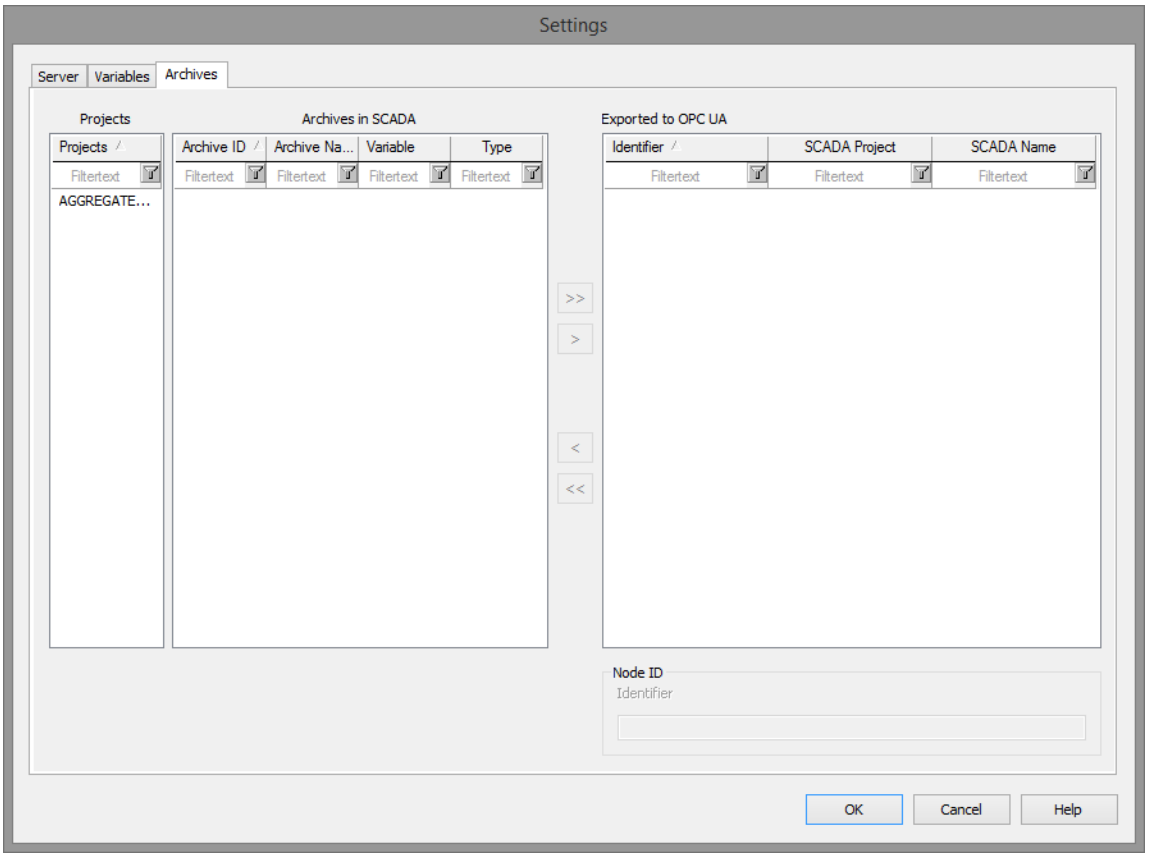
Bei zenon Variablen mit einem Slash (/) im Namen, wird der /durch einen Punkt (.) ersetzt. Die durch Slash getrennten Teile des Namens werden in eigenen Unterordnern gespeichert.

Beispiel: Die zenon Variable "`Static/Scalar/Int32`" wird aufgeteilt in:

- eine OPC UA Variable "`Int32`"
- in einem Ordner "`Scalar`",
- der ein Unterordner des Ordners "`Static`" ist

Das funktioniert mit allen Datentypen, mehrdimensionalen Arrays, Umlauten im Namen und Slashes am Ende.

13.2.3 Archives



PROJECTS

Liste des im zenon Editor aktivierten Projektes.

Parameter	Beschreibung
[Projektname]	Name des im zenon Editor geladenen Projekts. In **Archives in SCADA** werden die Inhalte des ausgewählten Projekts angezeigt.

ARCHIVES IN SCADA

Liste der Archiv-Variablen des ausgewählten Projekts.

Parameter	Beschreibung
`[Variablenliste]`	Liste der in zenon Runtime projektierten Archiv-Variablen. ▶ `Archive ID` Identifikation des vorhandenen zenon Archivs. ▶ `Archive Name` Projektierter Name des zenon Archivs. ▶ `Variable` Projektierter Name der Archiv-Variable ▶ `Type` Archivtyp Die Liste ist sortier- und filterbar: ▶ Sortierung durch Klick auf Spaltenüberschrift. Ein nochmaliger Klick ändert die Sortierrichtung. ▶ Filterung durch Eingabe des Filtertexts in Eingabefeld. Als Platzhalter sind die Zeichen * oder ? möglich.
Pfeil-Tasten	Zuordnung der Archiv-Variablen zu OPC UA: ▸ Klick auf die Schaltfläche **>** fügt ausgewählte Variable aus der Liste **Archives in SCADA** zur Liste **Exported to OPC UA** hinzu. ▸ Klick auf die Taste **<** entfernt ausgewählte Variable aus der Liste **Exported to OPC UA** und fügt sie wieder in die Liste `Archives in SCADA` ein. ▸ Klick auf die Schaltfläche >> fügt alle Variablen der Liste **Archives in SCADA** zur Liste **Exported to OPC UA**. ▸ Klick auf die Schaltfläche << entfernt alle Variablen aus der Liste **Exported to OPC UA**. Mehrfachauswahl ist mit Maustklick + CTRL-Taste möglich.
Exported to OPC UA	Liste der exportierten Variablen. Die Variablen können sortiert werden nach: ▸ `Identifier` - setzt sich zusammen aus: `Projektname + Archiv-ID + Archivvariable-Name + Archivtyp` ▸ `SCADA Project` Name des zenon Projekts

	▸ SCADA Name - setzt sich zusammen aus: `Archiv-ID + Name-Archivvariable + Archivtyp`
Node ID	Das Eingabefeld für **Node ID** ist nur dann verfügbar, wenn ein Eintrag in der Liste **Exported to OPC UA** ausgewählt ist.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

13.3 Start

Beim Start lädt das **Process Gateway** das konfigurierte Modul.

Beim ersten Start oder wenn keine Variablen explizit konfiguriert sind, erzeugt der **OPC UA Server** automatisch für sämtliche Variablen aus sämtlichen Projekten in der Runtime, das Datenmodell. Bei vielen Variablen oder vielen Variablen mit konfigurierten Alarmen, kann der Start entsprechend lange dauern. Es ist grundsätzlich zu empfehlen, nur solche Variablen über den **OPC UA Server** zur Verfügung zu stellen, welche tatsächlich vom **OPC UA Client** benötigt werden. Es können auch mehrere Instanzen vom **OPC UA Process Gateway** mit unterschiedlichem TCP Port und unterschiedlicher Konfiguration gestartet werden, was die Variablen betrifft.

