zenon
by COPA-DATA

zenon Treiber Handbuch

IEC870

v.8.20

© 2020 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 5

2 IEC870 5

3 IEC870 - Datenblatt 6

4 Konfiguration 7

4.1 Anlegen eines Treibers 8

4.2 Einstellungen im Treiberdialog 11

4.2.1 Allgemein 12

4.2.2 Com 16

4.2.3 Basic settings 17

4.2.4 Connections 18

5 Variablen anlegen 29

5.1 Variablen im Editor anlegen 29

5.2 Adressierung 33

5.3 Treiberobjekte und Datentypen 36

5.3.1 Treiberobjekte 36

5.3.2 Zuordnung der Datentypen 37

5.4 Variablen anlegen durch Import 40

5.4.1 XML Import 41

5.4.2 DBF Import/Export 42

5.4.3 Online Import 48

5.5 Kommunikationsdetails (Treibervariablen) 48

6 Treiberspezifische Funktionen 54

6.1 Mapping von Double Point Values 62

6.2 TLS-verschlüsselte Kommunikation 65

7 Funktion Treiberkommandos 69

8 Interoperability 74

9 Fehleranalyse 90

9.1 Analysetool 90

9.2 Treiberüberwachung 92

9.3 Checkliste 93

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 Customer Service, den 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 IEC870

Treiber für das IEC 60870-5-101 (seriell) und das IEC 60870-5-104 (TCP/IP) Protokoll.

Die Kommunikation zwischen zenon und der Steuerung basiert auf

- dem seriellen IEC 60870-5-101 Protokoll

  Hier agiert zenon als Master im unbalancierten Kommunikationsmodus. Der Kommunikationskanal kann zwischen einem 60870 Master (zenon) und mehreren 60870 Slaves (Steuerung) geteilt werden.

- oder dem TCP/IP Protokoll IEC 60870-5-104

  Hier agiert zenon auf der Protokollebene als Master und auf TCP-Ebene als Client.

GEMEINSAME VERWENDUNG VON SERIELLER UND TCP/IP VERBINDUNG

Verbindungen nach **870-104** (TCP/IP) und **870-101** (seriell) sollten nicht in einer Treiberinstanz gemeinsam angelegt werden. Die langsamere serielle Verbindung (101) würden auch die TCP/IP-Verbindung (104) bremsen. Eventuell könnte es sogar zu einem TCP/IP Timeout kommen.

Tipp: Wenn Sie die Verbindungen über beide Protokolle benötigen, legen Sie im zenon Projekt zwei Instanzen des Treibers an.

3 IEC870 - Datenblatt

Allgemein:	
Treiberdateiname	IEC870.exe
Treiberbezeichnung	IEC 60870-5-101_104
Steuerungs-Typen	IEC 60870-5-104 Slaves bzw. IEC 60870-5-101 Slaves (controlled stations).
Steuerungs-Hersteller	Alstom; Siemens; IEC; SAT; Sprecher Automation; Areva

Treiber unterstützt:	
Protokoll	IEC 60870-5-104; IEC 60870-5-101
Adressierung: Adress-basiert	Address based
Adressierung: Namens-basiert	--
Kommunikation spontan	X
Kommunikation pollend	X
Online Browsing	X
Offline Browsing	--
Echtzeitfähig	X

Treiber unterstützt:	
Blockwrite	--
Modemfähig	--
RDA numerisch	--
RDA String	--
Hysterese	X
erweiterte API	X
Unterstützung von Statusbit **WR-SUC**	X
alternative IP-Adresse	X

Voraussetzungen:	
Hardware PC	Standard Netzwerkkarte (für -104) oder serielle Schnittstelle RS-232 oder RS-485 (für -101).
Software PC	--
Hardware Steuerung	--
Software Steuerung	--
Benötigt v-dll	X

Plattformen:	
Betriebssysteme	Windows 10; Windows 7; Windows 8; Windows 8.1; Windows Server 2008 R2; Windows Server 2012; Windows Server 2012 R2; Windows Server 2016

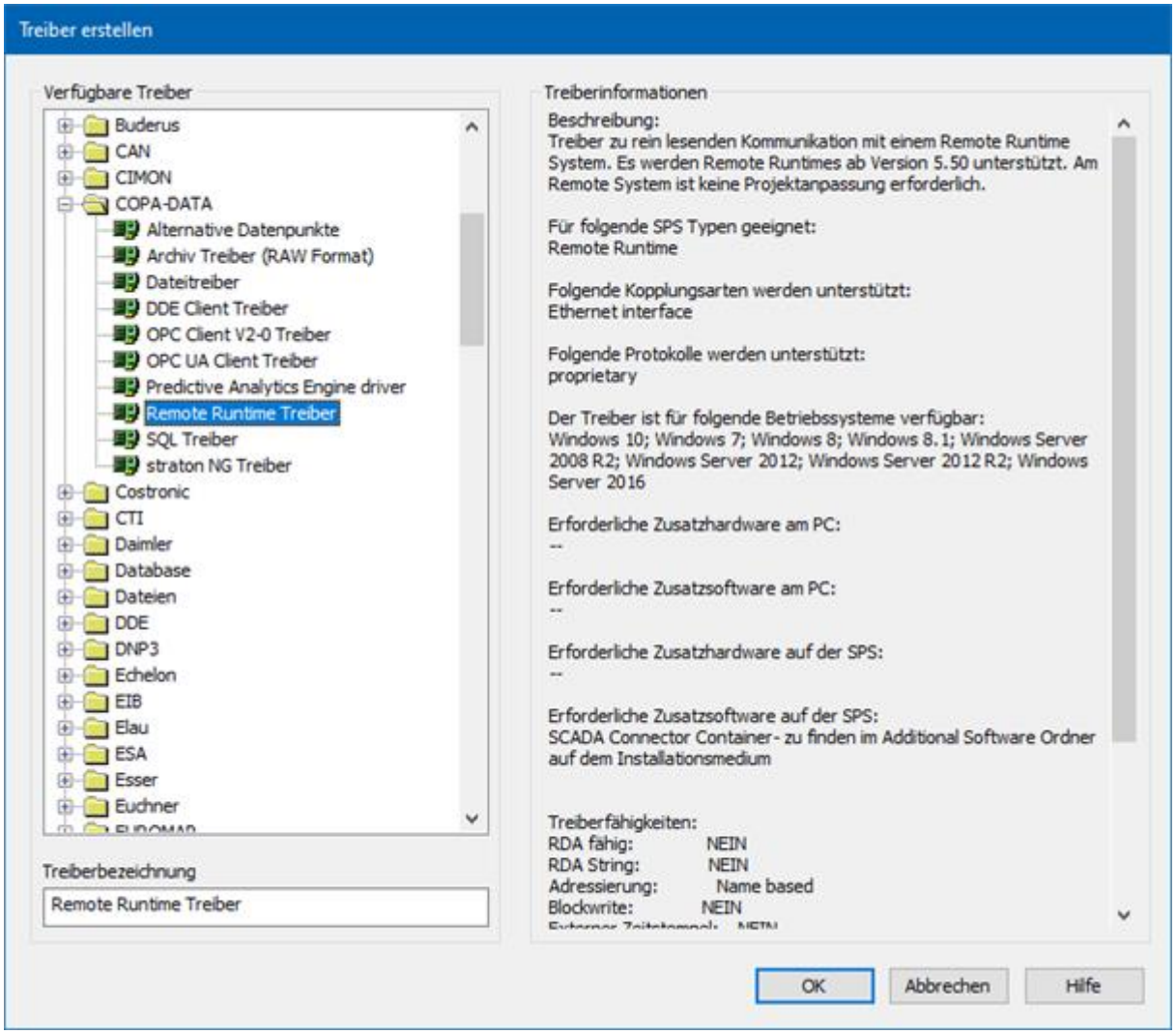
4 Konfiguration

In diesem Kapitel lesen Sie, wie Sie den Treiber im Projekt anlegen und welche Einstellungen beim Treiber möglich sind.

> **Info**
>
> Weitere Einstellungen, die Sie für Variablen in zenon vornehmen können, finden Sie im Kapitel Variablen der Online-Hilfe.

4.1 Anlegen eines Treibers

Im Dialog **Treiber erstellen** wählen Sie aus einer Liste jenen Treiber, den Sie neu anlegen wollen.



Parameter	Beschreibung
Verfügbare Treiber	Liste aller verfügbaren Treiber. Die Darstellung erfolgt in einer Baumstruktur: *[+]* erweitert die Ordnerstruktur und zeigt die darin

Parameter	Beschreibung
	enthaltenen Treiber. [-] reduziert die Ordnerstruktur Default: *keine Auswahl*
Treiberbezeichnung	Eindeutige **Bezeichnung** des Treibers. Default: *leer* Das Eingabefeld wird nach Auswahl eines Treibers aus der Liste der verfügbaren Treiber mit der vordefinierten **Bezeichnung** vorausgefüllt.
Treiberinformationen	Weiterführende Informationen über den gewählten Treiber. Default: *leer* Nach Auswahl eines Treibers werden in diesem Bereich die Informationen zum gewählten Treiber angezeigt.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt alle Einstellungen und öffnet den Treiberkonfigurationsdialog des ausgewählten Treibers.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

> **Info**
>
> Die Inhalte dieses Dialogs sind in der Datei Treiber_[Sprachkürzel].xml gespeichert. Sie finden diese Datei im Ordner *C:\ProgramData\COPA-DATA\zenon[Versionsnummer].*

TREIBER NEU ANLEGEN

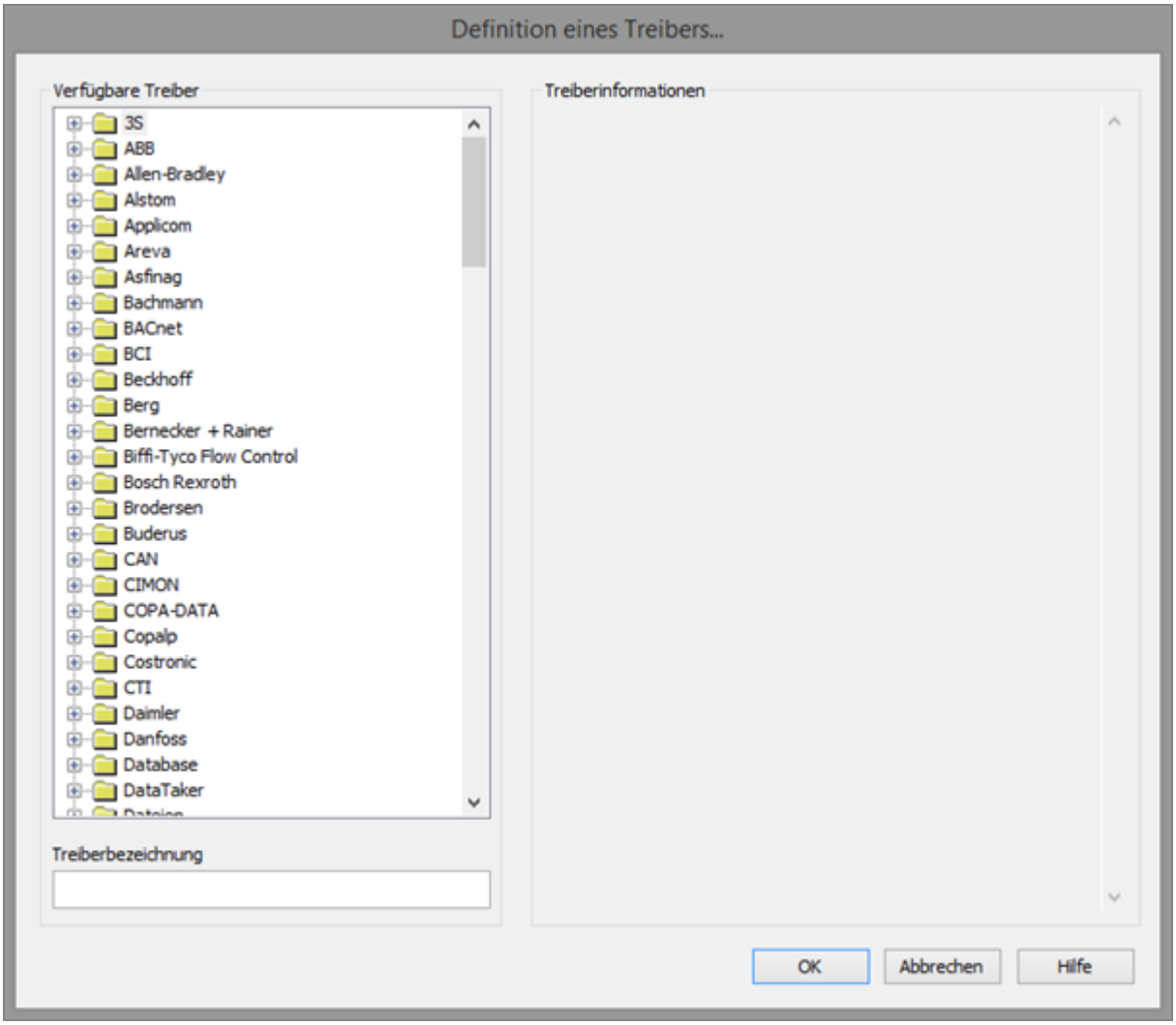
Um einen neuen Treiber anzulegen:

1. Klicken Sie mit der rechten Maustaste im Projektmanager auf **Treiber** und wählen Sie im Kontextmenü **Treiber neu** aus.

Optional: Wählen Sie die Schaltfläche **Treiber neu** aus der Symbolleiste der Detailansicht der **Variablen**.

Der Dialog **Treiber erstellen** wird geöffnet.

2. Der Dialog bietet eine Auflistung aller verfügbaren Treiber an.



3. Wählen Sie den gewünschten Treiber und benennen Sie diesen im Eingabefeld **Treiberbezeichnung**.
   Dieses Eingabefeld entspricht der Eigenschaft **Bezeichnung**. Per Default wird der Name des ausgewählten Treibers in diesem Eingabefeld automatisch eingefügt.
   Für die **Treiberbezeichnung** gilt:

   - Die **Treiberbezeichnung** muss eindeutig sein.
     Wird ein Treiber mehrmals im Projekt verwendet, so muss jeweils eine neue Bezeichnung vergeben werden.
     Dies wird durch Klick auf die Schaltfläche **OK** evaluiert. Ist die Treiber im Projekt bereits vorhanden wird dies mit einem Warndialog angezeigt.

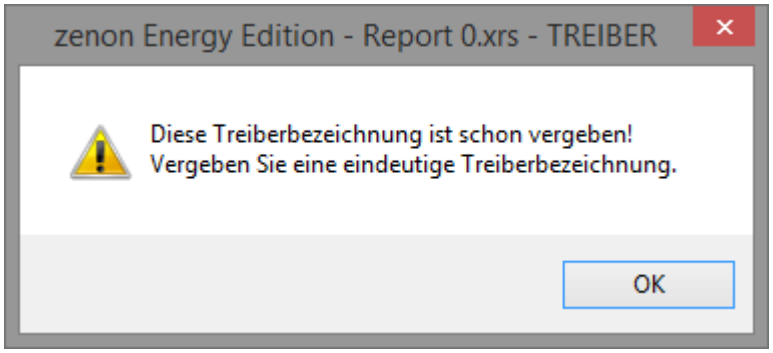
- Die **Treiberbezeichnung** ist Bestandteil des Dateinamens.
  Daher darf Sie nur Zeichen enthalten, die vom Betriebssystem unterstützt werden. Nicht gültige Zeichen werden durch einen Unterstrich (_) ersetzt.
- **Achtung:** Die Bezeichnung kann später nicht mehr geändert werden.

4. Bestätigen Sie den Dialog mit Klick auf die Schaltfläche **OK**.
   Der Konfigurationsdialog des ausgewählten Treibers wird geöffnet.

Hinweis: Treibernamen sind nicht sprachumschaltbar. Sie werden später immer in der Sprache angezeigt, in der sie angelegt wurden, unabhängig von der Sprache des Editors. Das gilt auch für Treiberobjekttypen.

DIALOG TREIBERBEZEICHNUNG BEREITS VORHANDEN

Ist ein Treiber bereits im Projekt vorhanden wird dies in einem Dialog angezeigt. Mit Klick auf die Schaltfläche **OK** wird der Warndialog geschlossen. Der Treiber kann korrekt benannt werden.



ZENON PROJEKT

Bei neu angelegten Projekten werden die folgenden Treiber automatisch angelegt:

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

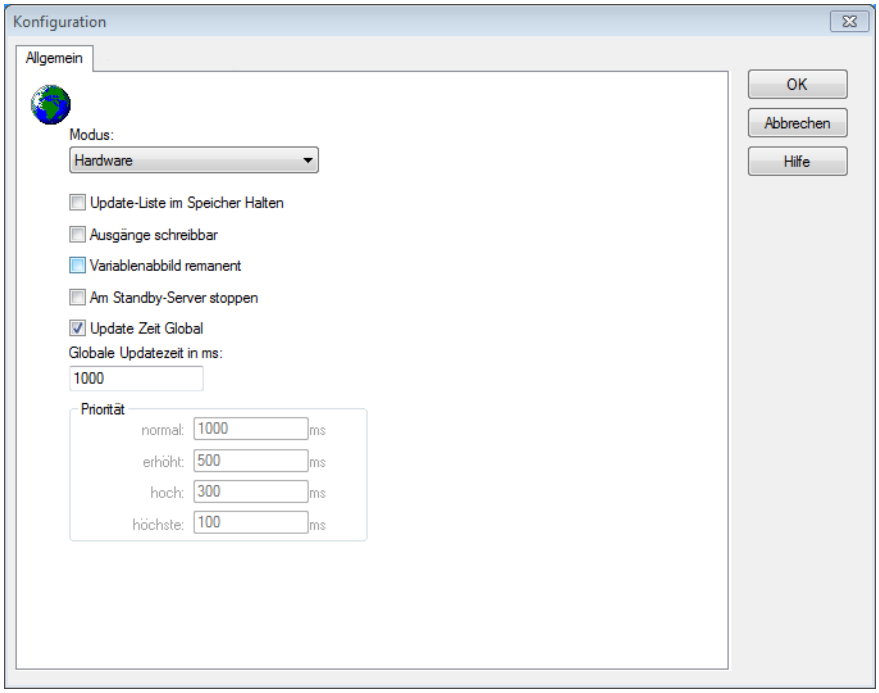
> **Info**
>
> In einem zenon Projekt müssen nur die benötigten Treiber vorhanden sein.
> Treiber können bei Bedarf zu einem späteren Zeitpunkt hinzugefügt werden.

