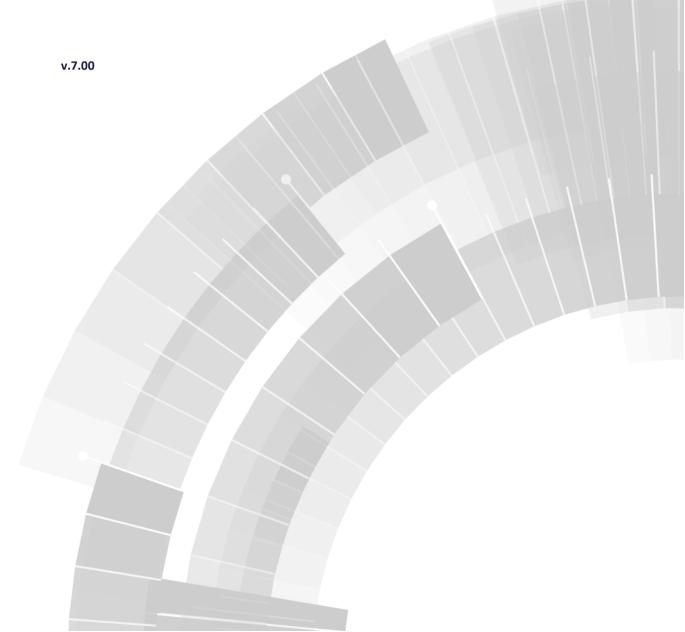


zenon Treiber Handbuch SNMP32





© 2012 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

Τ.	VVIIII	Comme	il bei dei COFA-DATA fillie		
2.	SNMP32				
3.	SNM	P32 - D	Patenblatt	7	
4.	Treiber-Historie				
5.	Vora	ussetzu	ungen	9	
	5.1	PC		g	
	5.2	Steuer	rung	9	
6.	Konf	iguratio	on	10	
	6.1	Anlege	en eines Treibers	10	
	6.2	Einstel	llungen im Treiberdialog	12	
		6.2.1	Allgemein	12	
		6.2.2	Konfiguration	15	
		6.2.3	SNMP Agents	17	
		6.2.4	Offline MIB-Liste	21	
7.	Varia	ablen ar	nlegen	33	
	7.1	Variab	olen im Editor anlegen	33	
	7.2			36	
	7.3	Treibe	erobjekte und Datentypen	37	
		7.3.1	Treiberobjekte	37	
		7.3.2	Zuordnung der Datentypen	38	
	7.4	Variab	olen anlegen durch Import	39	
		7.4.1	XML Import	40	
		7.4.2	Variablen vom Treiber importieren	40	
		7.4.3	DBF Import/Export	46	
	7.5	Treibe	ervariablen	52	
8.	Treik	erspezi	ifische Funktionen	57	
9.	Treib	erkomi	mandos	58	



10. Fehleranalyse 60			
10.1	Analysetool	60	
10.2	Fehlerbehandlung	62	
10.3	Checkliste	66	



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. SNMP32

BEGRIFFSDEFINITIONEN

SNMP	Simple Network Management Protocol
	Dieses Protokoll wird für die Fernverwaltung, Diagnose und den Schutz von Netzwerken und Hosts verwendet. SNMP lässt sich für das Management von Geräten verwenden, die einen sogenannten SNMP-Agent ausführen.
SNMP-Agent	Dient als sogenannter Provider, d.H: er stellt die Informationen eines Gerätes anderen SNMP-Management-Workstations (in unserem Fall zenon mit dem SNMP-Treiber) zur Verfügung. Diese fragen die SNMP-Agents zyklisch ab, um Informationen über die entsprechenden Geräteeigenschaften zu erhalten.
SNMP-Objekt	Bestandteile eines Gerätes, auf die der SNMP-Agent zugreifen, oder die ein SNMP-Agent modifizieren kann, werden als SNMP-Objekte bezeichnet.
MIB	Management Information Bases
	Stellt eine logische Datenbank dar, die eine Gruppe/Sammlung von SNMP- Objekten enthält. Da für verschiedene Gerätetypen und Protokolle unterschiedliche Netzwerkverwaltungsdienste verwendet werden können, besitzt jeder Dienst eine eigene MIB
OID	Objekt Identifier
	Stellt eine spezifische Detailinformation eines SNMP-Objektes dar.
TCP/IP	Transmission Control Protocol / Internet Protocol
	Dabei handelt es sich um einen in vier Schichten aufgebauten Satz von herstellerneutralen, häufig verwendeten Anwendungs- und Transportprotokollen. Wird vom zenon SNMP Treiber verwendet um Netzwerk-Informationen über SNMP auszulesen.
ICMP	Internet Control Message Protocol
	Dieses Protokoll ist für Fehler und Kontrollmeldungen an die beteiligten Rechner während des Übertragungsprozesses verantwortlich.
Ping	Dient zur Überprüfung der Erreichbarkeit eines anderen Computers.
TRAP	In SNMP wird damit eine Meldung bezeichnet, die ein Agent an ein Verwaltungssystem sendet. Damit wird das Auftreten eines Ereignisses auf dem Host angezeigt, auf dem der Agent ausgeführt wird. Der SNMP-Dienst kann beispielsweise so konfiguriert werden, dass er bei Empfang einer Informationsanforderung, die weder den korrekten Communitynamen noch

6



einen akzeptierten Hostnamen für den Dienst enthält, ein Trap sendet.

3. SNMP32 - Datenblatt

Allgemein:	
Treiberdateiname	SNMP32.exe
Treiberbezeichnung	SNMP Treiber
Steuerungs-Typen	SNMP fähige Geräte
Steuerungs-Hersteller	SNMP;

Treiber unterstützt:	
Protokoll	SNMP; SNMPv1; SNMPv2c;
Adressierung: Adress-basiert	х
Adressierung: Namens- basiert	x
Kommunikation spontan	х
Kommunikation pollend	х
Online Browsing	х
Offline Browsing	-
Echtzeitfähig	-
Blockwrite	-
Modemfähig	-

7



Serielles Logging	-
RDA numerisch	-
RDA String	-

Voraussetzungen:	
Hardware PC	Standard Netzwerkkarte
Software PC	- Der Windows "SNMP Trap" Dienst muss installiert und gestartet sein- UDP traffic für port 161 und port 162 sollte an einer etwaigen Firewall freigeschalten sein (oder Anwendung snmptrap.exe)
Hardware Steuerung	-
Software Steuerung	SNMP-Agent (Server)
Benötigt v-dll	-

Plattformen:	
Betriebssysteme	Windows XP, Vista, 7, Server 2003, Server 2008/R2;
CE Plattformen	-;

4. Treiber-Historie

Datum	Treiberversion	Änderung
07.07.08	1400	Treiberdokumentation wurde neu erstellt



5. Voraussetzungen

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

5.1 PC

HARDWARE

Netzwerkkarte.

SOFTWARE

TCP/IP-Protokoll

Installierter SNMP Dienst

5.2 Steuerung

HARDWARE

SNMP fähiger Netzwerkteilnehmer.

HUBs, Router, Bridges, Gateways, Server, Workstations, Drucker usw.

SOFTWARE

SNMP-Dienst und entsprechender SNMP-Agent

TCP/IP Protokoll.



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

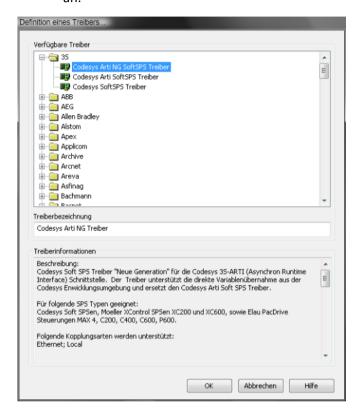
Anlegen eines Treibers 6.1

Um einen neuen Treiber anzulegen:

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

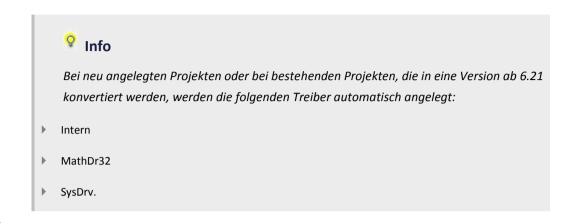


► In der folgenden Dialogbox bietet Ihnen das Programm eine Auflistung aller verfügbaren Treiber an.