Der Treiber versucht, die aktive Konfiguration aus der angegebenen INI-Datei auszulesen. Enthält die Konfiguration Fehler oder tritt beim Start ein Fehler auf, wird der Konfigurationsdialog (auf Seite 283) geöffnet und eine Fehlermeldung oder ein Fehlertext gezeigt. Wird die Konfiguration nicht mit **OK** bestätigt, wird das **Process Gateway** geschlossen.

Achtung

*Wir die Anwendung als `unsichtbar` gestartet, wird auch kein Konfigurationsdialog angezeigt. Ist die Konfiguration ungültig oder nicht vorhanden, wird das **Process Gateway** automatisch geschlossen.*

13.4 Verbindung

Das **OPC UA Process Gateway** unterstützt nur das **OPC UA TCP binary Protocol** und beinhaltet keinen **Discovery Server**. Der Client muss also direkt zum Server verbinden. Die Verbindungsadresse entspricht dem Schema: **opc.tcp://hostname.of.server:port.**

Die Verbindung kann gesichert (auf Seite 298) werden.

13.5 Sicherheit

Das Modul verwendet die **OPC UA** entsprechenden Sicherheitsprozeduren (signieren und verschlüsseln von Nachrichten) wie im Standard definiert. Zertifikate können, je nach Einstellung (auf Seite 283), überprüft oder pauschal akzeptiert werden.

SICHERE VERBINDUNGEN

Wird die Option **Accept secure connections only** (auf Seite 284) aktiviert, akzeptiert der Server ausschließlich Verbindungen, die den **Security Mode** `Sign` oder `Sign & Encrypt` verwenden. Zusätzlich muss vom Client als **Security Policy** `Basic128Rsa15` oder `Basic256` benutzt werden. Bei deaktivierter Option wird jede beliebige Kombination von **Security Mode** und **Security Policy** akzeptiert.

BENUTZERVERWALTUNG

Ist im zenon Projekt die Benutzerverwaltung aktiviert und mindestens ein Benutzer angelegt, dann steht die Option **Use SCADA user authentication** (auf Seite 284) zur Verfügung. Wird diese Option aktiviert, haben nur noch Clients die eine gültige Kombination aus `Benutzer` und `Passwort` senden, Zugriff auf die Daten des **OPC UA Servers**. Ist die Option inaktiv, findest keine Überprüfung statt. In diesem Fall werden sowohl anonyme Verbindungen wie auch alle Benutzer/Passwort-Kombinationen akzeptiert.

13.6 Alarms and Conditions

Das **OPC UA Process Gateway** unterstützt auch **OPC UA** Alarms and Conditions. Beim Start prüft der **OPC UA Server** automatisch die konfigurierte Variablen auf vorhandene Grenzwerte oder Reaktionsmatrix, mit Zustände welcher als Alarm definiert sind.

Ein **OPC UA Client** kann eine `Subscription` für `Event Notifications` erstellen, und das `Server Object` hinzufügen. Somit kann der **OPC UA Client** über Grenzwertverletzungen einer Variablen mit

aktiver Alarm, informiert werden. Der Client ist auch in der Lage, den Alarm - gleich wie in der zenon Runtime - zu bestätigen.

Achtung: Der Kommentar für Alarme bei dem **OPC UA** Methodenaufruf zur Quittierung wird nicht weitergereicht an die zenon Runtime. Wenn ein Alarm in zenon mit "Kommentar Erforderlich" konfiguriert ist, kann ein **OPC UA Client** diesen Alarm mit Kommentar nicht bestätigen.

Achtung: Ist ein Alarm so konfiguriert, dass eine Alarmursache erforderlich ist, kann der Alarm auf **OPC UA** Seite nicht quittiert werden.

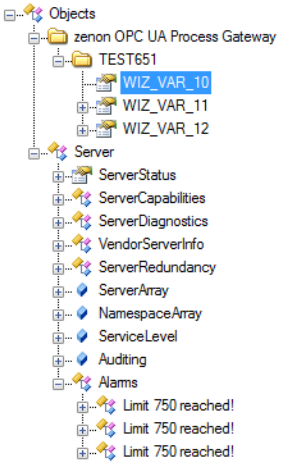
Die standardkonforme OPC Gewichtung wird auf den Index der mit dem Grenzwert verbundenen Alarm/Ereignis-Klassen abgebildet. Der Index einer Alarm/Ereignis-Klasse wird vom zenon Editor mit jeder neuen Alarm/Ereignis-Klasse automatisch hochgezählt und kann in den Eigenschaften überprüft werden. Es werden bis zu 20 Klassen mit folgender Zuordnung unterstützt:

OPC UA Bereich	zenon Alarm/Ereignis-Klasse	OPC Gewichtung
HIGH	20	1000
	19	950
	18	900
	17	850
MEDIUM HIGH	16	800
	15	750
	14	700
	13	650
MEDIUM	12	600
	11	550
	10	500
	9	450
MEDIUM LOW	8	400
	7	350
	6	300
	5	250
LOW	4	200
	3	150
	2	100
	1	50

Ist eine Variable mit keiner Alarm/Ereignis-Klasse verknüpft ist, wird die niedrigste mögliche Einstufung `1 (LOW)` verwendet.

ABBILDUNG ZENON ALARME ALS OBJEKT

zenon Variablen und darauf bezogene Alarme werden im **OPC UA Server** als Objekte abgebildet.



Die Abbildung zeigt die Einbindung des zenon Projekts (`TEST651`) und der zenon Variablen (`WIZ_VAR_10, WIZ_VAR_11, WIZ_VAR_12`) als Objekte im oberen Bereich. Darunter stehen jene Alarme, die durch Erreichen der Grenzwerte erzeugt wurden (`Limit 750 reached!` usw.).

NODE IDENTIFIER ALARM CONDITION

Der **Node Identifier** Typ von der `Alarm Condition` ist ein **String Type Node ID**.

Standard wird folgenden Aufbau verwendet:

```
<project GUID>-<Variablen ID hexadezimal>@<Variablenname>:_<Grenzwert
Bezeichnung oder "Rema">
```

Über den INI Eintrag: USE_LIMIT_TEXT_FOR_ALARM_NODES=TRUE im Abschitt OPCUA in der ini Datei für das **OPC UA Process Gateway**, kann einen alternativen Aufbau aktiviert werden, wo der Grenzwerttext verwendet wird. Dieser alternativen Aufbau steht nur für Grenzwerte zur Verfügung und nicht für Alarme über einen Reaktionsmatrix. Ein dynamischen Grenzwerttext wird hierbei nicht unterstützt. Es muss sichergestellt sein, dass der gleiche Grenzwerttext nicht für mehrere Grenzwerte für die gleiche Variable verwendet wird, da sonst keine Eindeutigkeit existiert:

```
<project GUID>-<Variablen ID hexadezimal>@<statischer Grenzwerttext>
```

PROTOKOLLIERUNG (LOGGING)

Der Treiber unterstützt eine umfangreiche LOG-Funktion. Diese LOG-Einträge können mit dem Tool Diagnosis Viewer ausgewertet werden.