4.2 Einstellungen im Treiberdialog

Folgende Einstellungen können Sie beim Treiber vornehmen:

4.2.1 Allgemein

Beim Anlegen eines Treibers wird der Konfigurationsdialog geöffnet. Um den Dialog später zum Bearbeiten zu öffnen, führen Sie einen Doppelklick auf den Treiber in der Liste aus oder klicken Sie auf die Eigenschaft **Konfiguration**.



Option	Beschreibung
Modus	Ermöglicht ein Umschalten zwischen Hardware und Simulationsmodus ▸ *Hardware*: Die Verbindung zur Steuerung wird hergestellt. ▸ Simulation - statisch: Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. In diesem Modus bleiben die Werte konstant oder die Variablen behalten die über zenon Logic gesetzen Werte. Jede Variable hat seinen eigenen Speicherbereich. Zum Beispiel zwei Variablen vom Typ Merker mit Offset 79, können zur Runtime unterschiedliche Werte haben und beeinflussen sich gegenseitig nicht. Ausnahme: Der Simulatortreiber. ▸ *Simulation - zählend*: Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert.

Option	Beschreibung
	In diesem Modus zählt der Treiber die Werte innerhalb ihres Wertebereichs automatisch hoch. ▸ *Simulation - programmiert*: Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden von einem frei programmierbaren Simulationsprojekt berechnet. Das Simulationsprojekt wird mit der zenon Logic Workbench erstellt und läuft in einer in den Treiber integrierten zenon Logic Runtime ab. Details siehe Kapitel Treibersimulation.
Update-Liste im Speicher Halten	Einmal angeforderte Variablen werden weiterhin von der Steuerung angefordert, auch wenn diese aktuell nicht mehr benötigt werden. Dies hat den Vorteil, dass z B. mehrmalige Bildumschaltungen nach dem erstmaligen Aufschalten beschleunigt werden, da die Variablen nicht neu angefordert werden müssen. Der Nachteil ist eine erhöhte Belastung der Kommunikation zur Steuerung.
Ausgänge schreibbar	▸ *Aktiv*: Ausgänge können beschrieben werden. ▸ *Inaktiv*: Das Beschreiben der Ausgänge wird unterbunden. **Hinweis**: Steht nicht für jeden Treiber zur Verfügungen.
Variablenabbild remanent	Diese Option speichert und restauriert den aktuellen Wert, den Zeitstempel und die Status eines Datenpunkts. Grundvoraussetzung: Die Variable muss einen gültigen Wert und Zeitstempel besitzen. Das Variablenabbild wird im Modus Hardware gespeichert, wenn einer dieser Status aktiv ist: ▸ Benutzerstatus *M1* (*0*) bis *M8* (*7*) ▸ *REVISION(9)* ▸ *AUS(20)* ▸ *ERSATZWERT(27)* Das Variablenabbild wird immer gespeichert wenn: ▸ die Variable vom Objekttyp

Option	Beschreibung
	Kommunikationsdetails ist ▸ der Treiber im Simulationsmodus läuft. (nicht programmierte Simulation) Folgende Status werden beim Start der Runtime nicht restauriert: ▸ *SELECT(8)* ▸ *WR-ACK(40)* ▸ *WR-SUC(41)* Der Modus **Simulation - programmiert** beim Treiberstart ist kein Kriterium, um das remanente Variablenabbild zu restaurieren.
Am Standby Server stoppen	Einstellung für Redundanz bei Treibern, die nur eine Kommunikationsverbindung erlauben. Dazu wird der Treiber am Standby Server gestoppt und erst beim Hochstufen wieder gestartet. **Achtung:** Ist diese Option aktiv, ist die lückenlose Archivierung nicht mehr gewährleistet. ▸ *Aktiv*: Versetzt den Treiber am nicht-prozessführenden Server automatisch in einen Stopp-ähnlichen Zustand. Im Unterschied zum Stoppen über Treiberkommando erhält die Variable nicht den Status **abgeschaltet**, sondern einen leeren Wert. Damit wird verhindert, dass beim Hochstufen zum Server nicht relevante Werte in AML , CEL und Archiv erzeugt werden. Default: *inaktiv* **Hinweis:** Nicht verfügbar, wenn CE Terminal als Datenserver dient. Weitere Informationen dazu erhalten Sie im Handbuch zenon Operator im Kapitel CE Terminal als Datenserver.
Update Zeit Global	Einstellung für globale Update-Zeiten in Millisekunden: ▸ *Aktiv*: Die eingestellte **Globale Update Zeit** wird für alle Variablen im Projekt verwendet. Die bei den Variablen eingestellte Priorität wird nicht

Option	Beschreibung
	verwendet. ▸ *Inaktiv*: Die eingestellten Prioritäten werden für die einzelnen Variablen verwendet. **Ausnahmen:** Spontane Treiber ignorieren diese Option. Sie nutzen in der Regel die kürzest mögliche Update Zeit. Details siehe Abschnitt **Update Zeit spontane Treiber**.
Priorität	Hier werden die Pollingzeiten der einzelnen Prioritätsklassen eingestellt. Alle Variablen mit der entsprechenden Priorität werden in der eingestellten Zeit gepollt. Die Zuordnung der Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften. Mit den Prioritätsklassen kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit oder benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast. **Achtung:** Prioritätsklassen werden nicht von jedem Treiber unterstützt, z.B. von spontan kommunizierenden zenon Treibern.

DIALOG BEENDEN

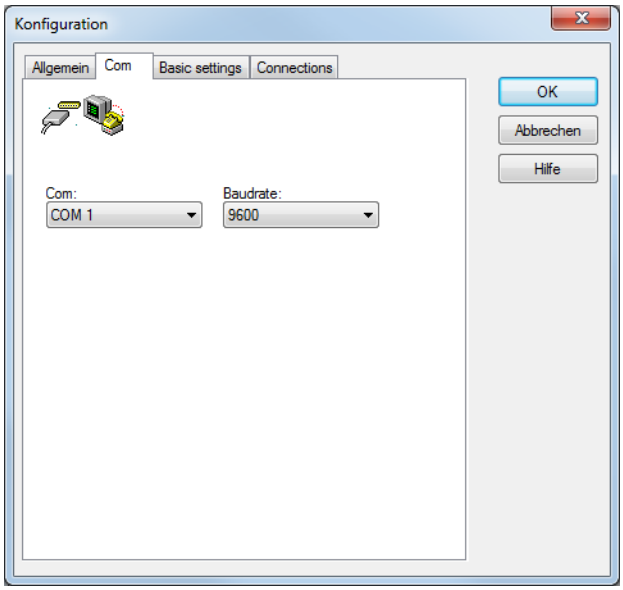
Option	Beschreibung
OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

UPDATE ZEIT SPONTANE TREIBER

Bei spontanen Treibern wird beim **Sollwert Setzen**, **Advisen** von Variablen und bei **Requests** sofort ein Lesezyklus ausgelöst - unabhängig von der eingestellten Update Zeit. Damit wird sicher gestellt, dass der Wert nach dem Schreiben in der Visualisierung sofort zur Verfügung steht. In der Regel beträgt die Updatezeit 100 ms.

Spontane Treiber sind **ArchDrv**, **BiffiDCM**, **BrTcp32**, **DNP3**, **Esser32**, **FipDrv32**, **FpcDrv32**, **IEC850**, **IEC870**, **IEC870_103**, **Otis**, **RTK9000**, **S7DCOS**, **SAIA_Slave**, **STRATON32** und **Trend32**.

4.2.2 Com



Für diesen Treiber, können nur COM-Port und Baudrate für die Kommunikation geändert werden. Alle anderen Kommunikationsparameter sind nach dem IEC 60870-5-101 Standard fest definiert.

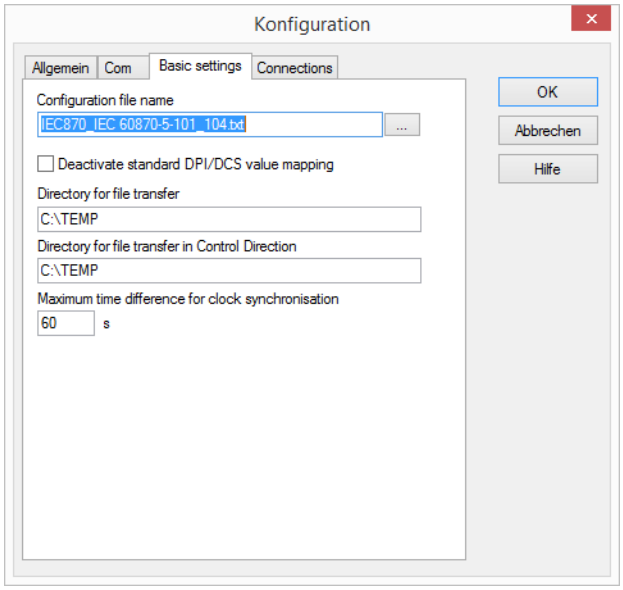
Werte:

- **Parität**: *even*
- **Stopbit**: *1*
- **Startbit**: *1*
- **Databit**: *8*

Details siehe: IEC 60870-5-1 "Part 5: Transmission protocols, Section One - Transmission frame formats".

4.2.3 Basic settings

Hinweis: Dieser Dialog steht nur in englischer Sprache zur Verfügung.



Parameter	Beschreibung
Configuration file name	Geben Sie hier den Namen des Konfigurationsdatei ein. Diese Datei ist für die Definition der Treiberverbindung notwendig.
Deactivate standard DPI/DCS value mapping	*Inaktiv* (Default): Die Werte von Doppelmeldungen (DPI und DCS) werden an die Bedienelemente von zenon angepasst. Verwenden Sie diese Konfiguration. wenn sie die Module von zenon Energy nutzen möchten. Der Treiber wandelt die Werte von Doppelmeldungen (DPI und DCS) für die Runtime um: ▸ *intermediate* | *off* | *on* | *fault* in: *2* | *0* | *1* | *3*, so dass in der Runtime z. B. der Wert *0* OFF und *1* ON bedeutet. *Aktiv*: Die Werte von Doppelmeldungen werden 1:1 an zenon weitergegeben: ▸ *intermediate* | *off* | *on* | *fault* = *0* | *1* | *2* | *3*). In diesem Fall kann allerdings z. B. die Befehlsgabe nicht verwendet werden, um die Doppelkommandos abzusetzen. **Hinweis:** Der Treiber wandelt nur einen Wert der Variable

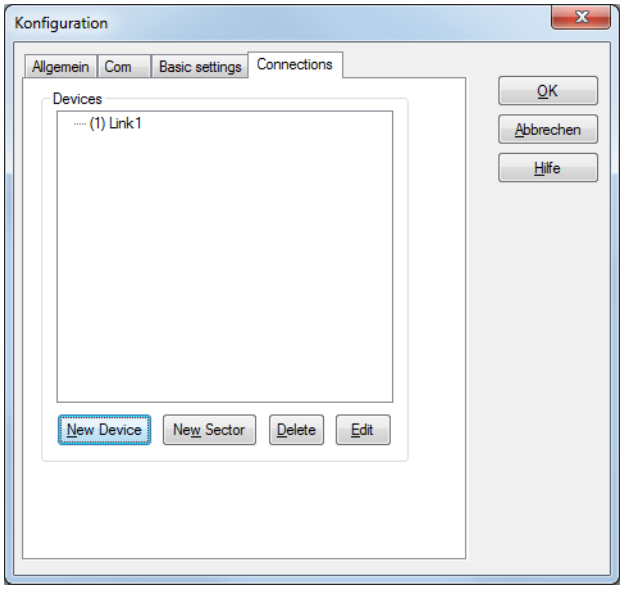
Parameter	Beschreibung
	um, der dem DPI/DCS Wertbereich (auf Seite 54) entspricht. DPI/DCS besteht laut Norm aus 2 Bits, alle anderen Bits der Variable werden nicht übertragen.
Directory for file transfer	Geben Sie hier den Ordner am Runtime-Rechner an, in dem die "file transfer" Dateien abgelegt werden sollen. In diesem Ordner liegen jene Dateien, welche vom IEC60870 Slave empfangen werden *(GET)*.
Directory for file transfer in Control Direction	Pfad des Verzeichnises, in welchem die Dateien für die Dateiübertragung liegen. In diesem Ordner werden jene Dateien abgelegt, welche zum IEC60870 Slave gesendet werden *(PUT)*.
Maximum time difference for clock synchronisation	Diese Parameter definiert ein maximales Zeitlimit bis zu dem die Uhrzeit-Synchronisierungen in Reverse Direction vom System automatisch vorgenommen wird. Ist die Differenz größer als die eingestellte Zeitdauer, wird die Synchronisierung nicht durchgeführt. Default: *60 s*

4.2.4 Connections

Hier können die Verbindungen definiert werden:

- Geräte

- Sektoren
  Achtung: Konfigurierte Sektoren müssen in der Steuerung vorhanden sein.



Parameter	Beschreibung
Devices	Liste der konfigurierten Geräte und Sektoren.
New Device	Öffnet Dialog zur Konfiguration neuer Geräte (auf Seite 20).
New Sector	Öffnet Dialog zur Konfiguration neuer Sektoren (auf Seite 28).
Delete	Löscht ausgewählten Eintrag aus der Liste.
Edit	Öffnet ausgewählten Eintrag zur Bearbeitung.

DIALOG BEENDEN

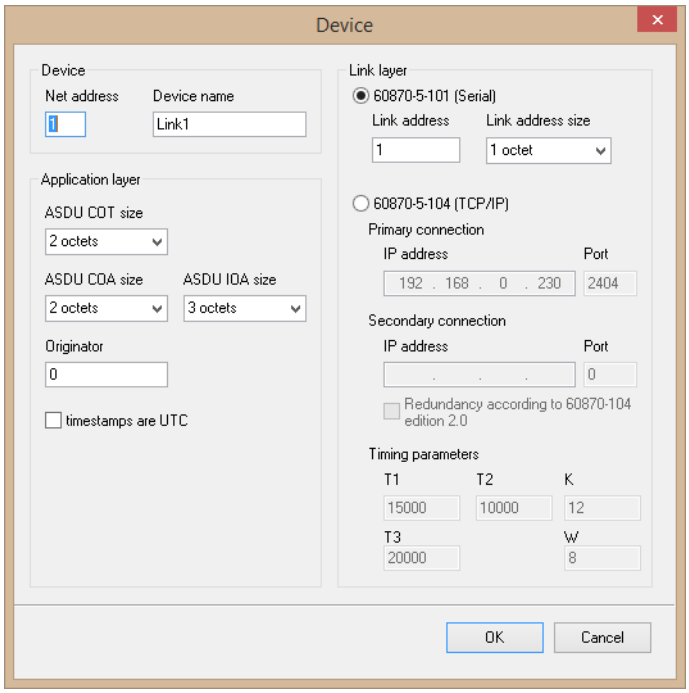
Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

4.2.4.1 Device

Eine Verbindung zu einem Gerät (Device) besteht aus einer physikalischen Verbindung des Geräts und mehren Sektoren (Common Address of ASDU) innerhalb dieses Geräts. Jedes Gerät wird identifiziert:

- auf der Runtime Seite durch die **Net address**. Diese muss mit der Eigenschaft **Netzadresse** der jeweiligen Variable stimmen.
- auf der Protokoll Seite durch die **Link address** oder **IP address**.

Hinweis: Dieser Dialog steht nur in englischer Sprache zur Verfügung.



DEVICE

Netzadresse, Name der Verbindung und Reverse Direction aktiv.

Parameter	Beschreibung
Net address	Netzadresse der Runtime. Laufende Nummer der projektierten Verbindung. **Achtung:** Jede Verbindung sollte eine eindeutige Nummer haben. Eine automatische Überprüfung, ob die Nummer bereits vergeben ist wird von zenon nicht durchgeführt.
Device name	Freigegebener Name des Geräts: Name des Geräts, wie es in der Liste angezeigt wird. **Hinweis:** Verwenden Sie kurze Namen. Beim

Parameter	Beschreibung
	Onlineimport wird dieser Name Teil der Variablenbenennung.
Reverse Direction	Bestimmt ob Reverse Direction aktiv ist für diese Verbindung. Default: inaktiv

APPLICATION LAYER

Anwendungsebene des Treibers. Stimmen Sie die Konfiguration in diesem Bereich mit ihrer SPS ab. Bei Fehlkonfiguration wird keine Kommunikation mit der Steuerung stattfinden.

Parameter	Beschreibung
ASDU COT size	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.
ASDU COA size	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.
ASDU IOA size	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.

Parameter	Beschreibung
Originator	Numerische Kennung des IEC60870 Masters. Die Kennung wird in Kommandos zur Steuerung geschickt. Somit kann die SPS Befehle von verschiedenen Master unterscheiden. Der Wert des Originators wird nur dann mitgeschickt, wenn bei **ASDU COT-Size** der Wert *2 octets* projektiert ist. Die Eingabe wird validiert. Bei ungültiger Eingabe wird ein Fehlerdialog aufgeschaltet. Default: *0* Eingabebereich: *0 - 255* Ausgegraut, wenn bei **ASDU COT-Size** der Wert *1 octet* projektiert ist.
Timestamps are UTC	Checkbox zur Auswahl des erwarteten Zeitformates. Ist diese Checkbox aktiviert, wird der Zeitstempel in UTC-Format erwartet oder für den Befehl (z.B. *T103* Zeitsynchronisation) verwendet. Eine nicht aktivierte Checkbox bedeutet, dass der Zeitstempel standardkonform als Lokalzeit interpretiert wird. Default: *inaktiv* **Achtung**: Laut IEC60870-Standard sollen alle Komponenten Lokalzeit verwenden.

LINK LAYER

Konfiguration der physikalischen Verbindung für Komminikation mit serieller Schnittstelle oder via TCP/IP.

SERIELLE KOMMUNIKATION

Parameter	Beschreibung
870-101	*Ausgewählt*: Verbindung wird seriell nach **60870-5-101** hergestellt.
Link adress	Serielle Adresse auf Protokollseite. Default: *1* **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-101 (TCP/IP)* ist.

