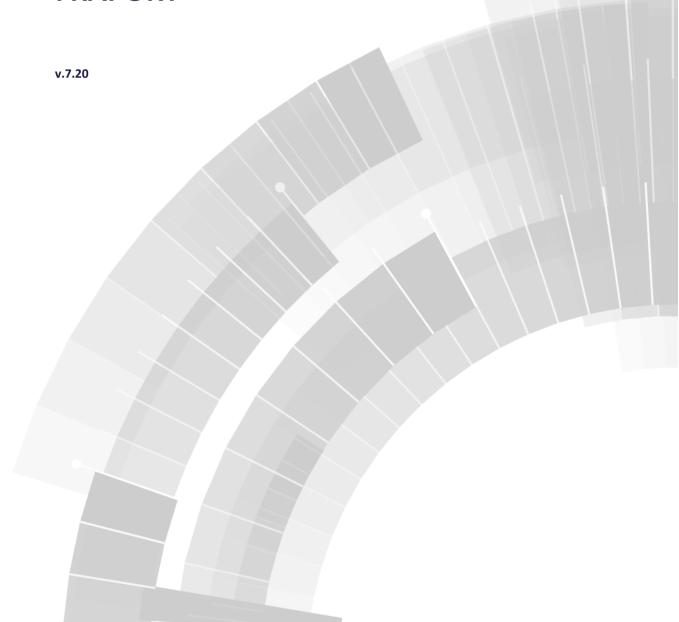


zenon Treiber Handbuch FRAPORT





©2015 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.



Inhalt

1.	. Willkommen bei der COPA-DATA Hilfe			
2.	FRAPORT - Datenblatt 6			
3.				
4.	1. Treiber-Historie			
5.	Konf	iguratio	on	8
	5.1	Anlege	en eines Treibers	8
	5.2	Einstel	llungen im Treiberdialog	10
		5.2.1	Allgemein	10
		5.2.2	Einstellungen	14
		5.2.3	Verbindungen	15
6.	Varia	ablen ar	nlegen	17
	6.1	Variab	len im Editor anlegen	18
	6.2 Adressierung			
		6.2.1	Fluganzeige für Zielbahnen	22
		6.2.2	Prozessvariablen	22
		6.2.3	Status	24
	6.3	Treibei	robjekte und Datentypen	27
		6.3.1	Treiberobjekte	28
		6.3.2	Zuordnung der Datentypen	28
	6.4	Variab	len anlegen durch Import	29
		6.4.1	XML Import	30
		6.4.2	DBF Import/Export	30
	6.5	Treibei	rvariablen	35
7.	. Treiberspezifische Funktionen			
8.	. Treiberkommandos4		43	
9.	Fehle	eranalys	se	46
	0.1	مراده ۸	ata al	4.0



9.2	Checkliste
9.2	Checkliste



1. Willkommen bei der COPA-DATA Hilfe

ALLGEMEINE HILFE

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

PROJEKTUNTERSTÜTZUNG

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

LIZENZEN UND MODULE

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

2. FRAPORT

Der FRAPORT-Treiber ermöglicht es, mit zenon im Gepäcksystemverbund (GSV) alle Meldungen und Anzeigen abzubilden und auf das System steuernd einzuwirken.



3. FRAPORT - Datenblatt

Allgemein:	
Treiberdateiname	FRAPORT.exe
Treiberbezeichnung	FRAPORT Treiber
Steuerungs-Typen	FRAPORT Gepäckverbundsystem
Steuerungs-Hersteller	Fraport AG;

Treiber unterstützt:	
Protokoll	proprietary;
Adressierung: Adress-basiert	-
Adressierung: Namens-basiert	x
Kommunikation spontan	x
Kommunikation pollend	-
Online Browsing	-
Offline Browsing	-
Echtzeitfähig	-
Blockwrite	-
Modemfähig	-
Serielles Logging	-
RDA numerisch	-
RDA String	-



Voraussetzungen:	
Hardware PC	Standard Netzwerk Adapter
Software PC	-
Hardware Steuerung	-
Software Steuerung	-
Benötigt v-dll	-

Plattformen:	
Betriebssysteme	Windows 7, 8, 8.1 Server 2008R2, Server 2012, Server 2012R2;
CE Plattformen	-;

4. Treiber-Historie

Datum	Treiberversion	Änderung
16.03.11	100	Treiberdokumentation wurde neu erstellt

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. 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 (main.chm::/15247.htm) der Online-Hilfe.

5.1 Anlegen eines Treibers

Um einen neuen Treiber anzulegen:

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



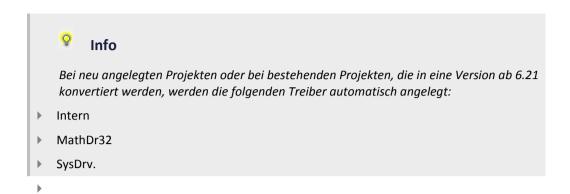
 In der folgenden Dialogbox bietet Ihnen das Programm eine Auflistung aller verfügbaren Treiber an.



- 3. Selektieren Sie den gewünschten Treiber und vergeben Sie eine Bezeichnung für diesen:
 - Die Treiberbezeichnung muss eindeutig sein, d.h. wird ein und derselbe Treiber mehrmals im Projekt verwendet, so muss jeweils eine neue Bezeichnung vergeben werden.
 - 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 ox. Im folgenden Dialog werden die einzelnen Konfigurationen der jeweiligen Treiber eingestellt.