- ▶ 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.
- ▶ 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.

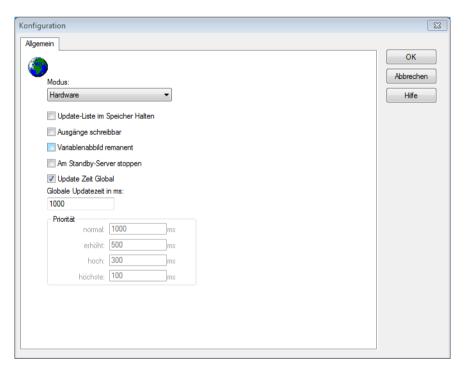




6.2 Einstellungen im Treiberdialog

Folgende Einstellungen können Sie beim Treiber vornehmen:

6.2.1 Allgemein





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. z.B. zwei Variablen vom Typ Merker mit Offset 79, können zur Laufzeit 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 Stop-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.
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 Prioritäten 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 Prioritäten kann die Kommunikation der einzelnen Variablen auf die Wichtigkeit bzw. benötigte Aktualität abgestuft werden. Daraus ergibt sich eine verbesserte Verteilung der Kommunikationslast.

UPDATE ZEIT ZYKLISCHE TREIBER

Für zyklische Treiber gilt:

Beim Sollwert Setzen, Advicen 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.

6.2.2 Konfiguration

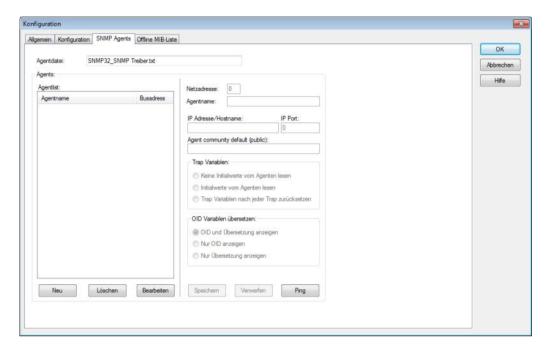




Parameter	Beschreibung
SNMP Konfigurationsdatei	Datei, in der die Einstellungen abgespeichert werden. Sie befindet sich im zenon Projektverzeichnis.
Fehlerwartezeit (ms)	Die Zeit in Millisekunden, die gewartet wird, bis ein Fehler als solcher gewertet wird. D.h.: Sind SNMP-Agents, z. B. durch eine kurzzeitige Auslastung des Netzwerks nicht erreichbar, wird diese Zeitspanne abgewartet, bevor der Treiber die Variablen
	als gestört kennzeichnet. Erreicht der Treiber die Variabien in dieser Zeit, übergeht er die Kurzzeit-Störung ohne Meldung.
	Hinweis: Gilt nicht für Traps, wenn bei der Agentkonfiguration kein Initialwert gelesen wird. Diese Variablen bleiben dann ohne Status leer, auch wenn keine Verbindung zum Agent besteht.
Adressierung über Variablenname oder Kennung	Aktiv: Die Adressierung erfolgt über den Namen der Variablen oder über die Kennung. Die Eigenschaften browse agent und receive online traps stehen dann nicht zur Verfügung.
	Inaktiv: Adressierung erfolgt über Offset in Zuordnungsdatei.
	Achtung: Die Art der Adressierung ist wichtig für den Variablenimport (auf Seite 40). Nachträgliche Änderungen der Adressierung von Variablenname oder Kennung auf Offset
	kann zu Kommunikationsproblemen führen.
Kennung für Adressierung verwenden	Kennung wird verwendet. Steht nur zur Verfügung, wenn "Adressierung über Variablenname oder Kennung" aktiviert wurde.



6.2.3 SNMP Agents





Parameter	Beschreibung
Agentfile	Die Datei in der die Konfigurationen der einzelnen SNMP-Agents gespeichert werden. Diese Datei entspricht dem SNMP Configurationfile, und kann daher in diesem Dialogfenster nicht verändert werden.
Netzadresse	Die zenon interne Netz Adresse der einzelnen Agents. Entspricht der Netzadresse in der Variablendefinition. Sie können diese Netzadresse z.B: zur logischen Gruppierung Ihrer Variablen verwenden.
Agentname	Der frei wählbare Name der einzelnen SNMP-Agents.
IP Adresse/Hostname	Die IP-Adresse der einzelnen Netzwerkteilnehmer/Agents. Es kann entweder die IP-Adresse oder der Computername (Voraussetzung dabei ist eine funktionierende Namensauflösung) verwendet werden. Wenn Traps verwendet werden, dürfen keinen Agents mit gleicher IP-Addresse konfiguriert sein.
IP Port	Portadresse
	Default:
	▶ SNMP: 161
	▶ Traps: 162
Agent community default (public)	Die Zugriffskennung der Agents. Ist standardmäßig immer "public" (nur lesen).
	Ein Community-Name fungiert als Kennwort, das für einen oder mehrere SNMP-Hosts freigegeben ist.
	Akzeptierte Community-Namen werden lediglich zur Authentifizierung eingehender Nachrichten verwendet. Dies kann in den SNMP-Dienst-Einstellungen des Agents konfiguriert werden.
Trap Variablen	Einstellungen für Traps.
▶ Keine Initialwerte vom	Für Traps wird keinen Initialwert gelesen.
Agent lesen	Wert und Status der Variable bleiben so lange leer, bis ein Trap empfangen wird.
	Empfehlung: Wählen Sie diese Option, wenn der Agent Traps versendet, die nicht mittels GET gelesen werden können und damit nicht in der MIBliste enthalten sind.



Initialwerte vom Agent lesen	Startwerte für Traps werden vom Agent gelesen. Startwert für die Trap-Variablen wird vom Agent mittels "GET" gelesen. Sie erhalten den letzten gültigen Wert vom letzten Trap bzw. den Initialwert bei Runtimestart. Wurde der Startwert erfolgreich gelesen, wird die Trap-Variable nicht mehr zyklisch gelesen, sondern nur noch spontan. Empfehlung: Wählen Sie diese Option, wenn Traps vom Agent einmal mittels GET gelesen werden, um dann spontane Meldungen (Traps) zu erwarten.
▶ Trap Variablen nach jeder Trap zurücksetzen	Für Traps wird keinen Initialwert gelesen. Wert und Status der Variable bleiben so lange leer, bis ein Trap empfangen wird. Wird ein Trap empfangen, wird der Wert an die Runtime gesendet, und anschließend sofort mit einem leeren String bzw. Wert 0 überschrieben. Trap-Wert kann durch Reaktionsmatrix erkannt werden und in AML und CEL geschrieben werden Ist bei aktiver Option der Wert der Trap-Variablen in der Runtime 0 oder ein leerer String mit Status Spontan, wurde bereits ein Trap empfangen. Empfehlung: Wählen Sie diese Option, um in der Runtime auf Traps zu reagieren, die immer den gleichen Wert haben, um einen CEL-Eintrag oder Alarm zu erzeugen.

OID Variablen übersetzen	Einstellungen für OID Variablen. Ermöglicht separate Konfiguration der Übersetzung für jeden Agenten.
	Hinweis: Nur verfügbar ab Version 6.50. Alle anderen Versionen werden automatisch mit dem Standard Nur OID anzeigen behandelt.
	Voraussetzungen:
	Die folgenden Eigenschaften definieren, wie die OID-Übersetzung auf eingehende Werte für Variablen vom SNMP-Treiber angewandt wird. Dazu müssen zwei Bedingungen erfüllt sein:
	Die Variable muss vom Treiber-Objekttyp SNMP Variable oder SNMP Trap sein.
	▶ Der eingehende Wert muss vom SNMP-Datentyp OID sein.
	Eingehende Werte, die diese Bedingungen nicht erfüllen, werden unbehandelt weitergeleitet.
OID und Übersetzung	Aktiv: OID und Übersetzung werden als Wert übermittelt.
anzeigen	Die OID wird in einen beschreibenden Text übersetzt und der Wert wird im folgenden Format an die Runtime weitergeleitet: [OID] ([Text])
	Beispiel: .1.3.6.1.6.3.1.1.5.3 (IF-MIB::linkDown).
	Wenn die Übersetzung nicht erfolgreich ist, wird nur die OID als Wert an die Runtime übermittelt.
Nur OID anzeigen	Aktiv: Nur OID wird als Wert übermittelt.
	Hinweis: Standard für alle Versionen vor zenon 6.50.
Nur Übersetzung anzeigen	Aktiv: Nur Übersetzung wird als Wert übermittelt.
	Die OID wird in einen beschreibenden Text übersetzt und nur der übersetzte Text wird an die Runtime weitergeleitet. Wenn die Übersetzung nicht erfolgreich ist, wird die OID als Wert an die Runtime übermittelt.
Neu	Neuen Agents definieren.
Löschen	Agent löschen.
	Agant haarbaitan
Bearbeiten	Agent bearbeiten.