Parameter	Beschreibung
Link address size	Auswahl der Adressgröße aus Dropdownliste: ▸ *1 octet* ▸ *2 octets* Default: *1 octet* **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-101 (TCP/IP)* ist.

KOMMUNIKATION VIA TCP/IP NETZWERK

Parameter	Beschreibung
870-104	*Ausgewählt*: Verbindung wird über das Netzwerk nach **60870-5-104** hergestellt. **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-104 (Serial)* ist.
Primary connection	Adressierung der primären Verbindung zur Steuerung.
IP address	IP-Adresse der Steuerung. **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-104 (Serial)* ist.
Port	Portnummer für primäre IP-Adresse. Default: 2404 **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-104 (Serial)* ist.
Secondary connection	Adressierung der alternativen Verbindung zur Steuerung.
IP address	IP-Adresse auf Protokollseite für reduntante Steuerungen. Alternative Verbindung bei Ausfall der primären Verbindung. **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-104 (Serial)* ist.
Port	Portnummer für redundante IP-Adresse. Default: *0*
Redundancy according to	Aktiv: Redundanz wird entsprechend **60870-104 edition 2**

Parameter	Beschreibung
870-104 ed. 2.0	ausgeführt. Aktivieren Sie diese Eigenschaft nur wenn Sie sicher sind, dass die Steuerung die Richtlinien der Edition 2 für Redundanz unterstützt. **Hinweis:** Nur aktiv wenn **Link layer** *60870-5-104 (Serial)* ist.

TIMING PARAMETERS

T0 entspricht dem Timeout beim TCP-Verbindungsaufbau und ist nicht einstellbar (vom Betriebssystem definiert).
Die Eigenschaften dieser Gruppe sind nur dann aktiv, wenn als Verbindungsart *60870-5-104 (TCP/IP)* gewählt ist.

Parameter	Beschreibung
T1	Timeout in Millisekunden für Bestätigung des gesendeten Frames. Wird innerhalb **T1** ein ASDU o. APCI Frame nicht bestätigt, wird es als Verbindungsausfall erkannt. Wertbereich: *0 - 4294967295* Default: *15000* ms
T2	Timeout in Millisekunden, innerhalb dessen eine Bestätigung via S-Frame (APCI) erfolgen soll, wenn kein Datenaustausch (via ASDU) stattfand. Wertbereich: *0 - 4294967295* Default: *10000* ms **Hinweis:** Laut IEC 60870-5-104 Standard soll **T2** kleiner **T1** sein; und gleich am Master und Slave.
T3	Timeout in Millisekunden für Test-Frames - periodische U-Frames (APCI) zwecks Test-Prozedur der Verbindung, wenn kein Datenaustausch (via ASDU) stattfindet. Wertbereich: *0 - 4294967295* Default: *20000* ms **Hinweis:** Die Test-Prozedur findet im Hintergrund (Data Link Layer) statt. Wird ein Test-Frame innerhalb **T1** nicht bestätigt, wird dies als Verbindungsausfall erkannt. Somit

Parameter	Beschreibung
	ist es ratsam **T3** > **T1** einstellen. Ausnahme: das redundante Data Link Layer nach Edition 2 des IEC 60870-5-104 Standards - es wird vorgeschlagen **T3** auf der Seite des Master zu verkürzen (z.B. *5s*).
K	Maximale Anzahl der I-Frames, für die noch keine Bestätigungen empfangen wurde. Wird die Anzahl **k** der ausstehenden Bestätigungen überschritten, wird es als Verbindungsausfall erkannt. Wertbereich: *0 - 4294967295* Default: *12* **Achtung:** sowohl Master wie Slave müssen der selbe **k**-Wert verwenden! **Hinweis:** in schnellen Ethernet-Netzen kann es ratsam sein die **k** und **w** Parametern (im Master und Slave) um Faktor 10 oder 100 zu erhöhen, z.B. **k**=*120*, **w**=*80*. Dies kann die Performance erhöhen.
W	Maximale Anzahl der erhaltenen I-Frames, bis eine Bestätigung gesendet wird. Der Wert **w** muss kleiner **k** sein. Wertbereich: *0 - 4294967295* Default: *8* **Achtung:** sowohl Master wie Slave müssen der selbe **w**-Wert verwenden! **Hinweis:** in schnellen Ethernet-Netzen kann es ratsam sein die **k** und **w** Parametern (im Master und Slave) um Faktor 10 oder 100 zu erhöhen, z.B. **k**=*120*, **w**=*80*. Dies kann die Performance erhöhen.

NAVIGATION

OK	**Übernimmt alle Änderungen und schließt den Dialog.**
Cancel	Verwirft alle Änderungen und schließt den Dialog.

OK	**Übernimmt alle Änderungen und schließt den Dialog.**

Achtung

OK **Übernimmt alle Änderungen und schließt den Dialog.**

Laut IEC 60870-5 Standard kann ein 870-Slave pro Sektor (**COA**) und Übertragungsrichtung der **Type ID** (monitoring oder control direction) jeweils nur ein Information Object (IO) mit bestimmter **IOA** Adresse haben. Ein IO kann abhängig von der Cause of Transmission (*COT*) ohne oder auch mit einem time tag übertragen werden.

Beispiel: Die Übertragung erfolgt einmal als T01 wegen einer Generalabfrage, einmal als T02 oder T30 bei spontaner Wertänderung. Alle diese drei **Type ID** sind single-point information, mal ohne und mal mit Zeitstempel (time tag).

Ein IO darf seinen Datentyp aber nicht ändern, z.B. von T01 - single auf T03 - double.

Werden im IEC870 Treiber (Master) mehrere Variablen angelegt, welche die gleiche COA, IOA und Übertragungsrichtung haben, kann das zu einem unerwünschten Verhalten in der Runtime führen.

Beispiel: Unterstützt wird das folgende Verhalten:

- Zwei Variablen T01 (monitoring direction) und T45 (command direction) mit gleicher COA und IOA Adresse.

  Es ist üblich und empfohlen, dass ein Befehl und eine Rückmeldung eines Objektes, die selbe COA und IOA Adressierung haben.

- Eine Variable T01, die dann den Wert eines Information Object beinhaltet (mit oder ohne Zeitstempel). Der Treiber wird den Wert und den Zeitstempel (wenn dabei) aus einer ASDU mit T01 oder T02 oder T30 (event with time tag) auf die Variable übertragen.

  Somit ist es nicht erforderlich, für Variablen, die über **Online Import** erstellt worden sind (also über die Generalabfrage, die IOs ohne Zeitstempel liefert), die **Type ID** manuell auf ein IO mit time tag abzuändern.

Bedingt unterstützt werden:

- Die zwei Variablen T01 (point information) und T30 (event with time tag) mit gleicher COA und IOA Adresse.

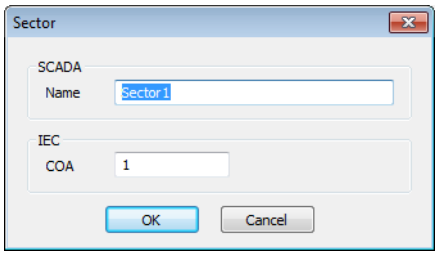
  In diesem Fall, wenn ein ASDU mit der COA und IOA eintrifft, dann wird der empfangenen Wert (T01) oder Wert und Zeitstempel (T02 oder T30) auf alle Variablen übertragen, die diese COA, IOA und die Übertragungsrichtung haben.
  Dies gilt ab zenon Version 8.10. In früheren Versionen wurde nur eine der Variablen mit Werten versorgt, und alle anderen wurden ignoriert.

4.2.4.2 Sector

Jedes Gerät (**Device**) kann mehrere Sektoren enthalten. Ein Sektor wird über sein Common Object Address (**COA**), auch Common Address of ASDU genannt, identifiziert und kann mehrere Datenpunkte - Information Objects - beinhalten.

Für jedes **Device** muss wenigstens ein **Sector** definiert werden. Der Treiber ignoriert alle ASDUs (Frames mit Daten), die ein **Device** (870-Slave) sendet, wenn das ASDU ein **COA** beinhaltet, die zu keinen vordefinierten **Sector** gehört. Beim Start der Kommunikation mit **Device** sendet der Treiber die General (Station) Interrogation - C_IC_NA_1 - ausschließlich für die **COA** der definierten Sektoren.

Hinweis: Dieser Dialog steht nur in englischer Sprache zur Verfügung.



Parameter	Beschreibung
SCADA	Einstellungen im SCADA.
Name	Frei definierbare Name o. Kurzbeschreibung des Sektors.
IEC	Einstellungen auf Protokollseite.
COA	Die Common Object Address, auch bekannt (IEC 60870-5-101 7.2.4) als Common Address of ASDU, unter welcher der Sektor angesprochen wird. Diese Nummer wird von Hersteller des Geräts (870-Slave) vorgegeben, liegt im Bereich *1..254* (*0xFE*) oder *1..65534* (*0xFFFE*), abhängig der **CAO Size** des Slaves. **Hinweis:** Die **COA** *0xFF(FF)* wird von machen Slaves als Broadcast interpretiert.

DIALOG BEENDEN

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen und schließt den Dialog.

Option	Beschreibung
OK	Übernimmt Einstellungen und schließt den Dialog.

Info

Alle im Treiberdialog konfigurierten Sektoren (**COAs**) müssen in der Steuerung vorhanden sein!

Wird ein Sektor definiert, der im Slave nicht existiert, so kann die initiale Generalabfrage (C_IC_NA_1) nicht abgeschlossen werden.

5 Variablen anlegen

So werden Variablen im zenon Editor angelegt:

5.1 Variablen im Editor anlegen

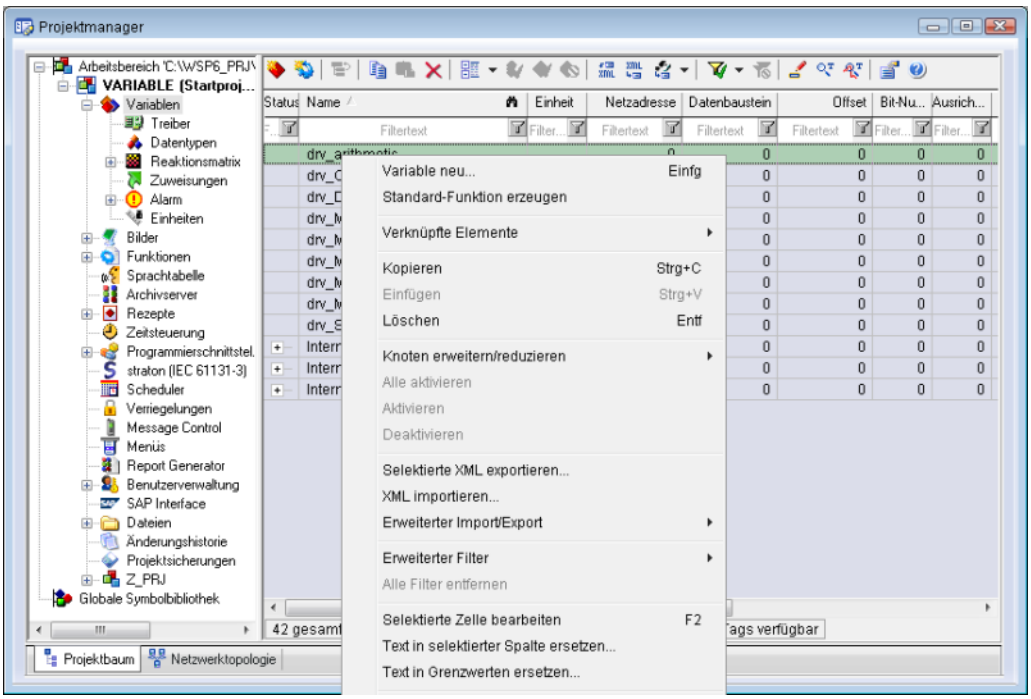
Variablen können angelegt werden:

- als einfache Variable
- in Arrays
- als Struktur-Variablen

DIALOG VARIABLE

Um eine neue Variable zu erstellen, gleich welchen Typs:

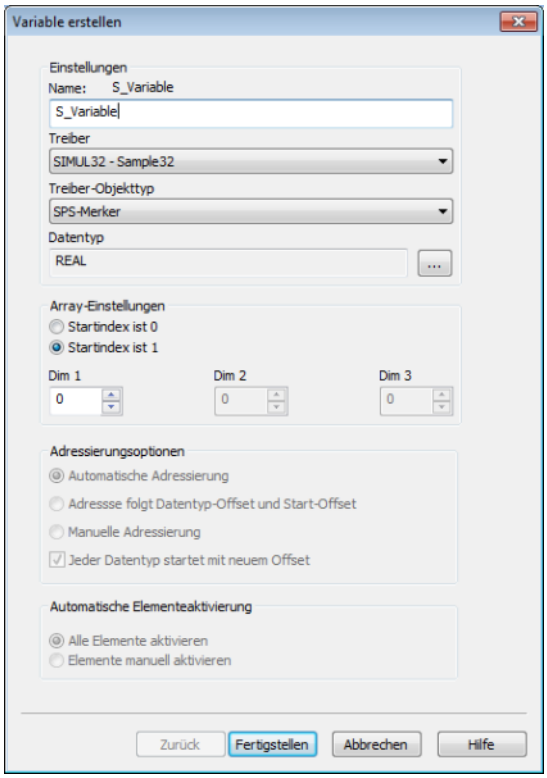
1. Wählen Sie im Knoten **Variablen** im Kontextmenü den Befehl **Variable neu**.



Der Dialog zur Konfiguration der Variable wird geöffnet.

2. Konfigurieren Sie die Variable.
3. Welche Einstellungen möglich sind, hängt ab vom Typ der Variablen.

DIALOG VARIABLE ERSTELLEN



Eigenschaft	Beschreibung
Name	Eindeutiger Name der Variablen. Ist eine Variable mit gleichem Namen im Projekt bereits vorhanden, kann keine weitere Variable mit diesem Namen angelegt werden. Maximale Länge: 128 Zeichen **Achtung:** Die Zeichen **#** und **@** sind für Variablennamen nicht erlaubt. Bei Verwendung nicht zugelassener Zeichen kann die Variablenerstellung nicht abgeschlossen werden, die Schaltfläche **Fertigstellen** bleibt inaktiv. **Hinweis:** Manche Treiber erlauben die Adressierung auch über die Eigenschaft **Symbolische Adresse**.
Treiber	Wählen Sie aus der Dropdownliste den gewünschten Treiber. **Hinweis:** Sollte im Projekt noch kein Treiber angelegt sein, wird automatisch der Treiber für interne Variable (**Intern.exe**) geladen.
Treiberobjekttyp	Wählen Sie aus der Dropdownliste den passenden Treiberobjekttyp aus.
Datentyp	Wählen Sie den gewünschten Datentyp. Klick auf die Schaltfläche

Eigenschaft	Beschreibung
	... öffnet den Auswahl-Dialog.
Array-Einstellungen	Erweiterte Einstellungen für Array-Variablen. Details dazu lesen Sie im Abschnitt Arrays.
Adressierungsoptionen	Erweiterte Einstellungen für Arrays und Struktur-Variablen. Details dazu lesen Sie im jeweiligen Abschnitt.
Automatische Elementeaktivierung	Erweiterte Einstellungen für Arrays und Struktur-Variablen. Details dazu lesen Sie im jeweiligen Abschnitt.

SYMBOLISCHE ADRESSE

Die Eigenschaft **Symbolische Adresse** kann für die Adressierung alternativ zu **Name** oder **Kennung** der Variablen verwendet werden. Die Auswahl erfolgt im Treiberdialog, die Konfiguration in der Variableneigenschaft. Beim Import von Variablen unterstützter Treiber wird die Eigenschaft automatisch eingetragen.

Maximale Länge: *1024* Zeichen.

Folgende Treiber unterstützen die **Symbolische Adresse**:

- **3S_V3**
- **AzureDrv**
- **BACnetNG**
- **IEC850**
- **KabaDPServer**
- **OPCUA32**
- **Phoenix32**
- **POZYTON**
- **RemoteRT**
- **S7TIA**
- **SEL**
- **SnmpNg32**
- **PA_Drv**
- **EUROMAP63**

ABLEITUNG VOM DATENTYP

Messbereich, **Signalbereich** und **Sollwert Setzen** werden immer:

- vom Datentyp abgeleitet
- beim Ändern des Datentyps automatisch angepasst

Hinweis Signalbereich: Bei einem Wechsel auf einen Datentyp, der den eingestellten **Signalbereich** nicht unterstützt, wird der **Signalbereich** automatisch angepasst. Zum Beispiel wird bei einem Wechsel von **INT** auf **SINT** der **Signalbereich** auf *127* geändert. Die Anpassung erfolgt auch dann, wenn der **Signalbereich** nicht vom Datentyp abgeleitet wurde. In diesem Fall muss der **Messbereich** manuell angepasst werden.

5.2 Adressierung

Die einzelnen Datenpunkte werden über eine COA (Common Object Address = Common Address of ASDU), einer IOA (Information Object Address) sowie dem IEC 60870 Typ adressiert. Dabei definiert der Typ - Type Identification - die Funktion der Variable (siehe Interoperapility Liste). Die COA entspricht dem Sektor, in dem sich die Variable am Device befindet. Die IOA legt den Offset im jeweiligen Sektor fest.

EIGENSCHAFTEN ZUR EINDEUTIGEN ADRESSIERUNG VON VARIABLEN

Eigenschaft	Beschreibung
Name	Frei vergebbarer Name. **Achtung:** Je zenon Projekt muss der Name eindeutig sein.
Kennung	Frei vergebbare Kennung. Z. B. für **Betriebsmittelkennung**, Kommentar usw.
Netzadresse	Netzadresse der Variable. Mit dieser Adresse wird die Zuordnung auf das in der Treiberkonfiguration angelegte **Device** festgelegt. Die dort verwendete Netzadresse muss bei den Variablen entsprechend eingetragen werden.
Datenbaustein	Wird für diesen Treiber nicht verwendet.
Offset	Wird für diesen Treiber nicht verwendet.
Ausrichtung	Wird für diesen Treiber nicht verwendet.
Bitnummer	Wird für diesen Treiber nicht verwendet.
Stringlänge	Nur verfügbar bei String-Variablen. Maximale Anzahl von Zeichen, die die Variable aufnehmen kann.
Symbolische Adresse	Wird für diesen Treiber nicht verwendet.