Für ein Projekt müssen nur die jeweils notwendigen Treiber eingebunden werden. Späteres Einbinden eines weiteren Treibers ist problemlos möglich.



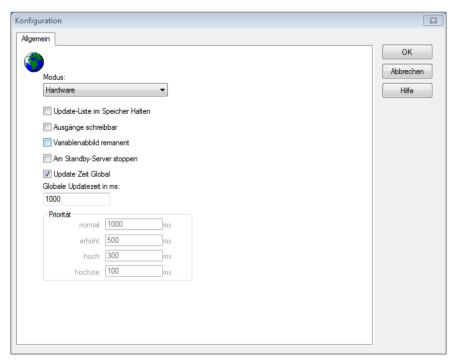


5.2 Einstellungen im Treiberdialog

Folgende Einstellungen können Sie beim Treiber vornehmen:

5.2.1 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**.





Parameter	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 bzw. die Variablen behalten die über zenon Logic gesetzen Werte. Jede Variable hat seinen eigenen Speicherbereich. Zum Beispiel zwei Variablen vom Typ Merker mit Offset 79, können zur Runtime unterschiedliche Werte haben und beeinflussen sich gegenseitig nicht. Ausnahme: Der Simulatortreiber.
	▶ Simulation - zählend
	Es wird keine Kommunikation zur Steuerung aufgebaut, die Werte werden vom Treiber simuliert. 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. (main.chm::/25206.htm)
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 der Status S_MERKER_1(0) bis S_MERKER8(7), REVISION(9), AUS(20) oder ERSATZWERT(27) aktiv ist
	Das Variablenabbild wird immer gespeichert wenn:
	die Variable vom Objekttyp Treibervariable 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 (statusverarbeitung.chm::/24150.htm), sondern einen leeren Wert. Damit wird verhindert, dass beim Hochstufen zum Server nicht relevante Werte in AML, CEL und Archiv erzeugt werden.
	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	Aktiv: Die eingestellte Globale Update Zeit in ms 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.
Priorität	Hier werden die Pollingzeiten der einzelnen Prioritätsklassen eingestellt. Alle Variablen mit der entsprechenden Priorität



werden in der eingestellten Zeit gepollt.
Die Zuordnung zu den Variablen erfolgt separat bei jeder Variablen über die Einstellungen in den Variableneigenschaften. Mit den Prioritätsklassen kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit bzw. benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast.
Achtung: Prioritätsklassen werden nicht von jedem Treiber unterstützt. Zum Beispiel unterstützen spontan kommunizierende Treiber dies nicht.

DIALOG BEENDEN

Parameter	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 ZYKLISCHE TREIBER

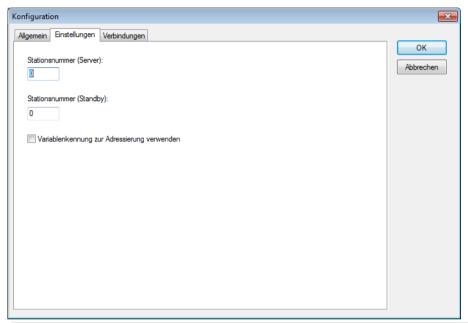
Für zyklische Treiber gilt:

Beim Sollwert Setzen, Advisen von Variablen und bei Requests wird sofort ein Lesezyklus für alle Treiber ausgelöst - unabhängig von der eingestellten Update Zeit. Damit wird sicher gestellt, dass der Wert nach dem Schreiben in der Visualisierung sofort zur Verfügung steht. Update-Zeiten können damit für zyklische Treiber kürzer ausfallen als eingestellt.



5.2.2 Einstellungen

In dieser Registerkarte werden allgemeine Einstellungen für die Kommunikation vorgenommen. Diese gelten für alle Verbindungen (auf Seite 15), die im Treiber angelegt sind.

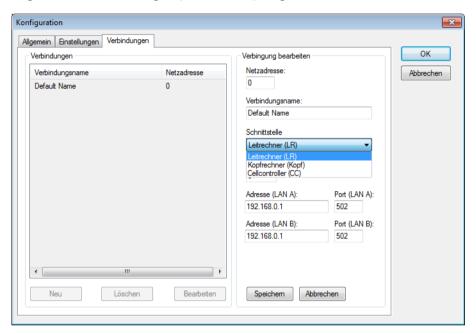


Eigenschaft	Beschreibung
Stationsnummer (Server):	Stationsnummer des zenon Servers.
Stationsnummer (Standby):	Stationsnummer des zenon Standby-Servers.
Variablenkennung zur Adressierung verwenden	Aktiv: Für die symbolische Adressierung wird die Kennung statt des Namens verwendet.



5.2.3 Verbindungen

In dieser Registerkarte erfolgen die Einstellungen für die Verbindungen. Im Treiber können beliebig viele Verbindungen angelegt werden. Alle Verbindungen nutzen die Globalen Einstellungen, die auf der Registerkarte Einstellungen (auf Seite 14) eingestellt wurden.





