zenon
by COPA-DATA

zenon Treiber Handbuch

DNP3_TG

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 DNP3_TG 5

3 DNP3_TG - Datenblatt 10

4 Treiber-Historie 11

5 Voraussetzungen 12

5.1 PC 12

6 Konfiguration 13

6.1 Anlegen eines Treibers 13

6.2 Einstellungen im Treiberdialog 17

6.2.1 Allgemein 17

6.2.2 Com 21

6.2.3 Options 23

6.2.4 Connections 27

7 Variablen anlegen 39

7.1 Variablen im Editor anlegen 39

7.2 Adressierung 43

7.3 Treiberobjekte und Datentypen 44

7.3.1 Treiberobjekte 45

7.3.2 Zuordnung der Datentypen 48

7.4 Variablen anlegen durch Import 64

7.4.1 XML Import 64

7.4.2 DBF Import/Export 65

7.4.3 Online-Import 71

7.4.4 Offline-Import 73

7.5 Kommunikationsdetails (Treibervariablen) 76

8 Treiberspezifische Funktionen 82

8.1 DNP3 File Transfer 86

9 Funktion Treiberkommandos 94

10 Fehleranalyse 99

10.1 Analysetool 99
10.2 Treiberüberwachung 100
10.3 Checkliste 101

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 DNP3_TG

Treiber für das Protokoll nach IEEE1815 -Distributed Network Protocol (DNP3). Der Treiber ist Master auf Protokollebene und unterstützt serielle Kommunikation mit mehrere Outstations, sowie IP-Kommunikation über TCP und UDP.

BEGRIFFSDEFINITIONEN

Zum besseren Verständnis dieses Dokuments finden Sie in der folgenden Liste die Definitionen wichtiger Begriffe.

Begriff	Definition
Event Class	Eine Art der Gruppierung in der **Outstation**. Es gibt: ▸ **Event Classes** *1*,*2*, *3*: Eine **Event Class** ist in der Regel einem Puffer zugeordnet, in dem Wertänderungen für konfigurierte Objektgruppen mit der konfigurierten Variation in der **Outstation** gespeichert werden. Ein Master kann mittels einer Lese-Anfrage für z.B. **Class** *1* alle Wertänderungen dieser **Event Class** von der **Outstation** anfordern. In der **Outstation** kann in der Regel konfiguriert werden, welche **Points** bzw. welche **Object Group** zu welcher **Event Class** gehören. Es gibt keine exakte Definition, allerdings wird in der Regel **Class** *1* für wichtige Meldungen verwendet, z. B. *Binary Inputs* für Alarme. ▸ **Static Class** *0*: Die **Static Class** *0* liefert bei einer Leseanfrage vom **Master** jeweils den letzten Wert für beinahe alle **Points** zurück.
Eventpoll	Lese-Anfrage vom **Master** an die **Outstation**, wobei der **Master** die **Outstation** auffordert, nur die geänderte Werte für eine bestimmte **Event Class** oder für alle **Event Classes** (*1*, *2* und *3*) zu senden.
Integrity Poll	Lese Anfrage vom **Master** an die **Outstation** für die **Static Class** *0*. Der **Master** fordert mit dieser Anfrage ein Initialabbild aller **Points** an.
Master	Kontrollierende Station. Ein **Master** sendet Leseanfragen und Steueranfragen an die **Outstation**.
Objekt Group	Datentyp oder Datenart in der **Outstation** mit einer definierten Funktionalität. Beispiele für Zuordnungen: ▸ *Object Group 30* steht für einen statischen Analogeingang. ▸ Eine Wertänderung für einen Analogeingang ist *Object Group 32* zugeordnet. ▸ *Object Group 40* dient zum Lesen eines Analogausgangs. ▸ *Object Group 41* dient zum Schreiben auf einen Analogausgang.
Outstation	**SPS** oder **RTU** in **DNP3**. Eine **Outstation** ist Slave auf Protokollebene und sendet auf Anfrage Daten an den **Master** .
Point	Äquivalent einer Variable in der Outstation. Ein **Point** wird angesprochen mit einer **Point Number** (Offset) pro **Objekt Group**, wobei z.B. gilt: *Point 12* für **Object Group** *30* und *Point 12* für **Object Group** *32* haben beide den gleichen Analogeingang als Basis, *Point 12* für **Object Group** *1* jedoch ist ein komplett

Begriff	Definition
	unabhängiger Binäreingang. Der allgemeine Begriff für eine Wert-, Zeit- oder Statusänderung, ist **DNP Object**.
Unsolicited Response	Meldung von Wertänderungen aus einer **Event Class**, die spontan von der **Outstation** an den **Master** gesendet wird. Dazu muss der **Master** jedoch zuerst in der **Outstation Unsolicited Responses** aktivieren. Die **Outstation** muss **Unsolicited Responses** unterstützen und diese müssen für sie konfiguriert sein.
Variation	Format, in dem die **Oustation** einen statischen Wert oder eine Wertänderung in **Class** *0* bzw. **Class** *1*, *2* oder *3* speichert. Dass kann in der **Outstation** pro **Object Group** oder pro **Point** konfiguriert werden. Die **Variation** bestimmt z.B. ob: ▸ ein statischer Wert (**Class** *0*) oder eine Wertänderung, eine Integer oder Floating Point ist ▸ ein Zeitstempel gespeichert wird oder nicht, oder, ob Object Flags gespeichert werden oder nicht

KOMMUNIKATION

Die Kommunikation erfolgt vor allem pollend. Dabei fragt der Treiber die Outstation in konfigurierbaren Abständen nach Events (Wertänderungen und Statusänderungen) ab. Diesen sind in der Outstation Klassen (*1*, *2* oder *3*) zugeordnet.

Je nach Konfiguration in der Outstation kann als Antwort auch mehr als ein Wert für ein einzelnes DNP-Objekt gesendet werden. (**Sequence Of Events Buffer** im Gegensatz zu **Latest Value-in** der Oustation).

Der Treiber verarbeitet die empfangenen Werte und sendet diese weiter zu der Runtime.

Sofern die Outstation es unterstützt, können im Treiber auch pro Eventklasse **unsolicited responses** aktiviert werden. Der Treiber sendet in diesem Fall am Ende der Startup-Routine, nachdem der Integrity Poll abgeschlossen worden ist, die Anforderung **unsolicited responses** für die jeweilige Klasse zu aktivieren. Die Outstation kann anschließend ohne Aufforderung des Masters (Polling von Eventklassen) Wertänderungen an den Master senden. In diesem Fall ist es nicht unbedingt erforderlich, regelmäßig nach Events zu pollen. Der Polling-Zyklus kann dann höher oder überhaupt auf *0* (Polling für Eventklassen deaktiviert) eingestellt werden. Das Pollen für Events kann auch von der Runtime aus manuell angestoßen werden (Class Poll Variablen).

Weitere Informationen finden Sie in der Dokumentation zum **DNP3/IEEE1815-2012 Standard**. Diese Dokumentation können Sie von der IEEE erwerben. Sie bekommen auch Zugang, wenn Sie der **DNP3 users group** (http://www.dnp.org/ (http://www.dnp.org/)) beitreten.

DEVICE PROFILE

Ein Device Profile ist ein Standard-Dokument, das beschreibt, welche Funktionalität aus dem DNP3 Standard unterstützt wird. Für den DNP3_TG Treiber finden Sie das XML Device Profile nach Installation von zenon im Ordner *%ProgramData%\COPA-DATA\zenon8.20\CommunicationProfiles\Dnp3\Driver\DNP3_TG.xml*.

INFORMATION ZU DNP332, DNP3_NG UND DNP3_TG TREIBER

Der DNP332 Treiber und der DNP3_NG Treiber werden ab der Version 7.20 durch den DNP3_TG Treiber ersetzt. Aus Kompatibilitätsgründen sind der DNP3_NG Treiber und der ältere DNP332 Treiber nach wie vor im Setup enthalten, werden jedoch per Default nicht länger in der Treiberauswahlliste angezeigt. Bestehende Projekte, die konvertiert werden, verwenden nach wie vor den ursprünglich projektierten Treiber. Der alte Treiber kann in konvertierten Projekte normal weiter verwendet werden. Ein Wechsel auf den neuen DNP3_TG Treiber ist jedoch auch möglich.

DNP332/DNP3_NG IN TREIBERLISTE ANZEIGEN

Wenn Sie den DNP332/DNP3_NG Treiber in einem neuen Projekt in 7.20 verwenden möchten, muss der Treiber wieder zu der Treiberliste hinzugefügt werden:

1. Starten Sie das Programm **Driverinfo.exe** vom zenon Installationsmedium, Unterordner *\AdditionalSoftware\COPA-DATA DriverXML Editor*.
2. Öffnen Sie mit dem Programm **Driverinfo** die Treiber XML-Datei.
   Beispiel: **TREIBER_DE.XML** aus dem Ordner *%CD_PROGRAMDATA7200%*
   Dabei steht **DE** für das Kürzel der Sprache im Editor und **7200** für die installierte Version 7.20.
3. Navigieren Sie zum Ordner *DNP3* und wählen Sie im Kontextmenü **New Driver**.
4. Geben Sie **DNP332** bzw. **DNP3_NG** in alle drei Felder des Dialogs ein und bestätigen Sie mit Klick auf **OK**.
5. Speichern Sie die Änderungen mit Klick auf das Symbol **Speichern** in der Symbolleiste.
   Ab diesem Zeitpunkt steht der Treiber im Editor wieder zur Auswahl.
6. Wiederholen Sie diese Schritte für jede Sprache, in der Sie den Editor verwenden.

AUF DEN NEUEN DNP3_TG TREIBER WECHSELN

Der DNP3_TG Treiber ist im Prinzip kompatibel mit dem alten DNP3_NG Treiber und den DNP332 Treiber. Mittels der Funktion **Treiber Austauschen** im Editor kann vom DNP332 Treiber oder vom DNP3_NG Treiber auf den DNP3_TG Treiber gewechselt werden. Nach dem Treiber Austausch müssen einige Einstellungen manuell neu gemacht werden. Einige Funktionen sind im DNP3_TG Treiber anders implementiert und erfordern eine Änderung der Projektierung.

Beachten Sie bei einem geplanten Treibertausch:

Allgemein:

- Machen Sie als erstes eine Projektsicherung
- Notieren Sie die Treiberkonfiguration des alten Treibers. Nach dem Treibertausch muss diese im neuen Treiber erneut eingegeben werden.
- Das Schreiben auf eine Variable vom Typ Frozen Counter führt kein Immediate Freeze No Ack mehr aus. Diese Funktionalität kann über eine Variable vom Typ Commando erreicht werden.

DNP332 auf DNP3_TG:

- Wenn Sie beim DNP332 Treiber Select Before Operate bei der Befehlsgabe verwenden, deaktivieren Sie die Eigenschaft **Select Before Operate** bei der Variable. Wählen Sie stattdessen für die Variablen vom Typ *Binary Ouput* und *Analog Output* bei der Eigenschaft **Command Mode** den Eintrag *Auto-SBO* aus.
- Falls Sie beim DNP332 Treiber Datentypen für **Analog Inputs** oder **Counter** verwendet haben, die es im DNP3_NG Treiber nicht mehr gibt, dann müssen Sie vor dem Wechsel die Datentypen der Variablen anpassen. Hintergrund: Mit dem DNP332 Treiber war es möglich, Datentypen zu wählen, die vom DNP3 Standard nicht vorgesehen waren. Im DNP3_NG können diese Datentypen nicht länger verwendet werden.

DNP3_NG auf DNP3_TG:

- Das Property **Command Mode** für Variablen vom Typ "*Analog Output*" und "*Binary Output*" muss manuell richtig gesetzt werden. Alternativ ist es möglich die Variablen vor dem Treiber Austausch in XML zu exportieren und nach dem Austausch wieder zu importieren. Der DNP3_TG Treiber unterstützt neben Direct Operate und Auto SBO auch Direct Operate No Ack.
- Das Property Variation bei der Variable wurde entfernt. Anstelle der Variation bei der Variable zu konfigurieren kann die Variable vom Class Poll ausgenommen werden über den Checkbox **Classless Read** und mittels Benutzerdefiniertem Kommando mit der gewünschte Variation explizit gelesen werden.
- Der Variablentyp "Class Scan" existiert bei dem DNP3_TG nicht mehr. Löschen Sie am besten Variablen dieses Typs vor dem Treiber Austausch. Um einen Class Poll zu triggern, einen explizites Lesen zu triggern oder auch ein Cold Restart zu triggern, erstellen Sie vom DNP3_TG Treiber variablen vom Variablentyp "Command".
-

3 DNP3_TG - Datenblatt

Allgemein:	
Treiberdateiname	DNP3_TG.exe
Treiberbezeichnung	DNP3 third generation Treiber
Steuerungs-Typen	DNP3 / IEEE 1815 Outstations
Steuerungs-Hersteller	DNP3

Treiber unterstützt:	
Protokoll	DNP3; IEEE Std 1815
Adressierung: Adress-basiert	Address based
Adressierung: Namens-basiert	--
Kommunikation spontan	X
Kommunikation pollend	X
Online Browsing	X
Offline Browsing	X
Echtzeitfähig	X
Blockwrite	--
Modemfähig	--
RDA numerisch	--
RDA String	--
Hysterese	--
erweiterte API	X
Unterstützung von Statusbit **WR-SUC**	X
alternative IP-Adresse	X

Voraussetzungen:	
Hardware PC	Serielle Schnittstelle; Standard Netzwerkkarte
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 Treiber-Historie

Datum	Build Nummer	Änderung
13.11.14	15670	Treiberdokumentation wurde neu erstellt.
16.03.15	20178	*Subset Level 4* Erweiterungen implementiert und dokumentiert.
20.07.15	21164	*File transfer* wurde implementiert und dokumentiert.

TREIBERVERSIONIERUNG

Mit zenon 7.10 wurde die Versionierung der Treiber verändert. Ab dieser Version gibt es eine versionsübergreifende Build-Nummer. Das ist die Zahl an der 4. Stelle der Dateiversion.
Zum Beispiel: **7.10.0.4228** bedeutet: Der Treiber ist für Version **7.10**, Service Pack **0** und hat die Build-Nummer **4228**.

Erweiterungen oder Fehlerbehebungen werden zukünftig in einem Build eingebaut und sind dann ab der nächsthöheren Build-Nummer verfügbar.

Beispiel

Eine Treibererweiterung wurde in Build **4228** implementiert. Der Treiber, den Sie im Einsatz haben, verfügt über die Build-Nummer **8322**. Da die Build-Nummer Ihres Treibers höher ist als die Build-Nummer der Erweiterung, ist die Erweiterung enthalten. Die Versionsnummer des Treiber (die ersten drei Stellen der Dateiversion) spielen dabei keine Rolle. Die Treiber sind versionsunabsabhängig

5 Voraussetzungen

Dieses Kapitel enthält Informationen zu den Voraussetzungen, die für die Verwendung des Treibers erforderlich sind.

5.1 PC

HARDWARE

- Serielle Schnittstelle
- Ethernet TCP/UDP

Für Dual Endpoint muss der konfigurierte Listening Socket in der Firewall entsprechend konfiguriert sein.

SOFTWARE

Wenn nicht bereits vorhanden, Treiber **DNP3_TG.exe** in das zenon Programmverzeichnis kopieren und sicherstellen, dass auch die **DNP3_TGV.dll** vorhanden ist.

CE

Treiber **DNP3_TG.dll** in das zenon CE-Programmverzeichnis kopieren.
Für die Runtime wird die DN3_TGV.dll nicht benötigt.

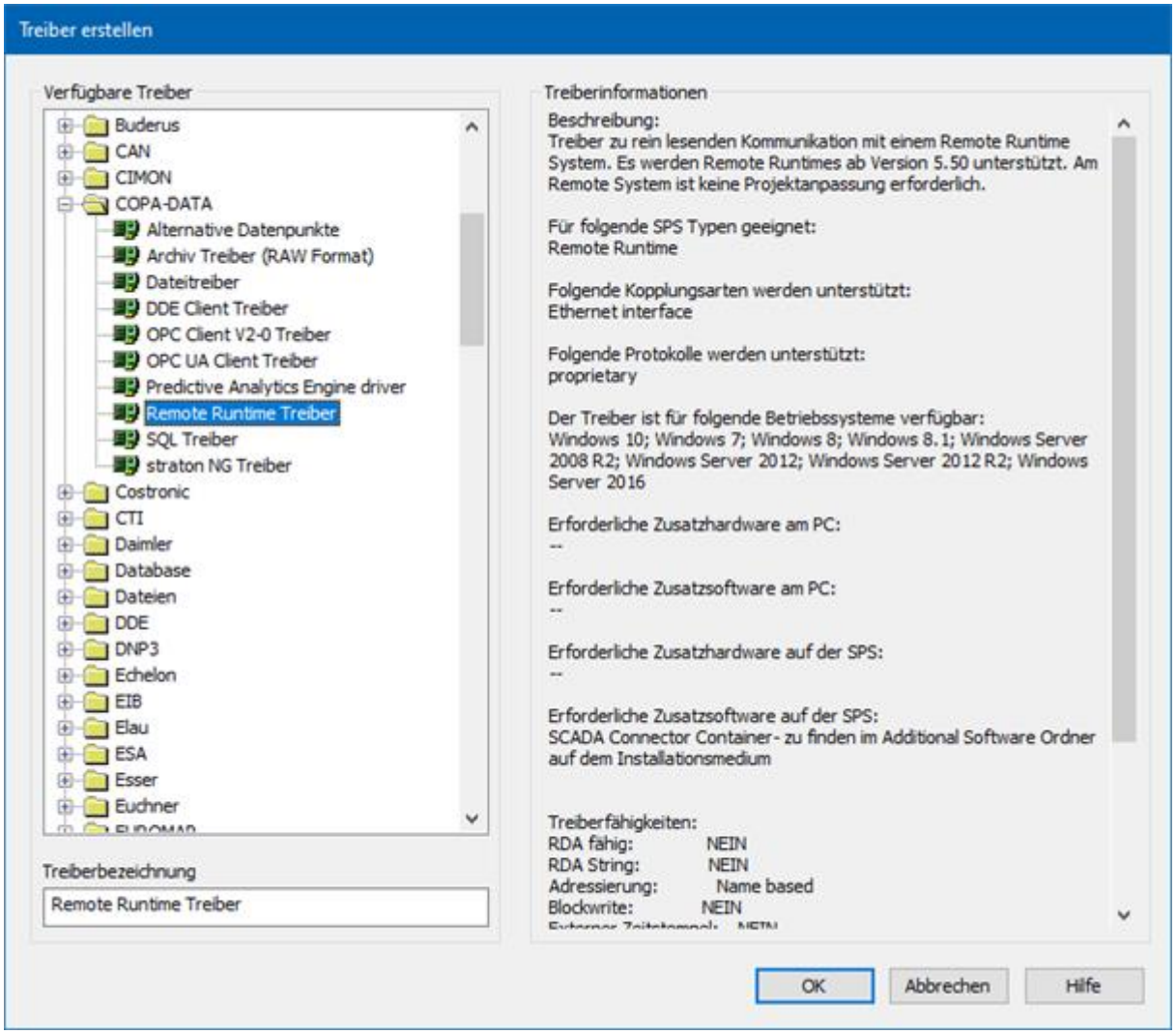
6 Konfiguration

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.