Eigenschaft	Beschreibung
IEC870 Type identification	Legt die Richtung der Kommunikation, Art und Funktion der Variable laut IEC 60870 Spezifikation fest. **Hinweis:** Sie können ein Typ in der Ausführung "with time tag", "with time tag CP56Time2a" und ohne Zeitstempel beliebig vergeben. **Beispiel:** Wurde die Variable mit Typ *T01 single-point information* angelegt, akzeptiert für sie der Treiber von SPS sowohl die Werte ohne Zeitstempel, wie mit - also auch *T02* oder *T30*. In zenon Runtime, die mit einen "time tag ..." empfangene Werte, haben dann doch einen externen Zeitstempel.
IEC870 COA	Common Object Address (Common Address of ASDU). Definiert den Sektor, in dem sich die Variable am **Device** befindet.
IEC870 IOA	Information Object Address. Adressierung (Offset) der Variable innerhalb eines Sektors (einer **COA**)
IEC870 private Index	Eigenschaft nur für Variablen mit **Type ID** in privaten Bereich, z.B. *T160*. Default: *0*
Treiber Anbindung /Treiberobjekt typ	Objekttyp der Variablen. Wird abhängig vom verwendeten Treiber beim Erstellen der Variablen ausgewählt und kann hier geändert werden. Default: *PLC marker*
Treiber Anbindung/Datentyp	Datentyp der Variablen. Wird beim Erstellen der Variablen ausgewählt und kann hier geändert werden. **ACHTUNG:** Wenn der Datentyp nachträglich geändert wird, müssen alle anderen Eigenschaften der Variablen überprüft bzw. angepasst werden.
Treiber Anbindung/Priorität	Wird für diesen Treiber nicht verwendet. Der Treiber unterstützt keine zyklisch pollende Kommunikation in Prioritätsklassen.
Nur von Standby Server anfordern	Quelle des Variablenwertes für die Anzeige in der Runtime, wenn das Projekt in einem zenon Netzwerk eingesetzt wird. Aktiv: Die Variable visualisiert den Wert vom Treiber auf Standby Server

Achtung

Laut IEC 60870-5 Standard kann ein 870-Slave pro Sektor (**COA**) und Übertragungsrichtung der **Type ID** (monitoring oder control direction) jeweils nur ein Information Object (IO) mit bestimmter **IOA** Adresse haben. Ein IO kann abhängig von der Cause of Transmission (*COT*) ohne oder auch mit einem time tag übertragen werden.

Beispiel: Die Übertragung erfolgt einmal als T01 wegen einer Generalabfrage, einmal als T02 oder T30 bei spontaner Wertänderung. Alle diese drei **Type ID** sind single-point information, mal ohne und mal mit Zeitstempel (time tag).

Ein IO darf seinen Datentyp aber nicht ändern, z.B. von T01 - single auf T03 - double.

Werden im IEC870 Treiber (Master) mehrere Variablen angelegt, welche die gleiche COA, IOA und Übertragungsrichtung haben, kann das zu einem unerwünschten Verhalten in der Runtime führen.

Beispiel: Unterstützt wird das folgende Verhalten:

- Zwei Variablen T01 (monitoring direction) und T45 (command direction) mit gleicher COA und IOA Adresse.

  Es ist üblich und empfohlen, dass ein Befehl und eine Rückmeldung eines Objektes, die selbe COA und IOA Adressierung haben.

- Eine Variable T01, die dann den Wert eines Information Object beinhaltet (mit oder ohne Zeitstempel). Der Treiber wird den Wert und den Zeitstempel (wenn dabei) aus einer ASDU mit T01 oder T02 oder T30 (event with time tag) auf die Variable übertragen.

  Somit ist es nicht erforderlich, für Variablen, die über **Online Import** erstellt worden sind (also über die Generalabfrage, die IOs ohne Zeitstempel liefert), die **Type ID** manuell auf ein IO mit time tag abzuändern.

Bedingt unterstützt werden:

- Die zwei Variablen T01 (point information) und T30 (event with time tag) mit gleicher COA und IOA Adresse.

  In diesem Fall, wenn ein ASDU mit der COA und IOA eintrifft, dann wird der empfangenen Wert (T01) oder Wert und Zeitstempel (T02 oder T30) auf alle Variablen übertragen, die diese COA, IOA und die Übertragungsrichtung haben.
  Dies gilt ab zenon Version 8.10. In früheren Versionen wurde nur eine der Variablen mit Werten versorgt, und alle anderen wurden ignoriert.

5.3 Treiberobjekte und Datentypen

Treiberobjekte sind in der Steuerung verfügbare Bereiche wie z. B. Merker, Datenbausteine usw. Hier lesen Sie, welche Treiberobjekte vom Treiber zur Verfügung gestellt werden und welche IEC-Datentypen dem jeweiligen Treiberobjekt zugeordnet werden können.

5.3.1 Treiberobjekte

Folgende Objekttypen stehen in diesem Treiber zur Verfügung:

TREIBER OBJEKTTYPEN UND UNTERSTÜTZTE IEC DATENTYPEN FÜR PROZESSVARIABLEN IN ZENON

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
SPS-Merker	8	X	X	*BOOL*, *SINT*, *USINT*, *INT*, *UINT*, *DINT*, *UDINT*, *REAL*, *STRING*	Variablen die in SPS den Information Objects entsprechen; und die mit *T00* (internal state). **Type ID**: ▸ *1..37*, *126* - nur lesen ▸ *45..103*, *122* - nur schreiben
Kommunikationsdetails	35	X	X	*BOOL*, *SINT*, *USINT*, *INT*, *UINT*, *DINT*, *UDINT*, *REAL*, *STRING*	Variablen für die statische Analyse der Kommunikation. Werte werden nur zwischen Treiber und Runtime übertragen, nicht zur SPS! **Hinweis**: Die Adressierung und das Verhalten ist bei den meisten zenon Treibern gleich.

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
					Weitere Informationen dazu finden Sie im Kapitel Kommunikationsdetails (Treibervariablen) (auf Seite 48).

Legende:

X: wird unterstützt

--: wird nicht unterstützt

KANALTYP

Der Begriff "**Kanaltyp**" ist die interne numerische Bezeichnung des Treiberobjekttyps. Diese wird auch für den erweiterten DBF Import/Export der Variablen verwendet.

"**Kanaltyp**" wird für den erweiterten CSV Import/Export der Variablen in der Spalte "**HWObjectType**" verwendet.

5.3.2 Zuordnung der Datentypen

Alle Variablen in zenon werden von IEC-Datentypen abgeleitet. In folgender Tabelle werden zur besseren Übersicht die IEC-Datentypen den Datentypen der Steuerung gegenübergestellt.

ZUORDNUNG DER DATENTYPEN AUS DER STEUERUNG AUF DIE ZENON DATENTYPEN

Steuerung	ASDU Type ID	zenon	Kommentar	Datenart
M_SP_NA_1	1	BOOL	SPI <0..1> single-point information	8
M_SP_TA_1	2	BOOL	SPI [1)] single-point information with time tag (CP24Time2a)	8
M_SP_TB_1	30	BOOL	SPI [2)]	8

Steuerung	ASDU Type ID	zenon	Kommentar	Datenart
			single-point information with time tag CP56Time2a	
M_DP_NA_1	3	USINT	DPI <0..3> mit Mapping von Double Point Values (auf Seite 62) double-point information	9
M_DP_TA_1	4	USINT	DPI [1] mit Mapping von Double Point Values (auf Seite 62)	9
M_DP_TB_1	31	USINT	DPI [2] mit Mapping von Double Point Values (auf Seite 62)	9
M_ST_NA_1	5	USINT	Entspricht gesamten VTI (IEC60870-5-101 7.2.6.5). Bit 8 ist das Transient Bit. step position information	9
M_ST_TA_1	6	USINT	VTI [1]	9
M_ST_TB_1	32	USINT	VTI [2]	9
M_BO_NA_1	7	UDINT	BSI (32 Bits) bitstring of 32 bits	4
M_BO_TA_1	8	UDINT	BSI [1]	4
M_BO_TB_1	33	UDINT	BSI [2]	4
M_ME_NA_1	9	REAL	NVA <–1..+1 –2^{-15}>, in Praxis <-1..0,9999> measured value, normalized value	5
M_ME_TA_1	10	REAL	NVA [1]	5
M_ME_TD_1	34	REAL	NVA [2]	5
M_ME_NB_1	11	INT	SVA <–2^{15}..+2^{15}–1> = <-32768..32767> measured value, scaled value	1
M_ME_TB_1	12	INT	SVA [1]	1
M_ME_TE_1	35	INT	SVA [2]	1
M_ME_NC_1	13	REAL	R32 measured value, short floating point number	5

Steuerung	ASDU Type ID	zenon	Kommentar	Datenart
M_ME_TC_1	14	REAL	R32 [1]	5
M_ME_TF_1	36	REAL	R32 [2]	5
M_IT_NA_1	15	DINT	BCR.Counter reading <-2^{31}..+2^{31}-1> integrated totals	3
M_IT_TA_1	16	DINT	BCR.Counter reading [1]	3
M_IT_TB_1	37	DINT	BCR.Counter reading [2]	3
C_SC_NA_1	45	BOOL	SCS single command	8
C_SC_TA_1	58	BOOL	SCS	8
C_DC_NA_1	46	USINT	DCS mit Mapping von Double Point Values (auf Seite 62) double command	9
C_DC_TA_1	59	USINT	DCS mit Mapping von Double Point Values (auf Seite 62)	9
C_RC_NA_1	47	USINT	RCS regulating step command	9
C_RC_TA_1	60	USINT	RCS	9
C_SE_NA_1	48	REAL	NVA set point command, normalized value	5
C_SE_TA_1	61	REAL	NVA	5
C_SE_NB_1	49	INT	SVA set point command, scaled value	1
C_SE_TB_1	62	INT	SVA	1
C_SE_NC_1	50	REAL	R32 set point command, short floating point number	5
C_SE_TC_1	63	REAL	R32	5
C_BO_NA_1	51	UDINT	BSI (32 Bits)	4

Steuerung	ASDU Type ID	zenon	Kommentar	Datenart
			bitstring of 32 bits command	
C_BO_TA_1	64	UDINT	BSI (32 Bits)	4
C_IC_NA_1	100	BOOL	1 während der Ausführung (general) interrogation command	8
C_CS_NA_1	103	BOOL	1 während der Ausführung clock synchronization command	8
F_SC_NA_1	122	STRING	Befehl für Filetransfer, z. B. "DIR" oder "GET"	12
F_DR_TA_1	126	STRING	Rückmeldevariable für Filetransfer	12
	136 - 141 und 160	numerisch	Adressierung für Siemens-spezifischen privaten **Type ID** Bereich.	

1) Time tag CP24Time2a beinhaltet nur mm:ss.ms; wird für Zeitstempel der Variable verwendet wobei der Treiber auf die PC-Uhr zurückgreift, um die fehlenden Zeitinformationen zu ergänzen. Bei einem höheren Minutenwert als der PC-Uhr setzt der Treiber automatisch den Zeitstempel eine Stunde zurück.
2) Time tag CP56Time2a wird für Zeitstempel der Variable verwendet.

ASDU Type ID: IEC60870-5-101 Type identification, entspricht der Eigenschaft **IEC870 Type identification** einer Variable.

DATENART

Der Begriff **Datenart** ist die interne numerische Bezeichnung des Datentyps. Diese wird auch für den erweiterten DBF Import/Export der Variablen verwendet.

5.4 Variablen anlegen durch Import

Variablen können auch mittels Variablenimport angelegt werden. Für jeden Treiber stehen XML- und DBF-Import zur Verfügung.

Info

Details zu Import und Export von Variablen finden Sie im Handbuch Import-Export im Abschnitt Variablen.

5.4.1 XML Import

Beim XML- Import von Variablen oder Datentypen werden diese erst einem Treiber zugeordnet und dann analysiert. Vor dem Import entscheidet der Benutzer, ob und wie das jeweilige Element (Variable oder Datentyp) importiert werden soll:

- *Importieren*:
  Das Element wird neu importiert.
- *Überschreiben*:
  Das Element wird importiert und überschreibt ein bereits vorhandenes Element.
- *Nicht importieren*:
  Das Element wird nicht importiert.

Hinweis: Beim Import werden die Aktionen und deren Dauer in einem Fortschrittsbalken angezeigt. In der folgenden Dokumentation wird der Import von Variablen beschrieben. Datentypen werden analog dazu importiert.

VORAUSSETZUNGEN

Beim Import gelten folgende Bedingungen:

- **Abwärtskompatibilität**

  Beim XML Import/Export ist keine Abwärtskompatibilität gegeben. Daten aus älteren zenon Versionen können übernommen werden. Die Übergabe von Daten aus neueren Versionen an ältere wird nicht unterstützt.

- **Konsistenz**

  Die zu importierende XML-Datei muss konsistent sein. Beim Import der Datei erfolgt keine Plausibilitätsprüfung. Weisen die importierten Daten Fehler auf, kann es zu unerwünschten Effekten im Projekt kommen.

  Dies muss vor allem auch beachtet werden, wenn in einer XML-Datei nicht alle Eigenschaften vorhanden sind und diese dann durch Default-Werte ersetzt werden. Z. B.: Eine binäre Variable hat einen Grenzwert von *300*.

- **Struktur-Datentypen**

Struktur-Datentypen müssen über die gleiche Anzahl von Strukturelementen verfügen. Beispiel: Ein Strukturdatentyp im Projekt hat 3 Strukturelemente. Ein gleichnamiger Datentyp in der XML-Datei hat 4 Strukturelemente. Dann wird keine der auf diesem Datentyp basierenden Variablen der Datei in das Projekt importiert.

Tipp

Weitere Informationen zum XML-Import finden Sie im Handbuch **Import - Export**, im Kapitel **XML-Import**.

5.4.2 DBF Import/Export

Daten können nach dBase exportiert und aus dBase importiert werden.

Info

Import und Export über CSV oder dBase unterstützt keine treiberspezifischen Variableneinstellungen wie z. B. Formeln. Nutzen Sie dafür den Export/Import über XML.

IMPORT DBF-DATEI

Um den Import zu starten:

1. Führen Sie einen Rechtsklick auf die Variablenliste aus.
2. Wählen Sie in der Dropdownliste von **Erweiterter Export/Import** ... den Befehl **dBase importieren**.
3. Folgen Sie den Anweisungen des Importassistenten.

Das Format der Datei ist im Kapitel Dateiaufbau beschrieben.

Info

Beachten Sie:

- Treiberobjekttyp und Datentyp müssen in der DBF-Datei an den Zieltreiber angepasst werden, damit Variablen importiert werden.
- dBase unterstützt beim Import keine Strukturen oder Arrays (komplexe Variablen).

EXPORT DBF-DATEI

Um den Export zu starten:

1. Führen Sie einen Rechtsklick auf die Variablenliste aus.
2. Wählen Sie im Dropdownliste von **Erweiterter Export/Import** ... den Befehl **dBase exportieren...** .
3. Folgen Sie den Anweisungen des Exportassistenten.

> **Achtung**
>
> DBF-Dateien:
>
> - müssen in der Benennung dem 8.3 DOS Format für Dateinamen entsprechen (8 alphanumerische Zeichen für Name, 3 Zeichen Erweiterung, keine Leerzeichen)
> - dürfen im Pfadnamen keinen Punkt (.) enthalten.
> Z. B. ist der Pfad *C:\users\Max.Mustermann\test.dbf* ungültig.
> Gültig wäre: *C:\users\MaxMustermann\test.dbf*
> - müssen nahe am Stammverzeichnis (Root) abgelegt werden, um die eventuelle Beschränkungen für Dateinamenlänge inklusive Pfad zu erfüllen: maximal 255 Zeichen

Das Format der Datei ist im Kapitel Dateiaufbau beschrieben.

> **Info**
>
> dBase unterstützt beim Export keine Strukturen oder Arrays (komplexe Variablen).

DATEIAUFBAU DER DBASE EXPORTDATEI

Für den Variablenimport und -export muss die dBaseIV-Datei folgende Struktur und Inhalte besitzen.