13.7 Service Sets

Service Set	Wird unterstützt
Discovery Service Set	
FindServers	ja
GetEndpoints	ja
RegisterServer	nein
SecureChannel Service Set	
OpenSecureChannel	Ja
CloseSecureChannel	Ja
Session Service Set	
CreateSession	ja
ActivateSession	ja
CloseSession	ja
Cancel	ja
NodeManagement Service Set	
AddNodes	nein
AddReferences	nein
DeleteNodes	nein
DeleteReferences	nein
View Service Set	
Browse	ja
BrowseNext	ja
TranslateBrowsePathsToNodeIds	ja
RegisterNodes	ja
UnregisterNodes	ja
Query Service Set	
QueryFirst	nein
QueryNext	nein
Attribute Service Set	
Read	ja

HistoryRead	ja
Write	ja
HistoryUpdate	nein
Method Service Set	
Call	ja
MonitoredItem Service Set	
CreateScreeenedItems	ja
ModifyScreenedItems	ja
SetMonitoringMode	ja
SetTriggering	ja
DeleteScreenedItems	ja
Subscription Service Set	
CreateSubscription	ja
ModifySubscription	ja
SetPublishingMode	ja
Publish	ja
Republish	ja
TransferSubscriptions	ja
DeleteSubscriptions	ja

13.8 Kommunikation

SUBSCRIPTIONS

Der **OPC UA Server** unterstützt die Kommunikation via `Subscriptions`. Pro Session können von einem Client mehrere `Subscriptions` mit unterschiedlichen Einstellungen erstellt werden. Die maximale Anzahl an Subscriptions pro Session ist per Default limitiert auf 10. Über den ini Eintrag "MAX_COUNT_OF_SUBSCRIPTIONS_PER_SESSION" in den Abschnitt "OPCUA" kann die maximale Anzahl zwischen "1" und "100" konfiguriert werden.

PUBLISHREQUESTS

Pro Session sind vom **OPC UA Server** maximal 10 `PublishRequests` für die Warteschlange erlaubt. Dies ist unabhängig von der Anzahl der `Subscriptions`.

- Reihenfolge für `publish requests` (First in, first out):
  Beim Versenden wird die Eingangsreihenfolge des `PublishRequest` berücksichtigt. Wird eine `Subscription` ausgeführt und muss eine `PublishResponse` gesendet werden für eine DataChangeNotification oder ein KeepAlive, wird immer der älteste `PublishRequest` aus der Warteschlange entnommen.
- `BadNoSubscription` für alle Publish-Anfragen, wenn die letzte `Subscription` gelöscht, oder die Session geschlossen wurde:
  Wurde die letzte `Subscription` vom Client am Server abgemeldet, werden für alle `PublishRequest`, die noch am Server angemeldet sind, `PublishResponse` mit dem Wert `OpcUa_BadNoSubscription` erzeugt und an den Client gesendet. Die offenen `PublishResponse` werden anschließend am Server gelöscht.
- timeoutHint für `PublishRequests`:
  Bevor die `PublishResponses` für die `Subscriptions` gesendet werden wird sichergestellt, dass nur gültige - d.h. nicht abgelaufene - `PublishRequests` vorhanden sind. Die Prüfung der Gültigkeit erfolgt über den Zeitstempel des `PublishRequests` sowie des im `Request` angegebenen timeoutHint. Die resultierende Endzeit muss kleiner als die aktuelle Zeit des Servers sein. Ist dies nicht der Fall, wird ein `Response` mit `OpcUa_BadTimeout` an den Client zurückgesendet. Der `Request` wird in diesem Falle gelöscht.
- SubscriptionAcknowledgement welche im PublishRequest enthalten sind, bearbeitet der **OPC UA Server** wenn der PublishRequest aus der Warteschlange entnommen wird

REPUBLISHREQUESTS

Der **OPC UA Server** unterstützt die `RepublishRequests` auch für den Fall, wenn kurzzeitig die TCP Verbindung zwischen **OPC UA Client** und **OPC UA Server** unterbrochen ist.

UHRZEIT SYNCHRON

Der OPC UA Server berücksichtigt für `PublishRequests` das Feld "`timeoutHint`".

Wenn der OPC UA Server ein `PublishResponse` senden muss, werden zuerst alle vorhandenen `PublishRequests` in der Warteschlange (max. 10) überprüft. Diese Prüfung validiert, ob der Zeitstempel neuer als die aktuelle Uhrzeit ist. Ist das der Fall, wird der älteste `PublishRequest` aus der Warteschlange verwendet.

Enthält die Warteschlange `PublishRequests` bei denen der Zeitstempel im PublishRequest + timeoutHint im PublishReuest gleich oder älter ist wie die aktuelle Uhrzeit im OPC UA Server, werden die `PublishRequests` aus der Warteschlange entfernt und für jedes `PublishRequest` wird eine `ServiceFault` mit "`Bad_Timeout`" gesendet.

Es ist daher wesentlich, dass die System-Uhrzeit des OPC UA Servers und die System-Uhrzeit des OPC UA Clients, entsprechend synchronisiert sind. Für diese Zeitsynchronisation wird empfohlen, sowohl für den OPC UA Server wie auch für den OPC UA Client eine gemeinsame Synchronisierungsquelle zu verwenden. Eine gemeinsame Quelle kann z.B. derselbe NTP-Server für Server und Client sein. Anderenfalls kann es dazu kommen, dass der OPC UA Server sämtliche `PublishRequests` vom OPC UA Client sofort verwirft und keine Wertänderungen übermittelt werden können.

Info

DER ZEITSTEMPEL SETZT SICH ZUSAMMEN AUS: `PUBLISHREQUEST` + `TIMEOUTHINT` AUS DEM `PUBLISHREQUEST`.

MONITORED ITEMS

Der OPC UA Server unterstützt das Hinzufügen, Entfernen und Modifizieren von `MonitoredItems`.

Wird von einem OPC UA CLient eine Warteschlangegröße größer 1 angefordert wird diese Anfrage unterstützt. Dadurch werden mehrere Wertänderungen für dieselbe Variable zwischen zwei Publish-Intervallen vom OPC UA Server gepuffert. Via einem `PublishResponse` werden die Wertänderungen an den OPC UA Client gesendet.

DATACHANGEFILTER - DEADBAND

Der OPC UA Server unterstützt den `absolute deadband`.

READ REQUESTS

Der OPC UA Server unterstützt neben der Kommunikation via `Subscriptions` auch `Read Requests`. Grundsätzlich ist die Kommunikation via `Subscriptions` zu bevorzugen über die Kommunikation mittels Read Requests. Wenn ein OPC UA Client keine Subscriptions unterstützt und zyklisch Read Requests sendet um jeweils den aktuellen Wert zu bekommen, wird stark empfohlen, dass der Client für solche Nodes die zyklisch gelesen werden, einmalig den Aufruf RegisterNodes verwendet.

Dies erlaubt den OPC UA Server eine Optimierung der Kommunikation mit der Runtime für zyklische `Read Requests`.

WRITE REQUESTS

Der OPC UA Server unterstützt write Requests. Die Sollwertgrenzen der Variable werden berücksichtigt.

