Modbus ASCII Serial-Treiber

© 2025 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

Modbus ASCII Serial-Treiber 2
Inhaltsverzeichnis 3
Modbus ASCII Serial-Treiber 5
Übersicht 5
Setup 5
Kanaleigenschaften – Allgemein 6
Tag-Zähler 7
Kanaleigenschaften – Serielle Kommunikation 7
Kanaleigenschaften – Schreiboptimierungen 10
Kanaleigenschaften – Erweitert 12
Kanaleigenschaften – Kommunikationsserialisierung 12
Geräteeigenschaften – Allgemein 14
Geräteeigenschaften – Scan-Modus 15
Geräteeigenschaften – Zeitvorgabe 16
Geräteeigenschaften – Automatische Herabstufung 17
Geräteeigenschaften – Tag-Generierung 18
Geräteeigenschaften – Blockgrößen 21
Geräteeigenschaften – Variablenimporteinstellungen 22
Geräteeigenschaften – Fehlerbehandlung 22
Geräteeigenschaften – Redundanz 23
Automatische Tag-Datenbankgenerierung 24
Datentypbeschreibung 25
Adressbeschreibungen 27
Modbus-ASCII-Adressierung 27
Beschreibung von Funktionscodes 30
Mengenumwerter-Adressierung 30
Flussautomatisierungs-Adressierung 31
Ereignisprotokollmeldungen 32
Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>. 32
Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 32
Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'. 32
Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. | Empfangene Länge = <Anzahl> (Byte), erwartete Länge = <Anzahl> (Byte). 32
Blockanfrage auf Gerät hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>. 33
In Adresse auf Gerät kann nicht geschrieben werden. Gerät hat mit Ausnahme geantwortet. | 33

Adresse = '<Adresse>', Ausnahme = <Code>. .....
Von Adresse auf Gerät kann nicht gelesen werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 33
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 33
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 34
Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Anzahl>. 34
Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 34
Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'. 34
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 35
Tag-Datenbank wird importiert. | Quelldatei = '<Pfad>'. 35
Modbus-Ausnahmecodes 36
Fehlermaskendefinitionen 37
Index **38**

Modbus ASCII Serial-Treiber

Hilfe-Version 1.062

INHALT

Übersicht
Was ist Modbus ASCII Serial-Treiber?

Setup
Wie konfiguriere ich Geräte für die Verwendung mit diesem Treiber?

Automatische Tag-Datenbankgenerierung
Wie kann ich Tags für Modbus ASCII Serial-Treiber konfigurieren?

Datentypbeschreibung
Welche Datentypen unterstützt dieser Treiber?

Adressbeschreibungen
Wie adressiere ich eine Datenposition auf einem Modbus-Gerät?

Ereignisprotokollmeldungen
Welche Meldungen können bei Modbus ASCII Serial-Treiber auftreten?

Übersicht

Modbus ASCII Serial-Treiber bietet eine zuverlässige Möglichkeit, serielle Modbus-ASCII-Geräte mit OPC-Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und benutzerdefinierten Anwendungen, zu verbinden. Es ist für die Verwendung mit seriellen Geräten vorgesehen, die das Modbus-ASCII-Protokoll unterstützen. Die Funktionen des Treibers ermöglichen die Kontrolle über Folgendes: die Datenmenge, die in einer einzelnen Anfrage von einem Gerät angefordert wird, die Wortreihenfolge von 32-Bit-Double-Registerwerten, die Byte-Reihenfolge von 32-Bit- und 16-Bit-Registerwerten und die Anpassung der Adressbasis. Der Treiber kann den RTS-Leitungsbetrieb für die Verwendung mit Funkmodems steuern, die eine bestimmte RTS-Zeitvorgabe erfordern.

Setup

Unterstützte Geräte

Modbus-ASCII-kompatible Geräte
Mengenumwerter mit Daniels/Omni/Elliot-Registeradressierung

Kommunikation Protokoll

Modbus-ASCII-Protokoll

Kanal- und Gerätegrenzwerte

Die von diesem Treiber unterstützte maximale Anzahl von Kanälen liegt bei 100. Die maximale Anzahl von Geräten, die von diesem Treiber unterstützt werden, liegt bei 247 pro Kanal.

Ethernet-Kapselung

Dieser Treiber unterstützt Ethernet-Kapselung, wodurch er mit seriellen Geräten kommunizieren kann, die über einen Terminalserver mit einem Ethernet-Netzwerk verbunden sind. Sie kann in den

Kanaleigenschaften über das Dialogfenster "COM-ID" aufgerufen werden. Weitere Informationen finden Sie in der Server-Hilfedatei.