> **Achtung**
>
> dBase unterstützt keine Strukturen oder Arrays (komplexe Variablen).
>
> DBF-Dateien müssen:
>
> - in der Benennung dem 8.3 DOS Format für Dateinamen entsprechen (8 alphanumerische Zeichen für Name, 3 Zeichen Erweiterung, keine Leerzeichen)
> - nahe am Stammverzeichnis (Root) abgelegt werden

STRUKTUR

Bezeichnung	Typ	Feldgröße	Bemerkung
KANALNAME	Char	128	Variablenname. Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
KANAL_R	C	128	Ursprünglicher Name einer Variablen, der durch den Eintrag unter VARIABLENNAME ersetzt werden soll (Feld/Spalte muss manuell angelegt werden). Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
KANAL_D	Log	1	Variable wird bei Eintrag *1* gelöscht (Feld/Spalte muss manuell angelegt werden).
TAGNR	C	128	Kennung. Länge kann über den Eintrag **MAX_LAENGE** in der **project.ini** eingeschränkt werden.
EINHEIT	C	11	Technische Maßeinheit
DATENART	C	3	Datentyp (z. B. Bit, Byte, Wort, ...) entspricht dem Datentyp.
KANALTYP	C	3	Speicherbereich in der SPS (z. B. Merkerbereich, Datenbereich, ...) entspricht Treiberobjekttyp.
HWKANAL	Num	3	Netzadresse
BAUSTEIN	N	3	Datenbaustein-Adresse (nur bei Variablen aus den Datenbereich der SPS)
ADRESSE	N	5	Offset
BITADR	N	2	Für Bit-Variablen: Bitadresse Für Byte-Variablen: 0=niederwertig, 8=höherwertig Für String-Variablen: Stringlänge (max. 63 Zeichen)
ARRAYSIZE	N	16	Anzahl der Variablen im Array für Index-Variablen ACHTUNG: Nur die erste Variable steht voll zur Verfügung. Alle folgenden sind nur über VBA oder den Rezeptgruppen Manager zugänglich
LES_SCHR	L	1	Lese-Schreib-Berechtigung

Bezeichnung	Typ	Feldgröße	Bemerkung
			0: Sollwert setzen ist nicht erlaubt 1: Sollwert setzen ist erlaubt
MIT_ZEIT	L	1	Zeitstempelung in zenon (nur wenn vom Treiber unterstützt)
OBJEKT	N	2	Treiberspezifische ID-Nummer des Primitivobjekts setzt sich zusammen aus TREIBER-OBJEKTTYP und DATENTYP
SIGMIN	Float	16	Rohwertsignal minimal (Signalauflösung)
SIGMAX	F	16	Rohwertsignal maximal (Signalauflösung)
ANZMIN	F	16	technischer Wert minimal (Messbereich)
ANZMAX	F	16	technischer Wert maximal (Messbereich)
ANZKOMMA	N	1	Anzahl der Nachkommastellen für die Darstellung der Werte (Messbereich)
UPDATERATE	F	19	Updaterate für Mathematikvariablen (in sec, eine Dezimalstelle möglich) bei allen anderen Variablen nicht verwendet
MEMTIEFE	N	7	Nur aus Kompatibilitätsgründen vorhanden
HDRATE	F	19	HD-Updaterate für hist. Werte (in sec, eine Dezimalstelle möglich)
HDTIEFE	N	7	HD-Eintragtiefe für hist. Werte (Anzahl)
NACHSORT	L	1	HD-Werte als nachsortierte Werte
DRRATE	F	19	Aktualisierung an die Ausgabe (für zenon DDE-Server, in sec, eine Kommastelle möglich)
HYST_PLUS	F	16	Positive Hysterese; ausgehend vom Messbereich
HYST_MINUS	F	16	Negative Hyterese; ausgehend vom Messbereich
PRIOR	N	16	Priorität der Variable
REAMATRIZE	C	32	Name der zugeordnete Reaktionsmatrix
ERSATZWERT	F	16	Ersatzwert; ausgehend vom Messbereich
SOLLMIN	F	16	Sollwertgrenze Minimum; ausgehend vom Messbereich

Bezeichnung	Typ	Feldgröße	Bemerkung
SOLLMAX	F	16	Sollwertgrenze Maximum; ausgehend vom Messbereich
VOMSTANDBY	L	1	Variable vom Standby Server anfordern; der Wert der Variable wird im redundanten Netzwerkbetrieb nicht vom Server sondern vom Standby Server angefordert
RESOURCE	C	128	Betriebsmittelkennung. Freier String für Export und Anzeige in Listen. Länge kann über den Eintrag MAX_LAENGE in der **project.ini** eingeschränkt werden.
ADJWVBA	L	1	Nichtlineare Wertanpassung: *0*: Nichtlineare Wertanpassung wird verwendet *1*: Nichtlineare Wertanpassung wird nicht verwendet
ADJZENON	C	128	Verknüpftes VBA-Makro zum Lesen der Variablenwerte für die nichtlineare Wertanpassung.
ADJWVBA	C	128	Verknüpftes VBA-Makro zum Schreiben der Variablenwerte für die nichtlineare Wertanpassung.
ZWREMA	N	16	Verknüpfte Zählwert-Rema.
MAXGRAD	N	16	Maximaler Gradient für die Zählwert-Rema.

Achtung

Beim Import müssen Treiberobjekttyp und Datentyp in der DBF-Datei an den Zieltreiber angepasst werden, damit Variablen importiert werden.

GRENZWERTDEFINITION

Grenzwertdefinition für Grenzwert *1* bis *4*, oder Zustand *1* bis *4*:

Bezeichnung	Typ	Feldgröße	Bemerkung
AKTIV1	L	1	Grenzwert aktiv (pro Grenzwert vorhanden)
GRENZWERT1	F	20	technischer Wert oder ID-Nummer der verknüpften Variable für einen dynamischen Grenzwert (siehe VARIABLEx) (wenn unter VARIABLEx *1* steht und hier *-1*, wird die bestehende Variablenzuordnung nicht überschrieben)

Bezeichnung	Typ	Feldgröße	Bemerkung
SCHWWERT1	F	16	Schwellwert für den Grenzwert
HYSTERESE1	F	14	wird nicht verwendet
BLINKEN1	L	1	Blinkattribut setzen
BTB1	L	1	Protokollierung in CEL
ALARM1	L	1	Alarm
DRUCKEN1	L	1	Druckerausgabe (bei CEL oder Alarm)
QUITTIER1	L	1	quittierpflichtig
LOESCHE1	L	1	löschpflichtig
VARIABLE1	L	1	dyn. Grenzwertverknüpfung der Grenzwert wird nicht durch einen absoluten Wert (siehe Feld GRENZWERTx) festgelegt.
FUNC1	L	1	Funktionsverknüpfung
ASK_FUNC1	L	1	Ausführung über die Alarmmeldeliste
FUNC_NR1	N	10	ID-Nummer der verknüpften Funktion (steht hier -1, so wird die bestehende Funktion beim Import nicht überschrieben)
A_GRUPPE1	N	10	Alarm/Ereignis-Gruppe
A_KLASSE1	N	10	Alarm/Ereignis-Klasse
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Farbe als Windowskodierung
GRENZTXT1	C	66	Grenzwerttext
A_DELAY1	N	10	Zeitverzögerung
INVISIBLE1	L	1	Unsichtbar

Bezeichnungen in der Spalte Bemerkung beziehen sich auf die in den Dialogboxen zur Definition von Variablen verwendeten Begriffe. Bei Unklarheiten, siehe Kapitel Variablendefinition.

5.4.3 Online Import

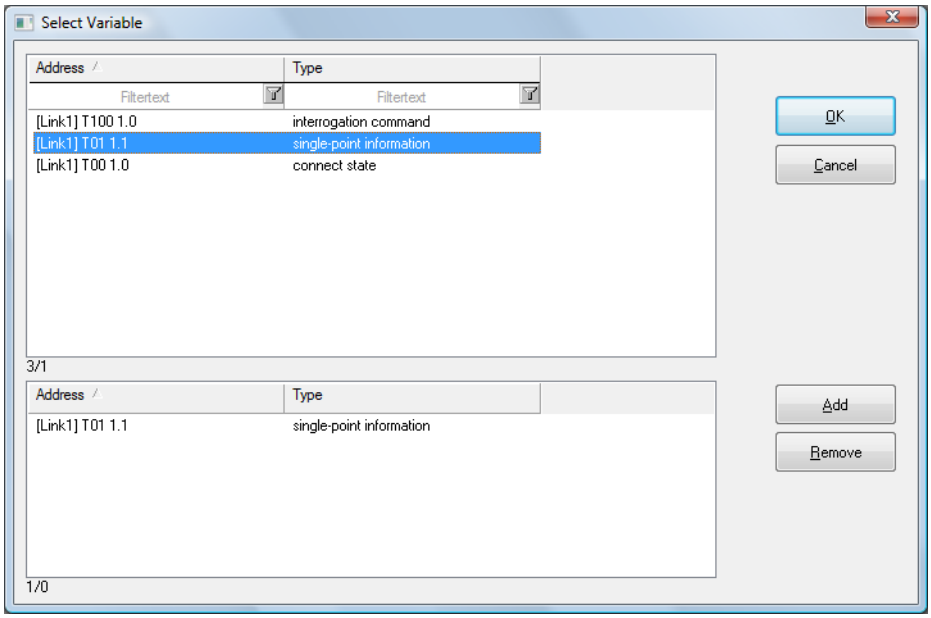
Variablen können durch den Treiber Online Import nur in Melderichtung (T01..T39) angelegt werden. Variablen für die Befehle (T45..T64) müssen Sie manuell anlegen, da im IEC 60870 Protokoll die Befehle *write-only* sind. Es ist nicht möglich, deren Adressierung von der Steuerung abzulesen.

Sie finden den Online Import in der Treiberliste im Kontextmenü des Treibers.

Zuerst muß ein Bauteil ausgewählt werden, von dem Sie die Variablen browsen wollen.

Dann wird ein „general interrogation command" (GI) an das ausgewählte Gerät gesendet und alle erhaltenen Variablen werden in einer Liste aufgezeigt.

Wenn die GI beendet ist, erhalten Sie den Dialog zur Auswahl der zu importierenden Variablen:



Durch Doppelklicken der Variablen bzw. durch Auswählen der Variable + Add Button wird die Variable in der zu importierenden Variablen hinzugefügt. Mit dem Button OK werden die ausgewählten Variablen in zenon angelegt. Die Adressierung wird aus dem Device übernommen.

5.5 Kommunikationsdetails (Treibervariablen)

Das Treiberkit implementiert eine Reihe von Treibervariablen, welche in dem Treiberobjekttyp *Kommunikationsdetails* zusammengefasst sind. Diese sind unterteilt in:

- Information
- Konfiguration
- Statistik und
- Fehlermeldungen

Die Definitionen der im Treiberkit implementierten Variablen sind in der Importdatei **DRVVAR.DBF** verfügbar und können von dort importiert werden.
Pfad zur Datei: *%ProgramData%\COPA-DATA\zenon<Versionsnummer>\PredefinedVariables*

Hinweis: Variablennamen müssen in zenon einzigartig sein. Soll nach einem Import der Variablen vom Treiberobjekttyp *Kommunikationsdetails* aus **DRVVAR.DBF** ein erneuter Import durchgeführt werden, müssen die zuvor importierten Variablen umbenannt werden.

Info

Nicht jeder Treiber unterstützt alle Treibervariablen des Treiberobjekttyps *Kommunikationsdetails*.

Zum Beispiel:

- Variablen für Modem-Informationen werden nur von modemfähigen Treibern unterstützt.
- Treibervariablen für den Polling-Zyklus stehen nur für rein pollende Treiber zur Verfügung.
- Verbindungsbezogene Informationen wie **ErrorMSG** werden nur von Treibern unterstützt, die zu einem Zeitpunkt nur eine Verbindung bearbeiten.

INFORMATION

Name aus Import	Typ	Offset	Erklärung
MainVersion	*UINT*	0	Haupt-Versionsnummer des Treibers.
SubVersion	*UINT*	1	Sub-Versionsnummer des Treibers.
BuildVersion	*UINT*	29	Build-Versionsnummer des Treibers.
RTMajor	*UINT*	49	zenon Hauptversionsnummer
RTMinor	*UINT*	50	zenon Sub-Versionsnummer
RTSp	*UINT*	51	zenon Service Pack-Nummer
RTBuild	*UINT*	52	zenon Buildnummer
LineStateIdle	*BOOL*	24.0	TRUE, wenn die Modemleitung belegt ist.
LineStateOffering	*BOOL*	24.1	TRUE, wenn ein Anruf rein kommt.
LineStateAccepted	*BOOL*	24.2	Der Anruf wird angenommen.
LineStateDialtone	*BOOL*	24.3	Rufton wurde erkannt.

Name aus Import	Typ	Offset	Erklärung
LineStateDialing	*BOOL*	24.4	Wahl aktiv.
LineStateRingBack	*BOOL*	24.5	Während Verbindungsaufbau.
LineStateBusy	*BOOL*	24.6	Zielstation besetzt.
LineStateSpecialInfo	*BOOL*	24.7	Spezielle Statusinformation empfangen.
LineStateConnected	*BOOL*	24.8	Verbindung hergestellt.
LineStateProceeding	*BOOL*	24.9	Wahl ausgeführt.
LineStateOnHold	*BOOL*	24.10	Verbindung in Halten.
LineStateConferenced	*BOOL*	24.11	Verbindung im Konferenzmodus.
LineStateOnHoldPendConf	*BOOL*	24.12	Verbindung in Halten für Konferenz.
LineStateOnHoldPendTransfer	*BOOL*	24.13	Verbindung in Halten für Transfer.
LineStateDisconnected	*BOOL*	24.14	Verbindung beendet.
LineStateUnknow	*BOOL*	24.15	Verbindungszustand nicht bekannt.
ModemStatus	*UDINT*	24	Aktueller Modemstatus.
TreiberStop	*BOOL*	28	Treiber gestoppt Bei *Treiberstop*, hat die Variable den Wert *TRUE* und ein **OFF**-Bit. Nach dem Treiberstart, hat die Variable den Wert *FALSE* und kein **OFF**-Bit.
SimulRTState	*UDINT*	60	Informiert über Status der Runtime bei Treibersimulation.
ConnectionStates	*STRING*	61	Interner Verbindungsstatus des Treibers zur SPS. Verbindungszustände: ▸ *0:* Verbindung OK ▸ *1:* Verbindung gestört ▸ *2:* Verbindung simuliert Formatierung: **<Netzadresse>:<Verbindungszustand>;...;...**

Name aus Import	Typ	Offset	Erklärung
			; Eine Verbindung ist erst nach dem ersten Anmelden einer Variablen bekannt. Damit eine Verbindung im String enthalten ist, muss einmal eine Variable dieser Verbindung angemeldet worden sein. Der Zustand einer Verbindung wird nur aktualisiert, wenn eine Variable der Verbindung angemeldet ist. Ansonsten wird nicht mit der entsprechenden Steuerung kommuniziert.

KONFIGURATION

Name aus Import	Typ	Offset	Erklärung
ReconnectInRead	*BOOL*	27	Wenn TRUE, dann wird beim Lesen automatisch ein Neuaufbau der Verbindung durchgeführt.
ApplyCom	*BOOL*	36	Änderungen an den Einstellungen der seriellen Schnittstelle zuweisen. Das Schreiben auf diese Variable hat unmittelbar den Aufruf der Methode SrvDrvVarApplyCom zur Folge (aktuell ohne weitere Funktion).
ApplyModem	*BOOL*	37	Änderungen an den Modemeinstellungen zuweisen. Das Schreiben auf diese Variable hat unmittelbar den Aufruf der Methode SrvDrvVarApplyModem zur Folge. Diese schließt die aktuelle Verbindung und öffnet eine neue entsprechend den Einstellungen **PhoneNumberSet** und **ModemHwAdrSet**.
PhoneNumberSet	*STRING*	38	Telefonnummer, welche verwendet werden soll.
ModemHwAdrSet	*DINT*	39	Hardwareadresse, welche zu der Telefonnummer gehört.
GlobalUpdate	*UDINT*	3	Updatezeit in Millisekunden (ms).
BGlobalUpdaten	*BOOL*	4	TRUE, wenn die Updatezeit global ist.

Name aus Import	Typ	Offset	Erklärung
TreiberSimul	*BOOL*	5	TRUE, wenn der Treiber in Simulation ist.
TreiberProzab	*BOOL*	6	TRUE, wenn das Prozessabbild gehalten werden soll.
ModemActive	*BOOL*	7	TRUE, wenn das Modem bei diesem Treiber aktiv ist.
Device	*STRING*	8	Name der seriellen Schnittstelle oder Name des Modem.
ComPort	*UINT*	9	Nummer der seriellen Schnittstelle.
Baudrate	*UDINT*	10	Baudrate der seriellen Schnittstelle.
Parity	*SINT*	11	Parität der seriellen Schnittstelle.
ByteSize	*USINT*	14	Bitanzahl pro Zeichen der seriellen Schnittstelle. Wert = *0*, wenn der Treiber keine serielle Kommunikation herstellen kann.
StopBit	*USINT*	13	Anzahl der Stoppbits der seriellen Schnittstelle.
Autoconnect	*BOOL*	16	TRUE, wenn die Modemverbindung automatisch beim Lesen/Schreiben aufgebaut werden soll.
PhoneNumber	*STRING*	17	Aktuelle Telefonnummer.
ModemHwAdr	*DINT*	21	Hardwareadresse zur aktuellen Telefonnummer.
RxIdleTime	*UINT*	18	Wenn länger als diese Zeit in Sekunden (s) erfolgreich kein Datenverkehr stattfindet, wird die Modemverbindung beendet.
WriteTimeout	*UDINT*	19	Maximale Schreibdauer bei einer Modemverbindung in Millisekunden (ms).
RingCountSet	*UDINT*	20	So oft läutet ein hereinkommender Anruf, bevor dieser angenommen wird.
ReCallIdleTime	*UINT*	53	Wartezeit zwischen Anrufen in Sekunden (s).
ConnectTimeout	*UINT*	54	Zeit in Sekunden (s) für Verbindungsaufbau.

STATISTIK

Name aus Import	Typ	Offset	Erklärung
MaxWriteTime	*UDINT*	31	Längste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MinWriteTime	*UDINT*	32	Kürzeste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MaxBlkReadTime	*UDINT*	40	Längste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
MinBlkReadTime	*UDINT*	41	Kürzeste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
WriteErrorCount	*UDINT*	33	Anzahl der Schreibfehler.
ReadSucceedCount	*UDINT*	35	Anzahl der erfolgreichen Leseversuche.
MaxCycleTime	*UDINT*	22	Längste Zeit in Millisekunden (ms), die zum Lesen aller angeforderten Daten benötigt wurde.
MinCycleTime	*UDINT*	23	Kürzeste Zeit in Millisekunden (ms), die zum Lesen aller angeforderten Daten benötigt wurde.
WriteCount	*UDINT*	26	Anzahl der Schreibversuche.
ReadErrorCount	*UDINT*	34	Anzahl der fehlerhaften Leseversuche.
MaxUpdateTimeNormal	*UDINT*	56	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Normal** in Millisekunden (ms).
MaxUpdateTimeHigher	*UDINT*	57	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Höher** in Millisekunden (ms).
MaxUpdateTimeHigh	*UDINT*	58	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Hoch** in Millisekunden (ms).
MaxUpdateTimeHighest	*UDINT*	59	Zeit seit letzter Aktualisierung der Prioritätsgruppe **Höchste** in Millisekunden (ms).
PokeFinish	*BOOL*	55	Geht für eine Abfrage auf *1*, wenn alle anstehenden Pokes ausgeführt wurden.

FEHLERMELDUNGEN

Name aus Import	Typ	Offset	Erklärung
ErrorTimeDW	*UDINT*	2	Zeit (in Sekunden seit 1.1.1970), wann der letzte

Name aus Import	Typ	Offset	Erklärung
			Fehler auftrat.
ErrorTimeS	*STRING*	2	Zeit (in Sekunden seit 1.1.1970), wann der letzte Fehler als String auftrat.
RdErrPrimObj	*UDINT*	42	Nummer des PrimObjektes, als der letzte Lesefehler verursacht wurde.
RdErrStationsName	*STRING*	43	Name der Station, als der letzte Lesefehler verursacht wurde.
RdErrBlockCount	*UINT*	44	Anzahl der zu lesenden Blöcke, als der letzte Lesefehler verursacht wurde.
RdErrHwAdresse	*DINT*	45	Hardwareadresse, als der letzte Lesefehler verursacht wurde.
RdErrDatablockNo	*UDINT*	46	Bausteinnummer, als der letzte Lesefehler verursacht wurde.
RdErrMarkerNo	*UDINT*	47	Merkernummer, als der letzte Lesefehler verursacht wurde.
RdErrSize	*UDINT*	48	Blockgröße, als der letzte Lesefehler verursacht wurde.
DrvError	*USINT*	25	Fehlermeldung als Nummer.
DrvErrorMsg	*STRING*	30	Fehlermeldung als Klartext.
ErrorFile	*STRING*	15	Name der Fehlerprotokolldatei.