6.1 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 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

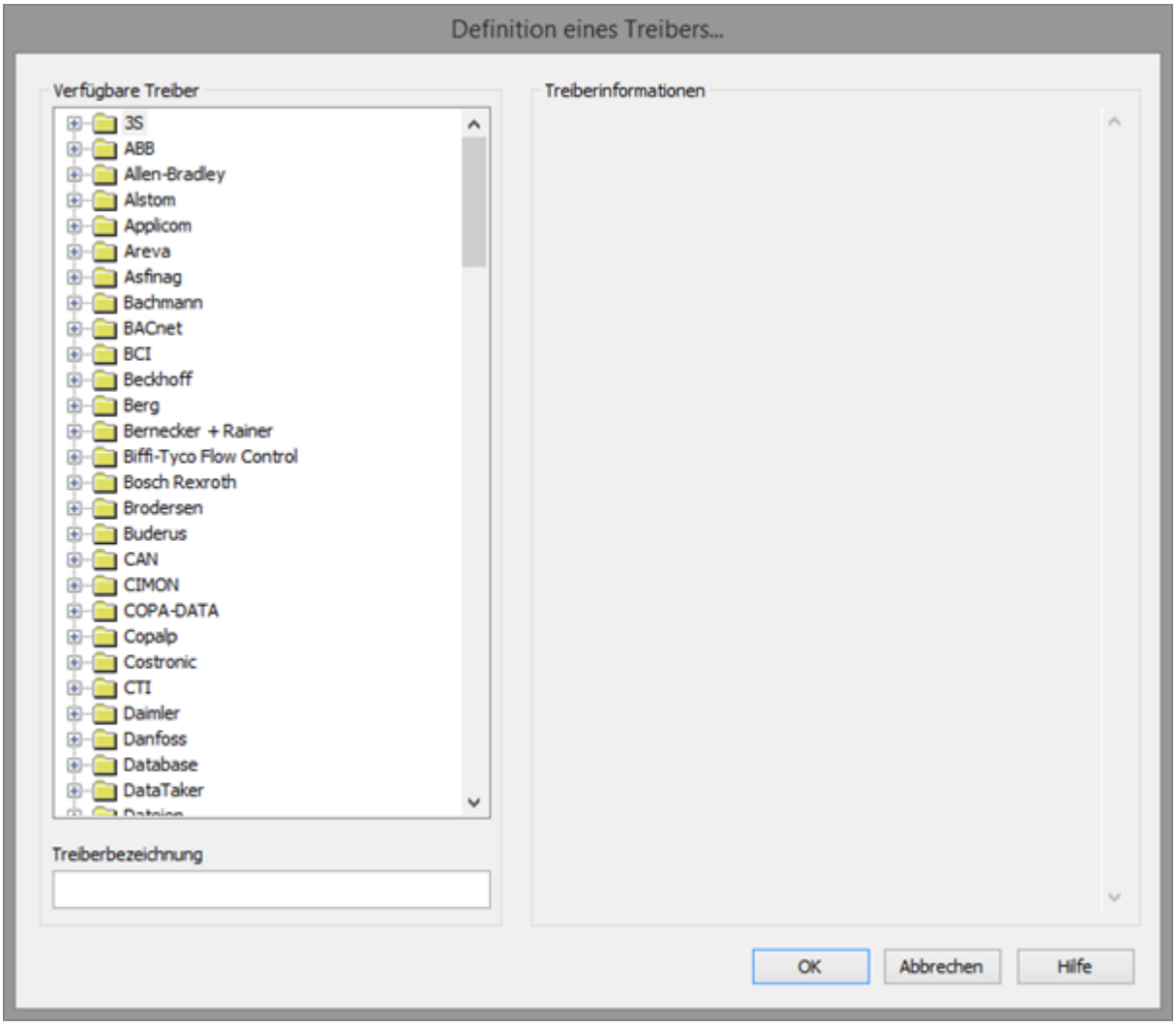
Um einen neuen Treiber anzulegen:

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.

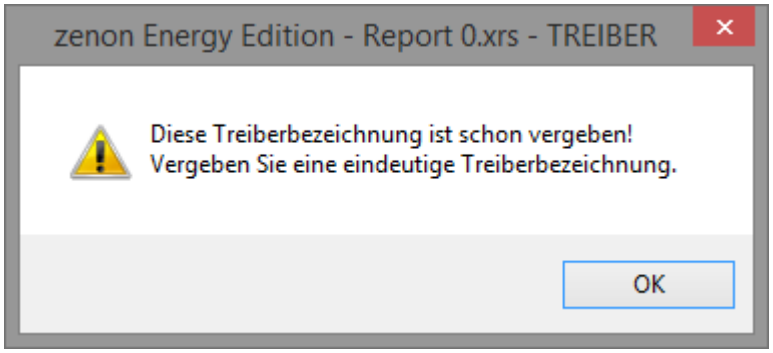
- Die **Treiberbezeichnung** ist Bestandteil des Dateinamens.
  Daher darf Sie nur Zeichen enthalten, die vom Betriebssystem unterstützt werden. Nicht gültige Zeichen werden durch einen Unterstrich (_) ersetzt.
- **Achtung:** Die Bezeichnung kann später nicht mehr geändert werden.

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**

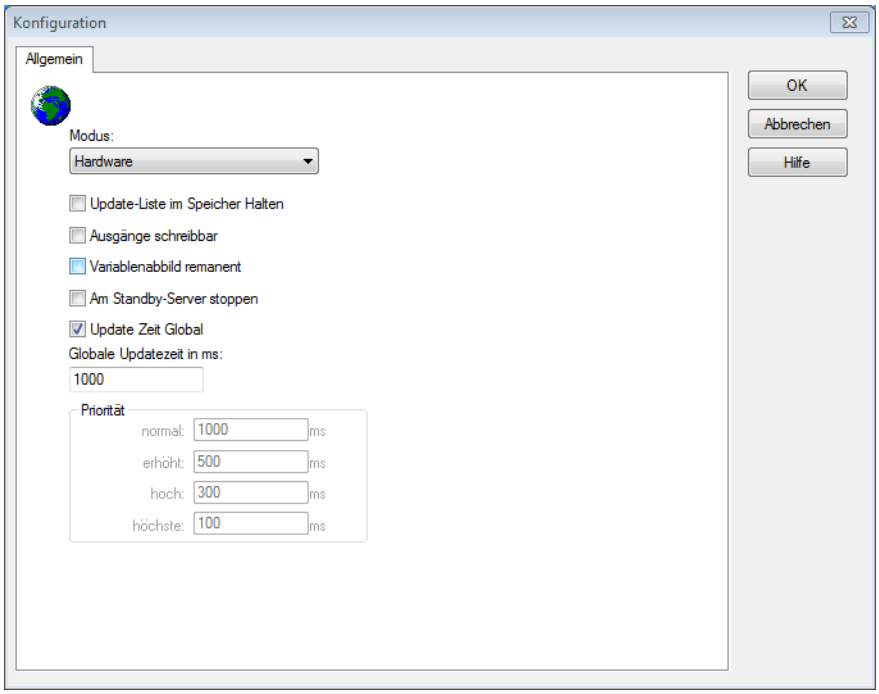
> **Info**
>
> In einem zenon Projekt müssen nur die benötigten Treiber vorhanden sein. Treiber können bei Bedarf zu einem späteren Zeitpunkt hinzugefügt werden.

6.2 Einstellungen im Treiberdialog

Folgende Einstellungen können Sie beim Treiber vornehmen:

6.2.1 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

Option	Beschreibung
	Simulatortreiber. - *Simulation - zählend*: Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. 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)*

Option	Beschreibung
	▸ *ERSATZWERT(27)* Das Variablenabbild wird immer gespeichert wenn: ▸ die Variable vom Objekttyp **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:

Option	Beschreibung
	▸ *Aktiv*: Die eingestellte **Globale Update Zeit** wird für alle Variablen im Projekt verwendet. Die bei den Variablen eingestellte Priorität wird nicht 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

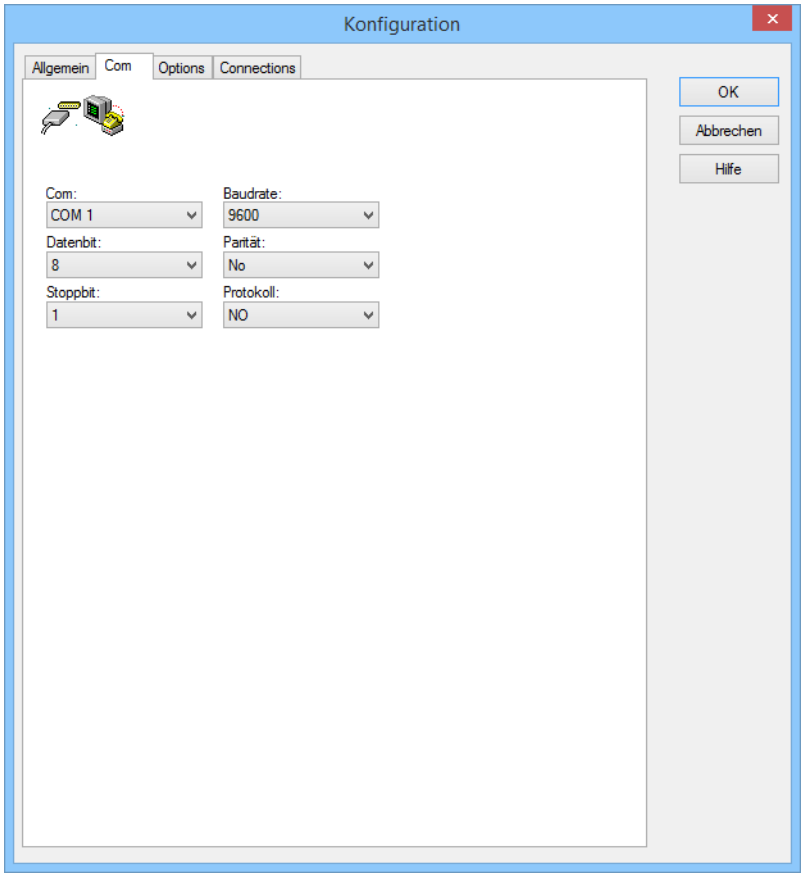
Option	Beschreibung
OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

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**.

6.2.2 Com



Parameter	Beschreibung
Com	Auswahl Com-Port. Default: *1*
Baudrate	Auswahl Baudrate. Anpassen an Steuerung. Auswahl aus Dropdownliste. Default: *9600*

Parameter	Beschreibung
	Eingabebereich: *110* bis *256000*
Datenbit	Anzahl Datenbits. Anpassen an Steuerung. Default: *8*
Stopbit	Auswahl Stopbit. Anpassen an Steuerung. Default: *1*
Parität	Auswahl Parity. Anpassen an Steuerung. Default: *No*
Protokoll	Auswahl Protokoll. Anpassen an Steuerung. Default: *No*

DIALOG BEENDEN

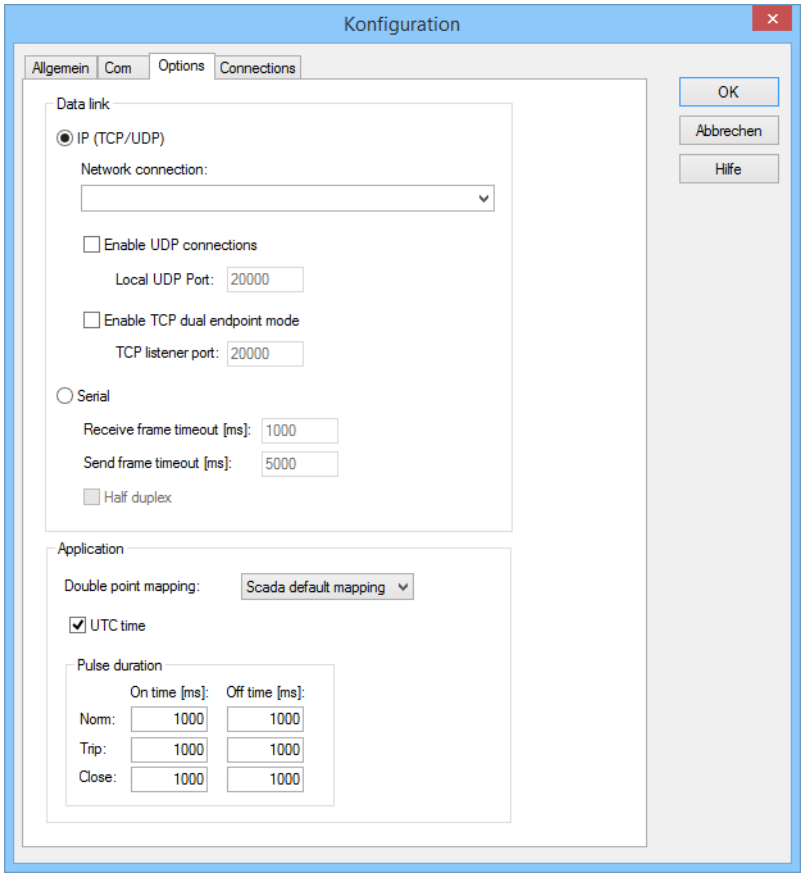
Option	Beschreibung
OK	Übernimmt alle Änderungen in allen Registerkarten und schließt den Dialog.
Abbrechen	Verwirft alle Änderungen in allen Registerkarten und schließt den Dialog.
Hilfe	Öffnet die Online-Hilfe.

Info

Die exakten Einstellungen hängen von der verwendeten Steuerung ab. Entnehmen Sie die gültigen Werte dem Handbuch der Steuerung.

6.2.3 Options

In dieser Registerkarte konfigurieren Sie die Kommunikationsparameter.



DATA LINK

Parameter	Beschreibung
IP (TCP/UDP)	*Aktiv*: Die Kommunikation erfolgt über TCP/UDP.
Network connection	Auswahl der Netzwerkverbindung, die für die Kommunikation genutzt wird. Eingabe in Feld oder Auswahl aus Dropdownliste von: ▸ Name oder ▸ Netzwerkverbindung oder ▸ IP-Adresse. In der Dropdownliste werden die Namen der am lokalen Rechner verfügbaren Netzwerkverbindungen angezeigt.

Parameter	Beschreibung
	Für Remote Runtime Rechner geben Sie den Namen der Netzwerkkarte oder die IP-Adresse des Remote Rechners ein. Diese Einstellung ist aktiv für UDP Kommunikation und TCP Dual Endpoint Kommunikation. Schlagt das Öffnen des LISTENING Ports für UDP / TCP für die konfigurierte Netzwerkverbindung fehl, erstellt der Treiber einen LISTENING Port für sämtliche Netzwerkkarten im System.
Enable UDP connections	*Aktiv*: Das UDP Protokoll kann zur Kommunikation mit den Steuerungen verwendet werden.
Local UDP port	Lokaler UDP Port. Dieser wird für alle UDP-Verbindungen verwendet. Default: *20000*
Enable TCP dual endpoint mode	*Aktiv*: TCP Dual-Endpoint Mode wird verwendet. In diesem Modus kann eine Steuerung bei Bedarf eine TCP-Verbindung zum Controller initiieren.
TCP listener port	Port, der für den TCP Dual-Endpoint Betrieb verwendet wird. **Hinweis:** Dieser Port wird für alle Verbindungen genutzt. Default: *20000*
Serial	*Aktiv*: Kommunikation erfolgt über die serielle Schnittstelle.
Recieve frame timeout [ms]	Timeout in Millisekunden für den Empfang eines kompletten Telegramms. Startzeit ist das erste Zeichen des Telegramms. Default: *1000*
Send frame timeout [ms]	Timeout in Millisekunden für das Senden eines Telegramms. Ist der serielle Ausgangspuffer länger als diese Zeitspanne gefüllt, wird das als Sendefehler betrachtet. Default: *5000*
Half duplex	*Aktiv*: Es wird erst wieder ein Telegramm versendet, wenn für das zuletzt auf der Schnittstelle gesendete Telegramm eine Antwort empfangen wurde.

APPLICATION

Parameter	Beschreibung
Double Point Mapping	Auswahl des Double Point Mappings auf einen Integerwert. Auswahl aus Dropdownliste: ▸ *SCADA default mapping* ▸ *DNP3 binary mapping* ▸ *Custom legacy mapping* Default: *SCADA default mapping*
UTC time	*Aktiv*: Alle Zeiten von und zur Steuerung werden als UTC und nicht als lokale Zeit behandelt. Default: *aktiv*
Puls duration	Einstellung der Puls-Dauer für **Norm**, **Trip** und **Close** je Verbindung. **Norm**, **Trip** und **Close** legen fest, welches Relais geschaltet wird.
Norm	Entspricht **NUL**. Pulsdauer Norm: ▸ **On time:** Zeitraum, in dem der Output auf *EIN* ist, in Millisekunden Default: *1000* ▸ **Off time:** Zeitraum, in dem der Output auf *AUS* ist, in Millisekunden Default: *1000*
Trip	Pulsdauer Trip: ▸ **On time:** Zeitraum, in dem der Output auf *EIN* ist, in Millisekunden Default: *1000* ▸ **Off time:** Zeitraum, in dem der Output auf *AUS* ist, in Millisekunden Default: *1000*
Close	Pulsdauer Close: ▸ **On time:**

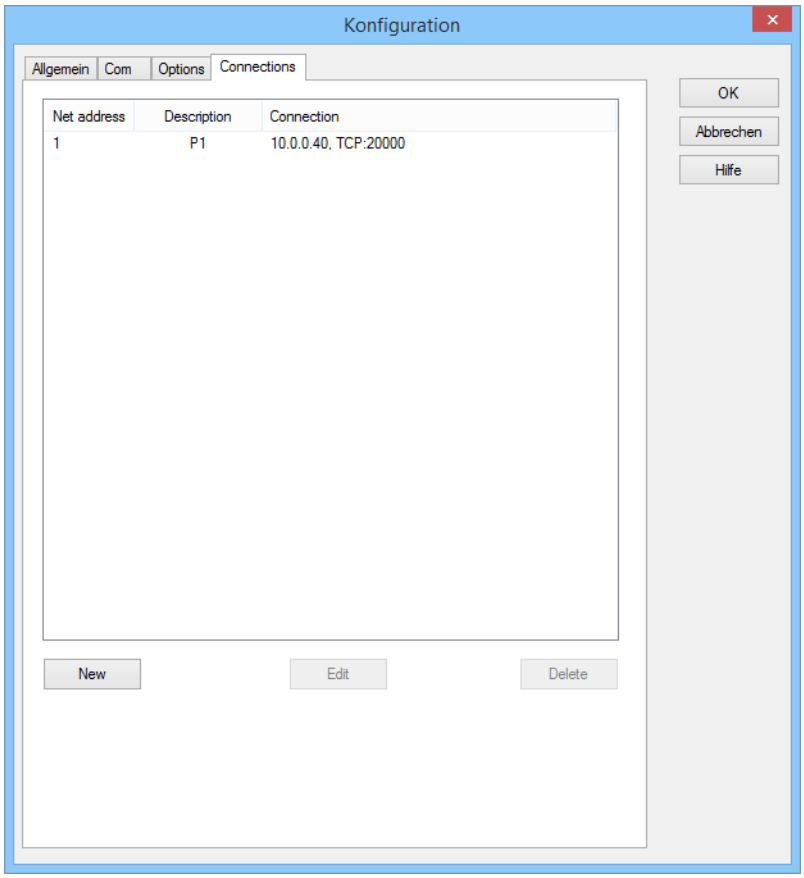
Parameter	Beschreibung
	Zeitraum, in dem der Output auf *EIN* ist, in Millisekunden Default: *1000* ▸ **Off time:** Zeitraum, in dem der Output auf *AUS* ist, in Millisekunden Default: *1000*

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.