Verwerfen	Verwirft die getroffenen Agent-Einstellungen.
Ping	Testet eine projektierten Agent IP Adress.

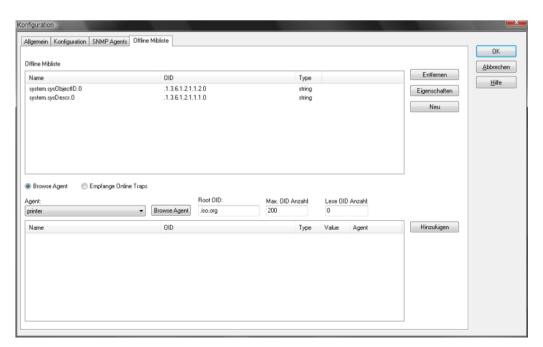
Ergebnis eines erfolgreichen Ping-Versuchs mit Namensauflösung:



Erhalten Sie keine positive Rückmeldung auf einen Ping, konnte entweder der Name nicht richtig aufgelöst werden. In diesem Fall versuchen Sie einen Ping mit der IP-Adresse. Schlägt auch dieser fehl, konnte der Netzwerkpartner/Agent nicht erreicht werden.



6.2.4 Offline MIB-Liste





Parameter	Beschreibung
Offline MIB-Liste	Beinhaltet alle Eigenschaften (=SNMP-Objekte) aller SNMP-Agents, die in der Variablendefinition verfügbar/auswählbar sein sollen. SNMP-Objekte, die nicht in diese Liste übernommen wurden, können später nicht als Variablen definiert werden.
Entfernen	Löscht ausgewählte OID aus der Liste.
Eigenschaften	Zeigt Eigenschaften des ausgewählten Eintrags an.
Neu	Ermöglicht das manuelle Erstellen eines neuen Eintrags (auf Seite 24).
Browse Agent	Aktiv: Agent wird nach OIDs durchsucht. Der ausgewählte Agent wird mittels GETNEXT so lange nach OIDS durchsucht, bis entweder die definierte maximale Anzahl an OIDs empfangen wurde oder der Agent auf GET mit endofMIBview antwortet als Kennzeichnung, dass es keine weitere OIDs mehr gibt. Hinweis: Für manche SNMP Agents kann konfiguriert werden, wie viele SNMP Pakete pro Sekunde beantwortet werden. Eine zu geringe Zahl kann sich auf das Auslesen der OIDs mittels Browse Agent auswirken.
Empfange Online Traps	Aktiv: Über den Windows Dienst SNMP Trap werden vom Treiber die SNMP Trap Meldungen von verschiedenen Agents empfangen und angezeigt. Dazu muss der Agent nicht zwingend konfiguriert sein, auch Traps von nicht konfigurierten Agents werden aufgelistet. Falls die Trapmeldung von einem bereits konfigurierten Agent kommt, wird der Name des Agents in der Spalte "Agent" angezeigt. Damit der Projektierungsrechner Traps empfängt, muss: die Firewall UDP Traffic auf Port 161 und 162 zulassen die Firewall den Prozess snmptrap. exe zulassen
	im Agent der Projektierungsrechner als Ziel für SNMP Traps definiert sein Hinweis: Erhalten Sie eine Fehlermeldung mit Bezug auf 0x2a oder 0x64, läuft möglicherweise der Windows Dienst SNMP Trap nicht.
Agent	Auswahl des Agents aus Dropdownliste, der durch den Treiber mit dem Befehl "GETNEXT" nach unterstützten Informationen durchsucht werden und dessen MIB-Objekte angezeigt werden sollen. Diese Liste beinhaltet alle Agents, die in der Registerkarte SNMP Agents (auf Seite 17) angelegt wurden.
Browse Agent	Durchsucht ausgewählten Agent.

Root OID	Hier ist die Definition ".iso.org" oder ".1.3" zu verwenden.
Max. OID Anzahl	Die Anzahl der OIDs die vom Browser ausgelesen werden sollen.
Lese OID Anzahl	Statusanzeige der bis zu diesem Zeitpunkt ausgelesenen OIDs.
Hinzufügen	Fügt ausgewählte OIDs in Offline MIB-Liste hinzu. Von hier aus können die OIDs zu verschiedenen Agents über Treiber-Online Import automatisch erstellt werden.

Achtung

Der SNMP-Treiber sucht auf Basis der Netzadresse erst den Agent und danach die zur OID gehörende Variable. Das bedeutet:

- Pro Agent darf nur eine Variable für eine OID definiert werden. Existieren für einen Agenten mehrere Variablen mit der gleichen OID, ist undefiniert, welche der Variablen bei Ankunft eines Traps aktualisiert wird.
- Mehrere Variablen mit der selben OID aber unterschiedlichen Agenten sind erlaubt.

Eine ähnliche Einschränkung gilt für die Agents. Für Trapvariablen wird über die Absender IP-Addresse der Agent zugeordnet. Mehreren Agents mit der gleiche IP-Addresse, oder mehrere Treiber mit Agents mit der gleiche IP-Addresse sind nicht unterstützt. Es ist undefiniert welcher Agent bzw. Treiber den Trapwert empfängt.

UNTERSCHIEDE MIB EDITOR UND ONLINE VARIBALEN-IMPORT

Werden Variablen Offline über den MIB Editor (auf Seite 21) angelegt oder über Online Variablenimport (auf Seite 40) angelegt, ergeben sich einige Unterschiede:

VARIABLEN UND AGENT

- Werden Traps für die offline MIB-Liste empfangen, werden Traps von allen IP-Adressen angezeigt. Beim Hinzufügen der empfangenen Traps zur offline MIB-Liste werden Informationen zum Agent nicht gespeichert.
- Werden Variablen vom Treiber importiert, muss darauf geachtet werden, die korrekten
 Variablen für den korrekten Agent als Trap zu erstellen.

KONFIGURIERTE UND UNKONFIGURIERTE AGENTS IM ONLINE-IMPORT

Wenn Traps über den Import-Dialog empfangen werden, werden SMNP-Traps von allen IP-Adressen angezeigt.

- ► Wenn ein Trap von einem Agent kommt, der bereits konfiguriert ist, wird der Agent beim Hinzufügen des Traps angepasst: die Net-Adresse wird korrekt gesetzt.
- ► Kommt der Trap von einem im Treiber nicht konfigurierten Agent (Source IP-Adresse wird in der Liste empfangener Traps angezeigt) und die Variable wird hinzugefügt, wird die Net Adresse 0 verwendet. Diese Net Adresse könnte allerdings bereits von einem anderen Agent verwendet werden und so zu Problemen führen.

ADRESSIERUNG

Beim Browsen eines Agents oder beim Empfangen von Traps im Treiber Online-Import, werden Variablen nicht zur offline MIB-Liste hinzugefügt und empfangen keinen Offset.

Wird die Art der Adressierung in der Treiberkonfiguration (auf Seite 15) von Variablenname oder Kennung auf Offset geändert, werden diese Variablen nicht mehr kommunizieren. Wählen Sie daher die Art der Kommunikation bevor Sie Variablen erstellen.

OID manuell anlegen

Wenn zum Zeitpunkt des Projektierens keine Verbindung zum gewünschten Netzwerkteilnehmer besteht, kann die OID auch ohne Browser Agent angelegt werden.





Parameter	Beschreibung
Name	Hier kann ein frei wählbarer Name verwendet werden.
OID	Die genaue Bezeichnung der OID. Bitte beachten Sie, dass hier die vollständige und korrekte Bezeichnung eingegeben wird, da der zenon SNMP-Treiber die Daten sonst nicht auslesen kann.
Datatype	Integer oder String, abhängig vom anzusprechenden SNMP-Datentyps.