6 Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:

WERTÄNDERUNGEN

Die Kommunikation mit Steuerungen ist Spontan. Das heißt alle Wertänderungen der Steuerung werden vom Treiber verarbeitet. Die allgemeine Einstellungen eines Treibers: **Update Zeit Global** und **Priorität** beeinflussen der IEC870 Treiber nicht.

Beispiel:
Die Steuerung sendet drei Wertänderungen im Abstand von 5 ms, diese Änderungen werden alle an zenon weitergegeben.
Legt man eine **Reaktionsmatrix** an, die auf jede Wertänderung reagiert und schreibt damit einen Eintrag in die **Chronologische Ereignis Liste (CEL)**, so werden in der **CEL** drei Einträge mit Abstand von 5 ms erstellt. Gleiches gilt auch für spontane Archivierung, oder Alarme etc.; es gehen keine Werte verloren.

GENERALABFRAGE

Wie im IEC 60870 Standard vorgesehen, sendet der Treiber unmittelbar nach dem Verbindungsaufbau automatisch die Generalabfrage (General Interrogation - *C_IC_NA_1*) an die Steuerungen. Der Treiber sendet die initiale Generalabfragen für alle Sektoren (**COA)** die in der Treiberkonfiguration definiert wurden.

Wird auf einem Sektor (**COA**), während einer laufenden Generalabfrage (COT_actterm noch nicht empfangen), ein End of Initialization (*M_EI_NA_1*) empfangen, wird die GI-Anforderung unterbrochen und erneut gesendet. Ist die initiale Generalabfrage noch nicht für den **COA** abgeschlossen, wird der Treiber jeden Versuch einen *T45..T64* Befehl auszulösen, verweigern. Die Fehlermeldung dazu im LOG ist „Poke to unconnected device". Erst wenn die initiale GI-Anforderung abgeschlossen wurde, wird ein "Sollwert-setzen-Befehl auf eine *T45..T64* Variable in dem Sektor", vom Treiber zum Slave ausgelöst.

Hinweis: Werden die initialen GI-Anforderungen abgeschlossen, setzt der Treiber der Wert der Kommunikationssstatus Variable (*T00*, *COA 0*, *IOA 0*) auf *5*.

Es kann nachträglich erneut eine Generalabfrage manuell ausgelöst werden. Dazu ist eine BOOL-Variable vom **Typ ID** *T100* zu beschreiben. Die **IOA** dieser Variablen muss *0* sein. Während die Generalabfrage läuft, ist der Wert dieser Variable *1*. Konnte die Generalabfrage nicht korrekt ausgeführt werden, so erhält diese Variable den Status *INVALID*. Variablen vom Typ *T100* können für jeden Sektor (**COA**) separat angelegt werden. Somit ist es möglich, die Erreichbarkeit der einzelnen Sektoren eines Slaves (**Device**) zu überprüfen.

KOMMUNIKATIONSSTATUS (APPLICATION LAYER)

Über eine *USINT*-Variable vom **Typ ID** *T00 internal state* und **COA** *0* und **IOA** *0* kann der aktuelle Kommunikationstatus zum 870-Slave ausgewertet werden. Ist der Wert dieser Variable *5*, dann besteht eine laufende, gültige IEC 60870 Application Layer Verbindung und die Generalabfrage wurde erfolgreich abgeschlossen.

Info

Indizien für eine ungültige Verbindung:

Der Wert des Verbindungsstatus erreicht auch nach längerer Zeit nicht den Wert 5 obwohl:

- eine Verbindung zum Treiber aufgebaut ist.
- Daten ausgetauscht werden und eine Kommunikation stattfindet.

Das kann zum Beispiel passieren, wenn Sektoren in der Treiberkonfiguration konfiguriert sind, die im Slave nicht vorhanden sind.

Hinweis: Wenn eine Verbindung hergestellt ist und kommuniziert wird, jedoch der Wert des Verbindungsstatus auch nach längere Zeit nicht den Wert 5 erreicht, kann das auf ein Kommunikationsproblem hindeuten.

Hinweis: Der Application Layer des Treibers ist sowohl für IEC 60870-5-101 wie auch für die IEC 60870-5-104 Kommunikation gleich.

TCP/IP VERBINDUNG (IEC 60870-5-104 DATA LAYER)

Der Treiber untersützt als IEC 60870-5-104 Master auch die Verbindungen zu 870-Slaves über duale Ethernet-Netze. Dies wird im Konfigurationsdialog des Treibers in der Option für **Secondary connection** konfiguriert. Wurde die **Secondary connection** konfiguriert, so kann der Treiber zusätzlich die Redundanz des Data Layers nach die Regeln der Edition 2 des IEC 60870-5-104 Standards verwenden. Dies wird optional mit der Check-Box **Redundancy according to 870-104 ed. 2.0** aktiviert.

Bei Verbindungsverlust führt der Treiber alle benötigte Prozeduren für die Wiederverbindungen oder der Umschaltung zwischen **Primary** und **Secondary connections** automatisch aus.

Beispiel:

Ohne die Redundanz nach Edition 2:
Wird die Verbindung via **Primary** unterbrochen, bekommen alle Variablen (Meldungen und Befehle) das Statusbit *INVALID*. Im Hintergrund versucht der Treiber automatisch diese Verbindung erneut aufzubauen. Ist dies nicht möglich, versucht der Treiber eine Verbindung über die **Secondary connection** mit der Steuerung aufzubauen. Bei erfolgreicher Verbindung sendet der Treiber die Generalabfrage (*C_IC_NA_1*). Nach erfolgreichem Verbindungsaufbau oder Wiederaufbau werden die Werte der Variablen der Meldungen (*T01-T37*) mit Statusbit *SPONT* oder *GI* gekennzeichnet. Die Variable jener Befehle, die ausschließlich *write-only* sind, bleiben unverändert (Wert 0, *INVALID*) bis jeweils ein Wert gesetzt (= ein Befehl gegeben) wird.

- Hat der Treiber erfolgreich auf die **Secondary connection** umgeschaltet, wird diese Verbindung zur Kommunikation benutzt, bis diese Verbindung abbricht. Bei nächsten

Verbindungsabbruch wird erneut automatisch versucht auf die **Primary connection** umzuschalten.

- Ist der 870-Slave über beide Kommunikationswege nicht erreichbar versucht der Treiber weiter alle 30 Sekunden eine Verbindung aufzubauen. Dabei werden abwechselnd die **Primary connection** und **Secondary connection** für den Verbindungsauffbau angewendet.

STATUS DES DL-LAYERS FÜR TCP/IP

Mithilfe einer Variablen vom **Typ ID** *T00 internal state* und **COA** *0* und **IOA** *1* kann der Status der Datensicherungsschicht-Verbindung (engl. Data Link Layer) zu einer IEC60870-5-104 Steuerung ausgewertet werden. Die Variable gibt Auskunft darüber, über welcher der beiden IPs der Treiber aktuell kommuniziert. Im Fall der Edition 2 Redundanz wird zusätzlich die Information über die Verfügbarkeit den Data Link Layer bereitgestellt.

Bit	Beschreibung
Bit 0	Verbindungsstatus der Primärverbindung IP
Bit 1	Zeigt an, ob die Primärverbindung aktiv ist (wird bei der Kommunikation verwendet)
Bit 4	Verbindungsstatus der Sekundärverbindung IP
Bit 5	Zeigt an, ob die Sekundärverbindung aktiv ist (wird bei der Kommunikation verwendet)

Dem IEC 60870-5-104 Standard folgend kann für die Editon 2 Redundanz die Verfügbarkeit beider Data Link Layers überprüft werden. Dazu sendet der Treiber im Zeitabstand des Timeouts **T3** (Defaultwert: *20 s*) die **APCI** Frames TESTFR act. Als Bestätigung wird vom Slave ein TESTFR con erwartet. Bleibt das TESTFR Frame unbestätigt für Timeout **T1** (Defaultwert *15s*), so wird entsprechend das Bit *0* o. *4* der Status DL-Layers Variable zurückgesetzt. Handelt sich dabei um aktuell aktive IP Verbindung, löst es die Umschaltung der IP - der Treiber sendet ein STARTDT act auf die andere IP.

Hinweis: Für die redundante Verbindungen über Festnetze schlägt die Edition 2 des Standards vor, das **T3** Timeout zu verkürzen.

MANUELLES UMSCHALTEN DER TCP/IP VERBINDUNG

Info

Steht nur zur Verfügung, wenn Sie in der Treibers die Option **Redundancy according to 60870-104 edition 2.0** aktiviert haben.

Mithilfe des treibspezifischen Kommandos **104_MANUAL_SWITCH** *'net address'* lösen Sie das manuelle Umschalten auf die redundante Verbindung aus. Der Wert *'net address'* steht dabei für die entsprechenden Netzadresse des Slaves.

Beispiel: legen Sie zenon Funktion **Treiberkommandos** an und wählen Sie in der Eigenschaft **Treiberkomando** die Option *Treiberspezifisches Kommando*. Dies öffnet zusätzliches Eingabefeld **Treiberspezifisches Kommando**. Schreiben Sie in dem Eingabefeld das Text: *104_MANUAL_SWITCH 1*.

Wird in der Runtime die Funktion aufgerufen, z.B. über ein **Button**, so wird der Treiber für die Verbindung, die in seiner Konfiguration die **Netzadresse** *1* hat, zwischen beiden redundanten IP Adressen umschalten. Auf der aktuell aktiven IP Verbindung zum Slave mit Netzadresse *1* sendet der Treiber ein APCI Frame mit STOPDT act und auf dem bisher passiven IP - ein STARTDT act.

FILETRANSFER

Für den Filetransfer sind drei Funktionen implementiert:

1. Ordnerinformation anfordern *(DIR)*
2. Datei von der Steuerung holen *(GET)*
3. Datei zur Steuerung schicken *(PUT)*
4. Datei auf der Steuerung löschen *(DEL)*

1. ORDNERINFORMATION ANFORDERN (DIR)

Um die Ordnerinformation anzufordern:

1. erzeugen Sie in Ihrem zenon Projekt zwei Stringvariablen:
   a) Die erste Variable ist vom Typ „**call directory, call file**" (T122) (im weiteren Verlauf als Kommandovariable bezeichnet). Sie kann auch dazu benutzt werden, Dateien zu holen und zu löschen.
   a) Die zweite Variable vom Typ **„directory**" (T126) wird nur für das Ergebnis der Ordneranfrage benutzt. Sie empfängt den Ordnerinhalt als lesbaren Text. Deshalb sollte ihre Größe (Stringlänge) der maximalen Größe des Dateinamens (inklusive des Ordners) in der Steuerung entsprechen.
2. setzen Sie für die Kommandovariable (T122) den Wert "*DIR*" (für das Root-Directory) bzw. "*DIR <IOA>.<NOF>*"

Wenn der Ordner erfolgreich empfangen wurde:

- ändert die Kommandovariable ihren Wert auf "*DIR OK*"
- enthält die Ordnervariable (T126) den empfangenen Ordnerinhalt.
  Eine Zeile dieses Texts hat das folgende Format:
  <IOA>.<NOF>;<Dateilänge>;<Zeitstempel>;SOF

2. DATEI VON DER STEUERUNG HOLEN (GET)

Um eine Datei von der Steuerung zu holen, setzen Sie bei der Kommandovariable (T122) den Wert "*GET <IOA>.<NOF>*".

Wenn die Datei erfolgreich empfangen wurde:

- wird sie in dem Ordner gespeichert, der als **Directory for file transfer** im Treiberdialog Basic settings (auf Seite 17) definiert wurde
- ändert die Kommandovariable ihren Wert auf "*GET OK*"
- wird für jede COA ein Unterordner angelegt

Ein aktiver File Transfer kann durch Absetzen des Wertes "*CANCEL*" auf die Kommandovariable (T122) abgebrochen werden. Somit erwartet der Treiber keine weiteren Segmente der Datei und wird auch keine weitere Sektion der Datei anfordern. Eine Deaktivierung wird an die Steuerung aber nicht gesendet. Die Datei, deren Übertragung abgebrochen wurde, wird nicht gespeichert.

Beispiel

Holen der Datei mit IOA 1100 und NOF 'transparent' (1) von Sektor 151, wenn der Ordner in der Treiberkonfiguration als "C:\TEMP\IEC870" definiert wurde:

- Senden des Sollwertes „*GET 1100.1*" auf eine Variable vom Typ T122, COA 151 und IOA 0.
- Die Datei wird in *C:\TEMP\IEC870\151\1100.1* abgelegt.

3. DATEI ZUR STEUERUNG SCHICKEN (PUT)

Um eine Datei zur Steuerung zu schicken, setzen Sie bei der Kommandovariable (T122) den Wert "*PUT <IOA>.<NOF>*".

Es wird eine Datei aus dem Ordner COA versendet. Dieser Ordner ist ein Unterverzeichnis jenes Verzeichnisses, welches im Treiberdialog Basic settings (auf Seite 17), im Eingabefeld **Directory for file transfer in Control Direction** definiert wurde.
Hinweis: Dieser Ordner muss vom Benutzer manuell angelegt werden.

Wenn die Datei erfolgreich gesendet wurde:

- ändert die Kommandovariable ihren Wert auf "*PUT OK*"

Wenn der Transfer läuft wird „*PUT BUSY*" angezeigt. Ein aktiver File Transfer kann durch Absetzen des Wertes "*CANCEL*" auf die Kommandovariable (T122) abgebrochen werden.

4. DATEI AUF DER STEUERUNG LÖSCHEN (DEL)

Um eine Datei auf der Steuerung zu löschen, setzen Sie bei der Kommandovariablen (T122) den Wert "*DEL <IOA>.<NOF>*".

Wenn die Datei erfolgreich gelöscht wurde, ändert die Kommandovariable ihren Wert auf "*DEL OK*"

FEHLERBEHANDLUNG

Wenn bei der Ausführung des Dateientransfers Fehler auftreten, ändert die Kommandovariable ihren Wert auf "*XXX ERROR*" (XXX = DIR, GET, PUT oder DEL) und der Treiber schreibt optional einen Eintrag in die Log-Datei (siehe auch Fehleranalyse (auf Seite 90)).

UHRZEITSYNCHRONISATION

Durch Beschreiben einer Variable vom Typ ID C_CS_NA_1 (*T103*) wird die aktuelle Uhrzeit des PCs zur Steuerung gesendet. Die **IOA** dieser Variablen muss *0* sein.

Uhrzeitsynchronisation in "monitoring direction":

Der Treiber bietet die Möglichkeit, ein T103 Telegramm in Monitoring Direction - vom Device - anzunehmen und auszuwerten. Wird vom Device ein ASDU<103> mit COT=3 gesendet, wird am PC die Uhrzeit vom Device übernommen jedoch nur, wenn die Device-Zeit nicht mehr als die eingestellte **Maximaldifferenz** (siehe TreiberDialog Basic Setting) von der PC Zeit abweicht.

STATUSBIT WR-SUC (BIT 41)

Wenn für eine **Sollwert setzen** bzw. **Rezept ausführen** Aktion oder eine Befehlsgabe eine Schreibbestätigung angefordert wurde (WR-ACK), wird bereits nach dem Versenden des Kommandos (**COT_act**) dieses Statusbit gesetzt. Es wird nicht auf **COT_actcon** gewartet.

STATUSBIT BL_870 (BIT 44)

Signalisiert IEC Status *blocked*. Der Wert ist für die Übertragung blockiert und bleibt in dem Status, den er inne hatte, bevor er blockiert wurde. Dieses Statusbit kann in Multireaktionsmatrizen, im Combi-Element und in der Verriegelungsformel ausgewählt werden.

In VBA können mit `StatusExtValue()` die oberen 32-Bits abgefragt werden, bei `SetValueWithStatusEx()` können alle 64 Statusbits mitgegeben werden.

STATUSBIT SB_870 (BIT 45)

Signalisiert IEC Status *substituted*. Der Wert wurde durch einen Bediener oder eine automatisierte Quelle gesetzt. Dieses Statusbit kann in Multireaktionsmatrizen, im Combi-Element und in der Verriegelungsformel ausgewählt werden.

In VBA können mit `StatusExtValue()` die oberen 32-Bits abgefragt werden, bei `SetValueWithStatusEx()` können alle 64 Statusbits mitgegeben werden.

STATUSBIT NT_870 (BIT 46)

Signalisiert IEC Status *not topical*. Der Wert wurde während einer bestimmten Zeitspanne nicht erfolgreich upgedatet oder ist nicht verfügbar. Dieses Statusbit kann in Multireaktionsmatrizen, im Combi-Element und in der Verriegelungsformel ausgewählt werden.

In VBA können mit `StatusExtValue()` die oberen 32-Bits abgefragt werden, bei `SetValueWithStatusEx()` können alle 64 Statusbits mitgegeben werden.

STATUSBIT OV_870 (BIT 47)

Signalisiert *Overflow*. Der Wert liegt außerhalb der vordefinierten Bandbreite.

STATUSBIT SE_870 (BIT 48)

Entspricht IEC Qualifier S/E und wird in Verbindung mit der **Select before operate** Funktionalität verwendet: dient zur Unterscheidung zwischen *Select*- und *Execute*-Zustand eines Befehls.

STATUSBITS COTX (BITS 32..37)

Der empfangene Wert der Übertragungsursache ("Cause of Transmission" - siehe *IEC60870-5-101 7.2.3*) wird auf die Statusbits *COTx* abgebildet.

STATUSBIT T_INVAL (BIT 49)

Das Statusbit T_INVAL (Echtzeit ungültig) wird vom IEC870 Treiber gesetzt, wenn der empfangene Echtzeitstempel als ungültig markiert ist. In diesem Fall wird mit der lokalen PC-Zeit gestempelt.

Hinweis: Im IEC870Slave des Process-Gateways wird dieses Statusbit in Melderichtung im Zeitstempel weitergeleitet.