6.2.4 Connections

In dieser Registerkarte konfigurieren Sie die Verbindungen.



Parameter	Beschreibung
Liste Verbindungen	Zeigt die konfigurierten Verbindungen an.
New	Öffnet Dialog zum Anlegen einer neuen Verbindung (auf Seite 29).
Edit	Öffnet Dialog zum Bearbeiten der ausgewählten Verbindung.
Delete	Löscht ausgewählte Verbindung.

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.

Option	Beschreibung
Hilfe	Öffnet die Online-Hilfe.

NEUE VERBINDUNG ANLEGEN

1. Klicken Sie auf die Schaltfläche **New**.
2. Tragen Sie die Verbindungsdetails ein.
3. Klicken Sie auf **OK**.

VERBINDUNG BEARBEITEN

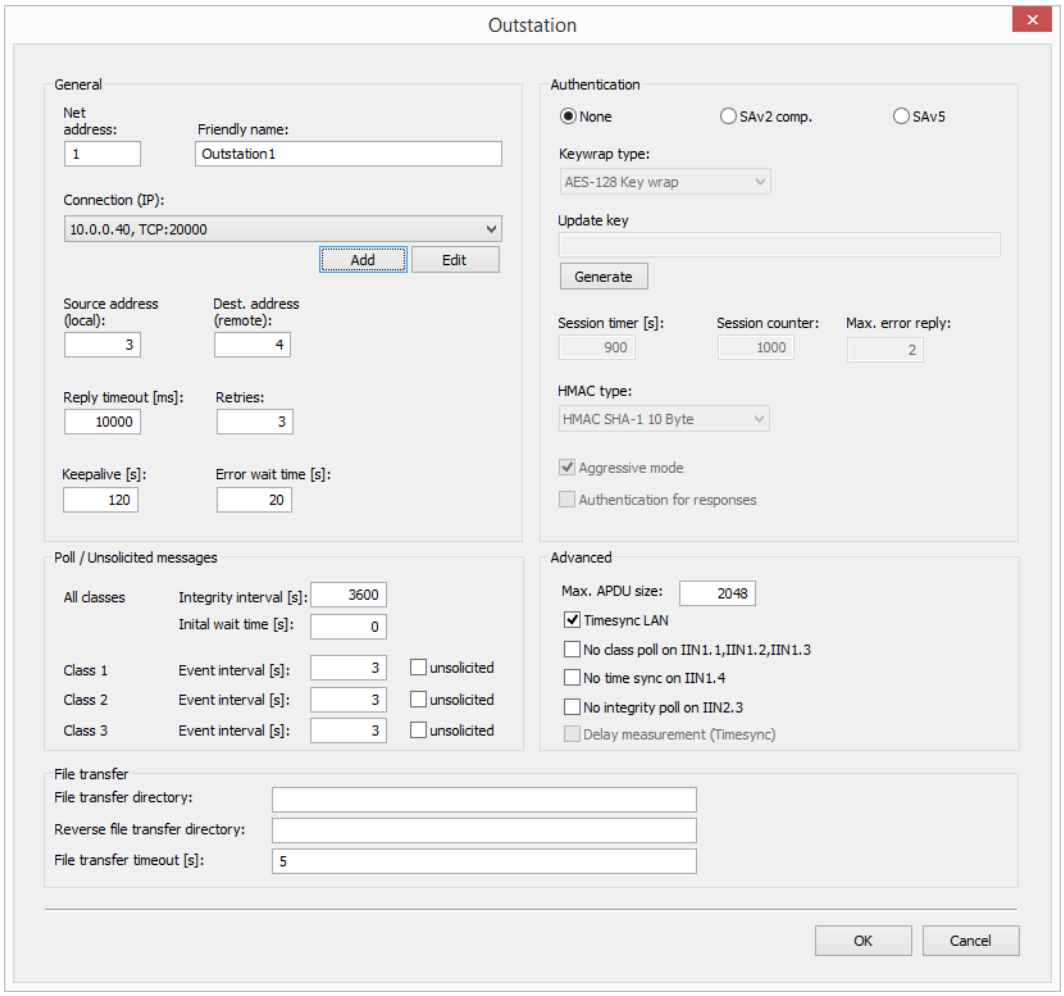
1. Wählen Sie in der Verbindungsliste die gewünschte Verbindung.
2. Klicken Sie auf die Schaltfläche **Edit**.
3. Ändern Sie die Verbindungsparameter.
4. Klicken Sie auf **OK**.

VERBINDUNG LÖSCHEN

1. Wählen Sie in der Verbindungsliste die gewünschte Verbindung.
2. Klicken Sie auf die Schaltfläche **Delete**.
3. Die Verbindung wird aus der Liste gelöscht.

6.2.4.1 Outstation

In diesem Dialog konfigurieren Sie Verbindungen. Der Dialog wird geöffnet, wenn Sie in der Registerkarte **Connections** (auf Seite 27) auf die Schaltfläche **New** oder **Edit** klicken.



GENERAL

Allgemeine Einstellungen.

Parameter	Beschreibung
Net address	Netzadresse der Verbindung. Wert zwischen *0* und *255*. Entspricht der Eigenschaft **Netzadresse** bei der Variable.
Friendly name	Name der Verbindung. Frei wählbar.
Connection (IP)	Auswahl des TCP/UDP Kanals über den mit dieser Outstation kommuniziert wird.

Parameter	Beschreibung
	Im Fall von virtuellen Outstations ist es möglich, dass sich mehrere Stationen einen Kanal teilen. (Nicht verfügbar für serielle Kommunikation.)
Add	Fügt eine neue TCP/UDP Link-Konfiguration (auf Seite 38) hinzu. Der Dialog TCP/UDP Links wird geöffnet.
Edit	Öffnet Dialog zum Editieren (auf Seite 36) aller vorhandenen TCP/UDP Links.
Source address (local)	Eigene Adresse des Treibers (DNP3 Master).
Dest. address (remote)	Linkadresse der Steuerung (DNP3 Outstation). Es kann die Addresse 65532 (Self Address) der Outstation verwendet werden, sofern die Outstation das unterstützt. Allerdings werden die Antworte von der Outstation mit der eigene Addresse, vom Treiber nicht akzeptiert.
Reply timeout [ms]	Zeit in Millisekunden, die auf eine Antwort der Steuerung gewartet wird. Ist die Zeit abgelaufen wird eine Wiederholung gesendet. Wird ein Fragment einer mehrteiligen Antwort empfangen. Beginnt das Intervall mit jedem empfangenem Fragment von Neuem. Default: *10000 ms*
Retries	Anzahl der Kommunikationswiederholungen nach Timeouts. ▸ *0*: deaktiviert
Keepalive [s]	Werden über die angegebene Zeit keine Daten empfangen oder gesendet, wird ein LINK_STATUS_REQUEST Telegramm gesendet. ▸ *0*: deaktiviert
Error wait time [s]	Fehlerwartezeit in Sekunden. Nach einem Fehler wird die angegebene Zeit gewartet, bevor erneut ein Verbindungsaufbau versucht wird ▸ *0*: deaktiviert

POLL/UNSOLICITED MESSAGES

Einstellungen für Polling und **Unsolicited Events**.

Parameter	Beschreibung
All classes	Integritätsintervall und Ereignisintervall für alle Klassen.
Integrity interval [s]	Integrity Poll des Treibers in Sekunden. Default: *3600*
Initial wait time [s]	Initiale Wartezeit vor einem neuerlichen Verbindungsaufbau Kann verwendet werden, um den Verbindungsaufbau zu unterschiedlichen Outstations zu Staffeln. Default: *0*
Class1 - Class 3	Einstellungen für Class 1, Class 2 und Class 3. Für jede Klasse werden die Optionen **Event interval [s]** und **unsolicited** extra konfiguriert.
Event interval [s]	Für die jeweilige Klasse spezifischer Intervall für das Polling von Events in Sekunden. Wert: *0 bis 99999 s* ▸ 0: *kein Polling* Default: *0*
unsolicited	*Aktiv*: Für diese Klasse werden Unsolicited Events akzeptiert. Siehe auch Kapitel Adressierung.

AUTHENTICATION

Einstellungen für Authentifizierung.

Parameter	Beschreibung
Authentication	Auswahl der Authentifizierungmethode: ▸ **None** ▸ **SAv2 compatible** ▸ **SAv5**
None	*Aktiv*: Es wird keine Authentifizierung verwendet.

Parameter	Beschreibung
SAv2 compatible	*Aktiv*: Authentifizierung im Kompatibilitätsmodus für Version 2.
SAv5	*Aktiv*: Authentifizierung für Version 5.
Keywrap type	Anzuwendender Keywrap-Algorithmus. Auswahl aus Dropdownliste: ▸ AES-128 ▸ AES-256
Update Key	Authentifizierungs-Schlüssel für die gesicherte Kommunikation. Eingabe in Feld oder automatischer Generierung über Klick auf Schaltfläche **Generate**. Es wird erwartet für: ▸ AES-128 Keywrap: 32 hexadezimale Digits ▸ AES-256 Keywrap: 64 hexadezimale Digits **Erlaubte Zeichen:** ▸ Ziffern: *0* bis *9* ▸ Buchstaben: *a* bis *f* und *A* bis *F* ▸ Sonderzeichen: Leerzeichen, Punkte und Doppelpunkte Alle übrigen Zeichen: Werden als *0* interpretiert und ergeben eventuell einen Fehler beim Erstellen des Sitzungsschlüssels.
Generate	Erzeugt einen zufälligen **Update Key**.
Session timer	Gültigkeitsdauer des Sitzungsschlüssels in Sekunden. Wert: *<700000 s* Default: *900 s*
Session counter	Vorgabe, wie oft ein Sitzungsschlüssel verwendet werden darf. Wert: *<10000 s* Default: *1000*
Max. error reply	Vorgabe, wie viele Fehler bei der Authentifizierung

Parameter	Beschreibung
	gemeldet werden. Wert: *0 bis 10* Default: *2*
HMAC type	Art des verwendeten HMACs (HASH Funktion), wenn der Treiber eine Challenge für ein Response (Authentication Requests) versendet. Auswahl aus Dropdownliste: ▸ HMAC SHA-1 4 Byte (Index 1) Kommunikation nur mit der Authentifizierungsmethode **SAv2 comp.** erlaubt. Gilt nur für serielle Verbindungen. Der Session timer muss kleiner als 1800 Sekunden (30 Minuten) gewählt sein. ▸ HMAC SHA-1 8 Byte (Index 5) ▸ HMAC SHA-1 10 Byte (Index 2) ▸ HMAC SHA-256 8 Byte (Index 3) ▸ HMAC SHA-256 16 Byte (Index 4)
Aggressive mode	*Aktiv*: Authentifizierung wird im aggressiven Modus durchgeführt.
Authentication for responses	Nur mit Authentifizierungsmethode **SAv5** erlaubt. ▸ Aktiv: Der Treiber sendet Authentication Requests für Antworten von der Outstation und Unsolicited Responses. Default: nicht aktiv.

ADVANCED

Parameter	Beschreibung
Max. APDU size	Maximale Größe eines DNP3 Application Layer Fragments ▸ Minimum: 249 ▸ Maximum: 65535 Default: 2048
Timesync LAN	▸ *Aktiv*: Zeitsynchronisation über das LAN. Nur für TCP-Verbindung projektierbar.

Parameter	Beschreibung
	Ist als Verbindungsart Seriell (auf Seite 23) gewählt, ist die Eigenschaft im Treiberdialog automatisch aktiviert und ausgegraut. **Hinweis:** Verwendet die Variation 3 des Zeitobjekts. Diese wird nicht von allen Stationen akzeptiert.
No class poll on IIN1.1, IIN1.2, IIN1.3	Aktiv: Treiber ignoriert die Internal Indication Flags IIN1.1, IIN1.2 und IIN1.3. Kompatibilitätseinstellung. Stellen Sie sicher, dass regelmäßig für Events gepollt wird, um eventuelle Pufferüberläufe in der Outstation zu vermeiden.
No time sync on IIN1.4	*Aktiv*: Treiber ignoriert das Internal Indication Flag IIN1.4. Kompatibilitätseinstellung. Zeitsynchronisation nötig.
No integrity poll on IIN2.3	*Aktiv*: Treiber ignoriert das Internal Indication Flag IIN2.3 (Buffer Overflow). Kompatibilitätseinstellung.
Delay Measurement (Timesync)	Wenn aktiv, führt der Treiber bei der Zeitsynchronisierung ein Delay Measurement durch. Aktiv wenn: ▸ Verbindungstyp Serial ▸ Verbindungstyp IP (TCP/UDP) **und Timesync LAN** nicht aktiv. **Hinweis:** Bei einer seriellen Verbindung ist zwar **Timesync_LAN** per default aktiv und ausgegraut, **Delay Measurement (Timesync)** aber wählbar.

FILE TRANSFER

Einstellungen für die Dateiübertragung

Parameter	Beschreibung
File transfer directory	Ordner für den Dateitransfer (lesend). Geben Sie hier den Ordner am Runtime-Rechner an, in dem die Dateien für die Dateiübertragung (*File transfer*) abgelegt werden sollen. Die Daten werden von der Outstation zum Master kopiert. Als Master wird jener Rechner bezeichnet, auf dem die zenon Runtime läuft. Dieser Pfad muss manuell eingegeben werden. Achten Sie

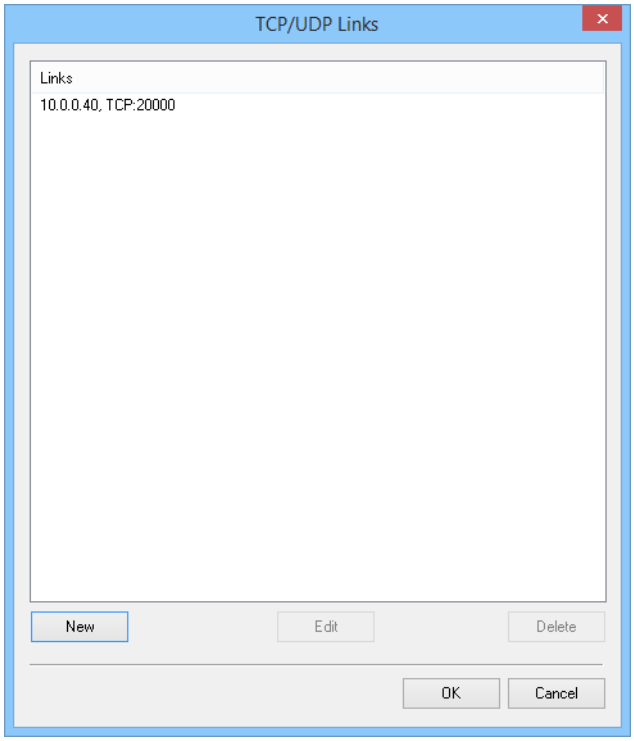
Parameter	Beschreibung
	darauf, dass dieser Pfad auch auf dem Zielsystem vorhanden sein muss. Existiert dieser Pfad auf dem Zielsystem nicht, wird die Dateiübertragung fehlschlagen. Es ist nur die Eingabe eines Absolupfades zulässig. Die Eingabe wird überprüft. Bei Fehlkonfiguration wird ein entsprechender Warndialog angezeigt. Achten Sie auch darauf, dass die entsprechenden Zugriffsrechte vorhanden sind. **Hinweis:** Für **File transfer directory** und **Reverse file transfer directory** kann auch dasselbe Verzeichnis verwendet werden.
Reverse file transfer directory	Geben Sie hier den Ordner am Runtime-Rechner an, in dem die Dateien für die Dateiübertragung in Schreibrichtung abelegt sind. In diesem Ordner werden jene Dateien abgelegt, welche vom Master (zenon) zur Outstation gesendet werden sollen. Dieser Pfad muss manuell eingegeben werden. Achten Sie darauf, dass dieser Pfad auch auf dem Zielsystem vorhanden sein muss. Existiert dieser Pfad auf dem Zielsystem nicht, wird die Dateiübertragung fehlschlagen. Es ist nur die Eingabe eines Absolupfades zulässig. Die Eingabe wird überprüft. Bei Fehlkonfiguration wird ein entsprechender Warndialog angezeigt. Achten Sie auch darauf, dass die entsprechenden Zugriffsrechte vorhanden sind. **Hinweis:** Für **File transfer directory** und **Reverse file transfer directory** kann auch dasselbe Verzeichnis verwendet werden.
File transfer timeout [s]	Zeitraum, in welchem versucht wird, die Dateiübertragung abzuschließen. Nach Ablauf dieser Zeit wird die Dateiübertragung mit einem Fehler auf dem Master abgebrochen. Default: *5 Sekunden*

DIALOG BEENDEN

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

6.2.4.1.1 TCP/UDP Links

Anzeige und Verwaltung aller TCP/UDP Links.



Parameter	Beschreibung
Links	Zeigt die konfigurierten TCP/UDP Kanäle an.
New	Öffnet Dialog zum Anlegen eines neuen TCP/UDP Kanals.
Edit	Öffnet Dialog zum Bearbeiten des ausgewählten TCP/UDP Kanals.
Delete	Löscht ausgewählten TCP/UDP Link.
OK	Übernimmt Änderungen und schließt Dialog.
Cancel	Verwirft alle Änderungen und schließt Dialog.

NEUEN TCP/UDP KANAL ANLEGEN

1. Klicken Sie auf die Schaltfläche **New**.
2. Tragen Sie die Verbindungsdetails ein.
3. Klicken Sie auf **OK.**

TCP/UDP KANAL BEARBEITEN

1. Wählen Sie in der Liste den gewünschten Link.
2. Klicken Sie auf die Schaltfläche **Edit**.
3. Ändern Sie die Linkparameter.
4. Schließen Sie mit **OK** ab.