OID-Variablen übersetzen

Die OID-Übersetzung steht zur Verfügung für Variablen von den Treiberobjekttypen:

- SNMP Trap
- **SNMP Variable**

Die Funktion zur OID-Übersetzung ist nur bei Treibern für zenon 6.50 und höher verfügbar. Die Liste aller Inhalte einer Trap wird in jeder Treiberversion unterstützt



Info

Hinweis zum Copyright

Die Funktion zur OID-Übersetzung wurde aus der Net-SNMP Open Source Library in der Version 5.6.1.1 extrahiert.

Copyright 1989, 1991, 1992 by Carnegie Mellon University. Derivative Work - 1996, 1998-2000.

Copyright 1996, 1998-2000 The Regents of the University of California. All Rights Reserved.

KONFIGURATION

Überblick über notwendigen Konfigurationen und der daraus resultierenden Ergebnisse:

- 1. <u>Treiberkonfiguration:</u>
 - a) Ein Treiber vom Typ snmp Treiber wird im Editor angelegt.
 - b) Am Treiber werden die benötigten Agenten konfiguriert (auf Seite 17).

- c) Bei den Agenten wird die gewünschte OID-Übersetzung eingestellt.
- d) In der Offline MIB-Liste (auf Seite 21) werden die Agenten gebrowst und die gewünschten OIDs zur Offline MIB-Liste hinzugefügt.
- e) Danach wird auf Empfange Online Traps umgeschaltet, um die Liste aller Inhalte der Traps in die Offline MIB-Liste einzufügen. Nachdem von den Agenten Traps empfangen wurden, werden die notwendigen OIDs aus den Traps zur Liste hinzugefügt. Die Treiberkonfiguration ist damit abgeschlossen.

2. Projektierung im Editor:

- a) Im Editor werden die Variablen vom Treiber importiert, wobei für jeden Agenten die benötigten Variablen aus der Offline MIB-Liste eingefügt werden.
- b) Die Variablen werden mit Elementen auf Bildern verknüpft, damit sie zur Laufzeit betrachtet werden können.
- c) Die Runtime-Dateien werden erzeugt und die Runtime wird gestartet.

3. Verhalten zur Runtime:

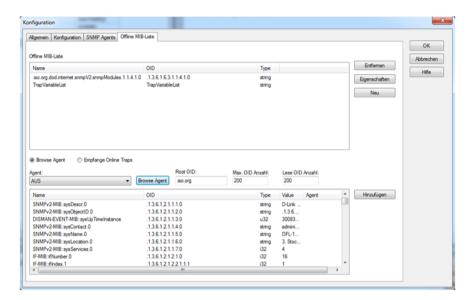
- a) Die Runtime startet den SNMP Treiber.
- b) Die SNMP-Treiberinstanz initialisiert die OID-Übersetzung.
- c) Nachdem die Variablen am Treiber angemeldet wurden, werden die ersten Werte gelesen:
 - Für alle Variablen der Treiber-Objekttypen Pingstatus, SNMP Counter, SNMP variable und SNMP Traps wenn Initialwerte für Traps lesen aktiv ist.
- d) Bei Variablen der Treiberobjekttypen snmp variable und snmp trap werden empfangene Werte vom SNMP-Datentyp oid gemäß der konfigurierten OID-Übersetzung für den jeweiligen Agenten behandelt.
- e) Bei jedem weiteren Polling und jeder empfangenen Trap kommt die OID-Übersetzung für Werte vom SNMP-Datentyp oid gemäß der jeweiligen Agenten-Konfiguration zum Einsatz. Bei Traps wird darüber hinaus noch die Liste aller Inhalte erstellt.

ANZEIGE IN OFFLINE MIB-LISTE

Die Liste aller Inhalte einer Trap wird im Dialog der Offline MIB-Liste angezeigt, wenn dieser auf Trapempfang eingestellt ist.



Bei einem Klick auf Empfange Online Traps wird ein Beispieleintrag für jeden konfigurierten Agenten erzeugt und in der Liste empfangener Traps angezeigt. Für jede empfangene Trap wird eine solche Liste aller Inhalte angelegt und als letzter empfangener Eintrag für diese Trap eingefügt.



Im Screenshot werden empfangene Variablen vom SNMP-Datentyp OID immer als OID und Übersetzung angezeigt. Als Name wird der aus der OID übersetzte beschreibende Text angezeigt.

Die Liste kann in einer Stringvariablen gespeichert werden. Diese muss am SNMP Treiber mit folgenden Patamteren angelegt werden:

- ► Treiber-Objekttyp: SNMP Trap
- ▶ Netzadresse: Index des Agenten, den sie betrifft
- Kennung: TrapVariableList
- Offset

Tipp: Fügen Sie den Beispieleintrag des Agenten in die Offline MIB-Liste ein und importieren Sie dann die Variable vom Treiber.

FORMAT DER STRINGVARIABLEN

Der String ist im gleichen Format wie eine INI-Datei formatiert:

Der erste Knoten [DEFAULT] enthält einen Eintrag COUNT, der die Anzahl der Variablen Bindings in einer Trap angibt. Danach folgt ein Knoten ENTRY_[Nr] für jedes Variable Binding mit den Inhalten OID, OID TRANSLATED, TYPE und VALUE.



SCHEMA

Parameter	Beschreibung
[DEFAULT]	
COUNT=x	Anzahl der Einträge.
[ENTRY_x]	Index
OID=x	OID im numerischen Format.
OID_TRANSLATED=x	Beschreibender Text zur OID, wenn die OID-Übersetzung verfügbar ist .
[TYPE]=x	zenon Datentyp des Werts
[VALUE]=x	Empfangener Wert. Empfangene OID-Werte werden entsprechend der Verfügbarkeit und Konfiguration der OID-Übersetzung hier eingetragen.

BEISPIEL

[DEFAULT]

COUNT=7

[ENTRY_000]

OID=.1.3.6.1.2.1.1.3.0

OID_TRANSLATED=DISMAN-EVENT-MIB::sysUpTimeInstance

TYPE=u32

VALUE=59769454

[ENTRY_001]

OID=.1.3.6.1.6.3.1.1.4.1.0

OID TRANSLATED=SNMPv2-MIB::snmpTrapOID.0

TYPE=string

VALUE=.1.3.6.1.6.3.1.1.5.3 (IF-MIB::linkDown)

[ENTRY_002]

OID=.1.3.6.1.2.1.2.2.1.1.6



```
OID TRANSLATED=IF-MIB::ifIndex.6
TYPE=i32
VALUE=6
[ENTRY 003]
OID=.1.3.6.1.2.1.2.2.1.7.6
OID TRANSLATED=IF-MIB::ifAdminStatus.6
TYPE=i32
VALUE=1
[ENTRY_004]
OID=.1.3.6.1.2.1.2.2.1.8.6
OID_TRANSLATED=IF-MIB::ifOperStatus.6
TYPE=i32
VALUE=2
[ENTRY_005]
OID=.1.3.6.1.3.1057.1
OID_TRANSLATED=SNMPv2-SMI::experimental.1057.1
TYPE=string
VALUE=192.168.0.120
[ENTRY_006]
OID=.1.3.6.1.6.3.1.1.4.3.0
OID_TRANSLATED=SNMPv2-MIB::snmpTrapEnterprise.0
TYPE=string
VALUE=.1.3.6.1.6.3.1.1.5 (SNMPv2-MIB::snmpTraps)
```

Beispielkonfiguration

Um übersetzte OID Variablen und Listen der Inhalte von Traps von einem Agenten zu empfangen und darzustellen:

- 1. konfigurieren Sie den SNMP Agent
- 2. legen Sie einen SNMP Treiber an
- 3. konfigurieren Sie den SNMP Agenten im Treiber
- 4. fügen Sie die benötigten OIDs zur Offline MIB-Liste hinzu
- 5. importieren Sie Variablen vom Treiber
- 6. stellen Sie die Variablen in einem Bild dar

SNMP AGENT KONFIGURIEREN

Der Agent muss der Standard-SNMP-Community public zumindest Leserechte einräumen.