ASDUS MIT "TIME TAG CP24TIME2A"

Die Zeitinformation vom Typ **CP24Time2a** beinhaltet nur Minuten, Sekunden und Millisekunden. Datumsinformationen (Jahr, Monat, Tag) und Informationen zur Stunde werden nicht übertragen. Der Treiber greift auf die PC-Uhr zurück, um die fehlenden Zeitinformationen (Jahr, Monat, Tag und Stunde) zu ergänzen. Wird ein **CP24Time2a** mit einem Unterschied von mehr als plus/minus 30 Minuten empfangen, korrigiert der Treiber den Zeitstempel automatisch um eine Stunde plus oder minus. Dabei wird auch ein eventueller Datumswechsel berücksichtigt.

Hinweis: Dies beeinflusst nicht die Zeitstempel vom Typ **CP56Time2a**.

HYSTERESE HANDLING

Der Treiber unterstützt im Allgemeinen auch Hysterese. Die Hysterese wird nur für numerischen Datentypen berücksichtigt, wenn Sie Hysterese<>0 projektiert haben. Für Variablen mit Hysterese=0 und für Variablen vom Typ BOOL oder STRING werden vom Treiber alle empfangenen Werte an die Runtime gesendet.

SELECT & EXECUTE

Um „select & execute" nutzen zu können, muss eine B**efehlsgabe** ausgeführt werden. Zusätzlich muss bei der Befehlsvariablen die Eigenschaft **Select Before Operate** aktiviert werden. Bei erste Stufe der Aktion sendet der Treiber ein *Select* Befehl - ein ASDU mit IEC Qualifier S/E gleich 1 (*select*). Auch das Statusbit *SE_870* der Befehlsvariable wird gesetzt. Dann folgt ein *Execute* Befehl - ein ASDU mit IEC Qualifier S/E gleich 0 (*execute*), bzw. ein *Cancel* Befehl - ein ASDU mit *COT 8*. Ein *Cancel* (*deactivation*) kann nur aus zweistufigen Aktion ausgelöst werden - wenn während der Ausführung der Aktion, der Benutzer die Befehlsgabe abbricht, statt zu bestätigen.

Wird der Wert direkt gesetzt, also nicht über eine Aktion der Befehlsgabe, dann:

- ignoriert der Treiber die Eigenschaft **Select Before Operate** der Befehlsvariable
- verwendet der Treiber "*direct execute*" und schickt den Befehl mit dem IEC Qualifier S/E = 0 (*execute*).

EINSCHRÄNKUNGEN

- Siehe IEC60870-5-101/104 Interoperabilitätsliste für Informationen zu unterstützten Kommunikationsparametern und Typenbezeichnung.
- „*select and execute*" kann nur bei Verwendung der Befehlsgabe genutzt werden. Ansonsten wird immer „*direct execute*" verwendet.

6.1 Mapping von Double Point Values

Double Point Value Mapping ist eine Standardfunktion der zenon Energy Treiber. Es beeinflusst nur die zenon Runtime und hat keine Auswirkung auf die Treiberkommunikationen mit einem Gerät. Die Konfiguration erfolgt in den Treibereinstellungen in der Registerkarte **Basic Settings**.

Hinweis: Es wird empfohlen, die Option **Deactivate standard double point value mapping** in der Treiberkonfiguration default auf inaktiv zu belassen.

Mittels Double Point Value Mapping wandelt der Treiber Werte so um, dass sie benutzerfreundlich dargestellt werden. Das gilt allerdings nur für HMI.

Der Treiber kommuniziert mit einem Gerät immer mit Werten für Double Points (Doppel-Punkte) mit 2-Bit-Informationen. Dies entspricht den Definitionen der Energie-Standards. Das bedeutet:

Parameter	Double Point	Wert	Bedeutung
Intermediate	*00b*	*0*	Schalter weder offen noch geschlossen, z. B. die End-Position noch nicht erreicht.
Off	*01b*	*1*	Schalter offen
On	*10b*	*2*	Schalter schließen/geschlossen
Fault	*11b*	*3*	Fehler

Double Points werden im Energiesektor aus historischen Gründen mit 2 Bits kodiert: Die Übertragung eines Telegramms auf einer serielle Verbindung (RS232) mit einer Serie von Werten, die nur *0* enthalten, war nicht gegen Übertragungsfehler gesichert. Um die Sicherheit zu erhöhen, wurde in den ersten Standards beschlossen, den Wert für *OFF* nicht als *0* zu senden, sondern als *01b*, was dezimal *1* entspricht. Diese Double Point Values spiegeln auch exakt die Art wieder, wie zwei Sensoren die physikalische Position eines Schalters erfassen.

Allerdings sind die damit übermittelten Werte für Menschen eventuell verwirrend:

- *OFF = 1*
- *ON = 2*

Menschen sind von allen anderen Geräten und Systemen gewohnt:

- *OFF = 0*
- *ON = 1*

  Gleichzeitig werden im gleichen Standard die Single Point Values mit *OFF = 0* und *ON = 1* definiert.

Um gefährliche Fehlhandlungen des Benutzers zu vermeiden, bieten die zenon Energy-Treiber ein eigenes Double Point Value Mapping. Ohne DPI Mapping musste der Benutzer sich immer bewusst sein, auf welcher technischen Ebene er gerade agiert und Informationen erhält bzw. übermittelt. In Stress-Situationen kann das sehr leicht zu schwerwiegenden Fehlhandlungen führen; zum Beispiel, wenn *ON* statt *OFF* gesendet wird.

MAPPING FÜR HMI

Mit dem Double Point Value Mapping haben in der zenon Runtime alle Double Points folgende Werte:

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

> **Info**
>
> Diese Funktion kann in den Treibereinstellungen deaktiviert werden. Allerdings können dann einige Funktionalitäten wie Befehlsgabe oder ALC nicht mehr verwendet werden.
>
> **Empfehlung:** Verzichten Sie auf Numerische Elemente und Zahlenwerte, um *OFF/ON* oder *OPEN/CLOSE* anzuzeigen. Verwenden Sie Combi Elemente mit grafischen Symbolen oder Text-Elemente.

DCS MAPPING IN DOUBLE COMMAND

Generell gilt: Wird ein Sollwert auf einer Befehlsvariable gesetzt, so wird der Wert zur SPS geschrieben. Normalerweise wird dabei der Wert 1:1 zur Steuerung geschrieben.

Ist diese Befehlsvariable jedoch:

- von Type ID T46 oder T59
- und ist der Wert <3

wird zur Steuerung gemappt kommuniziert:

Da DCS aus zwei Bits besteht, werden alle höheren Werte abgeschnitten und nicht gesendet. Das bedeutet:

zenon Wert	DCS Wert	Bedeutung
0 = 00b	*01b =* **1***	**Off**
1 = 01b	*10b =* **2***	**On**
2 = 10b	*00b =* **0***	**Intermediate**
3 = 11b	*11b = 3*	**Fault**
4 = 100b	*00b = 0*	**Intermediate**
5 = 101b	*01b = 1*	**Off**
6 = 110b	*10b = 2*	**On**
7 = 111b	*11b = 3*	**Fault**

* gemappter Wert

Werte *>4* werden ungemappt zur Steuerung kommuniziert.

6.2 TLS-verschlüsselte Kommunikation

Eine verschlüsselte Kommunikation des IEC870 Treibers mit der PLC wird in Konfigurations-Datei des Treibers konfiguriert. Diese Konfigurationsdatei liegt per Default im Projektordner im Unterordner *\zenon\custom\drivers*.

Der Ablageort und die Benennung der Konfigurationsdatei kann auch im Treiberdialog in der Registerkarte **Basic settings** (auf Seite 17) in der Option **Configuration File name** konfiguriert werden.

Tipp: Sie können im zenon Editor mit der Tastenkombination **Strg + Alt + E** in den Projektordner wechseln.

Diese Konfiguration muss folgende Einträge enthalten:

***** LINK *****

Die Parameter einer abgesicherten Verbindung können für jede Verbindung extra konfiguriert werden.

Parameter	Beschreibung
[TLS aktiviert]	Art der abgesicherten Kommunikation. ▸ *0: deaktiviert* Keine verschlüsselte Kommunikation. ▸ *1: aktiviert* Kommunikation erfolgt verschlüsselt, auf Basis der folgenden Parameter. Default: *0*
[Liste Verschlüsselungsverfahren]	Liste der unterstützten Verschlüsselungsverfahren. Die Liste enthält die Kürzel im openSSL-Format.
[Zertifikat]	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.
[Zertifikat Identifizierung]	Verwendete Zertifikats-Identifizierung des Clients. Das Client-Zertifikat muss das hier konfigurierte **SUBJECT** enthalten. Bei Nicht-Übereinstimmung wird die Verbindung abgebrochen.

Parameter	Beschreibung
[Zeit bis zur Erneuerung des Schlüssels]	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*
[Datenmenge bis zur Erneuerung des Schlüssels]	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*
[maximale Dauer einer verschlüsselten Verbindung]	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*
[maximale übertragene Datenmenge einer verschlüsselten Verbindung]	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*
[Überprüfungsintervall der REVOCATION LIST]	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.

Parameter	Beschreibung
	Default: *21600*

***** TLS CONTEXT *****

Die Konfiguration des Ablageortes der Zertifikate gelten für den Treiber und können nicht einzeln pro Verbindung projektiert werden.

Parameter	Beschreibung
[Ablageort der Zertifikate]	Absoluter Ablageort der Zertifikate. Dieses Basisverzeichnis muss die beiden Unterverzeichnisse **CA** und **PRIVATE** enthalten. ▸ **CA** für vertrauenswürdige Zertifikate und ▸ **PRIVATE** für eigene Zertifikate

BEISPIEL

1

*** LINK ***

29

1

Link1

0

192.168.0.230

2404

1

1

2

2

3

0.0.0.0

0

15000

10000

20000

12

8

0

0

0

0 [TLS aktiviert]

[Liste Verschlüsselungsverfahren]

[Zertifikat]

[Zertifikat Identifizierung]

0 [Zeit bis zur Erneuerung des Schlüssels] in Sekunden

0 [Datenmenge bis zur Erneuerung des Schlüssels]

0 [maximale Dauer einer verschlüsselten Verbindung]

0 [maximale übertragene Datenmenge einer verschlüsselten Verbindung]

0 [Überprüfungsintervall der REVOCATION LIST] in Sekunden

1

*** SECTOR ***

2

Sector1

1

1

*** TLS CONTEXT ***

1

C:\Certificates [Ablageort der Zertifikate]

7 Funktion Treiberkommandos

Die zenon Funktion **Treiberkommandos** dient dazu, Treiber über zenon zu beeinflussen.
Mit einem Treiberkommando können Sie einen Treiber:

- starten
- stoppen
- in einen bestimmten Treibermodus versetzen
- zu bestimmten Aktionen veranlassen

Hinweis: Dieses Kapitel beschreibt Standardfunktionalitäten, die für die meisten zenon Treiber gültig sind.
Nicht alle hier beschriebenen Funktionalitäten stehen für jeden Treiber zur Verfügung. Zum Beispiel enthält ein Treiber, der laut Datenblatt keine Modemverbindung unterstützt, auch keine Modem-Funktionalitäten.

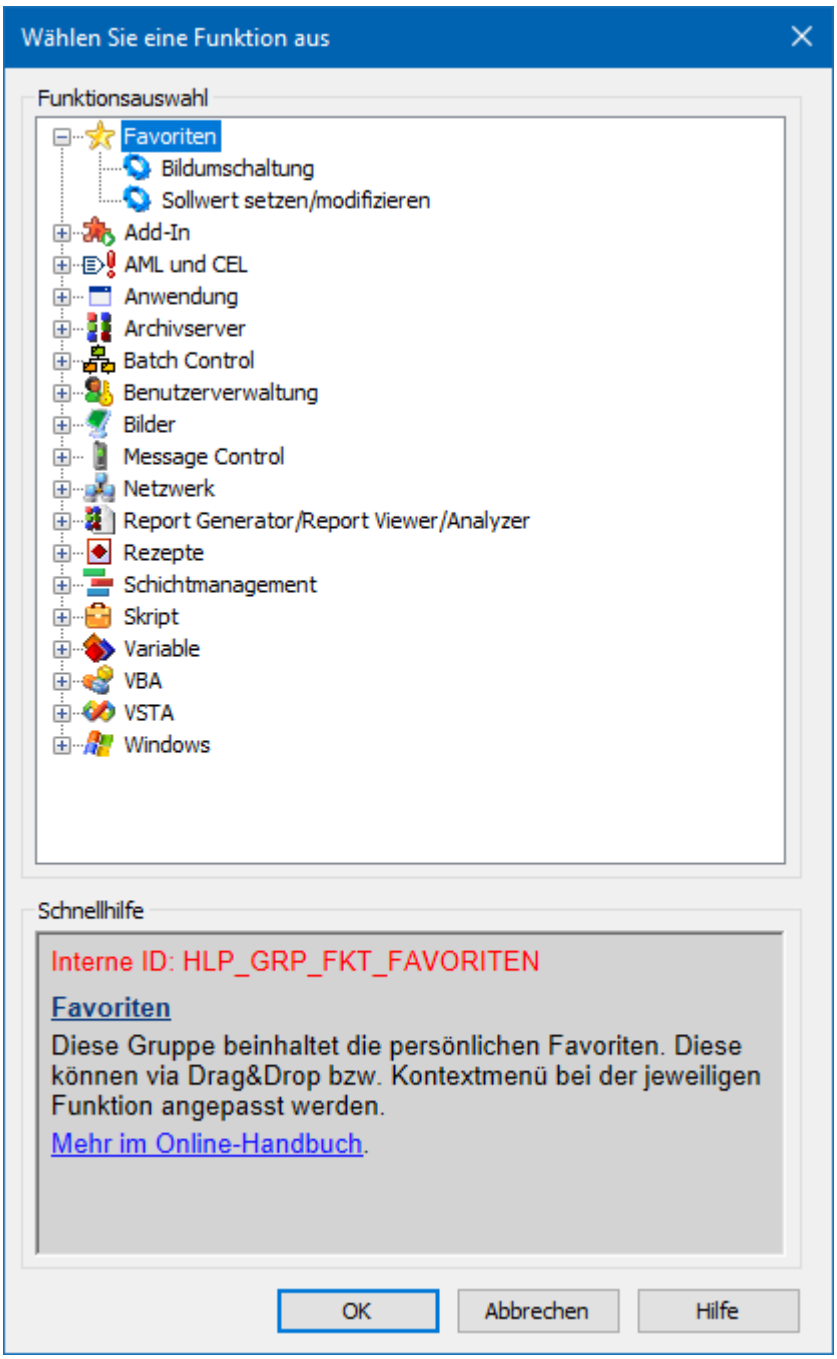
> **Achtung**
>
> Die zenon Funktion **Treiberkommandos** ist nicht ident mit den Treiberkommandos, die bei Energy-Treibern in der Runtime ausgeführt werden können!

PROJEKTIERUNG DER FUNKTION

Die Projektierung erfolgt über die Funktion **Treiberkommandos**.
Um die Funktion zu projektieren:

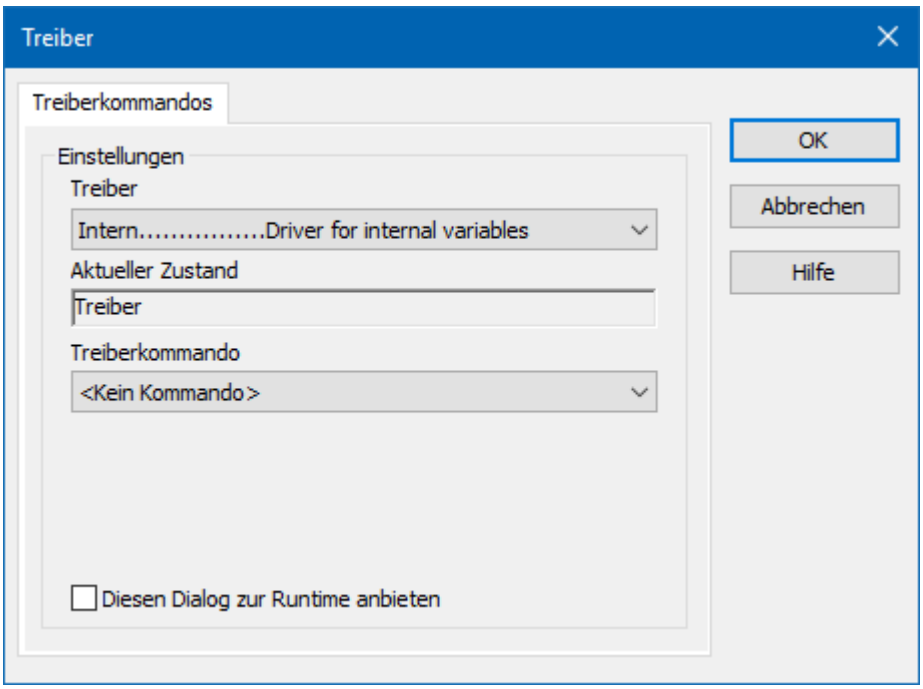
1. Legen Sie im zenon Editor eine neue Funktion an.

Der Dialog zur Auswahl einer Funktion wird geöffnet.