TCP/UDP KANAL LÖSCHEN

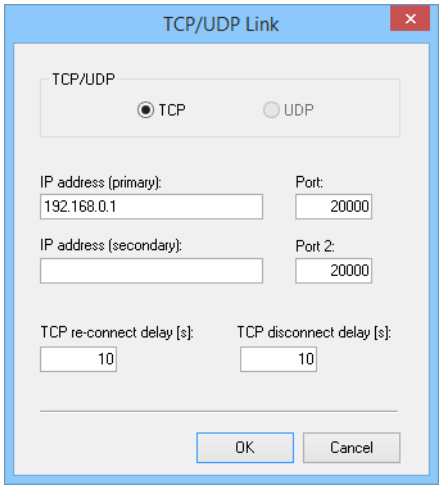
1. Wählen Sie in der Liste den gewünschten Link.
2. Klicken Sie auf die Schaltfläche **Delete**.
3. Die Verbindung wird aus der Liste gelöscht.

Achtung

Es kann pro IP Addresse nur eine Outstation mit einer Portnummer konfiguriert werden. Wenn zwei Outstations auf der gleiche IP Addresse mit unterschiedlicher Portnummer angesprochen werden müssen, verwenden Sie einen weiteren Treiber.

6.2.4.1.2TCP/UDP Link Konfiguration

Konfiguration von TCP/UDP Links.



Parameter	Beschreibung
TCP/UDP	Auswahl der Verbindungsart.
TCP	*Aktiv*: TCP-Verbindung.
UDP	*Aktiv*: UDP-Verbindung.
IP address (primary)	Eingabe der primären IP-Adresse.
Port	Primärer UDP/TCP Port der Steuerung. Default: *20000*
IP address (secondary)	Backup-Adresse bei redundanter Anbindung dieser Steuerung. Nur für TCP-Verbindungen erlaubt. Nicht in Kombination mit TCP Dual-Endpoint möglich.
Port (secondary)	Port der Backup-Adresse.
TCP re-connect delay [s]	Minimale Zeitspanne in Sekunden, die zwischen zwei TCP Verbindungsversuchen liegen muss.
TCP disconnect delay [s]	Werden über die Verbindung über die angegebene Zeitspanne keine Daten gesendet oder empfangen, wird die TCP Verbindung automatisch abgebaut. Kann auch ohne TCP Dual-Endpoint verwendet werden. In diesem Fall müssen die Polling Intervalle entsprechend konfiguriert werden.

Parameter	Beschreibung
	▸ *0*: deaktiviert

DIALOG BEENDEN

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

Verbindungsaufbau

Die konfigurierten Verbindungen der Eigenschaften für die primäre Verbindung **IP address (primary)** sowie für die sekundäre Verbindung **IP address (secondary)** verhalten sich folgendermaßen:

- Während des Starts:
  - Ist die primäre Verbindung nicht verfügbar, wird automatisch versucht eine Verbindung zur sekundären Verbindung aufzubauen.
  - Sind beide projektierten Verbindungen nicht verfügbar, versucht der Treiber kontinuierlich eine Verbindung zu einer der beiden Adressen aufzubauen. Dabei versucht er abwechselnd eine Verbindung zur primären oder sekundären Adresse herzustellen.
- Verbindungsausfall während des laufenden Betriebs:
  - Ist die primäre Verbindung nicht mehr verfügbar, versucht der Treiber eine Verbindung zur sekundären Verbindung aufzubauen.
  - Ist die sekundäre Verbindung nicht mehr verfügbar, versucht der Treiber eine Verbindung zur primären Verbindung aufzubauen.

7 Variablen anlegen

So werden Variablen im zenon Editor angelegt:

7.1 Variablen im Editor anlegen

Variablen können angelegt werden:

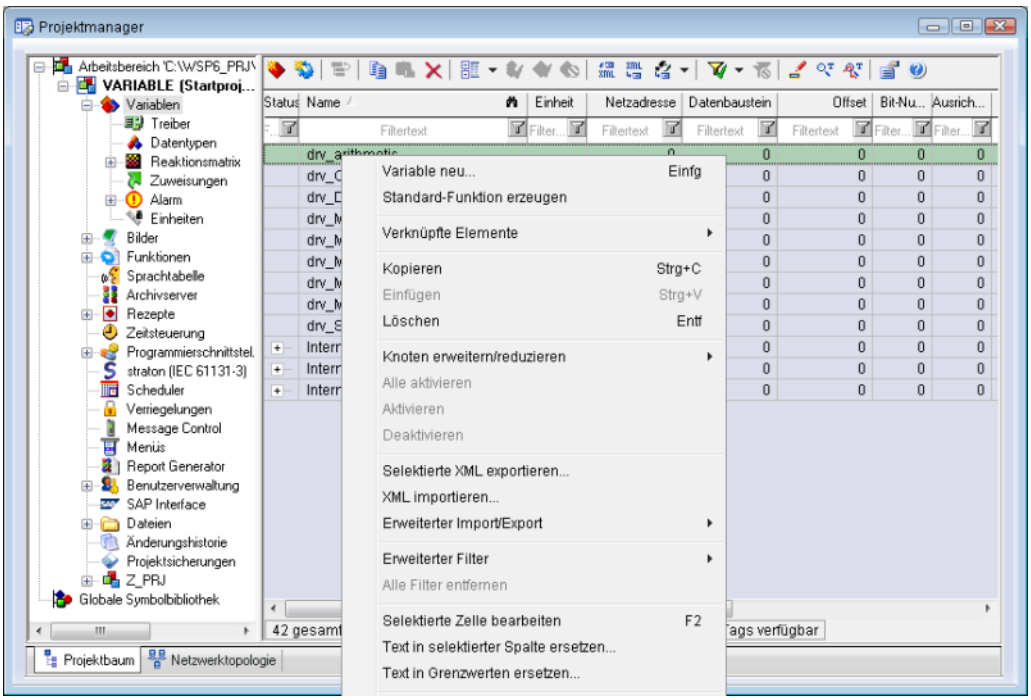
- als einfache Variable
- in Arrays

- als Struktur-Variablen

DIALOG VARIABLE

Um eine neue Variable zu erstellen, gleich welchen Typs:

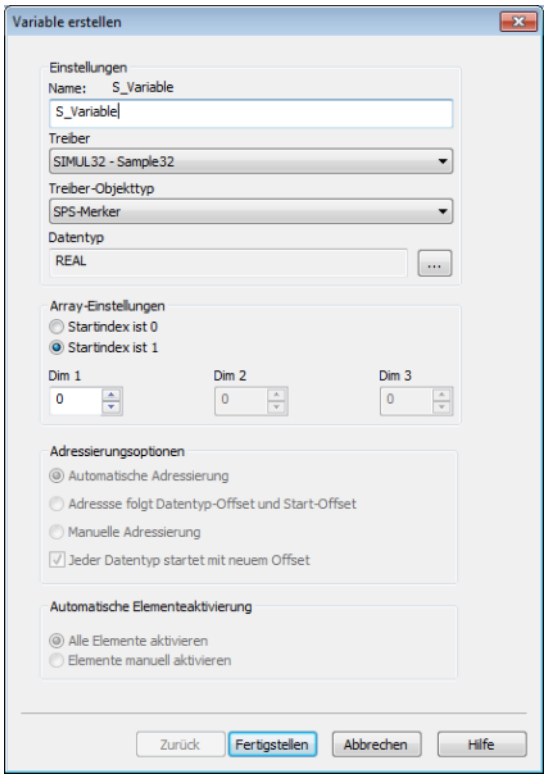
1. Wählen Sie im Knoten **Variablen** im Kontextmenü den Befehl **Variable neu**.



   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.

7.2 Adressierung

Der Offset bestimmt die DNP Point Nummer. In Kombination mit dem Treiberobjekttyp wird die DNP-Objektgruppe bestimmt.

Gruppe/Eigenschaft	Beschreibung
Allgemein	Gruppe mit allgemeinen Eigenschaften.
Name	Frei vergebbarer Name. **Achtung:** Je zenon Projekt muss der Name eindeutig sein.
Kennung	Frei vergebbare Kennung. Z. B. für **Betriebsmittelkennung**, Kommentar usw.
Adressierung	Eigenschaften der Variablenadressierung.
Netzadresse	Netzadresse der Variablen. Diese Adresse bezieht sich auf die Netzadresse der Verbindungsprojektierung im Treiber. Damit wird ausgewählt auf welcher Steuerung sich die Variable befindet. Achtung: Die DNP Adresse der Outstation wird bei der Verbindung separat konfiguriert.
Datenbaustein	Nur für Variablen des Treiberobjekttyps *Device Attribute*. In diesem Fall wird über den Datenbaustein der Index konfiguriert.
Offset	Für Variablen aller Treiberobjekttypen außer **Device Attribute:** DNP Point in der Outstation. Immer in Kombination mit dem Treiberobjekttyp zu sehen. Zum Beispiel: **Analog Input**, **Group** *20*, **Point** *0* entspricht hier **Offset** *0* mit einer Variable vom Typ *Analog Input*. Für Variablen des Treiberobjekttyps **Device Attribute**:

Gruppe/Eigenschaft	Beschreibung
	Variation (Art des Device Attribute).
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.
Treiber Anbindung	Treiberspezifische Eigenschaften.
Treiberobjekttyp	Objekttyp der Variablen. Wird abhängig vom verwendeten Treiber beim Erstellen der Variablen ausgewählt und kann hier geändert werden.
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.
Command Mode	Nur für *Binary Output* und *Analog Output*. Bestimmt, ob der Treiber ein *Direct operate*, *Direct operate no-ack* oder ein *Select Before Operate* durchführt. Default: *Direct Operate*
Classless read	Zuweisung der Werte an die Variable bei expliziten Leseanfragen: ▸ *Inaktiv*: Werte, die über Integrity-Poll, Event-Poll oder unsolicited responses empfangen wurden, werden der Variable zugewiesen. Werte, die als Antwort auf eine explizite Leseanfrage empfangen wurden, werden nicht zugewiesen ▸ *Aktiv*: Der Variablen werden nur Werte zugewiesen, die als Antwort auf eine explizite Leseanfrage empfangen wurden.

7.3 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.

7.3.1 Treiberobjekte

Folgende Objekttypen stehen in diesem Treiber zur Verfügung:

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
Absolute time	77	X	--	*UDINT,REAL*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit dem Befehl **Variablen von Treiber importieren** angelegt werden. Achten Sie auf die korrekte **Netzadresse**.
Analog input	68	X	--	*UDINT, DINT, REAL, LREAL, UINT, INT*	
Analog Input reporting deadbands	78	X	X	*UINT, UDINT, REAL*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit dem Befehl **Variablen von Treiber importieren** angelegt werden. Achten Sie auf die korrekte **Netzadresse**.
Analog output	69	X	X	*UDINT, DINT, REAL, LREAL, UINT, INT*	
Binary Input	64	X	--	*BOOL*	
Binary Output	65	X	X	*BOOL, USINT*	
Command	11	X	X	*STRING*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
					dem Befehl **Variablen von Treiber importieren** angelegt werden.
Counter	66	X	X	*BOOL, UDINT, DINT, USINT, UINT, INT, SINT*	
Device attributes	72			*INT, INT, INT, DINT, EAL, REAL, STRING*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit dem Befehl **Variablen von Treiber importieren** angelegt werden. Achten Sie auf die korrekte **Netzadresse**.
Double-bit binary input	74			*USINT, SINT*	
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. Weitere Informationen dazu finden Sie im Kapitel Kommunikationsdetails (Treibervariablen) (auf Seite 76).

Treiberobjekttyp	Kanaltyp	Lesen	Schreiben	Unterstützte Datentypen	Kommentar
File transfer result	36	X	--	*STRING*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit dem Befehl **Variablen von Treiber importieren** angelegt werden. Achten Sie auf die korrekte **Netzadresse**.
Frozen Counter	67	X	--	*BOOL, UDINT, DINT, USINT, UINT, INT, SINT*	
Internal indications	73	X	--	*BOOL*	
Legacy double bit binary input	71	X	--	*USINT*	**Hinweis:** Diese Variable muss manuell angelegt werden. Sie wird nicht angeboten, wenn Variablen im Treiber mit dem Befehl **Variablen von Treiber importieren** angelegt werden. Achten Sie auf die korrekte **Netzadresse**.
Security statistics	76	X	--	*DINT, UDINT*	
Statistics	9	X	--	*UDINT*	Lokale Kommunikationstatistiken
String	70	X	X	*STRING*	
Virtual terminal output block	75	X	X	*STRING*	

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.

7.3.2 Zuordnung der Datentypen

DATENART

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

TREIBER OBJEKTTYPEN UND UNTERSTÜTZTE IEC DATENTYPEN FÜR VARIABLEN IN ZENON

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
Binary Input *Static object group 1* *Event object group 2*	*64*	*BOOL*	X	--	
Legacy double-bit binary input	*71*	*USINT*	X	--	USINT-Mapping von zwei binären Status mit aufeinanderfolgenden Offsets. (Kein Double-Bit Binary Input! Group 3 / 4.)
Double-bit binary input *Static object group 3* *Event object group 4*	*74*	*USINT*	X	--	

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
Binary Output *Static object group 10* *Event object Group 11* *CROB object group 12*	*65*	*BOOL*	--	X	Wert 1 : LATCH_ON Wert 0 : LATCH_OFF
Binary Output *Static object group 10* *Event object group 11* *CROB object group 12*	*65*	*USINT*	--	X	Wert 1 : PULSE_ON Wert 2 : PULSE_OFF Wert 3 : LATCH_ON Wert 4 : LATCH_OFF Wert 65: CLOSE Wert 129: TRIP Wert 1 bei PULSE_ON, LATCH_ON und CLOSE, Wert 0 bei PULSE_OFF, LATCH_OFF und TRIP
Counters *Static object group 20* *Event object group 22*	*66*	*UINT, UDINT*	X	-	
Frozen Counters *Static object group 21* *Event object group 23*	*67*	*UINT, UDINT*	X	--	
Analog Input *Static object group 30*	*68*	*INT, DINT, REAL, LREAL*	X	--	

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
Event object group 32					
Analog Output *Static object group 40* *Event object group 42* *Analog Output Block object group 41*	*69*	*INT, DINT, REAL, LREAL*	--	X	geschriebener Wert wird nach erfolgreichem Schreiben als Rückmeldung gespiegelt
Octet strings *Static object group 110* *Event object group 111*	*70*	*STRING*	X	X	geschriebenen Wert wird nicht gespiegelt, somit erst nach einem Update aktualisiert.
Virtual terminal output block *Static object group 112* *Event object group 113*	*75*	*STRING*	X	X	geschriebenen Wert wird nicht gespiegelt, somit erst nach einem Update aktualisiert.
Device Attributes *Group 0*	72	INT, INT, INT, DINT, EAL, REAL, STRING	X	X	Device Attributes können geschrieben werden. Entnehmen Sie IEEE 1815 welchen Device Attributes aus dem Standard Satz geschrieben werden können.
Internal indications *Object group 80*	73	BOOL	X	--	
Security statistics	76	DINT, UDINT	X	--	

Treiber Objekttypen	Kanaltyp	Unterstützte Datentypen (Datenart)	Lesen	Schreiben	Kommentar
Static object group 121 *Event object group 122*					
Statistics	9	UDINT	X	--	Lokale Kommunikationstatistiken
Command	11	STRING	X	X	Initiiert ein explizites Kommando z.B. explizites Lesen, Counter Freeze, ...
Absolute time Object Group 50	77	LREAL, DINT	X	--	Wird nur explizit über Kommando gelesen. "Classless Read" muss bei der Variable aktiviert sein.
Analog Input reporting deadbands *Object Group 34*	78	REAL, UINT, UDINT	X	X	Wird nur explizit über Kommando gelesen. "Classless Read" muss bei der Variable aktiviert sein.
File transfer result	36	STRING	X	--	Auf dieser Variable wird das Ergebnis folgender Dateitransferaktionen abgebildet: ▸ *GET_FILE_INFORMATION* ▸ *READ_DIRECTORY*

BINARY OUTPUTS

Das Schreiben von *Binary Outputs* erfolgt immer über ein **CROB** (Gruppe 12), wahlweise mit *Direct Operate* oder *Select Before Operate*, oder *Direct Operate No Ack*. Dabei wird die Einstellung bei der variablenspezifischen Eigenschaft **Command Mode** (über XML-Export/-Import und VBA zugänglich über **COMMAND_MODE**) verwendet. Direktes Schreiben auf Gruppe *10* wird vom Treiber nicht unterstützt.

Nach dem erfolgreichem Schreiben enthält die Variable den Wert *0* für *Binary Outputs* bei *PULSE_OFF*, *LATCH_OFF* und *TRIP*, und Wert *1* bei *PULSE_ON*, *LATCH_ON* und *CLOSE*. Der Wert wird ebenfalls aktualisiert wenn ein *Binary Output Status* (Gruppe 10) oder *Binary Output Status Event* (Gruppe 11) Objekt empfangen wird.

Befehlsgabe mit *Binary Output* Variablen:

- Es dürfen ausschließlich Befehle vom Typ **Doppelkommando** bzw. **Schaltbefehl** verwendet werden.
- Die Eigenschaft **Select Before Operate** bei der Variable muss deaktiviert sein; der DNP3_TG Treiber verwendet bei Bedarf *Auto-SBO*.

Es wird empfohlen den **Qualifier of Command** bei der Befehlsgabe zu verwenden, um den Typ zu bestimmen:

Typ	QoC
PULSE OFF/PULSE ON	*1*
LATCH OFF/LATCH ON	*0*
TRIP/CLOSE	*2*

Die Option **Qualifier of Command** kann sowohl bei *Binary Output* Variablen vom Datentyp *BOOL* als auch vom Datentyp *USINT* verwendet werden.

ANALOG OUTPUTS

Das Schreiben von *Analog Outputs* erfolgt immer über ein Objekt der Gruppe 41, wahlweise mit *Direct Operate*, *Select Before Operate* oder *Direct Operate No Ack*. Dabei wird die Einstellung bei der variablenspezifischen Eigenschaft **Command Mode** (über XML-Export/-Import und VBA zugänglich über **COMMAND_MODE**) verwendet.

Nach dem erfolgreichen Schreiben enthält die Variable zuerst den in der Runtime geschriebenen Wert. Der Wert wird ebenfalls aktualisiert wenn ein *Analog Output Status* (Gruppe 40) oder *Analog Output Status* (Group 42) Objekt empfangen wird.