Achtung: Ist eine Variable in der Energy Edition mit einer Befehlsgruppe verknüpft und diese Variable wird von einem OPC UA Client geschrieben, wird dabei die Befehlsverriegelung nicht berücksichtigt, sondern die Befehlsvariable direkt geschrieben.

13.9 Konfigurationsdatei: INI-Einträge für OPC UA

Die Konfigurationsdatei `zenProcGateway.ini` enthält folgende Einträge für OPC UA:

[OPCUA]

Eintrag	Beschreibung
SERVER_NAME=	Name des **OPC UA Servers** zu dem das Gateway eine Verbindung aufbaut.
SERVER_PORT=	Port für die Kommunikation zum **OPC UA Server**. Default: `4841`
CERTIFICATES_PATH=	Pfad für den Stammordner mit den Zertifikatsdateien und den jeweiligen Unterordnern. Default: `%ProgramData%\PKI\CA\` Hinweis: Eintrag muss gelöscht werden, wenn beim Start des Process Gateway neue Zertifikat erstellt werden sollen.
SERVER_CERTIFICATE_LOCATION=	URI der Datei mit dem öffentlichem Schlüssel des Servers. Der Pfad wird immer absolut angegeben. Dieser öffentliche Schlüssel wird vom Client zum Verschlüsseln der Nachrichten verwendet. Default: `%ProgramData%\PKI\CA\certs\zenopcua.der` Hinweis: Eintrag muss gelöscht werden, wenn beim Start des Process Gateway neue Zertifikat erstellt werden sollen.
SERVER_PRIVATE_KEY_LOCATION=	URI der Datei mit dem privatem Schlüssel des Servers. Der Pfad wird immer absolut angegeben. Dieser private Schlüssel wird vom Server zum Entschlüsseln der Nachrichten des Clients verwendet. Default: `%ProgramData%\PKI\CA\private\zenopcua.pem` Hinweis: Eintrag muss gelöscht werden, wenn beim Start des Process Gateway neue Zertifikat erstellt werden sollen.
ONLY_SECURE_CONNECTIONS=	Auswahl, welche Art der Verbindung (sicher oder unsicher) für einen Verbindungsaufbau zugelassen ist. ▸ `TRUE` Eine Verbindung zum Server ist nur mit einer sicheren Verbindung möglich. ▸ `FALSE` Eine Verbindung zum Server ist auch mit einer unsicheren Verbindung möglich.

	Default: TRUE
TRUST_ALL_CLIENT_CERTIFICATES=	Auswahl, welche Art von Zertifikaten (Client- oder Serverzertifikate) für einen Verbindungsaufbau zugelassen ist. ▸ TRUE Alle Client-Zertifikate werden akzeptiert, die Verbindung wird aufgebaut. ▸ FALSE Alle Clients, die sich mit dem Server verbinden wollen, müssen ein gültiges Zertifikat übermitteln. Dieses Zertifikat wird zusätzlich darauf überprüft, ob es sich in der Liste der bekannten Client-Zertifikate (**Certificate trust list path**) befindet. Default: TRUE
USE_USER_AUTHENTICATION=	Auswahl, ob die Benutzerverwaltung des aktuellen zenon Projekts bei der Anmeldung am Server für die Identifizierung verwendet werden soll. ▸ TRUE Clients können sich am Server nur mit gültiger Benutzerkennung und Passwort anmelden. Hinweis: Dazu muss im zenon Editor die Benutzerverwaltung aktiviert und projektiert sein. ▸ FALSE Anonymes Login ist möglich. Default: FALSE
MAX_COUNT_OF_SUBSCRIPTIONS_PER_SESSION=	Anzahl der Subscriptions pro Session. Einstellbar zwischen minimal 1 und maximal 100 Subscriptions. Default: 10

[VARIABLES]

Eintrag	Beschreibung
PROJECT_x=	Name des Projekts aus dem die Variablen stammen. **x** steht für die Nummerierung innerhalb eines Projekts, beginnend bei 0. Beispiel: **PROJECT_0**=TEST
PARAM_x=	Parameter für Projekt **x**. Nummerierung erfolgt von 0 aufwärts. Beispiel: **PARAM_0**=80:INI.Driverforinternalvariables_ModemHwAdrSet

COUNT=	Anzahl der konfigurierten Variablen.
NAME_x=	Name der Variablen. **x** steht für die Nummerierung des Projekts, beginnend bei `0`. Beispiel: **NAME_0=**`Driverforinternalvariables_ModemHwAdrSet`

2 = Konstante für Namespace Index

Wird fix durch das System vorgegeben/vergeben und darf nicht geändert werden.

13.10 Fehlermeldungen

Fehlermeldungen in der LOG-Datei:

Fehlermeldung	Debug Level	Beschreibung
`Failed to validate client certificate`	**Error**	Das Client-Zertifikat konnte nicht validiert werden. Es ist entweder ungültig (eventuell Gültigkeitsdatum überschritten) oder es wurde in der Liste der bekannten Client-Zertifikate nicht gefunden. Eine zusätzliche Information inklusive eine Fehlernummer beschreibt den Fehler genauer.
`Wrong user/password supplied while trying to activate session (User: %s)`	**Error**	Benutzerauthentifizierung fehlgeschlagen. ▸ `%s`: Name des zu authentifizierenden Benutzers.
`Login of user '%s' succeeded.`	**Deep Debug**	Benutzerauthentifizierung erfolgreich. ▸ `%s`: Name des zu authentifizierenden Benutzers.
`Variable '%s' could not be added to OPC UA address space, because node '%s' with the same node identifier already exists`	**Error**	Eine vom Benutzer definierte Variable konnte nicht angelegt werden, da bereits eine andere Variable mit demselben `Identifier` existiert.
`Failed to initialize server architecture: An internal error occurred as a result of a programming or configuration error. (0x80020000)`	**Error**	Das Dateiformat für eine oder mehrere der folgenden Dateien liegt in keinem gültigen Format vor: ▸ Zertifikat ▸ `Private Key` ▸ `Certificate Revocation List`

%1 PublishResponse with OpcUa_BadNoScubscriptions sent because no subscriptions are created in session %2	**Error**	Für eine Publish-Anfrage liegt keine gültige Subscription vor oder die Session wurde gelöscht. Der OPC UA Server sendet eine Meldung an den OPC UA Client, für sämtliche ausstehende PublishRequests, wenn die letzte Subscription vom Server oder vom Client gelöscht wurde. ▸ %1 Anzahl der PublishResponse, die mit dem Resultatwert OpcUa_BadNoSubscription an einen Client gesendet wurden. ▸ %2 Name der Session, aus der die PublishResponse versendet werden.
%1 PublishResponse with OpcUa_BadTimeout sent in session %2	**Error**	Der Zeitstempel + timeoutHint im PublishRequest der vom Client zur Verfügung gestellt wurde, ist älter als die aktuelle Uhrzeit am OPC UA Server. Mögliche Ursachen: der timeoutHint im PublishRequest ist zu kurz, der OPC UA Client stellt mehr PublishRequests zur Verfügung als benötigt werden und die PublishRequests laufen ab, oder die Uhrzeit des Rechners mit dem OPC UA Server und die Uhrzeit des Rechners mit dem OPC UA Client, sind nicht synchron. ▸ %1 Anzahl der PublishResponse, die mit dem Resultatwert OpcUa_BadTimeout an einen Client gesendet wurden. ▸ %2 Name der Session, aus der die PublishResponse versendet werden. Hinweis: Der PublishRequest wird gelöscht.
%1 advise/unadvise thread procedure	**DEBUG**	Start oder Stop eines Advise- oder Unadvise-Threads: ▸ %1 Entering Thread gestartet ▸ %1 Leaving Thread gestoppt
%1 shutdown	**DEBUG**	Beginn und Ende einer Shutdown-Sequenz: ▸ %1 beginning Shutdown Beginn ▸ %1 finisched Shutdown beendet
%1 %2 request	**DEBUG**	Beginn oder Ende eines Request bei Nachladen der Runtime:

		▶ %1 Processing Beginn eines Request ▶ %1 Finished Ende eines Request ▶ %2 ACTIVE Request ist aktiv. ▶ % 2 INACTIVE Request ist inaktiv.
Server shut down.	**DEBUG**	Benachrichtung, dass der Server wurde vollständig heruntergefahren wurde.
The runtime does not provide archive variable [%1] of project [%2]	**ERROR**	Das Archiv einer konfigurierten Archivvariable ist in der zenon Runtime nicht mehr erreichbar. ▶ %1 Name der Archivvariable ▶ %2 Projektname
Reading configuration from %1	**MSG**	Die Konfiguration wird aus der INI-Datei gelesen. ▶ %1 Name der INI-Datei.
Linking configuration to the Runtime	**MSG**	Die im **Process Gateway** konfigurierten Variablen werden mit der zenon Runtime verknüpft.
some configured variables are not linked to the Runtime	**ERROR**	Bei der Verknüpfung wurde festgestellt, dass ein oder mehrere projektieren Variable nicht in der zenon Runtime verfügbar sind. Die nicht verfügbaren Projekte oder Variablen werden aufgelistet.
Project [%1] is not linked to the Runtime	**MSG**	Das mit den Variablen projektierte Projekt ist nicht in der zenon Runtime verfügbar. ▶ %1 Projektname Möglich Ursachen: Das Projekt ist nicht geladen oder wurde gelöscht.
Variable [%1] of Project [%2] is not linked to the Runtime	**MSG**	Die Variable ist in der zenon Runtime nicht verfügbar: ▶ 1% Variablenname ▶ %2 Projektname

`Could not write value [%1] to key [%2] in section [%3]`	**ERROR**	Fehler beim Schreiben eines Wertes. ▶ `%1` Wert, der geschrieben werden sollte. ▶ `%2` Schlüssel, in den der Wert geschrieben werden sollte. ▶ `%3` Sektion des Schlüssels, in die der Wert geschrieben werden sollte.
`No separator for archive flag [,] found in value [%s] of key [%s]`	**ERROR**	Fehlendes Separator-Zeichen bei einem Wert. Die betroffene Variable kann deshalb nicht in die Konfiguration übernommen werden: ▶ `%1` Name der Variable ▶ `%2` Schlüssel des Wertes

13.11 Status Mapping

In bestimmte Situationen kommuniziert der OPC UA Server einen abweichenden `StatusCode` beim Wert (`Value Attribute`). Dies Betrifft jenen Knoten (`Node`), welcher die Variable repräsentiert.

Unten stehende Tabelle listet den zenon Status mit entsprechendem **OPC UA StatusCode** auf.

status der Variable in der Runtime	StatusCode Value Attribute OPC UA Node
INVALID	`0x80050000 BadCommunicationError`
ALT_VAL	`0x40910000 UncertainSubstituteValue`
OFF	`0x808C0000 BadSensorFailure`
OV_870	`0x803C0000 BadOutOfRange`
N_UPDATE	`0x80310000 BadNoCommunication`

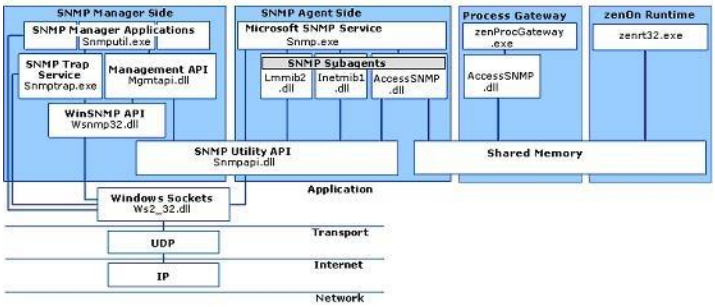
14. SNMP Agent

14.1 SNMP Architektur

Der Windows SNMP Agent unterstützt die SNMP Version 2 (SNMPv2c). Unterstützt werden die SNMP Funktionen (GET, GETNEXT, GETBULK, SET, TRAP).

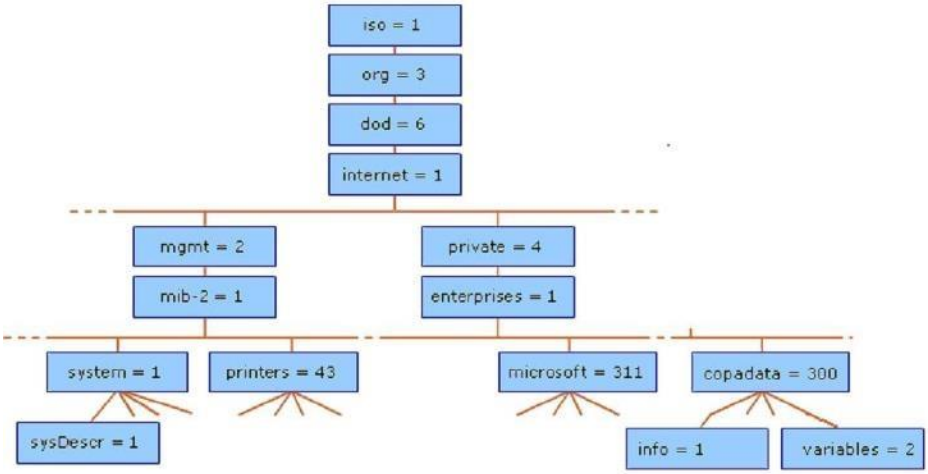
Nähere Details wie Microsoft SNMP implementiert hat finden sie auf der Microsoft Technet Website unter: “How SNMP Works” https://technet.microsoft.com/en-us/library/cc783142(v=ws.10).aspx (https://technet.microsoft.com/en-us/library/cc783142(v=ws.10).aspx)

WINDOWS BETRIEBSSYSTEMKOMPONENTEN MIT DER VERBINDUNG ZUM PROCESS GATEWAY UND DER RUNTIME:



14.2 Management Information Base (MIB)

Die Management Information Base (MIB) ist in Form einer Baumstruktur organisiert, in welcher der SNMP Agent die Variablen = (OID – Object Identifier) zur Verfügung stellt. Unterhalb ist der Aufbau der MIB Struktur mit den ergänzten Variablen ersichtlich, die über das Process Gateway bereitgestellt werden.