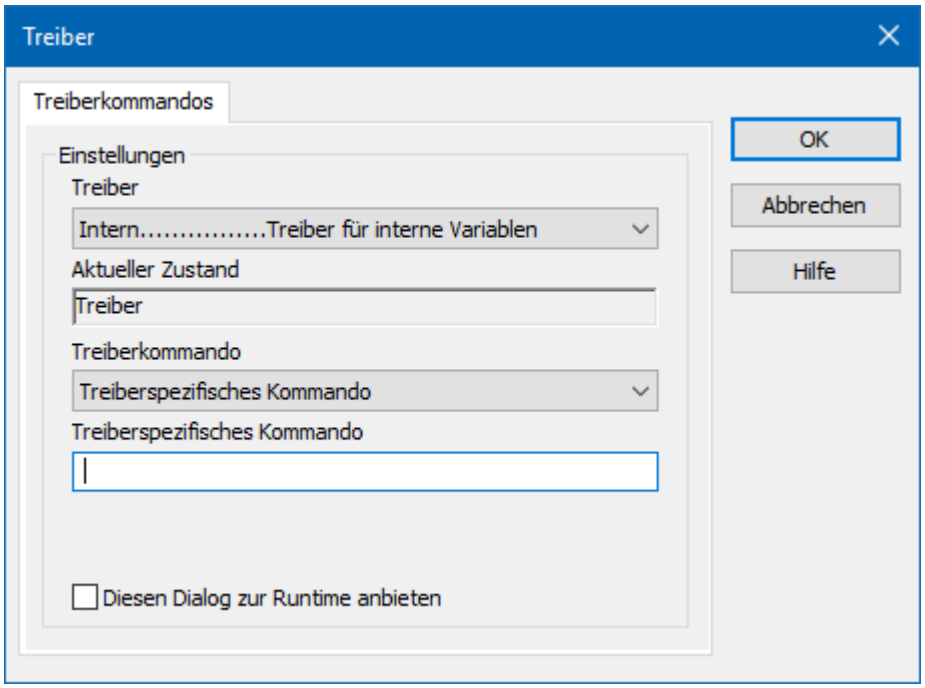
2. Navigieren Sie zum Knoten **Variable.**
3. Wählen Sie den Eintrag **Treiberkommandos**.

Der Dialog zur Konfiguration wird geöffnet.



4. Wählen Sie den gewünschten Treiber und das benötigte Kommando aus.
5. Schließen Sie den Dialog mit Klick auf **OK** und stellen Sie sicher, dass die Funktion in der Runtime ausgeführt wird.
   Beachten Sie die Hinweise im Abschnitt **Funktion Treiberkommandos im Netzwerk**.

DIALOG TREIBERKOMMANDOS



Option	Beschreibung
Treiber	Auswahl des Treibers aus der Dropdownliste. Diese enthält alle im Projekt geladenen Treibern.
Aktueller Zustand	Fixer Eintrag, der vom System gesetzt wird. In aktuellen Versionen ohne Funktion.
Treiberkommando	Auswahl des gewünschten Treiberkommandos aus Dropdownliste. Details zu den konfigurierbaren Treiberkommandos siehe Abschnitt **Verfügbare Treiberkommandos**.
Treiberspezifisches Kommando	Eingabe eines für den gewählten Treiber spezifischen Kommandos. **Hinweis:** Nur verfügbar, wenn für die Option **Treiberkommando** der Eintrag *Treiberspezifisches Kommando* gewählt wurde.
Diesen Dialog zur Runtime anbieten	Konfiguration, ob die Konfiguration in der Runtime geändert werden kann: ▸ *Aktiv*: Dieser Dialog wird in der Runtime vor dem Ausführen der Funktion geöffnet. Die Konfiguration kann damit in der Runtime vor der Ausführung noch geändert werden. ▸ *Inaktiv*: Die Editor-Konfiguration wird in der Runtime beim Ausführen der Funktion angewendet. Default: *inaktiv*

DIALOG BEENDEN

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

VERFÜGBARE TREIBERKOMMANDOS

Diese Treiberkommandos stehen - abhängig vom gewählten Treiber - zur Verfügung:

Treiberkommando	Beschreibung
Kein Kommando	Es wird kein Kommando gesendet. Damit kann auch ein bereits bestehendes Kommando aus einer projektierten Funktion entfernt werden.
Treiber starten (Onlinemodus)	Treiber wird neu initialisiert und gestartet. **Hinweis:** Wenn der Treiber bereits gestartet wurde, muss er gestoppt werden. Erst dann kann der Treiber wieder neu initialisiert und gestartet werden.
Treiber stoppen (Offlinemodus)	Treiber wird angehalten, es werden keine neuen Daten angenommen. **Hinweis:** Ist der Treiber im Offline-Modus, erhalten alle Variablen, die für diesem Treiber angelegt wurden, den Status *Abgeschaltet* (*OFF*; Bit *20*).
Treiber in Simulationsmodus	Treiber wird in den Simulationsmodus gesetzt. Die Werte aller Variablen des Treibers werden vom Treiber simuliert. Es werden keine Werte von der angeschlossenen Hardware (z. B. SPS, Bussystem, ...) angezeigt.
Treiber in Hardwaremodus	Treiber wird in den Hardwaremodus gesetzt. Für die Variablen des Treibers werden die Werte von der angeschlossenen Hardware (z. B. SPS, Bussystem, ...) angezeigt.
Treiberspezifisches Kommando	Eingabe eines treiberspezifischen Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.
Treiber Sollwertsetzen aktivieren	Sollwert setzen auf Treiber ist möglich.
Treiber Sollwertsetzen deaktivieren	Sollwert setzen auf Treiber wird verhindert.
Verbindung mit Modem aufbauen	Verbindung aufbauen (für Modem-Treiber). Öffnet Eingabefelder für Hardware-Adresse und Eingabe der zu wählenden Nummer.
Verbindung mit Modem trennen	Verbindung beenden (für Modem-Treiber).
Treiber in Simulationsmodus zählend	Treiber wird in den zählenden Simulationsmodus gesetzt. Alle Werte werden mit *0* initialisiert und in der eingestellten Updatezeit jeweils um *1* bis zum Maximalwert inkrementiert und beginnen dann wieder

Treiberkommando	Beschreibung
	bei *0*.
Treiber in Simulationsmodus statisch	Es wird keine Kommunikation zur Steuerung aufgebaut. Alle Werte werden mit *0* initialisiert.
Treiber in Simulationsmodus programmiert	Die Werte werden von einem frei programmierbaren Simulationsprojekt berechnet. Das Simulationsprojekt wird mit der zenon Logic Workbench erstellt und läuft in der zenon Logic Runtime ab.

FUNKTION TREIBERKOMMANDOS IM NETZWERK

Wenn sich der Rechner, auf dem die Funktion **Treiberkommandos** ausgeführt wird, im zenon Netzwerk befindet, werden zusätzlich weitere Aktionen ausgeführt:

- Ein spezielles Netzwerkkommando wird vom Rechner zum Server des Projekts gesendet. Dieser führt dann die gewünschte Aktion auf seinem Treiber aus.
- Zusätzlich sendet der Server das gleiche Treiberkommando zum Standby des Projekts. Der Standby führt die Aktion auch auf seinem Treiber aus.

Dadurch ist gewährleistet, dass Server und Standby synchronisiert sind. Dies funktioniert nur, wenn Server und Standby jeweils eine funktionierende und unabhängige Verbindung zur Hardware haben.

8 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 system. 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
[X] Controlling station definition (Master)
[ ] Controlled station definition (Slave)

2. NETWORK CONFIGURATION: 101 ONLY

(network-specific parameter, all configurations that are used are to be marked ' *X* ') Configuration types	
[X] Point-to-point	*[X]* Multipoint-partyline
[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
[X] 100 bit/s	*[X]* 2400 bit/s	*[ ]* 2400 bit/s
[X] 200 bit/s	*[X]* 4800 bit/s	*[ ]* 4800 bit/s

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
[X] 300 bit/s	*[X]* 9600 bit/s	*[ ]* 9600 bit/s
[X] 600 bit/s		*[ ]* 19200 bit/s
[X] 1200 bit/s		*[ ]* 38400 bit/s
		[ ] 56000 bit/s
		[ ] 64000 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
[X] 100 bit/s	*[X]* 2400 bit/s	*[ ]* 2400 bit/s
[X] 200 bit/s	*[X]* 4800 bit/s	*[ ]* 4800 bit/s
[X] 300 bit/s	*[X]* 9600 bit/s	*[ ]* 9600 bit/s
[X] 600 bit/s		*[ ]* 19200 bit/s
[X] 1200 bit/s		*[ ]* 38400 bit/s
		[ ] 56000 bit/s
		[ ] 64000 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	Address field of the link
[ ] Balanced transmission	[255] Maximum length L (number of octets)	*[ ]* not present (balanced transmission only)

Link transmission	Frame length	Address field of the link
[X] Unbalanced transmission		*[X]* One octet
		[X] Two octets
		[X] Structured
		[X] Unstructured

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>

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

Type identification	*Cause of transmission*
any*	any

*assignment of any ASDU type to class 2 as well to class 1 is supported

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* ')

[X] One octet	*[X]* Two octets

INFORMATION OBJECT ADDRESS

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

[X]	**One octet**	*[X]*	**Structured**
[X]	Two octets	*[X]*	Unstructured
[X]	Three octets		

CAUSE OF TRANSMISSION

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

[X]	**One octet**	*[X]*	**Two octets (with originator address). Originator address is set to zero if not used**

LENGTH OF APDU

(system-specific parameter, all configurations that are used are to be marked ' *X* ')
The maximum length of the APDU is 253 (default). 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).

[X]	<1>	:= Single-point information	M_SP_NA_1
[X]	<2>	:= Single-point information with time tag	M_SP_TA_1
[X]	<3>	:= Double-point information	M_DP_NA_1
[X]	<4>	:= Double-point information with time tag	M_DP_TA_1
[X]	<5>	:= Step position information	M_ST_NA_1
[X]	<6>	:= Step position information with time tag	M_ST_TA_1
[X]	<7>	:= Bitstring of 32 bit	M_BO_NA_1
[X]	<8>	:= Bitstring of 32 bit with time tag	M_BO_TA_1
[X]	<9>	:= Measured value, normalized value	M_ME_NA_1
[X]	<10>	:= Measured value, normalized value with time tag	M_ME_TA_1

[X] <1>	:= Single-point information	M_SP_NA_1
[X] <11>	:= Measured value, scaled value	M_ME_NB_1
[X] <12>	:= Measured value, scaled value with time tag	M_ME_TB_1
[X] <13>	:= Measured value, short floating point value	M_ME_NC_1
[X] <14>	:= Measured value, short floating point value with time tag	M_ME_TC_1
[X] <15>	:= Integrated totals	M_IT_NA_1
[X] <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
[X] <30>	:= Single-point information with time tag CP56Time2a	M_SP_TB_1
[X] <31>	:= Double-point information with time tag CP56Time2a	M_DP_TB_1
[X] <32>	:= Step position information with time tag CP56Time2a	M_ST_TB_1
[X] <33>	:= Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
[X] <34>	:= Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
[X] <35>	:= Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
[X] <36>	:= Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
[X] <37>	:= Integrated totals with time tag CP56Time2a	M_IT_TB_1
[ ] <38>	:= Event of protection equipment with time tag CP56Time2a	M_EP_TD_1

[X] <1>	:= Single-point information	M_SP_NA_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).

[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)

[ ] <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).

[X] <100>	:= Interrogation command	C_IC_NA_1
[ ] <101>	:= Counter interrogation command	C_CI_NA_1
[ ] <102>	:= Read command	C_RD_NA_1
[X] <103>	:= Clock synchronization command	C_CS_NA_1
[ ] <104>	:= Test command	C_TS_NA_1
[ ] <105>	:= Reset process command	C_RP_NA_1
[ ] <106>	:= Delay acquisition command	C_CD_NA_1
[ ] <107>	:= Test command with time tag CP56Time2a	C_TS_TA_1

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).

[ ] <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).

[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.
Black boxes: option not permitted in this companion standard
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
<1>	M_SP_NA_1		X	X								X	X		20		
<2>	M_SP_TA_1			X								X	X				
<3>	M_DP_NA_1		X	X								X	X		20		
<4>	M_DP_TA_1			X								X	X				
<5>	M_ST_NA_1		X	X								X	X		20		

Type identification		*Cause of transmission*															
<6>	M_ST_TA_1			X								X	X				
<7>	M_BO_NA_1		X	X											20		
<8>	M_BO_TA_1			X													
<9>	M_ME_NA_1	X	X	X											20		
<10>	M_ME_TA_1			X													
<11>	M_ME_NB_1	X	X	X											20		
<12>	M_ME_TB_1			X													
<13>	M_ME_NC_1	X	X	X											20		
<14>	M_ME_TC_1			X													
<15>	M_IT_NA_1			X													
<16>	M_IT_TA_1			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			X								X	X				
<31>	M_DP_TB_1			X								X	X				
<32>	M_ST_TB_1			X								X	X				
<33>	M_BO_TB_1			X													
<34>	M_ME_TD_1			X													
<35>	M_ME_TE_1			X													
<36>	M_ME_TF_1			X													
<37>	M_IT_TB_1			X													
<38>	M_EP_TD_1																

Type identification		*Cause of transmission*															
<39>	M_EP_TE_1																
<40>	M_EP_TF_1																
<45>	C_SC_NA_1						X	X	X	X	X						L
<46>	C_DC_NA_1						X	X	X	X	X						L
<47>	C_RC_NA_1						X	X	X	X	X						L
<48>	C_SE_NA_1						X	X			X						L
<49>	C_SE_NB_1						X	X			X						L
<50>	C_SE_NC_1						X	X			X						L
<51>	C_BO_NA_1						X	X			X						L
<58>	C_SC_TA_1						X	X	X	X	X						L
<59>	C_DC_TA_1						X	X	X	X	X						L
<60>	C_RC_TA_1						X	X	X	X	X						L
<61>	C_SE_TA_1						X	X			X						L
<62>	C_SE_TB_1						X	X			X						L
<63>	C_SE_TC_1						X	X			X						L
<64>	C_BO_TA_1						X	X			X						L
<70>	M_EI_NA_1*				X												
<100>	C_IC_NA_1						X	X			X						
<101>	C_CI_NA_1																
<102>	C_RD_NA_1																
<103>	C_CS_NA_1			X			X	X									
<104>	C_TS_NA_1																
<105>	C_RP_NA_1																
<106>	C_CD_NA_1																
<107>	C_TS_TA_1																

Type identification		*Cause of transmission*															
<110>	P_ME_NA_1																
<111>	P_ME_NB_1																
<112>	P_ME_NC_1																
<113>	P_AC_NA_1																
<120>	F_FR_NA_1													B			
<121>	F_SR_NA_1													B			
<122>	F_SC_NA_1					B								B			
<123>	F_LS_NA_1													B			
<124>	F_AF_NA_1													B			
<125>	F_SG_NA_1													B			
<126>	F_DR_TA_1*					X											

* Blank or X only

L - log

5. 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)

[ ] 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)

[X] 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	
[ ] group 6	*[ ]* group 12	Information object addresses assigned to each group must be shown in a separate table.

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
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

[configurable]* Supervision of maximum delay in command direction of commands and set point commands
[not limited]* Maximum allowable delay of commands and set point commands

**only toggether with Command Input module*

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).

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

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

[ ] 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).
[ ] Test procedure

FILE TRANSFER

(station-specific parameter, mark 'X' if function is used).
File transfer in monitor direction
[X] Transparent file
[X] Transmission of disturbance data of protection equipment
[X] Transmission of sequences of events
[X] Transmission of sequences of recorded analogue values

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

Maximum range of values for all time-outs: 1 to 255 s, accuracy 1 s.

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	configurable
w	8 APDUs	Latest acknowledge after receiving w I format APDUs	configurable

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

PORTNUMBER: 104 ONLY

Parameter	*Value*	*Remarks*
Portnumber	2404	configurable

REDUNDANT CONNECTIONS: 104 ONLY

[2] Number N of redundant connections according Edition 2 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.

9 Fehleranalyse

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

9.1 Analysetool

Alle zenon Module wie z. B. Editor, Runtime, Treiber, usw. schreiben Meldungen in eine gemeinsame LOG-Datei. Um sie korrekt und übersichtlich anzuzeigen, benutzen Sie das Programm Diagnosis

Viewer, das mit zenon mitinstalliert wird. Sie finden es unter **Start/Alle Programme/zenon/Tools 8.20 -> Diagviewer.**

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.

Weitere Informationen zum Diagnosis Viewer finden Sie im Handbuch Diagnosis Viewer.

9.2 Treiberüberwachung

Die Runtime überwacht die Verfügbarkeit des Treibers via Watchdog. Ist ein Treiber nicht mehr verfügbar, wird für alle angemeldeten Variablen des Treibers zusätzlich das Statusbit *INVALID* gesetzt.

Mögliche Ursachen für Auslösen des Watchdogs:

- Der Treiberprozess läuft nicht mehr.

  Überprüfen Sie im Task Manager ob die Treiber-Exe noch läuft.

- Betriebssystem ist mit höher priorisierten Prozessen ausgelastet.

  Überprüfen Sie die Konfiguration Ihres Systems auf zu wenig Arbeitsspeicher und zu geringe CPU-Leistung. In diesem Fall erfolgt das Rücksetzen des *INVALID* Statusbits durch den Treiber nur bei Wertänderung auf der Gegenstelle. Statische Werte behalten das *INVALID* Statusbit bis zum nächsten Start der Runtime oder des Treibers.

WATCHDOG KONFIGURATION

Für die Überwachung der Kommunikation zur Runtime wird die Verbindung zum Treiber in einem fix vorgegebenen Zeitraum von 60 Sekunden überprüft. Dieser Vorgang wird mehrmals wiederholt. Konnte nach 5 Versuchen (= innerhalb von 5 Minuten) keine valide Verbindung zum Treiber erkannt werden, wird das *INVALID*-Bit bei der angemeldeten (*advised*) Variablen gesetzt. Zusätzlich wird das *INVALID*-Bit auch gesetzt, wenn neue Variablen angemeldeten werden. Das *INVALID*-Bit wird nicht mehr zurückgesetzt.

Dafür werden entsprechende LOG-Einträge erstellt.

LOG-EINTRAG

Bei Auslösen des Watchdogs wird eine Fehlermeldung im LOG protokolliert:

Parameter	Beschreibung
Communication with driver:<drvExe>/<drvDesc>(id:<drvId>) timed out. No communication for <time> ms.	Keine Kommunikation mit Treiber innerhalb der angegeben Zeit. - *<time>*: Zeitangabe in Millisekunden - *<drvDesc>*: Treibername - *<drvExe>*: Treiber EXE-Name - *<drvId>*: Treiber-ID im zenon Projekt
Communication with %s timed out. Invalid-Bit will be set.	Die Kommunikation zum Treiber *%s* konnte bei 5 Versuchen nicht innerhalb von 60 Sekunden aufgebaut werden. Bei der Variable wird das *INVALID*-Bit gesetzt.

Parameter	Beschreibung
Communication with %s timed out. Timeout happened %d times	Die Kommunikation zum Treiber *%s* konnte *%d* Mal nicht innerhalb von 60 Sekunden aufgebaut werden.

9.3 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-Dateien des IEC870 Slaves sowie die LOG Datei des **Diagnosis Viewer** an Ihren zuständigen Support.