Eigenschaft	Beschreibung			
Verbindungen	Liste der erstellten Verbindungen und ihrer Netzadressen.			
Verbindungsname	Name der Verbindung.			
Neu	Fügt neuen Eintrag zur Liste hinzu. Einstellungen werden im Feld rechts der Liste konfiguriert.			
Löschen	Löscht ausgewählten Eintrag aus Liste.			
Bearbeiten	Ermöglicht Konfiguration des ausgewählten Eintrags. Der Bereich rechts der Liste zum Editieren der Verbindung wird aktiviert.			
	Achtung! Im Editiermodus kann der Treiberdialog nicht geschlossen werden. Erst wenn der Editiermodus mit Save oder Cancel verlassen wurde ist ein Schließen des Treiberdialogs möglich.			
Verbindung bearbeiten	Verbindungseinstellungen für eine neue oder die unter verbindung ausgewählte Verbindung.			
Netzadresse	Die Netzadresse identifiziert die Verbindung. Jede Verbindung muss daher eine eindeutige Netzadresse haben. Variablen werden einer Verbindung über die Netzadresse zugeordnet.			
	Achtung! Wenn Sie die Netzadresse hier ändern, muss auch die Netzadresse bei allen Variablen der Verbindung geändert werden. Sonst stimmt die Zuordnung nicht mehr und die Variablen können in der Runtime nicht kommunizieren!			
Verbindungsname	Verbindungsname.			
	Frei definierbarer Name.			
Schnittstelle	Auswahl der Verbindung aus Dropdownliste:			
	► Leitrechner (LR)			
	► Cellcontroller (CC)			
	► Kopfrechner (Kopf)			
Stationsnummer	Stationsnummer der entfernten Station.			
Adresse (LAN A)	IP-Adresse bzw. DNS-Name der entfernten Station im LAN A.			
Port (LAN A)	TCP-Port der entfernten Station im LAN A.			
Adresse (LAN B)	IP-Adresse bzw. DNS-Name der entfernten Station im LAN в.			
	Wird diese Eintrag leer gelassen, erfolgt die Kommunikation ohne Redundanz.			



Port (LAN B)	TCP-Port der entfernten Station im LAN B.	
	Wird diese Eintrag leer gelassen, erfolgt die Kommunikation ohne Redundanz.	
Speichern	Speichert Änderungen.	
Abbrechen	Verwirft Änderungen.	

Hinweis: Meldungen des Treibers können mit dem Diagnose Viewer (main.chm::/12464.htm) ausgelesen werden.

NEUE VERBINDUNG ANLEGEN

- 1. klicken Sie auf die Schaltfläche Neu
- 2. Tragen Sie die Verbindungsdetails ein
- 3. klicken Sie auf speichern

VERBINDUNG BEARBEITEN

- 1. wählen Sie in der Verbindungsliste die gewünschte Verbindung
- 2. klicken Sie auf die Schaltfläche Bearbeiten
- 3. ändern Sie die Verbindungsparameter
- 4. schließen Sie mit speichern ab

VERBINDUNG LÖSCHEN

- 1. wählen Sie in der Verbindungsliste die gewünschte Verbindung
- 2. klicken Sie auf die Schaltfläche Löschen
- 3. die Verbindung wird aus der Liste gelöscht

6. Variablen anlegen

So werden Variablen im zenon Editor angelegt:



6.1 Variablen im Editor anlegen

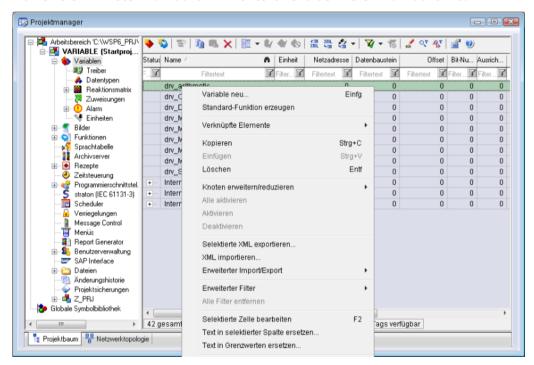
Variablen können angelegt werden:

- als einfache Variable
- ▶ in Arrays (main.chm::/15262.htm)
- ▶ als Struktur-Variablen (main.chm::/15278.htm)

DIALOG VARIABLE

Um eine neue Variable zu erstellen, gleich welchen Typs:

1. wählen Sie im Knoten Variablen im Kontextmenü den Befehl Variable neu



- 2. der Dialog zur Konfiguration der Variable wird geöffnet
- 3. konfigurieren Sie die Variable



4. welche Einstellungen möglich sind, hängt ab vom Typ der Variablen





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 (Main.chm::/Intern.chm::/Intern.htm)) geladen.	
Treiber-Objekttyp (cti.chm::/28685.htm)	Wählen Sie aus der Dropdownliste den passenden Treiber-Objekttyp aus.	
Datentyp	Wählen Sie den gewünschten Datentyp. Klick auf die Schaltfläche 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.	

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.