Verwenden Sie die den Aktionstyp *Sollwert Setzen* für *Analog Outputs*. Entsprechend der Option **Command Mode** bei der Variable wird auch für *Analog Outputs* entweder ein *direct operate* oder ein *select before operate* durchgeführt. Die Option **Select Before Operate** bei der Variable muss

deaktiviert sein. Die Option **Qualifier of Command** steht beim Aktionstyp *Sollwert Setzen* nicht zur Verfügung und hat keinen Einfluss bei *Analog Outputs*.

Bei zweistufiger Befehlsgabe wird erst bei der zweite Stufe der Befehl zur Outstation gesendet, inklusive **Select** bei *Auto-SBO*.

VERHALTEN VON COMMAND MODE

Für die Eigenschaft **Command mode** bei einer Variablen gilt:

- *direct Operate*: Der Wert wird direkt geschrieben. (Default)
- *automatic SBO*: Beim Schreiben vom Stack wird zunächst ein Select gesendet, und nach positiver Antwort mit sofortigem *Operate* bestätigt.

Das gilt für binäre und analoge Outputs.

ZUORDNUNG VON SOLLWERSETZEN AUF BINARYOUTPUT OHNE BEFEHLSGABE

- Bei einer Binary Output Variable vom Datentyp BOOL wird *LatchON* für *High* oder *LatchOFF* für *Low* gesendet.
- Bei einer Binary Output Variable vom Datentyp USINT wird das Sollwert setzen gemäß folgender Tabelle behandelt

Wert USINT	Aktion	Kommentar
0	*keine*	
1	*Pulse On*	
2	*Pulse Off*	Nicht vollständig kompatibel. Wird nicht zwingend von der Outstation unterstützt.
3	*Latch On*	
4	*Latch Off*	
65	*Close*	
129	*Trip*	

Auch für das direkte Sollwertsetzen ohne Befehlsgabe wird die Eigenschaft **Command Mode** berücksichtigt.

REGELN FÜR DAS SETZEN DER STATUSBITS BEI EINSATZ DER BEFEHLSGABE

Statusbits werden entsprechend den Regeln in der Tabelle gesetzt:

- Sobald ein *Select* gesendet wurde, werden *SE* und *CoT_act* gesetzt.
- Trat beim Senden ein Fehler auf oder wurde eine negative Antwort erhalten, werden *SE*, *P/N* und *CoT_actcon* gesetzt (*4*).
- Nach einer positiven Antwort ist der Status *SE* und *CoT_actcon* (*1*, *2*, *3*).
  Danach wird automatisch das *Operate* gesendet und der Status auf *CoT_act* gesetzt.
  Tritt jetzt beim Senden ein Fehler auf oder wurde eine negative Antwort erhalten, werden *P/N* und *CoT_actcon* gesetzt (*2*).
- Ist die Antwort auf das *Operate* positiv, wird der Status erst auf *CoT_actcon* und dann auf *CoT_actterm* gesetzt (*1*).
- Der Status nach einem *DirectOperate* wird ebenso behandelt (*1*, *2*, *3*).
- Bei einem Fehler im Senden wird auch das Invalid-Bit gesetzt (*2*, *4*).

Fall	Aktion	Status	Erfolg	Status	Aktion	Status	Erfolg	Status	Status
1	*select*	*act SE*	*ack*	*actcon SE*	*operate*	*act*	*ack*	*actcon*	*actterm*
2	*select*	*act SE*	*ack*	*actcon SE*	*operate*	*act*	*nack*	*actcon P/N*	
3	*select*	*act SE*	*ack*	*actcon SE*					
4	*select*	*act SE*	*nack*	*actcon SE P/N*					

Bedeutung der Begriffe in der Spalte **Erfolg**:

- ack: positiv
- nack: negativ
- t/o: Timeout

LEGACY DOUBLE-BIT BINARY INPUT

Mit dem Treiberobjekttyp *Legacy double-bit binary input* führt der Treiber ein Mapping durch, abhängig von der Einstellung in der Treiberkonfiguration.

Zum Beispiel: Bei einem USINT Offset 14 werden die binären Status mit den Offsets 14 und 15 kombiniert.

Achtung: Das Mapping zweier *Binary inputs* auf ein *Legacy double-bit binary input* ist eine treiberinterne Funktion und darf nicht verwechselt werden mit der DNP3 *Gruppe 3/4*, Treiberobjekttyp *Double-bit binary input*.

ERGEBNISSE

Binäre Status-Bits SCADA default mapping	Binäre Status-Bits DNP3 binary logic mapping	Binäre Status-Bits Custom legacy mapping	Status in zenon (USINT mit *Binary Inputs Double)*
▸ beide Bits: *off (false)*	▸ erstes Bit (**Offset** 14): *off (false)* ▸ zweites Bit (**Offset** 15): *on (true)*	▸ erstes Bit (**Offset** 14): *on (true)* ▸ zweites Bit (**Offset** 15): *off (false)*	*2 - intermediate*
▸ beide Bits: *on (true)*	▸ beide Bits: *on (true)*	▸ beide Bits: *on (true)*	*3 - faulty*
▸ erstes Bit (**Offset** 14): *on (true)* ▸ zweites Bit (**Offset** 15): *off (false)*	▸ beide Bits: *off (false)*	▸ beide Bits: *off (false)*	*0 - off*
▸ erstes Bit (**Offset** 14): *off (false)* ▸ zweites Bit (**Offset** 15): *on (true)*	▸ erstes Bit (**Offset** 14): *on (true)* ▸ zweites Bit (**Offset** 15): *off (false)*	▸ erstes Bit (**Offset** 14): *off (false)* ▸ zweites Bit (**Offset** 15): *on (true)*	*1 - on*

DEVICE ATTRIBUTES

Device Attributes müssen über ein explizites Lesekommando gelesen werden (siehe **Command**). Der Befehl "READ_VAR 1 0 254" liest alle Device Attributes für Netzaddresse "1"Der Index wird über die Datenbaustein Einstellung der Variable festgelegt. Die **Variation** entspricht dabei dem Offset der Variable. Der Benutzer muss selbst den richtigen Datentyp wählen. Beim Anlegen muss die Netzadresse entsprechend der Stationsnummer eingestellt werden. Die Eigenschaft "klassenloses Lesen" muss aktiviert werden. Device Attributes können derzeit nur gelesen werden.

LOKALE KOMMUNIKATIONSSTATISTIK (STATISTICS)

Über die Kommunikationsstatistik kann der Zustand des lokalen DNP3-Controllers überwacht werden. Die Adressierung der Variablen erfolgt über **Netzadresse** und **Offset** In der folgenden Tabelle sind alle verfügbaren Statistikvariablen mit deren Offset aufgelistet.

Wird ein physikalischer Data-Link Kanal von mehreren virtuellen Outstations verwendet, teilen sich alle diese virtuellen Outstations die Variablen der Klassen Data-link (phys) und TCP.

Die Adressierung erfolgt über die Netzadresse einer beliebigen virtuellen Outstation.

TCP state (Offset 204): Wert 0 bedeutet keine Verbindung, Wert 2 bedeutet Verbindung wird hergestellt, Wert 3 bedeutet Verbindung ist hergestellt

TCP flags (Offset 206): Wert 5 bedeutet Verbindung hergestellt über primäre IP Addresse, Wert 6 bedeutet Verbindung hergestellt über Sekundäre IP Addresse

Statistik Klasse	Statistik Detailtyp	Offset
Data-link (virt)	*Bytes received*	*500*
Data-link (virt)	*Bytes sent*	*501*
Data-link (virt)	*Frames received*	*502*
Data-link (virt)	*Frames sent*	*503*
Data-link (virt)	*Frames discarded*	*504*
Data-link (virt)	*Link status timeouts*	*505*
Data-link (virt)	*Link status errors*	*506*
Data-link (virt)	*Frame errors*	*507*
Transport	*Fragments received*	*600*
Transport	*Segments received*	*601*
Transport	*Assembling errors*	*602*
Transport	*Fragments sent*	*603*
Transport	*Segments sent*	*604*
Application layer (connection)	*Failures*	*700*
Application layer (connection)	*Sequence errors*	*701*
Application layer (connection)	*Retries*	*702*
Application layer (connection)	*Timeouts*	*703*
Application layer (connection)	*Invalid fragments*	*704*

Statistik Klasse	Statistik Detailtyp	Offset
Application layer (connection)	*Unsolicted responses*	*705*
Authentication	*Unexpected messages*	*800*
Authentication	*Authorization failures*	*801*
Authentication	*Authentication failures*	*802*
Authentication	*Reply timeouts*	*803*
Authentication	*Rekeys due to authentication failures*	*804*
Authentication	*Total messages sent*	*805*
Authentication	*Total messages received*	*806*
Authentication	*Critical messages sent*	*807*
Authentication	*Critical messages received*	*808*
Authentication	*Discarded messages*	*809*
Authentication	*Error messages sent*	*810*
Authentication	*Error messages received*	*811*
Authentication	*Successful authentications*	*812*
Authentication	*Session key changes*	*813*
Authentication	*Failed session key changes*	*814*
Authentication	*Rekeys due to restart*	*815*
Application layer (state)	*Integrity polls*	*900*
Application layer (state)	*Event class 1 polls*	*901*
Application layer (state)	*Event class 2 polls*	*902*
Application layer (state)	*Event class 3 polls*	*903*
Application layer (state)	*Commands succeeded*	*904*
Application layer (state)	*Commands failed*	*905*
Application layer (state)	*Response object count*	*906*
Application layer (state)	*Unsolicted response object count*	*907*

Statistik Klasse	Statistik Detailtyp	Offset
Data-link (phys)	*Bytes received*	*100*
Data-link (phys)	*Bytes sent*	*101*
Data-link (phys)	*Invalid bytes received*	*102*
Data-link (phys)	*Valid frames received*	*103*
Data-link (phys)	*Frames sent*	*104*
Data-link (phys)	*Frame CRC erros*	*105*
Data-link (phys)	*Send timeouts*	*106*
Data-link (phys)	*Receive timeouts*	*107*
TCP	*Connection succeeded primary*	*200*
TCP	*Connection succeeded secondary*	*201*
TCP	*Connection failed primary*	*202*
TCP	*Connection failed secondary*	*203*
TCP	*State*	*204*
TCP	*State counter*	*205*
TCP	*Flags*	*206*

EXPLIZITES KOMMANDO (COMMAND)

Variablen vom Treiberobjekttyp *Command* werden verwendet, um einmalig eine bestimmte Aktion anzustoßen. Die Aktion wird durch das Schreiben eines Stringwerts auf die Kommandovariable initiiert. Das Resultat der Aktion wird der Variablen zugewiesen.

- Syntax: Ein Kommando besteht aus Kommandoname und Parametern. Kommandoname und die einzelnen Parameter werden durch Leerzeichen getrennt.
  Beispiel: READ_VAR 1 0 254 Lesen für Netzaddresse 1 alle Device Attributes (Object Group 0, Variation 254)
  Beispiel: READ 1 1 Lesen für Netzaddresse 1 alle Binary Inputs (Object Group 1, Variation 0 (default variation))

Achtung: Über das zusammensetzen von Parameter für ein Explizites Kommando kann es zu Kombinationen kommen die kein gültiges Kommando im Sinne des IEEE 1815 Standards entsprechen. Kommandos werden ohne weitere Prüfung an die Outstation gesendet und können hier unter Umstände ein Fehlverhalten auslösen.

EXPLIZITES LESEN VARIATION 0

Kommando: READ

Alle Points einer Gruppe lesen: (Qualifier Code 06 - Sie können diese Parameteranzahl für statische Object Groups und Event Object Groups verwenden)

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe

N-Points einer Gruppe lesen: (Qualifier Code 07, 08, 09 - Sie können diese Parameteranzahl für Event Object Groups verwenden. Die Nutzung einer Anzahl grösser als 65535 (Qualifier Code 09) wird nicht empfohlen))

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe
- Parameter 3: Anzahl der zu lesenden Points

Bestimmte Points einer Gruppe lesen: (Qualifier Code 00, 01 - Sie können diese Parameteranzahl für statische Object Groups, zB. Group 1 - Binary Input verwenden)

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe
- Parameter 3: Start-Index
- Parameter 4: Stop-Index

EXPLIZITES LESEN

Kommando: READ_VAR

Alle Points einer Gruppe lesen: (Qualifier Code 06 - Sie können diese Parameteranzahl für statische Object Groups und Event Object Groups verwenden)

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe
- Parameter 3: Variation

N-Points einer Gruppe lesen: (Qualifier Code 07, 08, 09 - Sie können diese Parameteranzahl für Event Object Groups verwenden. Die Nutzung einer Anzahl grösser als 65535 (Qualifier Code 09) wird nicht empfohlen))

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe
- Parameter 3: Variation
- Parameter 4: Anzahl der zu lesenden Points

Bestimmte Points einer Gruppe lesen: (Qualifier Code 00, 01 - Sie können diese Parameteranzahl für statische Object Groups, zB. Group 1 - Binary Input verwenden)

- Parameter 1: **Netzadresse**
- Parameter 2: Gruppe
- Parameter 3: Variation
- Parameter 4: Start-Index
- Parameter 5: Stop-Index

COUNTER FREEZE

- Counter einfrieren - mit Rückmeldung (Function Code 7)
  Kommando: FREEZE
  Parameter 1: **Netzadresse**
- Counter einfrieren - ohne Rückmeldung (Function Code 8)
  Kommando: FREEZE_NR
  Parameter 1: Netzadresse
- Counter einfrieren und zurücksetzen - mit Rückmeldung (Function Code 9)
  Kommando: FREEZE_CLEAR
  Parameter 1: Netzadresse
- Counter einfrieren und zurücksetzen - ohne Rückmeldung (Function Code 10)
  Kommando: FREEZE_CLEAR_NR
  Parameter 1: **Netzadresse**

ASSIGN CLASS

Sendet eine Assign Class Anfrage an die Outstation (Function Code 22)

Zuordnen zu Class 0, deaktiviert die Events. Nach erfolgreichem Aufruf sendet der Treiber automatisch ein Read Request für Event Class 123.

- Klasse einer komplette Object Group (zum Beispiel Binary Input) zuordnen:
  - Kommando: ASSIGN_CLASS
  - Parameter 1: **Netzadresse**
  - Parameter 2: Object Group
  - Parameter 3: Class (1 = Class0, 2 = Event Class1, 3 = Event Class2, 4 = Event Class3)
- Klasse einzelner Points einer Object Group (zum Beispiel: Analog Input mit dem Point Index 4 bis 10) zuordnen:
  - Kommando: ASSIGN_CLASS
  - Parameter 1: **Netzadresse**

- Parameter 2: Object Group
- Parameter 3: Class (1 = Class0, 2 = Event Class1, 3 = Event Class2, 4 = Event Class3)
- Parameter 4: Start-Index
- Parameter 5: Stop-Index

UNSOLICITED RESPONSES AKTIVIEREN/DEAKTIVIEREN

Kommando: ENABLE_UNSOL

- Parameter 1: **Netzadresse**
- Parameter 2: 1 ... Enable / 0 ... Disable
- Parameter 3: Event-Class 1 (1 ... einschließen, 0 ... ausschließen)
- Parameter 4: Event-Class 2 (1 ... einschließen, 0 ... ausschließen)
- Parameter 5: Event-Class 3 (1 ... einschließen, 0 ... ausschließen)

CLASS POLL ANSTOßEN

Kommando: CLASS_POLL

- Parameter 1: **Netzadresse**
- Parameter 1: Class-0 (1 ... einschließen, 0 ... ausschließen)
- Parameter 2: Event-Class 1 (1 ... einschließen, 0 ... ausschließen)
- Parameter 3: Event-Class 2 (1 ... einschließen, 0 ... ausschließen)
- Parameter 4: Event-Class 3 (1 ... einschließen, 0 ... ausschließen)

CLASS POLL AKTIVIEREN/DEAKTIVIEREN

Kommando: CYCLIC_POLL

- Parameter 1: **Netzadresse**
  Wurde eine ungültige Netzadresse angegeben, wird der Befehl abgelehnt.
- Parameter 2: Polling:
  - 0 - deaktiviert
  - 1 - aktiviert

COLD RESTART

Sendet eine Cold Restart Anfrage an die Outstation (Function Code 13)

Nach erfolgreicher Rückmeldung, trennt der Treiber die Verbindung. Der Treiber wartet mit der neue Anfragen an die Outstation. Die Wartezeit ist die von der Outstation gemeldete Verzögerungszeit (in der Rückmeldung enthalten). Nach dieser Wartezeit wird die Verbindung wieder neu aufgebaut.

Kommando: COLD_RESTART

- Parameter 1: **Netzadresse**

ZEITSYNCHRONISATION ANSTOßEN

Die Zeitsynchronisierung wird nach der in den Treibereinstellungen (auf Seite 29) gewählte Methode durchgeführt:

- **Timesync LAN Netzadresse**aktiv
- **Timesync LAN** nicht aktiv
  - **Delay measurement (Timesync)** aktiv
  - **Delay measurement (Timesync)** nicht aktiv

DELAY MEASUREMENT

- Delay Measurement aktiv:
  Wenn die Option Delay measurement aktiv ist, wird beim Kommando **TIMESYNC** der Request für **Delay Measurement (FC23)** gesendet.
  Kommando: TIMESYNC
  Parameter 1: **Netzadresse**

TCP-REDUNDANZUMSCHALTUNG AUSLÖSEN

Achtung: Hat nur Effekt wenn gerade eine aktive Verbindung besteht (statisitics variable tcp State, offset 206, Wert 3). Wenn gerade keine Verbindung aktiv ist weil aufgrund des Idle disconnect die Verbindung getrennt wurde, wird der Befehl ignoriert. Der nächste Verbindungsversuch geht in diesem Fall wieder an die zuletzt verwendete IP-Addresse.

Über dieses Kommando kann aktiv von der Primäre IP-Addresse auf die Sekundäre IP-Addresse, oder umgekehrt, gewechselt werden. Die aktive Verbindung wird getrennt und beim nächsten Verbdinungsversuch wird die jeweils alternative IP Addresse verwendet.

KOMMANDO: RED_SWITCH

- Parameter 1: **Netzadresse**

Achtung: Im Fall von virtuellen Outstations gilt die Redundanzumschaltung für alle Outstations, die sich einen IP-Kanal teilen.

Achtung: Es wird davon ausgegangen, dass die Outstations verbunden über die primary IP-Addresse und secondary IP-Addresse über die gleiche Points Konfiguration verfügen und über einen eigenen Kanal oder mit dem Prozess synchronisiert sind.

KANAL STATISTIK ZURÜCKSETZEN

Setzt die Data-link (phys) und TCP Statistik für einen Kommunikationskanal zurück.

Kommando: RESET_LINK_STAT

- Parameter 1: **Netzadresse**

Achtung: Im Fall von virtuellen Outstations gilt das Rücksetzen der Statistik für alle Outstations, die sich einen IP-Kanal teilen.