Als Ziel für Traps muss die IP-Adresse des späteren Runtime-Rechners eingetragen werden. Wenn Engineering Station und Runtime-Rechner unterschiedliche Rechner sind, kann die IP-Adresse der Engineering Station hier zusätzlich eingetragen werden. Sofern der Agent das unterstützt, kann man so zur Designzeit auch Traps vom Agenten bekommen. Unterstützt der Agent keine multiplen Trapziele, kann hier die IP-Adresse der Engineering Station als einziges Trap-Ziel eingetragen werden. Dieser Eintrag muss aber vor dem Starten der Runtime durch die IP-Adresse des Runtimerechners ersetzt werden, damit dieser die Traps bekommt.

SNMP TREIBER ANLEGEN

Um einen neuen Treiber anzulegen:

- 1. öffnen Sie den Dialog zur Erstellung eines neuen Treibes über den Befehl Treiber neu:
 - im Kontextmenü im Projektknoten Treiber oder
 - Kontextmenü in der Treiberliste oder
 - Symbolleiste in der Treiberansicht
- 2. wählen Sie im Dialog den SNMP Treiber



SNMP AGENTEN IM TREIBER KONFIGURIEREN

In der Treiberkonfiguration legen Sie in der Registerkarte SNMP Agents (auf Seite 17) die Einstellungen für die Agenten fest. Beachten Sie dabei:

- ▶ Passen Sie den Port an: der Port für SNMP-Abfragen ist 161, für Traps 162.
- ▶ Die Standardcommunity public muss zulässig sein.
- ▶ Als IP-Adresse muss die IP-Adresse des SNMP Agenten eingetragen werden.
- ▶ Über die Optionen im Konfigurationsbereich Trap Variablen steuern Sie das Lesen von Initialwerten für Trapvariablen und ob diese als Wischer behandelt werden.
- ▶ Über die Optionen im Konfigurationsbereich OID Variablen übersetzen steuern die OID-Übersetzung. Hier wird eine der übersetzenden Varianten (alles außer "Nur OID anzeigen") ausgewählt.

BENÖTIGTE OIDS ZUR OFFLINE MIB-LISTE HINZUFÜGEN

Da in diesem Beispiel nur Traps von Interesse sind, aktivieren Sie in der Registerkarte Offline MIB-Liste (auf Seite 21) die Option Empfange Online Traps:

- ▶ Der automatisch für den Agenten erzeugte Eintrag TrapVariableList wird zur Offline MIB-Liste hinzugefügt.
- Nach dem Empfang einer Trap werden die benötigten OIDs zur Liste hinzugefügt.

Schließen Sie den Treiber-Konfigurationsdialog.

VARIABLEN VOM TREIBER IMPORTIEREN

Öffnen Sie das Kontextmenü des SNMP Treibers in der Treiberliste und wählen Sie den Eintrag Variablen vom Treiber importieren.

Im sich öffnenden Dialog (auf Seite 40):

- ▶ wählen Sie die die Einträge in der Offline MIB-Liste aus
- ▶ stellen Sie den zu verwendenden Agenten ein
- importieren Sie die Variablen mit Klick auf OK in die Variablenliste

VARIABLEN IN EINEM BILD DARSTELLEN

Die gewählten Variablen stellen Sie abhängig von ihrem Datentyp in einem dynamischen Zahlenwert oder in einem dynamischen Text dar. Für die Liste aller Inhalte einer Trap sollte ein großes Element projektiert werden, da diese sehr lang werden kann.

Optional kann noch eine Pingstatus-Variable für den Agenten angelegt und mit einem Schalter verknüpft werden, um die Verfügbarkeit des Agenten anzuzeigen.

VERHALTEN ZUR RUNTIME

Nachdem die Runtime gestartet wurde, startet diese den SNMP Treiber. Ab der Treiberinstanzerzeugung durch die Runtime werden folgende Schritte durchgeführt:

- Der Treiber durchläuft seine Initialisierung.
 Dabei werden auch die OID-Übersetzung initialisiert und der Index für die MIBs angelegt.
- Die Runtime meldet die benötigten Variablen beim Treiber an.
 Auf Basis der Netzadresse entscheidet der Treiber, welche Variable welchem Agenten zugeordnet wird und liest die Konfiguration des Agenten aus dem Treiberkonfigurationsfile.
- 3. Für pollende Variablen werden die ersten Werte gelesen und der Polling-Zyklus wird gestartet. Empfangene Werte vom SNMP-Datentyp oid werden entsprechend der Agentenkonfiguration übersetzt.
- 4. Traps erfassen.
 - Bei eingehenden Traps werden zuerst die einzelnen Variable Bindings abgearbeitet OIDs übersetzen, Werte an Runtime schicken und in einer Liste erfasst. Auf Basis dieser Liste wird der String für die Variable mit der Liste aller Inhalte einer Trap generiert und zur Runtime geschickt.
- 5. Beim Beenden der Runtime wird dem Treiber das Kommando zum Beenden übermittelt.
- 6. Der Treiber durchläuft seine Deinitialisierung und löscht dabei den zuvor angelegten Index.



7. Variablen anlegen

So werden Variablen im zenon Editor angelegt:

7.1 Variablen im Editor anlegen

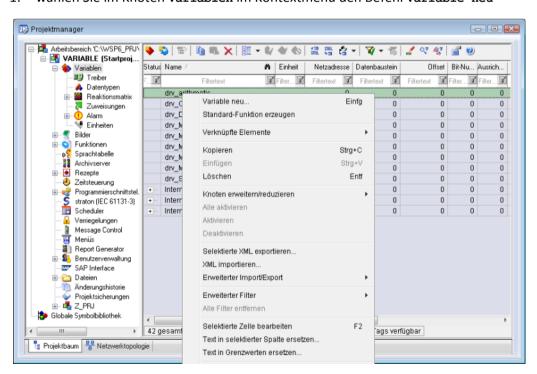
Variablen können angelegt werden:

- als einfache Variable
- ▶ in Arrays 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.
	Achtung: Das Zeichen # ist im Variablennamen nicht zugelassen. Bei Verwendung nicht zugelassener Zeichen kann die Variablenerstellung nicht abgeschlossen werden, die Schaltfläche Fertigstellen bleibt inaktiv.
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.h tm)	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.
Adressierungsoptio nen	Erweiterte Einstellungen für Arrays und Struktur-Variablen. Details dazu lesen Sie im jeweiligen Abschnitt.
Automatische Elementeaktivierun g	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.



7.2 Adressierung

Eigenschaft	Beschreibung
Name	Frei vergebbarer Name.
	ACHTUNG: je Leitsystemprojekt muss der Name eindeutig sein.
	Wenn über den Variablennamen adressiert wird muss hier die OID eingetragen werden.
Kennung	Frei vergebbare Kennung, für Betriebsmittelkennung, Kommentar
	Wenn über die Variablenkennung adressiert wird muss hier die OID eingetragen werden.
Netzadresse	Busadresse oder Netzadresse der Variable.
	Diese Adresse bezieht sich auf die Busadresse der Verbindungsprojektierung im Treiber. Damit wird ausgewählt auf welcher Steuerung sich die Variable befindet.
Datenbaustein	wird für diesen Treiber nicht verwendet
Offset	Offset der Variable, die Speicheradresse der Variable in der Steuerung. Einstellbar [0 4294967295]
Ausrichtung	wird für diesen Treiber nicht verwendet
Bitnummer	Nummer des Bits innerhalb des eingestellten Offsets.
	Mögliche Eingabe [0 65535].
Stringlänge	Nur verfügbar bei String-Variablen: Maximale Anzahl von Zeichen, die die Variable aufnehmen kann.
Treiberobjektt yp	Abhängig vom verwendeten Treiber wird bei der Erstellung der Variable ein Objekttyp ausgewählt und kann hier später geändert werden.
Datentyp	Datentyp der Variable; wird beim Erstellen der Variable ausgewählt und kann hier später verändert werden.
	ACHTUNG: Wenn der Datentyp nachträglich geändert wird, müssen alle anderen Eigenschaften der Variable überprüft bzw. angepasst werden.