6.2 Adressierung

Gruppe/Eigenschaft	Beschreibung		
General			
Name	Name der Variable. ACHTUNG: Je zenon Projekt muss der Name eindeutig sein. Abhängig von der Einstellung Kennung als externen Namen verwenden im Dialog Einstellungen (auf Seite 14) wird die Variable über ihren Namen bzw. über die eingestellte Kennung adressiert.		
Identification	Name der Variable. ACHTUNG: Je zenon Projekt muss der Name eindeutig sein. Abhängig von der Einstellung Kennung als externen Namen verwenden im Dialog Einstellungen (auf Seite 14) wird die Variable über ihren Namen bzw. über die eingestellte Kennung adressiert.		
Addressing			
Net address	Busadresse oder Netzadresse der Variablen.		
	Diese Adresse bezieht sich auf die Busadresse der Verbindungsprojektierung im Treiber. Damit wird ausgewählt, auf welcher Steuerung sich die Variable befindet.		
Data block	Wird für diesen Treiber nicht verwendet.		
Offset	Wird für diesen Treiber nicht verwendet.		
Alignment	Wird für diesen Treiber nicht verwendet.		
Bit number	Nummer des Bits innerhalb des eingestellten Offsets.		
	Mögliche Eingabe: 0 65535		
String length	Nur verfügbar bei String-Variablen: Maximale Anzahl von Zeichen, die die Variable aufnehmen kann.		
Driver connection/Driver object type	Objekttyp der Variablen. Wird abhängig vom verwendeten Treiber beim Erstellen der Variablen ausgewählt und kann hier geändert werden.		
Driver connection/Datatype	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.		

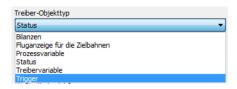
ADRESSIERUNG

Detaillierte Informationen zur Adressierung stehen zur Verfügung für die Objekttypen:

- ► Fluganzeige für die Zielbahnen (auf Seite 22)
- ► Prozessvariablen (auf Seite 22)
- ► Status (auf Seite 24)



Generell stehen als Objekttypen (auf Seite 28) zur Verfügung:



6.2.1 Fluganzeige für Zielbahnen

Fluganzeigen werden über die Zielnummer im System adressiert.

Die komplette Adresse setzt sich zusammen aus:

- einem beliebigen Präfix
- einem Rufzeichen (!) als Trenner
- ▶ der Zeichenfolge Flug_
- ▶ der Zielnummer

Beispiel: wE052!FLUG 1234

6.2.2 Prozessvariablen

Die Adressierung von Prozessvariablen unterscheidet sich am Leitrechner und am Cellcontroller.



Achtung

Wird eine Variable geschrieben, die nur einen Teil einer Prozessvariable abbildet, werden alle übrigen Bits $\,0$ gesetzt.

LEITRECHNER (LR)

Die Adresse setzt sich zusammen aus

- einem beliebigen Präfix
- ▶ einem Rufzeichen (!) als Trenner
- ▶ der Bereichskennung
- einem Unterstrich (_) als Trenner
- ▶ dem eigentlichen Prozessvariablennamen



- ▶ einem Punkt (.) als Trenner
- ▶ einem suffix

Info: Präfix und Suffix sind optional

Beispiel:

```
Mit Präfix und Suffix: weo52!w_pvxx1.Qword oder weo52!w_pvxx1._[0] Kurz: w pvxx1
```

CELLCONTROLLER (CC)

Die Adresse setzt sich zusammen aus

- ▶ einem beliebigen Präfix
- einem Rufzeichen (!) als Trenner
- ▶ dem eigentlichen Prozessvariablennamen
- ▶ einem Punkt (.) als Trenner
- ▶ einem suffix

Info: Präfix und Suffix sind optional

Beispiel:

```
Mit Präfix und Suffix: weo52!pvxx1.gword Oder weo52!pvxx1._[0] Kurz: pvxx1
```

SUFFIXE

Über das Suffix kann ein Teil der 64 Bit Prozessvariable adressiert werden. Folgende Suffixe können für Leitrechner und Cellcontroller verwendet werden:

- ▶ QWord (64 Bit)
- ► _[0], _[1] (32 Bit, Doppelwort)
- ▶ Word_[0] ... Word_[3] (16 Bit, Wort)
- ▶ Byte_[0] ... Byte_[7] (8 Bit, Byte)
- ▶ Bit_[0] ... Bit_[63] (Bit)



6.2.3 Status

Die Adresse setzt sich zusammen aus

- ▶ einem beliebigen Präfix
- ▶ einem Rufzeichen (!) als Trenner
- ▶ den im folgenden angeführten Kennungen für Statusinformationen

LEITRECHNER (LR), KOPFRECHNER (KOPF)

▶ LEITRECHNERKENNUNG

Adressierungsbeispiel: weo52!Leitrechnerkennung Die Leitrechnerkennung wird wie folgt gesetzt:

- 'H', 72 ... HAUS
- '0', 79 ... STO
- 'V', 86 ... V3
- 'W', 87 ... A-Plus
- 'K', 75 ... Kopfrechner
- BETRIEBSZUSTAND

Adressierungsbeispiel: weo52!Betriebszustand Folgende Leitrechnerbetriebszustände sind möglich:

- '1', 31 ... Betriebsbereit
- '2', 32 ... Nicht betriebsbereit
- LOG LINKZUSTAND[0] ... LOG LINKZUSTAND[6]

Adressierungsbeispiel: wEO52!LOG LINKZUSTAND[2]

Der logische Verbindungszustand zeigt an, ob der Datenaustausch mit einem Rechner erlaubt ist oder nicht. Nur der lokale logische Verbindungszustand (lokale Sichtweise) kann durch einen Bedieneingriff im Dialog verändert werden. Das Character-Array enthält für jedes System einen entsprechenden Buchstaben. Der Buchstabe des eigenen Systems ist immer gesetzt.