OUTSTATION STATISTIK ZURÜCKSETZEN

Kommando: RESET_OUTSTATION_STAT

- Parameter 1: **Netzadresse**

KOMMANDO STATUSCODES

Statuscode	Beschreibung
3	Kommando wurde versandt
4	Kommando erfolgreich abgeschlossen
5	Kommando konnte nicht verarbeitet werden
6	Kommando wurde negativ beantwortet
7	Fehler beim Ausführen des Kommandos (Timeout, Authentication failed)
8	Ausführung des Kommandos / File Transfer wurde abgebrochen
10	File Transfer: Timeout aufgetreten
11	File Transfer: Speichern der lokalen Datei fehlgeschlagen
12	File Transfer: Abort fehlgeschlagen
13	File Transfer: Abbruch, mehr Daten empfangen als gemeldete Dateigröße
16	Syntaxfehler
17	Unbekanntes Kommando
18	**Netzadresse** nicht vorhanden

Statuscode	Beschreibung
19	File Transfer: wird bereits ausgeführt
20	File Transfer: Verzeichnis Fehler oder nicht konfiguriert
21	File Transfer: Abort fehlgeschlagen, kein File Transfer aktiv

7.4 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.

7.4.1 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**.

7.4.2 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

Bezeichnung	Typ	Feldgröße	Bemerkung
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 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)

Bezeichnung	Typ	Feldgröße	Bemerkung
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
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)
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

Bezeichnung	Typ	Feldgröße	Bemerkung
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.

7.4.3 Online-Import

REGELN FÜR DEN DNP3_TG ONLINE-IMPORT

Beim Online-Import gilt:

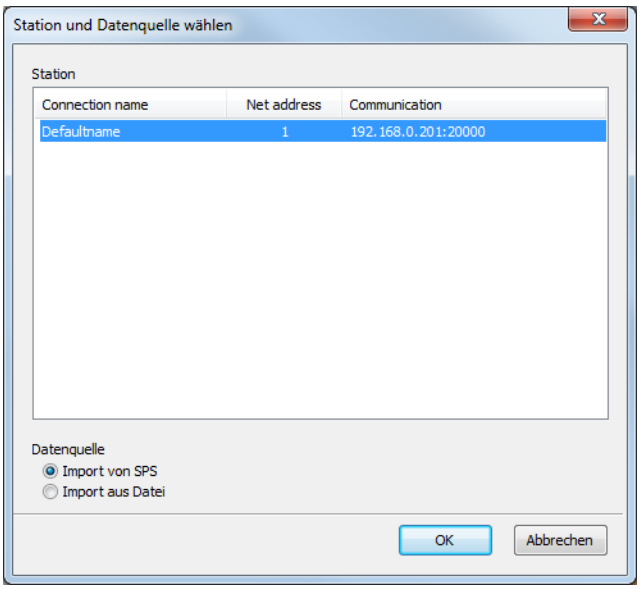
- Die Antwort auf einen Integritypoll wird ausgewertet.
- Der Variablenname wird aus Netzadresse, Verbindungsname, Gruppennummer und Index gebildet.
- Die Kennung enthält Netzadresse, Index und Beschreibung des Objekttyps.
- Achten Sie darauf, dass die Runtime nicht aktiv ist, wenn sie einen Online-Import starten. Die Outstation unterstützt unter Umständen nur einen Master oder nur eine Verbindung vom gleichen Rechner.
- Achten Sie darauf die Antwort-Timeout in der Treiberkonfiguration entsprechend höher einzustellen, wenn Sie bei langsamer (serieller) Verbindung eine Outstation mit großer Point Database verwenden .

ONLINE-IMPORT ALLGEMEIN

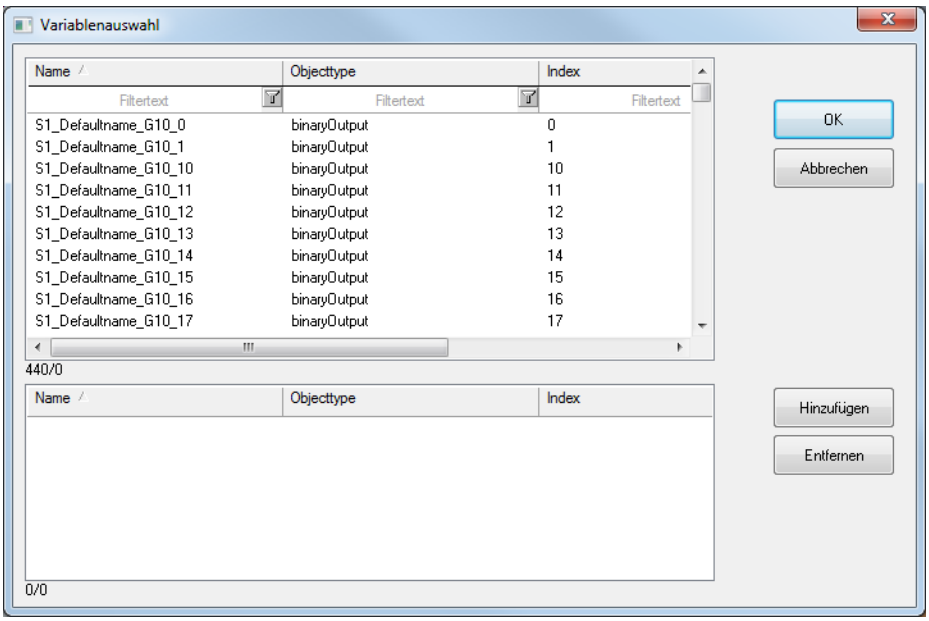
Um Variablen online von der Steuerung zu importieren:

1. Wählen Sie den Treiber aus.
2. Wählen Sie in der Symbolleiste oder im Kontextmenü **Variablen vom Treiber importieren**.

3. Der Dialog für den Import wird geöffnet:

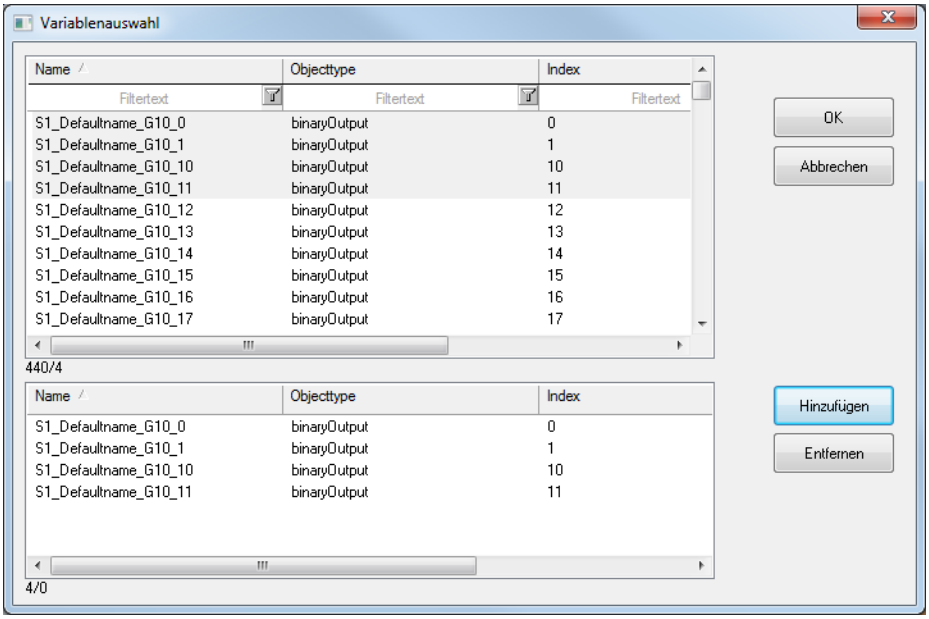


4. Wählen Sie die gewünschte Verbindung.
5. Wählen Sie **Auswahl von SPS**.
6. Bestätigen Sie die Auswahl mit Klick auf **OK.**
7. Der Dialog zur Variablenauswahl wird geöffnet:



8. Wählen Sie die gewünschten Variablen (Mehrfachauswahl ist möglich).

9. Fügen Sie die Variablen durch Klick auf die Schaltfläche **Hinzufügen** zur Liste der zu importierenden Variablen hinzu.



10. Mit Klick auf **Entfernen** können Sie Variablen auch wieder abwählen.
11. Starten Sie den Import durch Klick auf die Schaltfläche **OK**.

Die ausgewählten Variablen werden beim Import automatisch im zenon Projekt erstellt und dem ausgewählten Treiber zugeordnet. Dabei wird die Netzadresse der Variablen entsprechend der ausgewählte Station aus der Treiberkonfiguration konfiguriert.

7.4.4 Offline-Import

Der Treiber unterstützt den Offline-Import von Variablen aus einer **DNP3 XML Device Profile** Datei für die Versionen:

- 2.07 (Januar 2012)
- 2.08 (Juli 2012)

REGELN FÜR DEN DNP3_TG OFFLINE-IMPORT

Beim Offline-Import gilt:

- Die Variablendefinition muss einen Namen enthalten.

  Der Variablenname wird aus einem zusammengesetzten Namen aus dem XML Device Profile Dokument nach folgendem Schema gebildet:
  Gerätename_Variablenname
  Es muss sichergestellt sein, dass alle DNP Variablen, auch gruppenübergreifend, im Dokument einen eindeutigen Namen besitzen.

- Falls die Variablendefinition im Dokument ein Feld für **Description** enthält, wird diese Information beim Import in der **Kennung** der Variablen gespeichert. Der Variablenname und die Variablenkennung können nach dem Import geändert werden.
- Ist im Projekt bereits eine Variable mit dem gleichen Namen vorhanden, erhalten Sie beim erneuten Import eine Fehlermeldung. Die Variable wird nicht überschrieben oder gemerged. Diese Fehlermeldung kann auch angezeigt werden, wenn:
  - im **XML Device Profile** Dokument keine eindeutigen Namen verwendet wurden
  - wenn der Gerätename im Dokument identisch mit bereits importierten Variablen ist
- Es werden nur Variablen aus dem **XML Device Profile** Dokument zum Import angeboten, die vom Treiber unterstützt werden.
- **Frozen Counters** sind im **XML Device Profile** Dokument nicht explizit vorhanden. Wenn jedoch beim Counter der Wert für **frozenCounterExists** auf *True* (bzw. *1*) gesetzt ist, wird beim Import auch die Möglichkeit angeboten, Variablen für **Frozen Counter** zu importieren.

NICHT IMPORTIERBARE VARIABLEN

Folgende Variablen werden nicht importiert und müssen manuell erstellt werden:

- *Device attributes*
- Steuervariablen für Klassenpolls, klassenloses Lesen und *Cold Restart*
- *Binary inputs double*

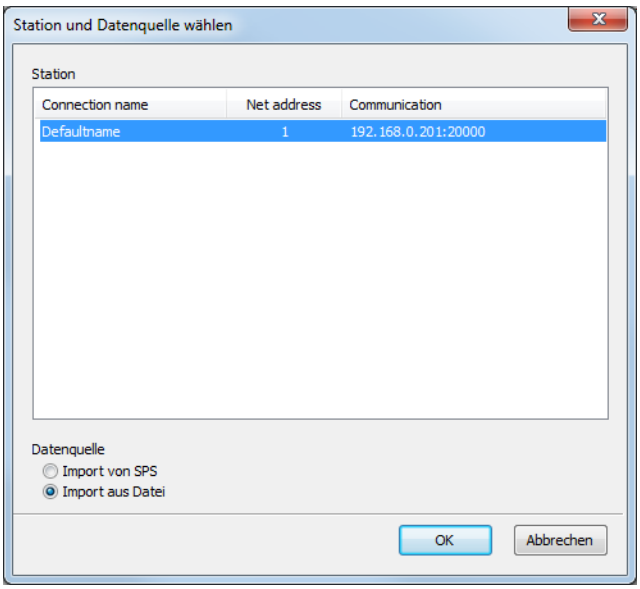
Achten Sie beim manuellen Erstellen dieser Variablen auf die korrekte **Netzadresse**.

OFFLINE-IMPORT ALLGEMEIN

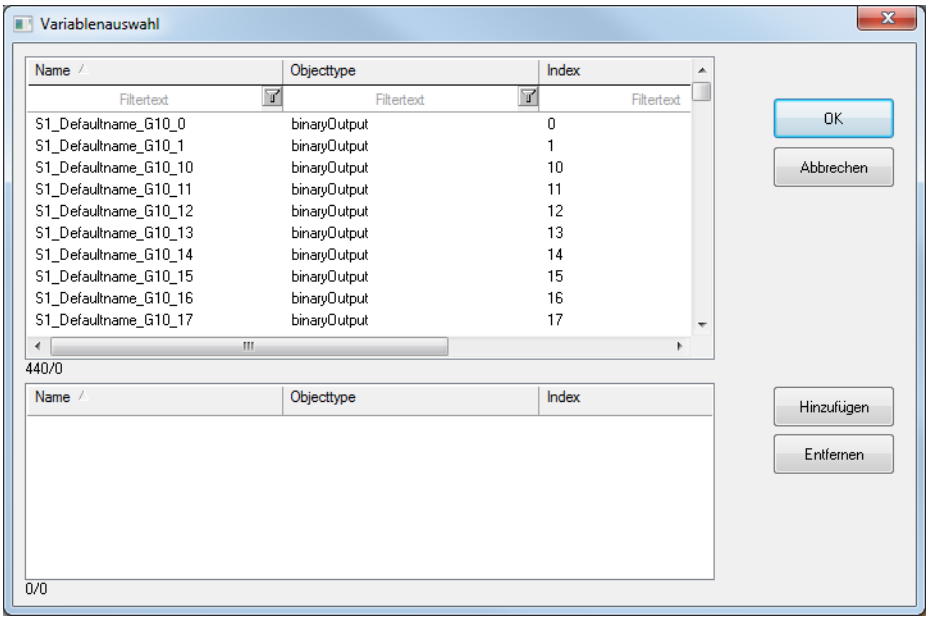
Um Variablen aus einer **DNP3 XML Device Profile** Datei zu importieren:

1. Wählen Sie den Treiber aus.
2. Wählen Sie in der Symbolleiste oder im Kontextmenü **Variablen vom Treiber importieren**.

3. Der Dialog für den Import wird geöffnet:

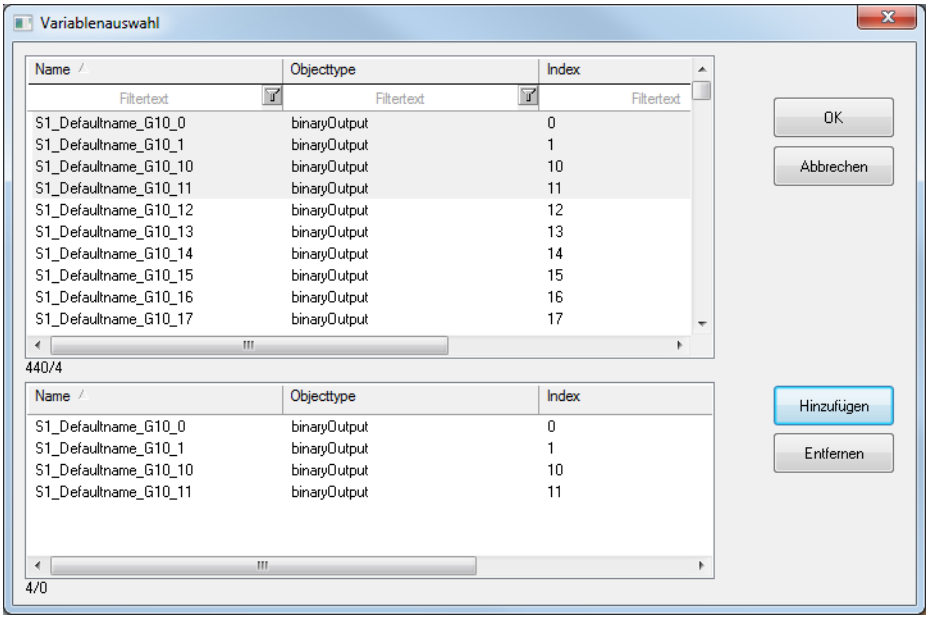


4. Wählen Sie die gewünschte Verbindung.
5. Wählen Sie **Import aus Datei**.
6. Der Dialog zur Dateiauswahl wird geöffnet.
7. Wählen Sie die gewünschte Datei und bestätigen Sie die Auswahl mit Klick auf **OK**.
8. Der Dialog zur Variablenauswahl wird geöffnet.



9. Wählen Sie die gewünschten Variablen (Mehrfachauswahl ist möglich).

10. Fügen Sie die Variablen durch Klick auf die Schaltfläche **Hinzufügen** zur Liste der zu importierenden Variablen hinzu.



11. Mit Klick auf **Entfernen** können Sie Variablen auch wieder abwählen.
12. Starten Sie den Import durch Klick auf die Schaltfläche **OK**.

Die ausgewählten Variablen werden beim Import automatisch im zenon Projekt erstellt und dem ausgewählten Treiber zugeordnet. Dabei wird die Netzadresse der Variablen entsprechend der ausgewählte Station aus der Treiberkonfiguration konfiguriert.

7.5 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.
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.

Name aus Import	Typ	Offset	Erklärung
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>;...;...;** 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.

Name aus Import	Typ	Offset	Erklärung
			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.
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.

Name aus Import	Typ	Offset	Erklärung
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.

Name aus Import	Typ	Offset	Erklärung
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 Fehler auftrat.
ErrorTimeS	*STRING*	2	Zeit (in Sekunden seit 1.1.1970), wann der letzte Fehler als String auftrat.

Name aus Import	Typ	Offset	Erklärung
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.

8 Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:

Funktion	Beschreibung
Blockwrite	Nicht unterstützt.
Redundanz	Wenn die Outstation mehrere Master unterstützt.
RDA	Nicht unterstützt. **Sequence Of Events** wird unterstützt, wenn in der Outstation aktiviert.
Echtzeitstempelung	Bei entsprechend konfigurierter **Variation** in der Outstation.
Browsen	Online und Offline.

Funktion	Beschreibung
Polling	Polling für Eventklassen, in konfigurierbaren Zyklen oder manuell gesteuert.
Spontan	Ja. Nur Wertänderungen werden übertragen.
Anzahl der Steuerungen	Es können mit einem Treiber beliebig viele Outstations angesprochen werden. Bei serielle Kommunikation können mehrere Outstations für eine serielle Schnittstelle (Bus) konfiguriert werden. Mit einem Treiber ist ebenfalls ein Mischbetrieb *seriell/TCP* möglich.

FEHLERDATEI

Der Treiber unterstützt das zentrale Logging am Diagnosis Server (auf Seite 99).

TREIBERVARIABLEN

Treiberstatistikvariablen werden von dem **DNP3_TG** Treiber nicht gesetzt.
Kommunikationsüberwachung ist über das **INVALID** Statusbit möglich.