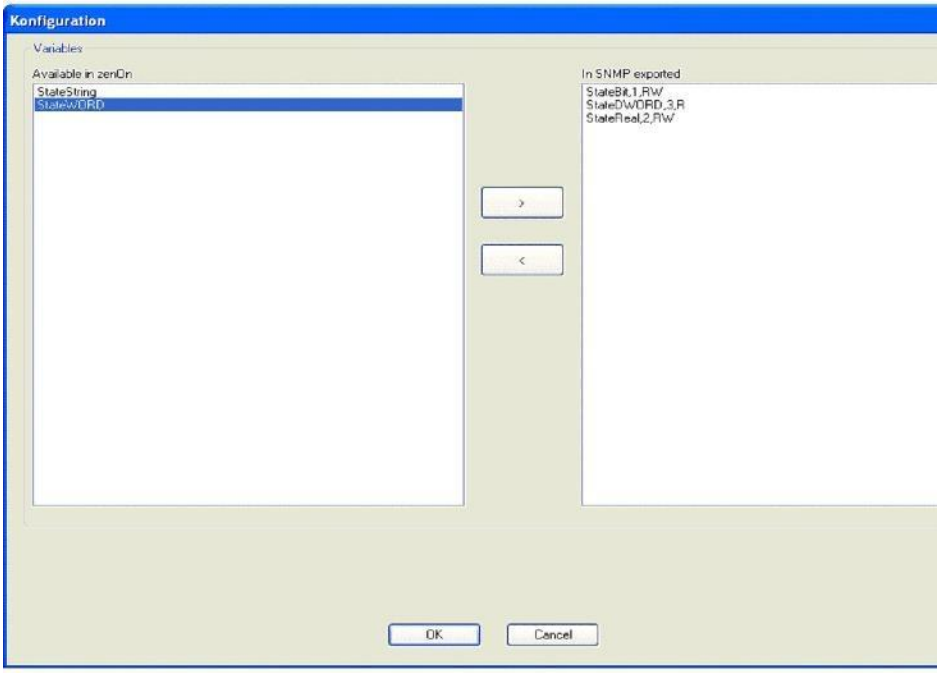
14.3 MIB Aufbau

Infoblockstruktur ist immer vorhanden				
OID	**Beschreibung**	**Datentyp**	**Zugriff**	**Änderung wird als Trap versendet**
1.3.6.1.4.1.300.1.1.0	copadata.info.statusinfo, Runtime ist (Online, Offline)	string	lesen	ja
1.3.6.1.4.1.300.1.2.0	copadata.info.statusval, Runtime ist 1=Online 0=Offline	integer	lesen	ja
1.3.6.1.4.1.300.1.3.0	copadata.info.watchdog	integer	lesen	nein
1.3.6.1.4.1.300.1.4.0	copadata.info.project, Projektname von wo das Process Gateway die Variablen liest	string	lesen	nein

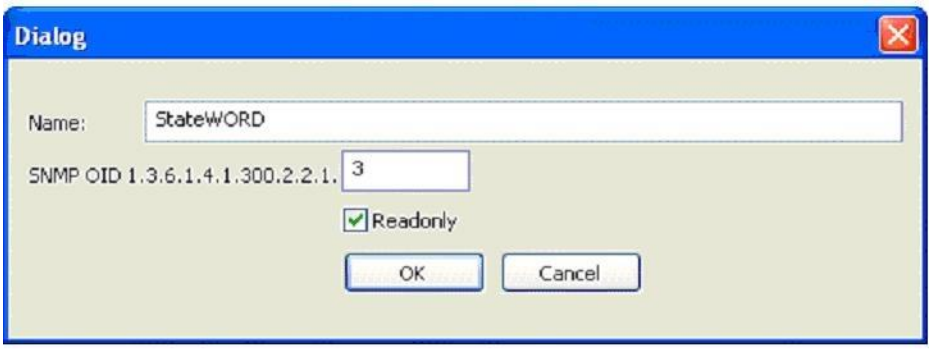
Variablenstruktur (Tabelle) wird je nach Konfiguration vom Process Gateway dynamisch angepasst				
OID	**Beschreibung**	**Datentyp**	**Zugriff**	**Änderung wird als Trap versendet**
1.3.6.1.4.1.300.2.1.0	copadata.variables.ifNumber, Anzahl der konfigurierten Variablen=OID	integer	lesen	ja
1.3.6.1.4.1.300.2.2.1.X	copadata.variables.ifTables.ifIndex, Konfigurierter OID-Index aus der Process Gateway Konfiguration	integer	lesen	nein
1.3.6.1.4.1.300.2.2.2.X	copadata.variables.ifTables.ifName, Variablenname	string	lesen	nein
1.3.6.1.4.1.300.2.2.3.X	copadata.variables.ifTables.ifValue, Variablenwert als Integer. Alle numerischen Datentypen werden auf einen SNMP Integerdatentyp formatiert. ACHTUNG bei real und double gehen die nachkommerstellen verloren.	integer	je nach Konfiguration im Process Gateway	nein
1.3.6.1.4.1.300.2.2.4.X	copadata.variables.ifTables.ifValueAsString, Variablenwert als string. Alle numerischen Datentypen werden auf einen SNMP Stringdatentyp formatiert.	string	je nach Konfiguration im Process Gateway	ja
1.3.6.1.4.1.300.2.2.5.X	copadata.variables.ifTables.ifStatus, Variablen Statuswert	integer	lesen	ja
1.3.6.1.4.1.300.2.2.6.X	copadata.variables.ifTables.ifTimestamp, Variablen Zeitstempel	integer	lesen	ja

14.4 Modulkonfiguration

Nach Klick auf die Schaltfläche **Settings** öffnet sich der folgende Dialog:



Hier können die Variablen ausgewählt werden, die verbunden werden sollen. Auf der linken Seite sind alle in zenon verfügbaren Variablen aufgelistet. Dort können sie ausgewählt und mit dem Schaltfläche **>** in die SNMP Exportliste verschoben werden. Mit dem Schaltfläche **<** können sie wieder zurück verschoben werden. Wenn eine Variable in die Exportliste verschoben wird, öffnet sich ein Dialog für die Namenszuweisung der zenon Variable zu einer OID Adresse in der SNMP Struktur:



Die so ausgewählten Variablen sind dann auf dem SNMP Server verfügbar und werden kontinuierlich zwischen zenon und der SNMP OID Struktur synchronisiert. Man kann pro OID Adresse definieren, ob man diese per SNMP SET schreiben darf. Der genau Aufbau pro OID Adresse ist im Kapitel “MIB Aufbau” beschrieben.

14.5 Konfigurationsdatei: spezifische Einträge für Access SNMP

Die Konfigurationsdatei muss im Systemordner liegen.

Parameter	Beschreibung
[SNMP]	
REFRESHRATE	Updatezeit in Millisekunden

Parameter	Beschreibung
[VARIABLES]	
COUNT	Anzahl der zu exportierenden Variablen
NAME_n	Name der Variable mit der Nummer n (0<=n<COUNT)
PARAM_n	Parameter SNMP OID Tabellenoffset und der Information R=nur lesen RW=lesen und schreiben mit der Nummer n (0<=n<COUNT)

EXPORT VON VARIABLEN

Für den erfolgreichen Export muss der Projektname in der `zenProcgateway.ini` im Pfad %CD_SYSTEM% korrekt eingetragen werden. Der Eintrag erfolgt in:

[DEFAULT]
PROJECT=

Achtung: Bei falschen oder fehlenden Einträge wird das Gateway ohne weitere Fehlermeldung geschlossen. Die Variablen stehen damit nicht zur Verfügung.