- 'H', 72 ... HAUS
- 'O', 79 ... STO
- 'V', 86 ... V3
- 'W', 87 ... A-Plus
- 'K', 75 ... Kopfrechner
- 'I', 73 ... Info-Plus Rechner
- 'Z', 90 ... Visualisierungsserver
- ', 95 ... Verbindung inaktiv



▶ PHYS LINKZUSTAND[0] ... PHYS LINKZUSTAND[6]

Adressierungsbeispiel: wEO52!PHYS LINKZUSTAND[6]

Der physikalische Verbindungszustand wird PROCOM-intern geführt und kann nicht beeinflusst werden. Er gibt Auskunft darüber, ob der Datenaustausch mit einem Rechner technisch möglich ist.

- 'H', 72 ... HAUS
- 'O', 79 ... STO
- 'V', 86 ... V3
- 'W', 87 ... A-Plus
- 'K', 75 ... Kopfrechner
- 'I', 73 ... Info-Plus Rechner
- 'Z', 90 ... Visualisierungsserver
- '_', 95 ... Verbindung inaktiv
- STATUS NODE A

Adressierungsbeispiel: weo52!status node a

- 'M', 77 ... Node A im Status Master
- 'S', 83 ... Node A im Status Standby
- ▶ STATUS NODE B

Adressierungsbeispiel: weo52!status node B

- 'M', 77 ... Node A im Status Master
- 'S', 83 ... Node A im Status Standby
- ► GSV_MODE_NODE_A

Adressierungsbeispiel: wEO52!GSV_MODE_NODE_A

- 'N', 78 ... Node A im Status Neustart
- ► GSV_MODE_NODE_B

Adressierungsbeispiel: weo52!gsv mode node в

- 'N', 78 ... Node A im Status Neustart
- REPL_NODE_A

Adressierungsbeispiel: wEO52!REPL NODE A

- 'H', 72 ... Replikation auf Node A aktiv (hot)
- 'C', 67 ... Replikation auf Node A inaktiv (cold)
- ▶ REPL_NODE_B

Adressierungsbeispiel: Adressierungsbeispiel: WEO52!REPL NODE B

- 'H', 72 ... Replikation auf Node A aktiv (hot)
- 'C', 67 ... Replikation auf Node A inaktiv (cold)



► AKTUELLE ZEIT

Adressierungsbeispiel: Adressierungsbeispiel: weo52!AKTUELLE_ZEIT Aktuelle Zeit am System

VERBINDUNGS STATUS

Adressierungsbeispiel: Adressierungsbeispiel: weo52!verbindungs_status
Diese Variable gibt den Verbindungsstatus des Treibers zum Leit bzw. Kopfrechner an

- 0 ... Keine Aktion
- 1 ... Generalabfrage nach Neuverbindung aktiv
- 2 ... Durch Benutzer initierte Generalabfrage aktiv
- 3 ... Bilanzabfrage aktiv

▶ VERBINDUNGS ZUSTAND:

Adressierungsbeispiel: Adressierungsbeispiel: weo52!verbindungs_zustand Diese Variable gibt den Verbindungszustand des Treibers zum Leit bzw. Kopfrechner an Diese Variable gibt den Verbindungsstatus des Treibers zum Leit bzw. Kopfrechner an

- 3 ... Unbekannt
- 2 ... Beide fehlerhaft
- 1 ... Eine fehlerhaft
- 0 ... Beide verbunden

CELLCONTROLLER (CC)

▶ BETRIEBSZUSTAND

Adressierungsbeispiel: Adressierungsbeispiel: weo52!verbindungs_zustand Folgende Leitrechnerbetriebszustände sind möglich:

- '1', 31 ... Betriebsbereit
- '2', 32 ... Nicht Betriebsbereit
- '3', 33 ... Initialisierung aktiv
- ► BETRIEBSZUSTAND_WEXPERTE[0] ... BETRIEBSZUSTAND_WEXPERTE[99]
 Adressierungsbeispiel: Adressierungsbeispiel: weo52!BETRIEBSZUSTAND WEXPERTE[0]
 - '1', 31 ... Betriebsbereit
 - '2', 32 ... Nicht Betriebsbereit

▶ LOG LINKZUSTAND

Adressierungsbeispiel: Adressierungsbeispiel: weo52!log_linkzustand

Der logische Verbindungszustand zeigt an, ob der Datenaustausch mit einem Rechner erlaubt ist oder nicht. Nur der lokale logische Verbindungszustand (lokale Sichtweise) kann durch einen Bedieneingriff im Dialog verändert werden. Das Character-Array enthält für jedes System einen entsprechenden Buchstaben. Der Buchstabe des eigenen Systems ist immer gesetzt.

• 'L', 76 ... Verbindung aktiv



- '?', 63... Verbindungszustand unbekannt
- '_', 95 ... Verbindung inaktiv

▶ PHYS LINKZUSTAND

Adressierungsbeispiel: Adressierungsbeispiel: weo52!PHYS_LINKZUSTAND

Der physikalische Verbindungszustand wird PROCOM-intern geführt und kann nicht beeinflusst werden. Er gibt Auskunft darüber, ob der Datenaustausch mit einem Rechner technisch möglich ist.