INTEGRITY POLL

Nach dem Start des Treibers wird automatisch ein Integrity Poll vom Treiber an die Outstation gesendet (**Read Request Group** *60*, **Variation** *2,3,4* und *0*, **Range** *all*). Die empfangenen Werte stehen entsprechend in der Runtime zur Verfügung. Die Outstation unter Umstände eine andere Variation als Antwort auf den Integrity Poll sendet als bei einer normale Wertänderung. (Mit/ohne Zeitstempel, mit/ohne Flags.)

Über die Steuervariable vom Typ *Klassenpoll* kann in der Runtime auch explizit ein Integrity Poll angestoßen werden. Wird der Treiber mittels Treiberkommando-Funktionen gestoppt und gestartet, löst das ebenfalls einen Integrity Poll aus.

INTERNAL INDICATIONS (IIN)

Die Outstation kann ihren Status über **Internal Indication** Statusbits dem Master mitteilen. Über Variablen der Treiberobjektgruppe **IIN** können diese Flags überwacht werden.
Der DNP3_TG Treiber wertet **Internal Indication** Bits wie folgt aus:

- IIN 1.1 CLASS_1_EVENTS:
  Die Outstation setzt dieses Bit, wenn der Eventbuffer für Klasse 1 Events weitere DNP Objekte beinhaltet, die nicht in der aktuelle Antwort enthalten sind. Der DNP3_TG Master reagiert auf dieses Bit, indem sofort eine Leseanfrage für Gruppe 60, Variation 2 an die Outstation

gesendet wird. Wenn die Outstation dieses Statusbit häufig setzt, kann es zu einem höheren Lesezyklus, als in der Treiberkonfiguration definiert, führen.

- IIN 1.2 CLASS_2_EVENTS:
  Die Outstation setzt dieses Bit, wenn der Eventbuffer für Klasse 2 Events weitere DNP Objekte beinhaltet, die nicht in der aktuellen Antwort enthalten sind. Der DNP3_TG Master reagiert auf dieses Bit, indem sofort eine Leseanfrage für Gruppe 60, Variation 3 an die Outstation gesendet wird. Wenn die Outstation dieses Statusbit häufig setzt, kann es zu einem höheren Lesezyklus, als in der Treiberkonfiguration definiert, führen.
- IIN 1.3 CLASS_3_EVENTS:
  Die Outstation setzt dieses Bit, wenn der Eventbuffer für Klasse 3 Events weitere DNP Objekte beinhaltet, die nicht in der aktuelle Antwort enthalten sind. Der DNP3_TG Master reagiert auf dieses Bit, indem sofort eine Leseanfrage für Gruppe 60, Variation 4 an die Outstation gesendet wird. Wenn die Outstation dieses Statusbit häufig setzt, kann es zu einem höheren Lesezyklus, als in der Treiberkonfiguration definiert, führen.
- IIN 1.4 NEED_TIME:
  Die Outstation setzt dieses Bit, wenn eine Zeitsynchronisierung vom Master verlangt wird. Der DNP3_TG Master reagiert sofort und antwortet der Outstation abhängig von der gewählte Zeitsynchronisierungsoption in der Treiberkonfiguration.
- IIN 2.3 BUFFER_OVERFLOW:
  Die Outstation setzt dieses Bit, wenn es zu einem Überlauf der Eventpuffer kam. Der DNP3_TG Master reagiert darauf mit einem automatischen Integrity Poll.
- IIN 1.7 DEVICE_RESTART:
  Wird von der Outstation bei einem Neustart gesetzt. Die Verbindung zur Outstation wird in diesem Fall neu initialisiert. Das Bit wird vom DNP3_TG Master in diesem Prozess zurück gesetzt.

Wenn die Option **No class poll on IIN1.1, IIN1.2, IIN1.3** in der Treiberkonfiguration bei der Station aktiv ist, ignoriert der Treiber die **Internal Indication** Flags *IIN1.1*, *IIN1.2* und *IIN1.3*. Der Treiber sendet keine automatische Lese-Anfrage für Gruppe 60, Variation 2, 3, 4. Diese Option kann aktiviert werden, um Kompatibilitätsprobleme zu umgehen, wenn die Outstation eines dieser Flags nicht zeitgerecht zurücksetzt, was dazu führt, dass der Treiber ausschließlich Event Polls sendet.

In der Regel setzt die Outstation diesen Flags um den Master nochmals zum Lesen aufzufordern, weil weiteren Daten verfügbar sind und um somit eventuell einen Pufferüberlauf in der Outstation vermeiden zu können. Achten Sie darauf, wenn Sie diese Option aus Kompatibilitätsgründen aktivieren, und stellen Sie sicher, dass der Treiber in regelmäßigen Intervallen die Outstation für Events pollt.

Über die Optionen **No time sync on IIN1.4** und **No integrity poll on IIN2.3** kann der Automatismus für Zeitsynchronisation und Integrity Poll abgeschaltet weren.

DNP3 OBJECT FLAGS MAPPING

Die DNP Objects Flags **ONLINE** und **COMM_LOST** werden vom DNP3_TG Treiber ausgewertet. Bei **ONLINE** = *false* oder **COMM_LOST** = *true*, ist in der Runtime das *INVALID* Bit bei der Variable gesetzt. Im umgekehrten Fall wird entweder das *SPONTAN* oder *GA* Bit gesetzt. Das *GA* Bit wird für Werte aus nicht Event-Variations gesetzt, das *SPONTAN* wird für Werte aus Event-Variations.

Die konfigurierte Variation in der Outstation bestimmt, ob für eine Objektgruppe Objekte mit oder ohne Flags gesendet werden.

Außerdem werden die nachfolgend angeführten Flags/IIN Bits auf die Variablen Statusbits übertragen:

DNP3 Flag / IIN	Statusbit
Restart	*NT* Bit
Remote forced	*SB* Bit
Local forced	*SB* Bit
Rollover	*OV* Bit
IIN.1.4 Need time	*Time-inval* Bit

ZEITSYNCHRONISIERUNG

Der DNP3_TG Treiber unterstützt die Zeitsynchronisierung von Outstations mit der Zeit der Master Station. Meldet die Outstation den Bedarf nach einer Zeitsynchronisierung mittels **Internal Indication** Flag 1.4 an, sendet der DNP3_NG Treiber die aktuelle Systemzeit entsprechend den Optionen für UTC/lokale Zeit und LAN Zeitsynchronisierung. Die Zeitsynchronisation kann auch manuell über ein explizites Kommando angestoßen werden.

Die Outstation kann die Zeit mit einer andere Quelle synchronisieren, z. B. via GPS Empfänger. In diesem Fall würde die Möglichkeit der Zeitsynchronisation nicht genutzt werden. Stellen Sie in diesem Fall sicher , dass der Runtime Rechner mit der DNP3_TG Master Station mit der gleiche Zeitquelle synchronisiert wird.

Das DNP3 Protokoll sieht nicht vor, dass der Master seine eigene Zeit mit der Zeit der Outstation synchronisiert.

SELECT UND CANCEL (BEFEHLSGABE)

Der DNP3_TG Treiber beantwortet ein Select und Cancel sofort positiv mit passender COT. Das Execute wird nach einem Select mit angepasster COT durchgeführt. Automatisches Select und Execute werden dabei bevorzugt behandelt.
Desweiteren haben Select, Cancel und Execute ein zusätzliches Statusbit. Dadurch ist eine saubere Laufzeitüberwachung bei projektiertem Routing möglich.

SELECT BEFORE OPERATE - DIRECT OPERATE

Der DNP3_TG Treiber verwendet wahlweise *Direct Operate* (Default), *Direct Operate NoAck* oder *Select Before Operate* für das Schreiben von Variablen vom Typ *Binary Output Status* oder *Analog Output*. Die Konfiguration erfolgt über die treiberspezifische Variableneigenschaft **Command Mode**.

Achten Sie darauf, dass bei der Variable die Eigenschaft **Select Before Operate** inaktiv bleibt! Diese Eigenschaft ändert das Verhalten der zweistufigen Befehlsgabe und ist nicht kompatibel mit dem DNP3 Standard. Wenn diese Eigenschaft doch aktiv ist:

- Blockieren Sie die Befehlsgabe für weitere Befehle, wenn *Direct Operate* bei der Variable gesetzt ist.
- Führen Sie den Befehl bei der erste Stufe und auch bei der zweite Stufe durch, wenn *Auto-SBO* bei der Variable gesetzt ist.

Hinweis: Beim DNP3 32 Treiber kann nur über diese Option ein **Select Before Operate** durchgeführt werden. Dies ist allerdings nicht konform mit den im DNP3 Standard definierten Regeln für **Select Before Operate**.

HYSTERESE

Der Treiber unterstützt die Hysterese für Spontanwerte (unsolicited responses). Die Hysterese wird nicht berücksichtigt von:

- Werten, die als Antwort auf eine Leseanforderung für eine Eventklasse empfangen werden
- Variablen, die als klassenlose Variablen explizit gelesen werden

Wird ein identer Wert, aber mit neuerem Zeitstempel, empfangen, dann wird dieser Wert vom Treiber als neu zur Runtime geschickt.

8.1 DNP3 File Transfer

Der DNP3_TG Treiber unterstützt File Transfer nach IEEE1815 Standard. Folgende Funktionalitäten werden dabei unterstützt:

- Datei,- oder Verzeichnis Informationen von der Outstation lesen.
- Verzeichnisinhalt von der Outstation lesen.
- Datei von der Outstation lesen und lokal speichern.
- Datei von der Outstation entfernen.
- Lokale Datei in die Outstation schreiben.
- Eine laufende Dateiübertragung (File Transfer) abbrechen

Es wird empfohlen für die Dateiübertragung mit File transfer eine eigene Kommando-Stringvariable zu verwenden. Neben dieser Kommando-Stringvariable muss pro Netzadresse eine Stringvariable vom Typ "*File Transfer Result*" mit entsprechender Länge erstellt werden um das Ergebnis der Anfragen zum Lesen von Dateiinformationen oder Verzeichnisinhalten abzubilden.

File transfer Aktionen werden asynchron ausgeführt. Der Treiber kann ein File transfer-Kommando an die Outstation senden, wenn nicht gerade ein zyklischer Poll aufgerufen wird oder wenn nicht gerade ein Kommando gesendet wird. Es kann immer nur eine einzige File transfer-Aktion aktiv sein. Bis die initiale Antwort von der Outstation erhalten wurde, werden keine weiteren Anfragen an die Outstation gesendet. Hier gilt das normale *Response Timeout*. Kommt eine initiale Antwort mit *Null Response* läuft der normale Pollingzyklus weiter. Es können auch neue Kommandos, wie z.B. zum expliziten Lesen von Events, gesendet werden.

Pro Verbindung muss das Verzeichnis für das Lesen einer Datei aus der Outstation und das Verzeichnis zum Senden einer Datei an die Outstation, konfiguriert werden, damit File Transfer genutzt werden kann. Es können ausschließlich lokale Verzeichnisse angegeben werden. Der Pfad darf kein Relativpfad sein. Die Verzeichnisse müssen auf dem Rechner, auf dem die Runtime läuft, vorhanden sein.

Info

Wenn die Runtime unter Windows CE verwendet werden soll, gibt es unter Umstände beispielsweise den Absolutpfad c:\temp nicht, sondern muss mittels \temp angesprochen werden. Da die Treiberkonfiguration die Eingabe eines Absolut Pfades erfordert, muss in diesem Fall der Pfad manuell in der Treiberkonfigurationsdatei angepasst werden.

Pro Verbindung kann auch das dazugehörige *Timeout* konfiguriert werden. Dieses *Timeout* sorgt dafür, daß File transfer-Aktionen die asynchron laufen, auch dann automatisch beendet werden, wenn die Outstation nicht rechtzeitig eine Antwort liefert. Dadurch wird sichergestellt, daß eine neue File transfer-Aktion auch wieder gestartet werden kann.

NULL RESPONSE

Lt. IEEE1815 muss die Outstation nicht sofort eine Antwort senden in der die angeforderten Daten enthalten sind. Sie darf auch initial mit *Null response* antworten, und die Antwort mit den angeforderten Daten später als Event senden. Die Outstation kann die Antwort mit den angeforderten Daten als Event auf verschiedene Arten senden:

- Als Antwort auf ein *Event Class Poll* (Class 1, Class 2 oder Class 3)
- Als Antwort auf ein *Read Request* für G70v4 oder G70v7
- Als Antwort auf ein *Read Request* für G70v0 (offiziell lt. IEEE1815 nicht unterstützt)
- Als *Unsolicited Response*

Für einen Master der eine File transfer-Aktion für eine Outstation startet - welche initial einen *Null Response* liefert - bedeutet dies, daß der Master nach dem initiale *Null Response* explizit zyklisch die Outstation nach diesen Events fragen muss.

Der **DNP3_TG Treiber** macht diese Anfrage für diese Events nicht automatisch, allerdings unterstützt der **DNP3_TG Treiber** alle oben genannte Arten und kann diese unabhängig vom projektierten Polling Zyklus explizit über Kommandos auf der allgemeinen Kommando-Stringvariable anfordern. Der Projektant für ein solches Projekt muss in diesem Fall Mechanismen projektieren, die während dem Warten auf eine Antwort, zyklisch entsprechende Anfragen an die Outstation sendet, bis entweder die Antwort eintrifft und so lange bis die File transfer -Aktion läuft oder das *File transfer timeout* abgelaufen ist.

Eine Möglichkeit das zu realisieren ist mit Hilfe von zenon Logic, wo eine Kommando Stringvariable überwacht werden kann. Wurde nach einem File transfer-Kommando noch keine Antwort mit Daten empfangen, können zyklisch über die allgemeine Kommando-Stringvariable die Kommandos zum Lesen der Events gesendet werden. Dies wird so lange ausgeführt, bis entweder die Antwort eintrifft und File transfer abgeschlossen ist, oder die Anfrage in das *File transfer timeout* läuft und fehlschlägt.

DATEI ODER VERZEICHNIS INFORMATION ANFORDERN

Über das Kommando *GET_FILE_INFO* sendet der DNP3_TG Master eine File Transfer Anfrage an die Outstation. Die Antwort von der Outstation wird auf eine Variable vom Treiber Objekt Typ "File Transfer Result" abgebildet, welche manuell mit der entsprechende Netzaddresse erstellt werden muss. Das Ergebnis vom Kommando wird auf die Kommando-Stringvariable abgebildet.

Kommando: *GET_FILE_INFO*

- Parameter 1: **Netzadresse**
- Parameter 2: Remote Pfad

Remote Pfad:

- Der Remote Pfad muss zwischen "" aufgelistet werden.
- Die "" werden bei der Anfrage an die Outstation entfernt.
- Der Remote Pfad muss zumindest ein Zeichen enthalten und darf nicht nur aus Leerzeichen bestehen.
- Der Remote Pfad darf Leerzeichen enthalten.

Ist die Anfrage erfolgreich, wird das Ergebnis formatiert auf der Stringvariable "*File Transfer Result*" abgebildet:

Ergebnis: **<Typ>;<Name>;<Grösse>;<Datum und Uhrzeit>**

Der Typ enthält entweder "*F*" für eine Datei, oder "*D*" für ein Verzeichnis (Directory).

Beispiel

Drei Beispiele für den Wert der String Variable "**File Transfer Result**" als Ergebnis dieser Aktion:

- F;hallo.txt;234;2015-03-20T13:58:34:22.387Z
  Datei mit einer Grösse von 234 bytes
- D;Dir1;3;2015-03-20T13:58:34:22.387Z
  Verzeichnis mit 3 Objekte (Dateien oder Unterverzeichnisse)
- D;/;0;2015-03-20T13:58:34:22.387Z
  (Leeres Root Verzeichnis)

VERZEICHNISINHALT ANFORDERN

Über das Kommando *READ_DIRECTORY* sendet der **DNP3_TG Master** zunächst eine *File Open*-Anfrage an die Outstation und wartet auf die Antwort. Anschließend wird anhand des empfangenen *File Handle* für das Verzeichnis, der Inhalt gelesen. Dabei kann der **DNP3_TG Treiber** auch mehrere *Read Requests* senden. Die Antwort wird auf eine Variable vom Treiberobjekt Typ "**File Transfer Result**" abgebildet, welche manuell mit der entsprechende Netzaddresse im Projekt erstellt werden muss. Das Ergebnis des Kommandos wird auf die Kommando-Stringvariable abgebildet.

Kommando: *READ_DIRECTORY*

- Parameter 1: **Netzadresse**
- Parameter 2: Remote Pfad

Remote Pfad:

- Der Remote Pfad muss zwischen "" aufgelistet werden.
- Die "" werden bei der Anfrage an die Outstation entfernt.
- Der Remote Pfad muss zumindest ein Zeichen enthalten und darf nicht nur aus Leerzeichen bestehen.
- Der Remote Pfad darf Leerzeichen enthalten.

Ist die Anfrage erfolgreich, wird die Antwort auf der Stringvariable "*File Transfer Result*" abgebildet:

Ergebnis:

<Typ>;<Name>;<Grösse>;<Datum und Uhrzeit>CR LF

..

<Typ>;<Name>;<Grösse>;<Datum und Uhrzeit>

Der Typ enthält entweder "*F*" für eine Datei, oder "*D*" für ein Verzeichnis (Directory).

Für jede Datei und für jedes Verzeichnis wird die Information auf eine eigene Zeile abgebildet. Jede Zeile ist mit einem **Carriage Return / Line Feed** getrennt.

DATEI LESEN

Über das Kommando *READ_FILE* sendet der **DNP3_TG Master** zunächst eine *File Open*-Anfrage an die Outstation und wartet auf die Antwort. Darauffolgend wird anhand des empfangenen *File Handle*, die Datei in einem oder mehreren *Read Requests* gelesen und im konfigurierten Verzeichnis gespeichert.

Dieses Kommando hat zwei Varianten:

Kommando: READ_FILE

- Parameter 1: **Netzadresse**
- Parameter 2: Remote Pfad

Kommando: READ_FILE

- Parameter 1: **Netzadresse**
- Parameter 2: Remote Pfad
- Parameter 3: lokaler Pfad

Bei der Variante mit zwei Parameter werden nur der Remote Pfad und der Dateiname spezifiziert. Der lokale Dateiname und der Pfad sind identisch zu den Remote Dateiname und Pfad.

Bei der Variante mit drei Parameter kann explizit der Pfad und Dateiname für die lokale Datei spezifiziert werden.

Achtung: Für den Parameter 3 (lokaler Pfad) sind die Zeichenfolgen "..", "\\", und ":" nicht erlaubt. Das verhindert, dass Dateien lokal ausserhalb vom konfigurierten Verzeichnis gespeichert werden.

Info

Wenn im Remote Pfad zB. ein ":" oder ein "\" enthalten ist, muss explizit die Variante mit 3 Parameter verwendet werden.