14.6 Konfiguration des Windows SNMP Agent

Damit der SNMP Agent von Windows (**SNMP.exe**) die zenon MIB Teilstruktur zur Verfügung stellt, muss die Process Gateway Datei **AccessSNMP.dll** in der Registry entsprechend konfiguriert werden:

1. Installation des Windows SNMP Service über die Systemsteuerung:
   - über Modul: Programm und Funktionen -> Windows-Funktionen aktivieren oder deaktivieren

   Hinweis: Unter Windows 8 muss nach dem Hinzufügen des SNMP Service der Rechner neu gestartet werden.
2. Installation des Process Gateway mit der Datei `AccessSNMP.dll` in den Ordner: `%Program Files (x86)%\COPA-DATA\zenon 8.00`

3. Mit dem Registry Editor folgende Schlüssel und Zeichenfolge hinzufügen. Die Pfade und Inhalte unterscheiden sich teilweise bei 32-Bit und 64-Bit Systemen:

   32-Bit:

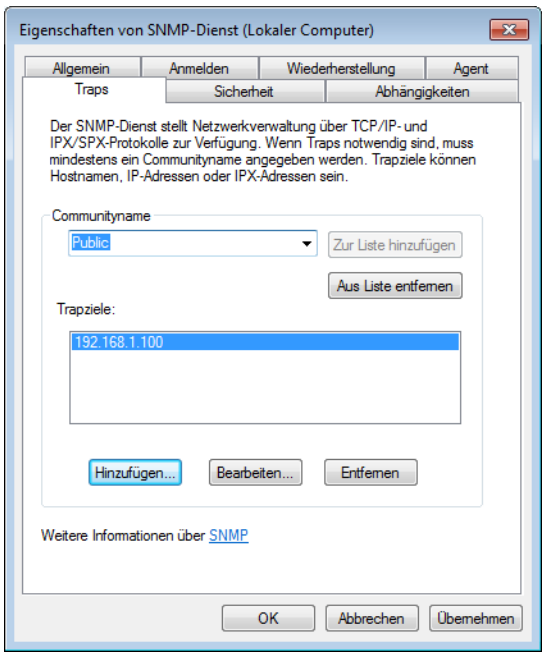
   - **[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents]**
     Name=`ProcessGateway`
     Wert=`SOFTWARE\COPA-DATA\SNMP\CurrentVersion`
   - **[HKEY_LOCAL_MACHINE\SOFTWARE\COPA-DATA\SNMP\CurrentVersion]**
     Name=`Pathname`
     Wert=`C:\Program Files\COPA-DATA\zenon 8.00 SP0\AccessSNMP.dll`

   64-Bit:

   - **[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents]**
     Name=`ProcessGateway`
     Wert=`SOFTWARE\COPA-DATA\SNMP\CurrentVersion`
   - **[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\COPA-DATA\SNMP\CurrentVersion]**
     Name=`Pathname`
     Wert=`C:\Program Files (x86)\COPA-DATA\zenon 8.00 SP0\AccessSNMP.dll`

4. Start des SNMP-Dienst.
   Dieser lädt über die Registrykonfiguration die Process Gateway Datei `AccessSNMP.dll`.
   Weitere Konfigurationseinstellungen für den Windows SNMP Dienst: Systemsteuerung -> Verwaltung -> Dienste -> SNMP-Dienst -> Eigenschaften

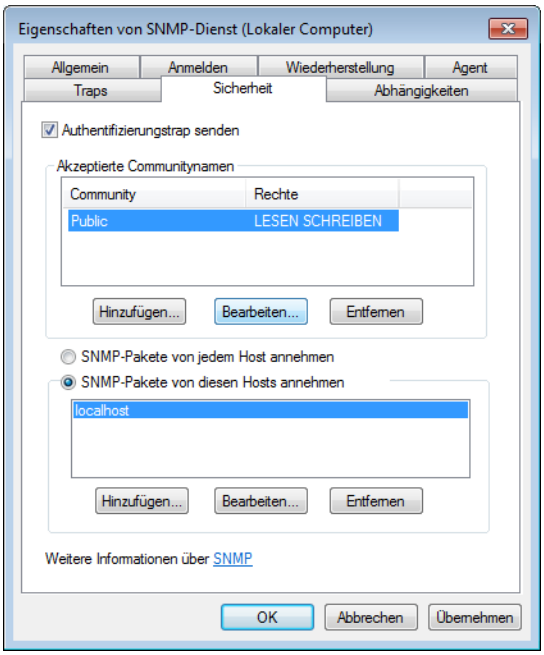
   a) Registerkarte **Traps:** Festlegung, an welche IP-Adressen im Netzwerk Traps versendet werden:



   - **Communityname**: muss auf Manager und Agent ident sein

- **Trapziele**: Adressen der Ziele als Hostnamen, IP-Adressen oder IPX-Adressen

a) Registerkarte **Sicherheit:** Festlegung der Rechte, mit denen ein SNMP Client auf den SNMP Agent zugreift:



- **Rechte**: Muss auf "`LESEN SCHREIBEN`" gesetzt sein
- **Community**: muss auf Manager und Agent identisch sein

15. SQL

15.1 Modulkonfiguration

Die Konfiguration der Datenbankanbindung besteht aus zwei Teilen.

1. OLE-DB Ankopplung: Diese OLE-DB Ankopplung kann im Konfigurationsdialog über die Auswahl-Schaltfläche für den Connect-String definiert werden. Dazu wird der Standarddialog für Datenverknüpfungseigenschaften aufgerufen.
2. Zusätzlich kann der Name der Tabelle definiert werden, falls der Standardname ONLINE_VALUES nicht verwendet werden soll.
   Dies kann über den Eintrag **TABLE** in der Sektion **[DATABASE]** in der Konfigurationsdatei

(`zenProcGateway.ini`) eingestellt werden. Die Einstellungen werden beim Start des Add-Ons übernommen.