- 'L', 76 ... Verbindung aktiv
- '?', 63... Verbindungszustand unbekannt
- '_', 95 ... Verbindung inaktiv

▶ AKTUELLE_ZEIT

Adressierungsbeispiel: Adressierungsbeispiel: wEO52!AKTUELLE_ZEIT Aktuelle Zeit am System

▶ VERBINDUNGS STATUS

Adressierungsbeispiel: Adressierungsbeispiel: weo52!verbindungs_status

Diese Variable gibt den Verbindungsstatus des Treibers zum Leit- bzw. Kopfrechner an

- 0 ... Keine Aktion
- 1 ... Generalabfrage nach Neuverbindung aktiv
- 2 ... Durch Benutzer initierte Generalabfrage aktiv

► VERBINDUNGS ZUSTAND:

Adressierungsbeispiel: Adressierungsbeispiel: weo52!verbindungs_zustand Diese Variable gibt den Verbindungszustand des Treibers zum Cellcontroller an.

- 3 ... Unbekannt
- 2 ... Beide fehlerhaft
- 1 ... Eine fehlerhaft
- 0 ... Beide verbunden

6.3 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.



6.3.1 Treiberobjekte

Folgende Objekttypen stehen in diesem Treiber zur Verfügung:

Treiberobjekttyp	Kanaltyp	Lesen / Schreiben	Unterstützte Datentypen	Beschreibung
Prozessvariable	8	L/S	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, LINT, LWORD	Prozessvariablen.
Status	64	L	USINT, STRING, DATE_AND_TIME	Status.
Fluganzeige für die Zielbahnen	65	L	STRING	Fluganzeige für die Zielbahnen.
Bilanzen	66	L	STRING	Bilanzen des Kopfrechners
Trigger	67	S	UINT	Trigger zum manuellen Ausführen einer Generalabfrage bzw. Kopfabfrage (Bilanzabfrage).
Treibervariable	35	L/S	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	Variable für die statische Analyse der Kommunikation; wird zwischen Treiber und Runtime übertragen (nicht zur SPS).
				Hinweis: Die Adressierung und das Verhalten ist bei den meisten zenon Treibern gleich
				Weitere Infos finden Sie bei den Treibervariablen (auf Seite 35).

6.3.2 Zuordnung der Datentypen

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



Steuerung	zenon	Datenart
ULINT	BOOL	8
ULINT	USINT	9
ULINT	SINT	10
ULINT	UINT	2
ULINT	INT	1
ULINT	UDINT	4
ULINT	DINT	3
ULINT	ULINT	27
ULINT	LINT	26
-	REAL	5
-	LREAL	6
-	STRING	12
-	WSTRING	21
-	DATE	18
-	TIME	17
-	DATE_AND_TIME	20
-	TOD (Time of Day)	19

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

6.4 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 (main.chm::/13028.htm) im Abschnitt Variablen (main.chm::/13045.htm).



6.4.1 XML Import

Für den Import/Export von Variablen gilt:

- ▶ Der Import/Export darf nicht aus dem Globalprojekt gestartet werden.
- ▶ Der Start erfolgt über:
 - Kontextmenü zu Variablen bzw. Datentyp im Projektbaum
 - oder Kontextmenü einer Variablen bzw. eines Datentyps
 - oder Symbol in der Symbolleiste Variablen



Achtung

Beim Import/Überschreiben von existierenden Datentypen werden alle Variablen geändert, die auf diesem existierenden Datentyp basieren.

Beispiel:

Es existiert ein Datentyp XYZ abgeleitet vom Typ INT mit Variablen, die auf diesem Datentyp basieren. Ihre zu importierende XML-Datei enthält ebenfalls einen Datentyp mit Namen XYZ, allerdings abgeleitet vom Typ STRING. Wird dieser Datentyp importiert, so wird der existierende Datentyp überschrieben und bei allen auf ihm basierenden Variablen der Typ angepasst. D.h. die Variablen sind jetzt STRING- und keine INT-Variablen mehr.

6.4.2 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
- wählen Sie in der Dropdownliste von Erweiterter Export/Import ... den Befehl dBase importieren
- 3. folgen Sie dem 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.
- b 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 dem 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.



Δ

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	Тур	Feldgröße	Bemerkung	
KANALNAME	Char	128	Variablenname.	
			Länge kann über den Eintrag MAX_LAENGE in der project.ini eingeschränkt werden.	
KANAL_R	С	128 Ursprünglicher Name einer Variablen, der durch den Eintrag unter VARIABLENNAME ersetzt werden soll (Feld/Spalte mus manuell angelegt werden).		
			Länge kann über den Eintrag MAX_LAENGE in der project.ini eingeschränkt werden.	
KANAL_D	Log	1	Variable wird bei Eintrag ${\mathbb 1}$ gelöscht (Feld/Spalte muss manuell angelegt werden).	
TAGNR	С	128	Kennung.	
			Länge kann über den Eintrag MAX_LAENGE in der project.ini eingeschränkt werden.	
EINHEIT	С	11	Technische Maßeinheit	
DATENART	С	3	Datentyp (z. B. Bit, Byte, Wort,) entspricht dem Datentyp.	
KANALTYP	С	3	Speicherbereich in der SPS (z. B. Merkerbereich, Datenbereich,) entspricht Treiber-Objekttyp.	
HWKANAL	Num	3	Bus-Adresse	
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)	
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	С	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	С	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	С	128	Verknüpftes VBA-Makro zum Lesen der Variablenwerte für die nichtlineare Wertanpassung.
ADJWVBA	С	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, bzw. Zustand 1 bis 4:



Bezeichnung	Тур	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	
A_KLASSE1	N	10	Alarm/Ereignis-Klasse	
MIN_MAX1	С	3	Minimum, Maximum	
FARBE1	N	10	Farbe als Windowskodierung	
GRENZTXT1	С	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.