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:

OBJEKTE VON PROZESSVARIABLEN IN ZENON

Treiberobjekttyp	Kanaltyp	Lesen / Schreiben	Unterstützte Datentypen	Kommentar
Pingstatus	64	L/S	BOOL	
SNMP-Traps	67	L/S	DINT, UDINT, STRING	Es werden V1 und V2 Traps unterstützt.
SNMP- Variablen	65	L/S	DINT, UDINT, STRING	
SNMP-Zähler	66	L/S	DINT, UDINT	
Treibervariab le	35	L/S	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	Variablen zur statistischen Auswertung der Kommunikation. Weitere Infos finden Sie bei den Treibervariablen (auf Seite 52)



Objekt	Lesen	Schreiben	Kommentar
Konfiguration			Ruft das Treiberkonfigurationsmenü auf
Pingstatus	J		Bit.
+i/u32Bit	J		
String	J		

FUNKTIONSWEISE SNMP-ZÄHLER

Ein Zähler berechnet den Durchschnitt der Wertänderung über Zeit (in Sekunden) zwischen zwei Lesezyklen und sendet diesen an die Runtime. Der Wert aus der OID wird allerdings nicht an die Runtime gesendet. Ist der neue Wert aus der OID kleiner als den vorigen Wert (overflow), wird dieser Wert ignoriert, und solange kein neuer Wert für den Zähler an die Runtime gesendet bis ein neuer Wert berechnet werden kann.

Über diesen Zähler kann z.B. der Netzwerktraffic gemessen werden: Der Agent muss dafür z.B. die Anzahl der gelesenen Bytes ständig hochzählen. Mit einem Lesezyklus von 1000 ms erhält man die Anzahl der Bytes pro Sekunde.

BEISPIEL

► Treiber-Updatezyklus: 30 s

▶ Wert 1: 2500

▶ **Wert 2**: 7500

► Ergebnis für Zählervariable: 166

Denn: 5000 Wertdifferenz über 30 Sekunden = 166

7.3.2 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
	BOOL	8
	USINT	9
	SINT	10
	UINT	2
	INT	1
	UDINT	4
	DINT	3
	ULINT	27
	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.

7.4 Variablen anlegen durch Import

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



7.4.1 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.

7.4.2 Variablen vom Treiber importieren

Um Variablen vom Treiber zu importieren:

- 1. wählen Sie im Kontextmenüs des Treibers den Befehl Variablen vom Treiber importierten
- 2. der Konfigurationsdialog wird geöffnet

Die Optionen des Dialogs hängt von den Einstellungen in der Konfiguration (auf Seite 15) ab

- a) Adressierung über Offset
- b) Adressierung über Variablenname oder Kennung



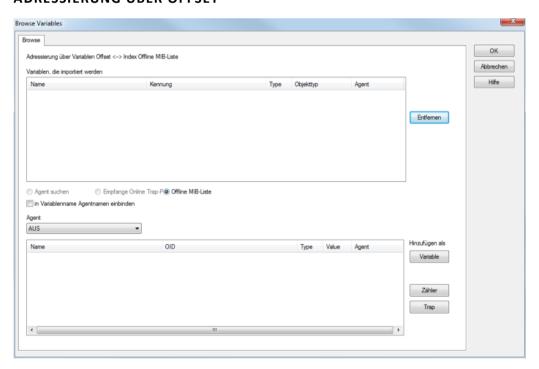
Achtung

Der SNMP-Treiber sucht auf Basis der Netzadresse erst den Agent und danach die zur OID gehörende Variable. Das bedeutet:

- Pro Agent darf nur eine Variable für eine OID definiert werden. Existieren für einen Agenten mehrere Variablen mit der gleichen OID, ist undefiniert, welche der Variablen bei Ankunft eines Traps aktualisiert wird.
- Mehrere Variablen mit der selben OID aber unterschiedlichen Agenten sind erlaubt.

Eine ähnliche Einschränkung gilt für die Agents. Für Trapvariablen wird über die Absender IP-Addresse der Agent zugeordnet. Mehreren Agents mit der gleiche IP-Addresse, oder mehrere Treiber mit Agents mit der gleiche IP-Addresse sind nicht unterstützt. Es ist undefiniert welcher Agent bzw. Treiber den Trapwert empfängt.

ADRESSIERUNG ÜBER OFFSET

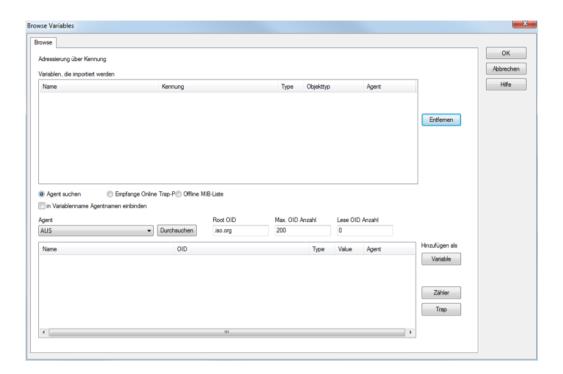




Parameter	Beschreibung			
Adressierung über Variablen Offset <> Index Offline MIB-Liste	Hinweis auf die unter Konfiguration (auf Seite 15) definierte Adressierung.			
zenon Variable zum Erzeugen	Liste der durch Import anzulegenden Variablen in zenon.			
Entfernen	Entfernt Einträge aus der Liste.			
Browse Agent	Kann nicht genutzt werden bei Addressierung über Offset.			
Empfange Online Traps	Kann nicht genutzt werden bei Addressierung über Offset.			
Offline MIB-Liste	Aktiv: Variablen werden aus der Offline MIBliste (auf Seite 21) importiert. Einträge der Liste werden im Listenfeld unterhalb angezeigt.			
In Variablenname Agentnamen einbinden.	Aktiv: Dem Variablennamen wird in der Importliste der Name des Agent vorangestellt, beide Einträge werden durch einen Unterstrich getrennt: Agent Variablenname.			
Agent	Dropdownliste der Agents.			
Liste	Liste der Variablen, die in der Offline MIB-Liste enthalten sind.			
Hinzufügen als:	Auswahl über Schaltflächen, wie zu importierende Variablen zur Liste "zenon Variablen zum Erzeugen" hinzugefügt werden sollen:			
Variable	Hinzufügen als Variable.			
Zähler	Hinzufügen als Zähler.			
Trap	Hinzufügen als Trap.			
OK	Übernimmt alle Einstellungen und beendet den Dialog. Die Variablen in der Liste "zenon Variablen zum Erzeugen" werden importiert.			
Abbrechen	Alle Einstellungen werden verworfen und der Dialog geschlossen.			
Hilfe	Öffnet die Online-Hilfe.			

ADRESSIERUNG ÜBER VARIABLENNAME ODER KENNUNG







Parameter	Beschreibung			
Adressierung über Variablen Offset <> Index Offline MIBliste	Hinweis auf die unter Konfiguration (auf Seite 15) definierte Adressierung.			
zenon Variable zum Erzeugen	Liste der durch Import anzulegenden Variablen in zenon.			
Entfernen	Entfernt Einträge aus der Liste.			
Browse Agent	Aktiv: Agent wird nach OIDs durchsucht.			
	Der ausgewählte Agent wird mittels GETNEXT so lange nach OIDS durchsucht, bis entweder die definierte maximale Anzahl an OIDs empfangen wurde oder der Agent auf GET mit endofMIBview antwortet als Kennzeichnung, dass es keine weitere OIDs mehr gibt.			
	Hinweis: Für manche SNMP Agents kann konfiguriert werden, wie viele SNMP Pakete pro Sekunde beantwortet werden. Eine zu geringe Zahl kann sich auf das Auslesen der OIDs mittels Browse Agent auswirken.			
Empfange Online Traps	Aktiv: Über den Windows Dienst SNMP Trap werden vom Treiber die SNMP Trap Meldungen von verschiedenen Agents empfangen und angezeigt. Dazu muss der Agent nicht zwingend konfiguriert sein, auch Traps von nicht konfigurierten Agents werden aufgelistet. Falls die Trapmeldung von einem bereits konfigurierten Agent kommt, wird der Name des Agents in der Spalte "Agent" angezeigt.			
	Damit der Projektierungsrechner Traps empfängt, muss:			
	b die Firewall UDP Traffic auf Port 161 und 162 zulassen			
	die Firewall den Prozess snmptrap.exe zulassen			
	im Agent der Projektierungsrechner als Ziel für SNMP Traps definiert sein			
	Hinweis: Erhalten Sie eine Fehlermeldung mit Bezug auf 0x2a oder 0x64, läuft möglicherweise der Windows Dienst SNMP Trap nicht.			
Agent	Auswahl des Agents aus Dropdownliste, der durch den Treiber mit dem Befehl "GETNEXT" nach unterstützten Informationen durchsucht werden und dessen MIB-Objekte angezeigt werden sollen. Diese Liste beinhaltet alle Agents, die in der Registerkarte SNMP Agents (auf Seite 17) angelegt wurden.			
Browse Agent	Durchsucht ausgewählten Agent.			
Root OID	Hier ist die Definition ".iso.org" oder ".1.3" zu verwenden.			
Max. OID Anzahl	Die Anzahl der OIDs die vom Browser ausgelesen werden sollen.			