> **Achtung**
>
> *Werden Tabellennamen händisch in der Konfigurationsdatei (zenProcGateway.ini) definiert, müssen diese Tabellennamen auch per Hand in der Datenbank geändert oder erstellt werden.*

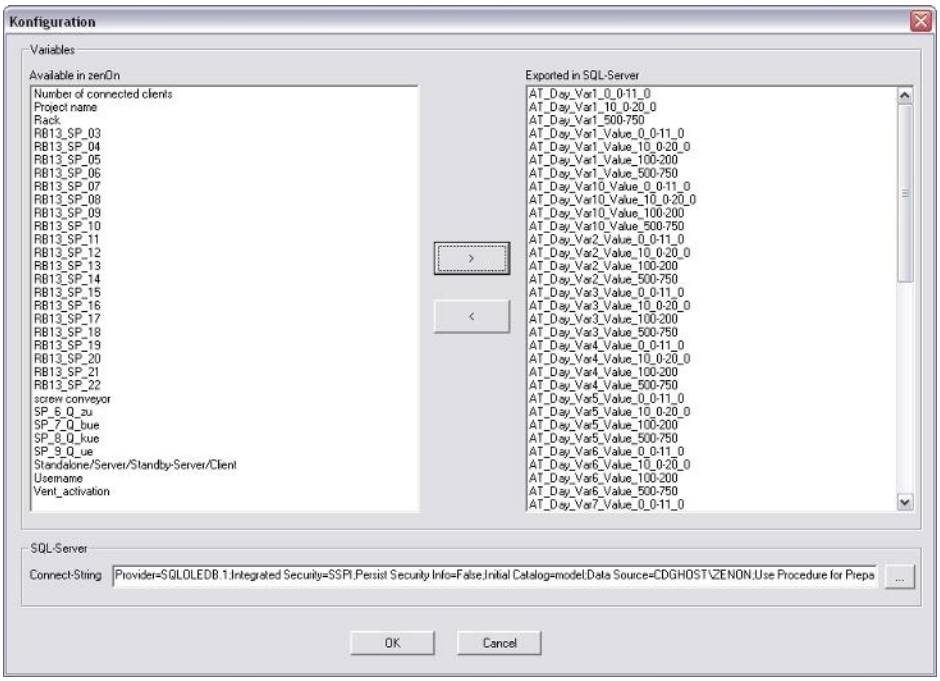
ORACLE DATENBANKEN

Die Verwendung des Process Gateways und der `AccessSQL.dll` in Verbindung mit Oracle Datenbanken kann Fehler verursachen: Mit manchen ODBC-Treibern kommt es vor, dass keine Online-Werte in die Datenbank geschrieben werden. Die erweiterte ODBC Protokollierung gibt dann die Fehlermeldung "**1843**" aus.

Das Problem kann durch den Austausch des ODBC-Treibers behoben werden. Beispiel: ODBC-Treiber 9.02.00.05 für Oracle 9.2.

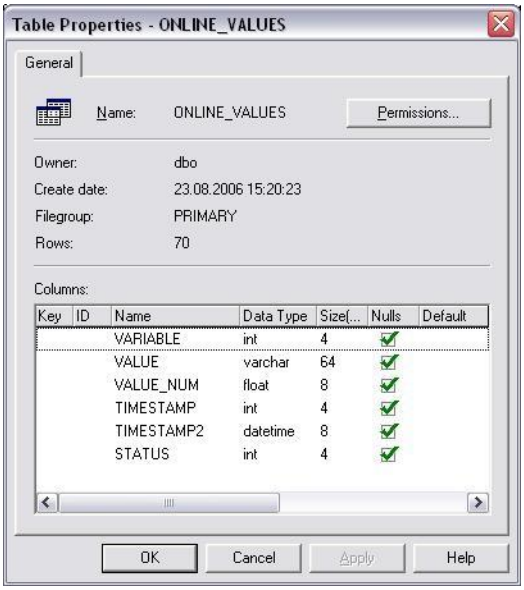
15.2 Variablenauswahl

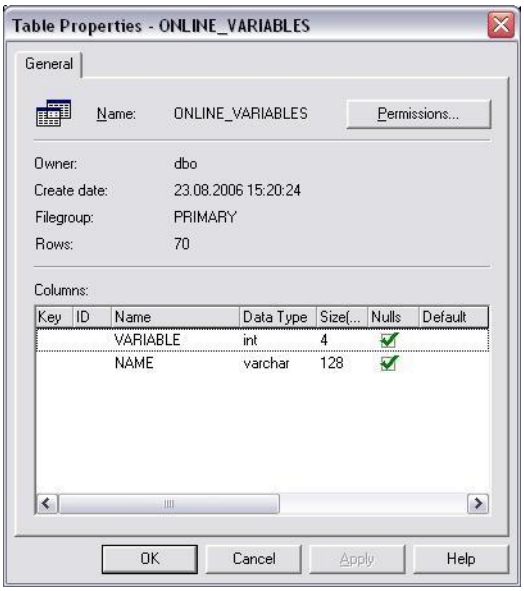
Im Konfigurationsdialog können die in das SQL-Prozessabbild zu schreibenden zenon Variablen definiert werden. Dazu müssen die im zenon Projekt verfügbaren Variablen - aufgelistet in der linken Liste - in das zu exportierende Abbild - aufgelistet in der rechten Liste - übernommen werden.



15.3 Tabellenerzeugung

Die Tabellenstruktur der SQL-Datenbank wird nach dem Verlassen des Konfigurationsdialogs mit **OK** automatisch erzeugt. Für den Fall, dass sie händisch angelegt werden, gilt:





15.4 Betrieb im Redundanzsystem

Im redundanten Betrieb stehen zwei verschiedene Varianten zur Auswahl. In der ersten Variante läuft die SQL-Datenbank auf beiden Servern lokal und das Process Gateway läuft auf ebenfalls auf beiden

Servern. Da es sich ausschließlich um Online-Werte handelt, ist kein zusätzliche Abgleich zwischen den Datenbanken notwendig. Großer Nachteil dieser Variante ist die hohe Netzwerklast, die durch die zusätzliche Anforderung der Onlinewerte am Standby Server erzeugt wird.

In der zweiten Variante läuft das Process Gateway nur am Runtime-Server und schreibt seine Daten in eine lokale oder entfernte SQL-Datenbank. In dieser Variante wird zwischen Runtime-Server und Standby Server keine zusätzliche Netzwerklast erzeugt und es wird im Falle der entfernten SQL-Datenbank der Server nicht durch externe Abfragen in dieser Datenbank belastet.

15.5 Konfigurationsdatei: spezifische Einträge für AccessSQL

Die Konfigurationsdatei muss im Systemordner liegen.

Parameter	Beschreibung
[DATABASE]	
INITSTRING	OLE-DB Connect-String zur SQL-Datenbank
TABLE_ONL	Name der Tabelle, welche das Prozessabbild empfängt
TABLE_VAR	Name der Tabelle, welche die Variablen-IDs empfängt
REFRESHRATE	maximale Refreshrate in Millisekunden

Parameter	Beschreibung
[VARIABLES]	
COUNT	Anzahl der zu exportierenden Variablen
NAME_n	Name der n-ten Variable. Regeln: ▸ beginnt mit 0 ▸ muss ohne Unterbrechung fortlaufend sein

15.6 Tabellenformat

15.6.1 Report für Prozessabbild

VARIABLE	int[4]	ID der zenon Variable
VALUE	varchar[64]	aktueller Wert als String
VALUE_NUM	float	aktueller Wert als float
TIMESTAMP	int[4]	Zeitstempel des aktuellen Wertes (als UNIX Zeit)
TIMESTAMP2	datetime	Zeitstempel des aktuellen Wertes (als datetime)
STATUS	int[4]	Statuswort des aktuellen Wertes

15.6.2 Tabelle für Variablen-IDs

VARIABLE	int[4]	ID der zenon Variable
NAME	varchar[128]	Name der zenon-Variable