6.5 Treibervariablen



Das Treiberkit implementiert eine Reihe von Treibervariablen. Diese sind unterteilt in:

- **▶** Information
- Konfiguration
- Statistik und
- Fehlermeldungen

Die Definitionen der im Treiberkit implementierten Variablen sind in der Importdatei drvvar.dbf (auf der CD im Verzeichnis: CD_Laufwerk:/Predefined/Variables) verfügbar und können von dort importiert werden.

Hinweis: Variablennamen müssen in zenon einzigartig sein. Soll nach einem Import der Treibervariablen aus druvar. dbf ein erneuter Import durchgeführt werden, müssen die zuvor importierten Variablen umbenannt werden.





Info

Nicht jeder Treiber unterstützt alle Treibervariablen.

Zum Beispiel werden:

- Variablen für Modem-Informationen nur von modemfähigen Treibern unterstützt
- ▶ Treibervariablen für den Polling-Zyklus nur für rein pollenden Treibern
- verbindungsbezogene Informationen wie ErrorMSG nur von Treibern, die zu einem Zeitpunkt nur eine Verbindung bearbeiten

INFORMATION

Name aus Import	Тур	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.
·			
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.

KONFIGURATION

Name aus Import	Тур	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.
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	Тур	Offset	Erklärung
MaxWriteTime	UDINT	31	Längste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MinWriteTime	UDINT	32	Kürzeste Zeit in Millisekunden (ms), die zum Schreiben benötigt wird.
MaxBlkReadTime	UDINT	40	Längste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
MinBlkReadTime	UDINT	41	Kürzeste Zeit in Millisekunden (ms), die zum Lesen eines Datenblocks benötigt wird.
WriteErrorCount	UDINT	33	Anzahl der Schreibfehler.



ReadSucceedCount	UDINT	35	Anzahl der erfolgreichen Leseversuche.
MaxCycleTime	UDINT	22	Längste Zeit in Millisekunden (ms), die zum Lesen aller angeforderten Daten benötigt wurde.
MinCycleTime	UDINT	23	Kürzeste Zeit in Millisekunden (ms), die zum Lesen aller angeforderten Daten benötigt wurde.
WriteCount	UDINT	26	Anzahl der Schreibversuche.
ReadErrorCount	UDINT	34	Anzahl der fehlerhaften Leseversuche.
MaxUpdateTimeNormal	UDINT	56	Zeit seit letzter Aktualisierung der Prioritätsgruppe Normal in Millisekunden (ms).
MaxUpdateTimeHigher	UDINT	57	Zeit seit letzter Aktualisierung der Prioritätsgruppe нöher in Millisekunden (ms).
MaxUpdateTimeHigh	UDINT	58	Zeit seit letzter Aktualisierung der Prioritätsgruppe нось in Millisekunden (ms).
MaxUpdateTimeHighest	UDINT	59	Zeit seit letzter Aktualisierung der Prioritätsgruppe носьste in Millisekunden (ms).
PokeFinish	BOOL	55	Geht für eine Abfrage auf 1, wenn alle anstehenden Pokes ausgeführt wurden.

FEHLERMELDUNGEN

Name aus Import	Тур	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.
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.

7. Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:

MANUELLES INITIIEREN EINER GERNERALABFRAGE

- ► Anlegen einer Variable vom Treiberobjekttyp Trigger und dem Namen bzw. Kennung bestehend aus einem optionalen Präfix und der Zeichenfolge GA.

 Zum Beispiel: MeinPräfix! GA bzw. GA
- ▶ Die Zuordnung zu einer Verbindung erfolgt über die Netzadresse.
- ▶ Durch Schreiben eines Werts auf die Variable wird eine Generalabfrage angestoßen. Der geschriebene Wert wird dabei als GA-Typ verwendet.
 - GA-Typ: 1 ... Alle PVs
 - GA-Typ: 2 ... Nur spezielle PVs





Info

Der Status der Generalabfrage kann über die Statusvariable VERBINDUNGS_STATUS abgefragt werden.

INITIEREN EINER BILANZABFRAGE VOM KOPFRECHNER

- ► Anlegen einer String-Variable vom Treiberobjekttyp Bilanzen.
- Anlegen einer Variable vom Treiberobjekttyp Trigger und dem Namen bzw. der Kennung bestehend aus einem optionalen Präfix und der Zeichenfolge BILANZEN.