Lese OID Anzahl	Statusanzeige der bis zu diesem Zeitpunkt ausgelesenen OIDs.
Liste	Liste der Variablen, die in der Offline MIB-Liste enthalten sind.
Hinzufügen als:	Auswahl über Schaltflächen, wie zu importierende Variablen zur Liste "zenon Variablen zum Erzeugen" hinzugefügt werden sollen:
> Variable	Hinzufügen als Variable.
Zähler	Hinzufügen als Zähler.
▶ Trap	Hinzufügen als Trap.
OK	Übernimmt alle Einstellungen und beendet den Dialog. Die Variablen in der Liste "zenon Variablen zum Erzeugen" werden importiert.
Abbrechen	Alle Einstellungen werden verworfen und der Dialog geschlossen.
Hilfe	Öffnet die Online-Hilfe.

UNTERSCHIEDE MIB EDITOR UND ONLINE VARIBALEN-IMPORT

Werden Variablen Offline über den MIB Editor (auf Seite 21) angelegt oder über Online Variablenimport (auf Seite 40) angelegt, ergeben sich einige Unterschiede:

VARIABLEN UND AGENT

- ► Werden Traps für die offline MIB-Liste empfangen, werden Traps von allen IP-Adressen angezeigt. Beim Hinzufügen der empfangenen Traps zur offline MIB-Liste werden Informationen zum Agent nicht gespeichert.
- Werden Variablen vom Treiber importiert, muss darauf geachtet werden, die korrekten Variablen für den korrekten Agent als Trap zu erstellen.

KONFIGURIERTE UND UNKONFIGURIERTE AGENTS IM ONLINE-IMPORT

Wenn Traps über den Import-Dialog empfangen werden, werden SMNP-Traps von allen IP-Adressen angezeigt.

- Wenn ein Trap von einem Agent kommt, der bereits konfiguriert ist, wird der Agent beim Hinzufügen des Traps angepasst: die Net-Adresse wird korrekt gesetzt.
- ► Kommt der Trap von einem im Treiber nicht konfigurierten Agent (Source IP-Adresse wird in der Liste empfangener Traps angezeigt) und die Variable wird hinzugefügt, wird die Net Adresse 0



verwendet. Diese Net Adresse könnte allerdings bereits von einem anderen Agent verwendet werden und so zu Problemen führen.

ADRESSIERUNG

Beim Browsen eines Agents oder beim Empfangen von Traps im Treiber Online-Import, werden Variablen nicht zur Offline MIB-Liste hinzugefügt und empfangen keinen Offset.

Wird die Art der Adressierung in der Treiberkonfiguration (auf Seite 15) von Variablenname oder Kennung auf Offset geändert, werden diese Variablen nicht mehr kommunizieren. Wählen Sie daher die Art der Kommunikation bevor Sie Variablen erstellen.

7.4.3 **DBF Import/Export**

Daten können nach dBase exportiert und aus dBase importiert werden.

IMPORT DBF-DATEI

Um den Import zu starten:

- 1. führen Sie einen Rechtsklick auf die Variablenliste aus
- 2. wählen Sie im Dropdownmenü 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.
- 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 Dropdownmenü 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.



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 KANALNAME 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	С	128	Kennung.
			Länge kann über den Eintrag MAX_LAENGE in der project.ini eingeschränkt werden.
EINHEIT	С	11	Technische Maßeinheit
DATENART	С	3	Datenart (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)
7 D D 7 V C T 7 E	N	16	
ARRAYSIZE	N	10	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 KANALTYP und DATENART
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
		1	<u> </u>



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 Alarmverwaltung	
FUNC_NR1	N	10	ID-Nummer der verknüpften Funktions (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.

7.5 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 druvar.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.



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 Servicepack-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	SINT	14	Bitanzahl pro Zeichen der seriellen Schnittstelle.
			Wert = 0, wenn der Treiber keine serielle Kommunikation herstellen kann.
StopBit	SINT	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	UDINT	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 ноћет 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 Höchste 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	UDINT	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	SINT	25	Fehlermeldung als Nummer.
DrvErrorMsg	STRING	30	Fehlermeldung als Klartext.
ErrorFile	STRING	15	Name der Fehlerprotokolldatei.

8. Treiberspezifische Funktionen

Dieser Treiber unterstützt folgende Funktionen:



GET, SET, GETNEXT

Mit den Funktionen GET, SET und GETNEXT werden Daten aus SNMP fähigen Geräten ausgelesen.

PING-STATUS

Über den Ping-Status wird festgestellt, ob das Endgerät per ICMP-Protokoll erreichbar ist.

Dazu wird das Statusbit INVALID (main.chm::/24148.htm) über das Combi-Element oder Reaktionsmatrizen abgefragt.

Wert 1: Gerät ist erreichbar.

Wert mit Status INVALID: Kommunikation ist gestört.

9. Treiberkommandos

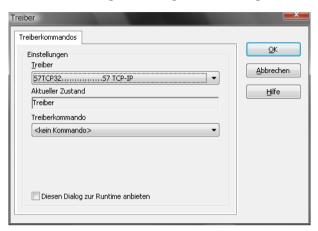
Dieses Kapitel beschreibt Standardfunktionalitäten, die für die meisten zenon Treiber gültig sind. Nicht alle hier beschriebenen Funktionalitäten stehen für jeden Treiber zur Verfügung. Zum Beispiel enthält ein Treiber, der laut Datenblatt keine Modemverbindung unterstützt, auch keine Modem-Funktionalitäten.

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.
<pre>Treiber stoppen (Offline-Modus)</pre>	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).
	esen Dialog zur ntime 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.

10. Fehleranalyse

Sollte es zu Kommunikationsproblemen kommen, bietet dieses Kapitel Hilfen, 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 Diagnose Viewer (main.chm::/12464.htm), das mit zenon mitinstalliert wird. Sie finden es unter Start/Alle Programme/zenon/Tools 7.00 -> 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\zenon\zenon \zenon700\LOG für die zenon Version 7.00 SPO. 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).

Fehlerbehandlung 10.2

FEHLERMELDUNGEN

Fehlertext	Beschreibung
Error on SnmpMgrRequest GET oid=33.1.3.1.0, specified. Busadress=0 Error=0x28 Error=0x	Fehlercode als hexadezimal Zahl à 0x28 = 40 Dec



GET Error: errorStatus=2, errorIndex=1 oid=.1.3.6.1.4.1.171.20.1.2.1.1.1.7 .0, specified. Busadress=0 Error=0x0	Variablenname oder Kennung in zenon ist ungültig. (Die OID gibt es nicht in der MIB am Agent) Mögliche weitere Ursache: Initialwerte für Traps sollen vom Agent ausgelesen werden, aber der Agent stellt auf das GET hin keine OIDs zur Verfügung, sondern sendet diese OIDs nur spontan als Traps. errorStatus=2: = no such item
error Trap Start 0x64!	Fehlermeldung im Editor. Windows Trap Service wurde nicht gestartet. Hinweis: Tritt auf, nachdem Empfange Online Traps in der Treiberkonfiguration (auf Seite 21) oder im Importassistenten aktiviert wurde.
Init TRAP Error 0x64	Meldung aus dem Diagnoseserver. Es werden keine Traps empfangen. Ursache: der Windows Service "SNMP Trap" wurde nicht gestartet. Hinweis: Der Service "SNMP Trap" (snmptrap.exe) benötigt die UDP-Ports 161 und 162. Dieser muss in der Firewall frei geschaltet sein. Wurde der Service gestartet, aber eine andere Anwendung belegt den Port 162, dann: kann der Treiber keine Traps empfangen wird keine Fehlermeldung angezeigt