Beispiel: *READ_FILE 1 "c:\temp\text1.txt" "text1.txt"*

DATEI LÖSCHEN

Über das Kommando *DELETE_FILE* sendet der **DNP3_TG Master** eine Anfrage an die Outstation um eine Datei zu löschen und wartet auf die Antwort. Das Ergebnis des Kommandos wird auf die Kommando-Stringvariable abgebildet.

Kommando: DELETE_FILE

- Parameter 1: **Netzadresse**

- Parameter 2: Remote Pfad

Remote Pfad:

- Der Remote Pfad muss zwischen "" aufgelistet werden.
- Die "" werden bei der Anfrage an die Outstation entfernt.
- Der Remote Pfad muss zumindest ein Zeichen enthalten und darf nicht nur aus Leerzeichen bestehen.
- Der Remote Pfad darf Leerzeichen enthalten.

DATEI SCHREIBEN

Über das Kommando *WRITE_FILE* sendet der **DNP3_TG Master** zunächst eine *File Open*-Anfrage an die Outstation und wartet auf die Antwort. Darauffolgend schreibt der Treiber anhand des empfangenen *File Handle*, die Datei in einem oder mehreren *Write Requests* in die remote outstation. Das Kommando hat zwei Varianten:

Kommando: WRITE_FILE

- Parameter 1: **Netzadresse**
- Parameter 2: lokaler Pfad
- Parameter 3: Outstation Dateiname

Kommando: WRITE_FILE

- Parameter 1: **Netzadresse**
- Parameter 2: lokaler Pfad
- Parameter 3: Outstation Dateiname
- Parameter 4: Berechtigung
- Parameter 5: Timeout

Bei der Variante mit zwei Parameter wird die Berechtigung "*World Read, Group Read, Owner Read*" verwendet und die default Einstellung für das File-Transfer Timeout.

Bei der Variante mit fünf Parameter kann die Berechtigung anhand untenstehende Tabelle gesetzt werden und muss auch ein Timeout in Sekunden definiert werden.

Parameter	Beschreibung
1	*World Execute*
2	*World Write*
4	*World Read*

Parameter	Beschreibung
8	*Group Execute*
16	*Group Write*
32	*Group Read*
64	*Owner Execute*
128	*Owner Write*
256	*Owner Read*

Die Zahlen für die Berechtigung (Parameter 4) können addiert werden um die entsprechende Berechtigung zu setzen. Zum Beispiel Wert 438 für Read/Write für World, Group und Owner. Der maximale Wert wo alle Bits für die Berechtigung gesetzt sind, beträgt somit 511.

FILE TRANSFER ABBRECHEN

Unter Umstände kann es erforderlich sein, eine laufende *File transfer*-Aktion abzubrechen. Zum Beispiel vor einer geplanten Redundanzumschaltung. Mit dem Abbruch soll die Outstation das *File Handle* wieder freigeben und für eine nächste *File transfer*-Aktion bereit sein. Es kann sinnvoll sein, für das Abort Kommando eine separate Kommandovariable zu verwenden. Das Ergebnis vom Abort Kommando wird somit auf die "Abort" Kommando-Stringvariable abgebildet (Erfolgreich, status 4). Die Kommando-Stringvariable für den laufenden File transfer-Aktion enthält den Wert "8".

Kommando: FILE_TRANSFER_ABORT

- Parameter 1: **Netzadresse**

Wenn der *File transfer* aus einem anderem Grund bereits abgebrochen ist, zB. weil die Verbindundung verloren gegangen ist und die Anzahl an *Retries* überschritten, kann der *File transfer* nicht mehr explizit abgebrochen werden.

FILE TRANSFER SPEZIFISCHE STATUSCODES

Die Kommando-Stringvariable enthält als Statuscode das Ergebnis der Kommandoausführung. Neben den allgemeinen Kommando Statuscodes, gibt es spezifische Statuscodes für *File transfer*, die die Antwort von der Outstation im Fehlerfall abbilden. Der *File transfer*-Statuscode der Antwort der Outstation, welche im Standard definiert sind, wird mit 1000 addiert.

Parameter	Beschreibung
1001	*Permission Denied*
1003	*File not found*

Parameter	Beschreibung
1004	*File locked*
1005	*too many open handles*
1006	*invalid handle or handle timed out*
1007	*write block size error*
1008	*communication failed*
1009	*abort failed*
1016	*file not open*
1017	*file handle expired*
1018	*buffer overrun on writing a file*
1019	*general error with file transfer*
1020	*block / sequence error during file transfer*
1255	*other error during file transfer occurred*

KOMMANDO STATUSCODES

Statuscode	Beschreibung
3	Kommando wurde versandt
4	Kommando erfolgreich abgeschlossen
5	Kommando konnte nicht verarbeitet werden
6	Kommando wurde negativ beantwortet
7	Fehler beim Ausführen des Kommandos (Timeout, Authentication failed)
8	Ausführung des Kommandos / File Transfer wurde abgebrochen
10	File Transfer: Timeout aufgetreten
11	File Transfer: Speichern der lokalen Datei fehlgeschlagen
12	File Transfer: Abort fehlgeschlagen
13	File Transfer: Abbruch, mehr Daten empfangen als gemeldete Dateigröße
16	Syntaxfehler

Statuscode	Beschreibung
17	Unbekanntes Kommando
18	**Netzadresse** nicht vorhanden
19	File Transfer: wird bereits ausgeführt
20	File Transfer: Verzeichnis Fehler oder nicht konfiguriert
21	File Transfer: Abort fehlgeschlagen, kein File Transfer aktiv

9 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.

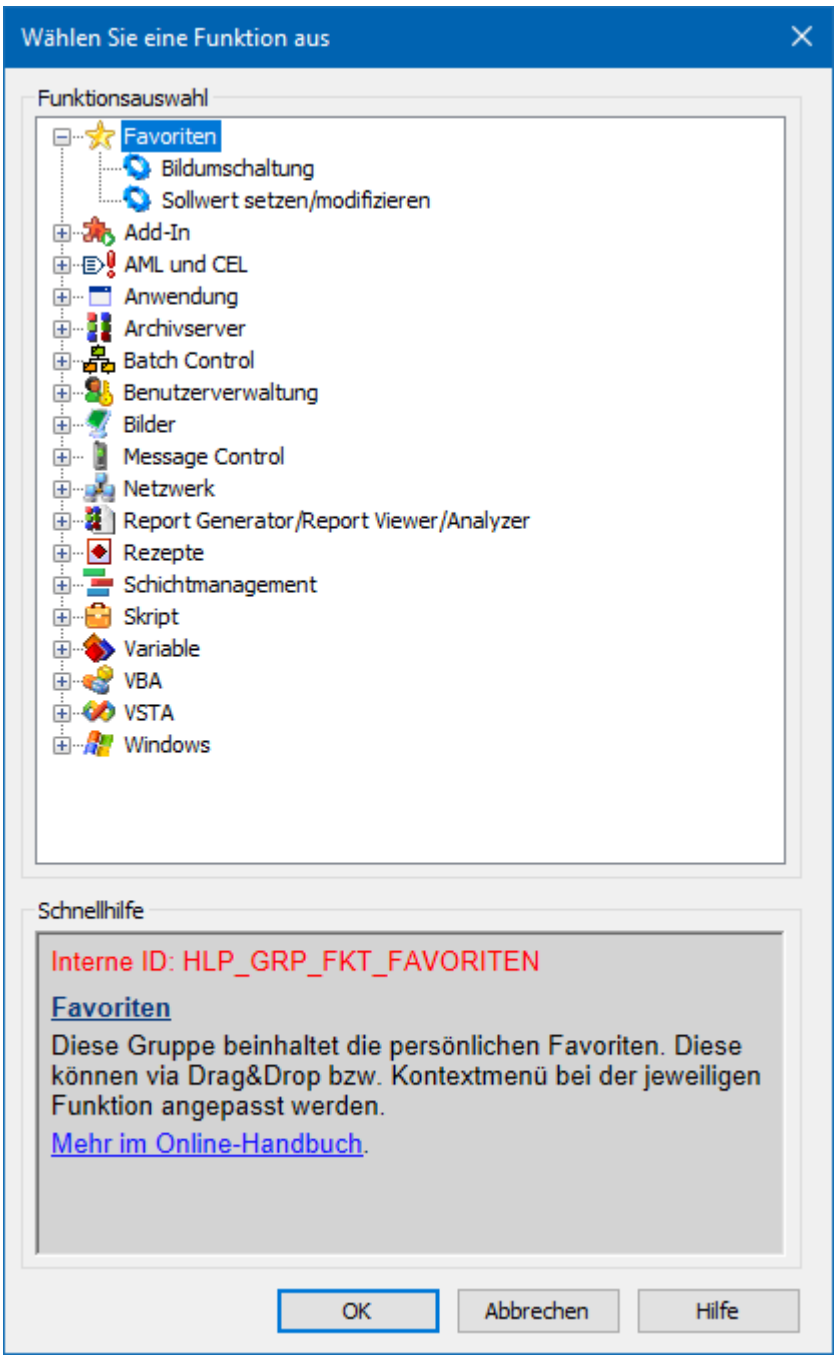
> **Achtung**
>
> Die zenon Funktion **Treiberkommandos** ist nicht ident mit den Treiberkommandos, die bei Energy-Treibern in der Runtime ausgeführt werden können!

PROJEKTIERUNG DER FUNKTION

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

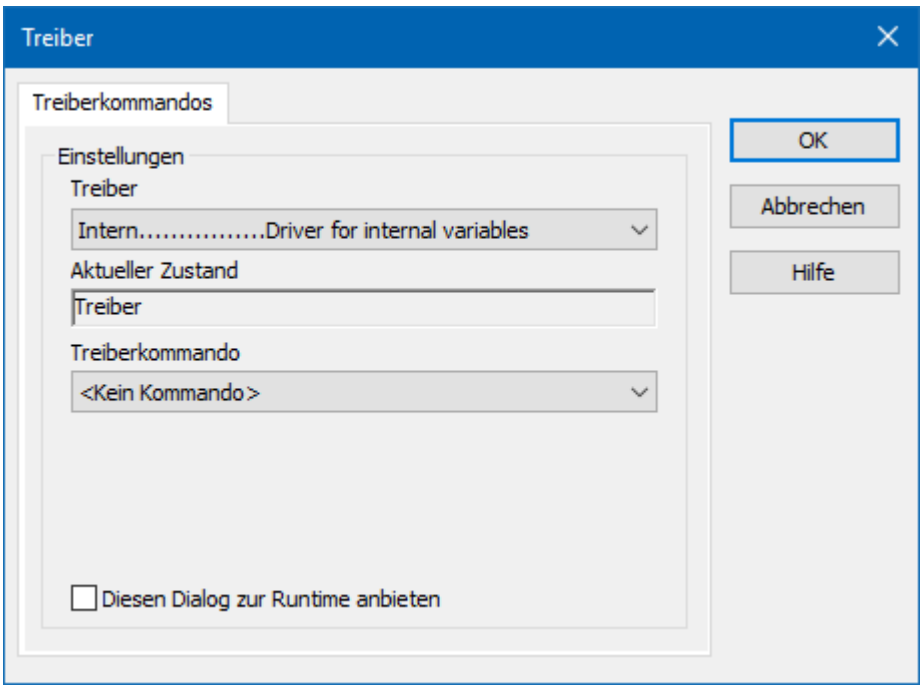
1. Legen Sie im zenon Editor eine neue Funktion an.

Der Dialog zur Auswahl einer Funktion wird geöffnet.



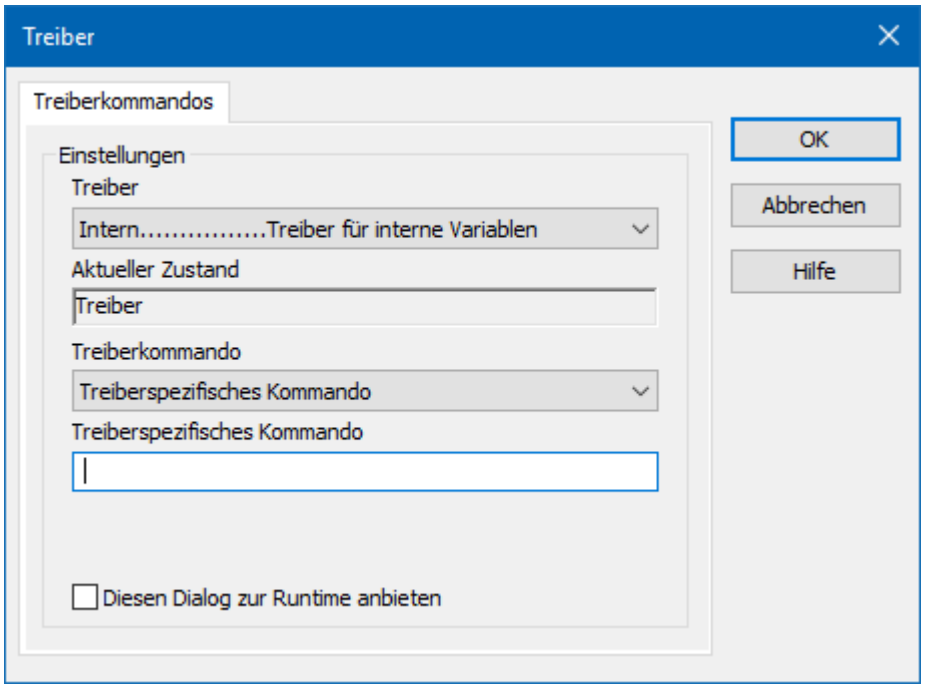
2. Navigieren Sie zum Knoten **Variable.**
3. Wählen Sie den Eintrag **Treiberkommandos**.

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.

10 Fehleranalyse

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

10.1 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.

10.2 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.
Communication with %s timed out. Timeout happened %d times	Die Kommunikation zum Treiber *%s* konnte *%d* Mal nicht innerhalb von 60 Sekunden aufgebaut werden.

10.3 Checkliste

Fragen und Hinweise zur Fehlereingrenzung:

ALLGEMEINE FEHLERSUCHE

- Ist die Steuerung an die Stromversorgung angeschlossen?
- Analyse mit Hilfe des Diagnosis Viewers (auf Seite 99):
  -> Welche Meldungen werden angezeigt?
- Sind die Teilnehmer im **TCP/IP**-Netz verfügbar?
- Kann die Steuerung über den Befehl *Ping* erreicht werden?

  Ping: **Kommandozeile öffnen -> ping <IP-Adresse> (z.B.: ping 192.168.0.100) -> Eingabe-Taste drücken.**

  Kommt eine Antwort mit Zeitangabe oder ein Timeout?
- Kann die Steuerung auf dem entsprechenden Port über *Telnet* erreicht werden?

  Telnet: **Kommandozeile öffnen, telnet <IP-Adresse Port-Nummer> eingeben (z.B. telnet 192.168.0.100 20000) -> Eingabe-Taste drücken .**

  Wird der Bildschirm schwarz, und blinkt der Cursor konnte eine Verbindung aufgebaut werden.
- Analyse mit Hilfe eines Programms zur Analyse des Netzwerkverkehrs (Sniffer, zB. Wireshark, Microsoft Network Monitor / Microsoft Message Analyzer)
- Wird für die Verbindung von Steuerung und PC das korrekte, vom Hersteller empfohlene Kabel verwendet?
- Wurde der richtige COM Port ausgewählt?
- Stimmen die seriellen Kommunikationsparameter (Baudrate, Parität, Start/Stop Bits,...) überein?
- Wird der COM Port durch eine andere Anwendung blockiert?
- Wurde die Netzadresse in den Adresseigenschaften der Variable korrekt eingestellt?
  - Stimmt die Adressierung mit der Konfiguration im Treiberdialog überein?
  - Entspricht die Netzadresse der Adresse der Zielstation?
- Wird in der Variable der richtige Objekttyp verwendet?
  Beispiel: Variablen vom Treiberobjekttyp **Kommunikationsdetails** sind reine Statistikvariablen und kommunizieren nicht mit der Steuerung.
  Weitere Informationen dazu finden Sie im Kapitel Kommunikationsdetails (Treibervariablen) (auf Seite 76).
- Stimmt die Offset-Adressierung der Variable mit der in der Steuerung überein?

MANCHE VARIABLEN MELDEN INVALID

- INVALID Bits beziehen sich immer auf eine Netzadresse.
- Mindestens eine Variable der Netzadresse ist gestört.

WERTE WERDEN NICHT ANGEZEIGT, ZAHLENWERTE BLEIBEN LEER

Unter Umstände wurde auf eine Lese-Anfrage an die Steuerung noch keine Antwort Empfangen.

Treiber läuft nicht. Überprüfen Sie die:

- Installation von zenon
- Installation des Treibers
- Installation aller Komponenten
  -> Achten Sie auf Fehlermeldungen beim Start der Runtime.

VARIABLEN WERDEN MIT BLAUEM PUNKT ANGEZEIGT

Die Kommunikation im Netzwerk ist gestört:

- Bei einem Netzwerkprojekt:
  Läuft das Netzwerkprojekt auch auf dem Server?
- Bei einem Standalone-Projekt oder Netzwerkprojekt, das auf dem Server läuft:
  Deaktivieren Sie die Eigenschaft Nur von Standby Server anfordern im Knoten Treiber Anbindung/Adressierung.

WERTE WERDEN FALSCH ANGEZEIGT

Überprüfen Sie die Angaben zur Berechnung im Knoten Wertberechnung der Variablen-Eigenschaften?

Überprüfen Sie die Konfiguration der Outstation,ob tatsächlich die gewünschte Variation gesendet wird (z.B. **Float** mit Nachkommsstellen).

WERTE WERDEN NICHT SCHNELL GENUG ANGEZEIGT

Überprüfen Sie in der Outstation, ob für die gewünschten Werte Events generiert werden und in welche Klasse. Prüfen Sie das Pollen diese Eventklasse in der Treiberkonfiguration.

Überprüfen Sie in der Outstation die Konfiguration von **unsolicited responses**.

ZEITSTEMPEL DER VARIABLE IST FALSCH

Überprüfen Sie in der Konfiguration der Outstation, ob die DNP3-Objekte tatsächlich mit Zeitstempel gesendet werden. (Entsprechende Variation gewählt mit Zeitstempel.)

Überprüfen Sie, ob die Outstation lokale Zeit verwendet oder UTC und setzen Sie die Option in der Treiberkonfiguration entsprechend,

Überprüfen Sie die Uhrzeit des lokalen Rechners und die Uhrzeit der Outstation, inklusive Einstellungen für die Zeitzone.

TREIBER FÄLLT SPORADISCH AUS

Analyse mit Hilfe des Diagnosis Viewers (auf Seite 99):
-> Welche Meldungen werden angezeigt?

Überprüfen Sie die Antwortzeit-Timeout in der Treiberkonfiguration, speziell bei Outstations mit vielen Datenpunkte und langsame Verbindung.

Bei *Secure Authentication*, überprüfen Sie ob der **pre-shared Update Key** im Treiber und in der Outstation identisch ist.