 Zum Beispiel: MeinPräfix!BILANZEN bzw. BILANZEN
- ▶ Die Zuordnung der Variablen zu einer Verbindung erfolgt über die Netzadresse.
- ▶ Durch Schreiben eines Wertes auf die Variable wird eine Kopfabfrage angestoßen. Der geschriebene Wert wird dabei als statistik-Typ verwendet
 - Statistik-Typ: 1 ... Statistiken für den laufenden Betrieb
 - Statistik-Typ: 2 ... Betriebsendestatistik



Info

Der Status der Kopfabfrage kann über die Statusvariable VERBINDUNGS_STATUS abgefragt werden.

Die Bilanzinformation werden in einer String-Variable vom Treiberobjekttyp Bilanzen abgelegt. Die Zuordnung zu einer Verbindung erfolgt über die Netzadresse.

8. Treiberkommandos

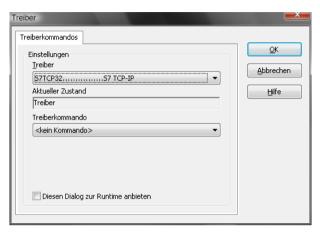
Dieses Kapitel beschreibt Standardfunktionalitäten, die für die meisten zenon Treiber gültig sind. Nicht alle hier beschriebenen Funktionalitäten stehen für jeden Treiber zur Verfügung. Zum Beispiel enthält ein Treiber, der laut Datenblatt keine Modemverbindung unterstützt, auch keine Modem-Funktionalitäten.

Treiberkommandos dienen dazu, Treiber über zenon zu beeinflussen, z. B. starten und stoppen. Die Projektierung erfolgt über die Funktion Treiber Kommandos. Dazu:

- ▶ legen Sie eine neue Funktion an
- ▶ wählen Sie Variablen -> Treiberkommandos



▶ der Dialog zur Konfiguration wird geöffnet





Parameter	Beschreibung		
Treiber	Dropdownliste mit allen im Projekt geladenen Treibern.		
Aktueller Zustand	Fixer Eintrag, in aktuellen Versionen ohne Funktion.		
Treiberkommando	Dropdownliste zur Auswahl des Kommandos.		
▶ Treiber starten (Online-Modus)	Treiber wird neu initialisiert und gestartet.		
▶ Treiber stoppen (Offline-Modus)	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 treiberspezifischer Kommandos. Öffnet Eingabefeld für die Eingabe eines Kommandos.		
▶ Treiber Sollwertsetzen aktivieren	Sollwert setzen auf Treiber ist erlaubt.		
▶ 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).		
Diesen Dialog zur Runtime anbieten	Dialog wird zur Runtime für Änderungen angeboten.		

TREIBERKOMMANDOS IM NETZWERK

Wenn sich der Rechner, auf dem die Funktion Treiberkommandos ausgeführt wird, im zenon Netzwerk befindet, werden zusätzliche Aktionen ausgeführt. Ein spezielles Netzwerkkommando wird vom Rechner zum Server des Projekts gesendet, der dann die gewünschte Aktion auf seinem Treiber durchführt. 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.



9. Fehleranalyse

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

9.1 Analysetool

Alle zenon Module wie z. B. Editor, Runtime, Treiber, usw. schreiben Meldungen in eine gemeinsame Log-Datei. Um sie korrekt und übersichtlich anzuzeigen, benutzen Sie das Programm Diagnose Viewer (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter Start/Alle Programme/zenon/Tools 7.20 -> Diagviewer.

zenon Treiber protokollieren alle Fehler in Log-Dateien. Der Standardordner für die Log-Dateien ist der Ordner Log unterhalb des Ordners ProgramData, zum Beispiel:

C:\ProgramData\COPA-DATA\LOG. Log-Dateien sind Textdateien mit einer speziellen Struktur.

Achtung: Mit den Standardeinstellungen zeichnet ein Treiber nur Fehlerinformationen auf. Mit dem Diagnose 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 Diagnose Viewer kann man auch:

- eben erstellte Einträge live mitverfolgen
- die Aufzeichnungseinstellungen anpassen
- ▶ den Ordner, in dem die Log-Dateien gespeichert werden, ändern

Hinweise:

- 1. Unter Windows CE werden aus Ressourcegründen auch Fehler standardmäßig nicht protokolliert.
- 2. Der Diagnose Viewer zeigt alle Einträge in UTC (Koordinierter Weltzeit) an und nicht in der lokalen Zeit.
- 3. Der Diagnose 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.
- 4. 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.
- 5. 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.



6. 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 Diagnose Viewers weiter aufgezeichnet.

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

9.2 Checkliste

Fragen und Hinweise zur Fehlereingrenzung:

ALLGEMEINE FEHLERSUCHE

- ► Analyse mit Hilfe des Diagnose Viewers (auf Seite 46):
 - -> Welche Meldungen werden angezeigt?
- ► Sind die Teilnehmer im тср/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) -> Taste Eingabe 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. für Modbus: telnet 192.168.0.100 502) -> Taste Eingabe drücken .

Wird der Bildschirm schwarz, konnte eine Verbindung aufgebaut werden.

- 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: Treibervariablen sind reine Statistikvariablen und kommunizieren nicht mit der Steuerung. (Siehe Kapitel Treibervariablen (auf Seite 35).)

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

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?

TREIBER FÄLLT SPORADISCH AUS

Analyse mit Hilfe des Diagnose Viewers (auf Seite 46):

-> Welche Meldungen werden angezeigt?