EINTRÄGE IN LOG-DATEI

Unexpected Error during OID Translation INIT. Tanslation of OIDs will not be available.	Bei der Initialisierung der OID-Übersetzung trat ein Fehler auf. Die OID-Übersetzung ist für diese Treiberinstanz deaktiviert.
Unexpected Error during translation of OID [numerische OID]	Bei der Übersetzung der angegebenen OID trat ein unerwarteter Fehler auf.
Could not translate OID [numerische OID]	Die angegebene OID konnte nicht übersetzt werden. Möglicherweise ist sie nicht im Standard-MIB-Baum auffindbar.
OID Translation successfully initialized.	Die OID-Übersetzung wurde erfolgreich initialisiert.
Translation of OID [numerische OID]	Die angegebene OID wurde erfolgreich übersetzt.

successfull	

FEHLERNUMMERN

Fehlernummern zu SnmpMgrRequest:



ID	Beschreibung	Beschreibung
0	SNMPAPI_FAILURE	/* Generic error code */
1	SNMPAPI_SUCCESS	/* Generic success code */
2	SNMPAPI_ALLOC_ERROR	/* Error allocating memory */
3	SNMPAPI_CONTEXT_INVALID	/* Invalid context parameter */
4	SNMPAPI_CONTEXT_UNKNOWN	/* Unknown context parameter */
5	SNMPAPI_ENTITY_INVALID	/* Invalid entity parameter */
6	SNMPAPI_ENTITY_UNKNOWN	/* Unknown entity parameter */
7	SNMPAPI_INDEX_INVALID	/* Invalid VBL index parameter */
8	SNMPAPI_NOOP	/* No operation performed */
9	SNMPAPI_OID_INVALID	/* Invalid OID parameter */
10	SNMPAPI_OPERATION_INVALID	/* Invalid/unsupported operation */
11	SNMPAPI_OUTPUT_TRUNCATED	/* Insufficient output buf len */
12	SNMPAPI_PDU_INVALID	/* Invalid PDU parameter */
13	SNMPAPI_SESSION_INVALID	/* Invalid session parameter */
14	SNMPAPI_SYNTAX_INVALID	/* Invalid syntax in smiVALUE */
15	SNMPAPI_VBL_INVALID	/* Invalid VBL parameter */
16	SNMPAPI_MODE_INVALID	/* Invalid mode parameter */
17	SNMPAPI_SIZE_INVALID	/* Invalid size/length parameter */
18	SNMPAPI_NOT_INITIALIZED	/* SnmpStartup failed/not called */
19	SNMPAPI_MESSAGE_INVALID	/* Invalid SNMP message format */
20	SNMPAPI_HWND_INVALID	/* Invalid Window handle */
40	SNMP_MGMTAPI_TIMEOUT	
41	SNMP_MGMTAPI_SELECT_FDERRORS	
42	SNMP_MGMTAPI_TRAP_ERRORS	
43	SNMP_MGMTAPI_TRAP_DUPINIT	
44	SNMP_MGMTAPI_NOTRAPS	



45	SNMP_MGMTAPI_AGAIN	
46	SNMP_MGMTAPI_INVALID_CTL	
47	SNMP_MGMTAPI_INVALID_SESSION	
48	SNMP_MGMTAPI_INVALID_BUFFER	
99	SNMPAPI_OTHER_ERROR	/* For internal/undefined errors */
100	SNMPAPI_TL_NOT_INITIALIZED	/* TL not initialized */
101	SNMPAPI_TL_NOT_SUPPORTED	/* TL does not support protocol */
102	SNMPAPI_TL_NOT_AVAILABLE	/* Network subsystem has failed */
103	SNMPAPI_TL_RESOURCE_ERROR	/* TL resource error */
104	SNMPAPI_TL_UNDELIVERABLE	/* Destination unreachable */
105	SNMPAPI_TL_SRC_INVALID	/* Source endpoint invalid */
106	SNMPAPI_TL_INVALID_PARAM	/* Input parameter invalid */
107	SNMPAPI_TL_IN_USE	/* Source endpoint in use */
108	SNMPAPI_TL_TIMEOUT	/* No response before timeout */
109	SNMPAPI_TL_PDU_TOO_BIG	/* PDU too big for send/receive */
199	SNMPAPI_TL_OTHER	/* Undefined TL error */

10.3 Checkliste

- ▶ Ist der SNMP-Dienst korrekt installiert ? (Muss auf allen Geräten installiert sein, die ausgelesen werden sollen außer bei Pingstatus-Abfrage)?
- ▶ Ist der korrekte Key installiert und ausgewählt? (default=public)?
- ► Ist das TCP/IP Protokoll installiert?



TRAPS UND OIB-ÜBERSETZUNG

Zur Diagnose kann der Netzwerkverkehr mit einem Tool wie Wireshark aufgezeichnet und analysiert werden. Wireshark übersetzt OIDs für die gleichen Standard-MIBs wie der SNMP-Treiber. Daher kann die OID-Übersetzung durch Analyse der Wireshark-Captures kontrolliert werden.

Problem	Diagnose	Mögliche Ursache
Der Agent sendet keine Traps.	Ereignisse, die Traps verursachen, werden ausgelöst. Wireshark zeigt aber keine eingehenden Traps an.	Der Trapversandt ist nicht korrekt konfiguriert. Die am Agenten eingestellte Zieladresse muss die IP-Adresse des Rechners mit dem SNMP Treiber sein.
Traps werden nicht in der Runtime angezeigt.	Wireshark zeigt eingehende Traps, angemeldete Variablen werden jedoch nicht aktualisiert.	 Der SNMP Trap Dienst von Windows läuft nicht. Dieser Dienst muss laufen, damit Traps empfangen werden können. Der SNMP Trap Dienst läuft zwar, hat jedoch keinen Zugriff auf das Netzwerk. Die Windows Firewall (oder andere verwendete Firewall) muss dem Dienst das Empfangen von UDP Paketen auf dem Port 162 erlauben. Der Agent ist nicht korrekt konfiguriert. Die IP-Adresse des Agenten in der SNMP-Treiberkonfiguration muss der tatsächlichen IP-Adresse des SNMP Agenten entsprechen. Auf Basis der IP-Adresse erfolgt im Treiber die Zuordnung einer eingehenden Trap zu einem konfigurierten Agenten. Die Variablen haben falsche OIDs. Innerhalb eines Agenten werden die in der Trap enthaltenen Variable Bindings auf Basis ihrer jeweiligen OIDs den derzeit für den Agenten angemeldeten Trap-Variablen zugewiesen. Eine Trap-Variable wird erst aktualisiert, wenn eine Trap mit einem Variable Binding mit der gleichen OID ankommt. Die Traps sind als Wischer eingestellt. Wenn bei einem Agenten die Option Trap Variablen nach jeder Trap zurücksetzen aktiv ist, werden für alle von einer Trap aktualisierten Variablen unmittelbar nach dem Hochsenden der Werte die Variablen auf die Werte 0 bzw. leerer String zurückgesetzt. In einem Text-Element oder numerischen Element in einem Bild kann das so



		aussehen, als wären die Variablen nie aktualisiert worden. Lösungen: - Grenzwerte auf Wertänderungen der betroffenen Variablen - die betroffenen Variablen in Archiven aufzeichnen - Deaktivierung des Zurücksetzens
Die OID Übersetzung funktioniert nicht.	Obwohl die OID-Übersetzung aktiviert ist und Variablenwerte mit übersetzbaren OIDs ankommen (siehe Wireshark), werden in der Runtime keine Übersetzungen angezeigt.	Die MIB-Files sind nicht da. Die OID-Übersetzung funktioniert nur, wenn im zenon Installationsverzeichnis ein Ordner SNMP-MIBS mit den MIB-Dateien als Inhalt vorhanden ist. Ist dies nicht der Fall, können keine OIDs übersetzt werden. Die einzige Übersetzung, die durchgeführt wird, ist das Ersetzen der ersten Ziffer (immer 1) durch "iso", das Root-Element aller OIDs.