Unterstützte Kommunikationseigenschaften

Baudrate: 1200, 2400, 9600, 19200
Parität: "Ungerade", "Gerade", "Keine"
Daten-Bits: 8
Stopp-Bits: 1, 2

Hinweis: Einige Geräte unterstützen möglicherweise nicht die aufgelisteten Konfigurationen.

Flusssteuerung

Bei Verwendung eines RS232/RS485-Konverters hängt die Art der erforderlichen Flusssteuerung von den Anforderungen des Konverters ab. Einige Konverter benötigen keine Flusssteuerung, bei anderen ist RTS-Flusssteuerung erforderlich. Der Dokumentation zum Konverter entnehmen Sie seine Flussanforderungen. Ein RS485-Konverter, der automatische Flusssteuerung bietet, ist empfehlenswert.

Hinweise:

1. Bei Verwendung des vom Hersteller mitgelieferten Kommunikationskabels ist es manchmal erforderlich, in den Kanaleigenschaften eine Flusssteuerungseinstellung **RTS** oder **RTS immer** zu wählen.
2. Modbus ASCII Serial-Treiber unterstützt die Flusssteuerungsoption "RTS manuell". Diese Auswahl wird verwendet, um den Treiber für den Betrieb mit Funkmodems zu konfigurieren, die spezielle RTS-Zeitvorgabe-Merkmale erfordern. Weitere Informationen zur Flusssteuerungsoption "RTS manuell" finden Sie in der Server-Hilfedatei.

Kommunikationsserialisierung

Modbus ASCII Serial-Treiber unterstützt Kommunikationsserialisierung, die angibt, ob Datenübertragungen auf jeweils nur einen Kanal begrenzt werden sollten.

Kanaleigenschaften – Allgemein

Dieser Server unterstützt die Verwendung von mehreren gleichzeitigen Kommunikationstreibern. Jedes Protokoll oder jeder Treiber, das/der in einem Serverprojekt verwendet wird, wird als Kanal bezeichnet. Ein Serverprojekt besteht unter Umständen aus vielen Kanälen mit demselben Kommunikationstreiber oder mit eindeutigen Kommunikationstreibern. Ein Kanal fungiert als grundlegender Baustein eines OPC-Links. Diese Gruppe wird verwendet, um allgemeine Kanaleigenschaften (wie z.B. die ID-Attribute und den Betriebsmodus) anzugeben.

Identifikation

Name: Geben Sie die benutzerdefinierte ID dieses Kanals an. Bei jedem Serverprojekt muss jeder Kanalname eindeutig sein. Zwar können Namen bis zu 256 Zeichen lang sein, doch haben einige Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers ein eingeschränktes Anzeigefenster. Der Kanalname ist ein Teil der OPC-Browserinformationen. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
Informationen über reservierte Zeichen finden Sie in der Serverhilfe unter „So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig".

Beschreibung: Geben Sie benutzerdefinierte Informationen über diesen Kanal an.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

Treiber: Geben Sie das Protokoll/den Treiber für diesen Kanal an. Geben Sie den Gerätetreiber an, der während der Kanalerstellung ausgewählt wurde. Es ist eine deaktivierte Einstellung in den Kanaleigenschaften. Die Eigenschaft ist erforderlich, um einen Kanal zu erstellen.
Hinweis: Beim Online-Vollzeitbetrieb des Servers können diese Eigenschaften jederzeit geändert werden. Dies schließt das Ändern des Kanalnamens ein, um zu verhindern, dass Clients Daten am Server registrieren. Wenn ein Client bereits ein Element vom Server abgerufen hat, bevor der Kanalname geändert wurde, sind die Elemente davon nicht beeinflusst. Wenn die Client-Anwendung das Element nach der Änderung des Kanalnamens freigibt und versucht, es mit dem alten Kanalnamen erneut abzurufen, wird das Element nicht akzeptiert. Es sollten keine Änderungen an den Eigenschaften erfolgen, sobald eine große Client-Anwendung entwickelt wurde. Verwenden Sie die richtige Benutzerrollen- und Berechtigungsverwaltung, um zu verhindern, dass Operatoren Eigenschaften ändern oder auf Serverfunktionen zugreifen.

Diagnose

Diagnoseerfassung: Wenn diese Option aktiviert ist, stehen die Diagnoseinformationen des Kanals für OPC-Anwendungen zur Verfügung. Da für die Diagnosefunktionen des Servers eine minimale Mehraufwandsverarbeitung erforderlich ist, wird empfohlen, dass sie bei Bedarf verwendet werden und ansonsten deaktiviert sind. Die Standardeinstellung ist deaktiviert.
Hinweise: Diese Eigenschaft ist nicht verfügbar, wenn der Treiber Diagnosen nicht unterstützt.
Weitere Informationen dazu finden Sie in der Serverhilfe unter "Kommunikationsdiagnosen".

Tag-Zähler

Statische Tags: Gibt die Gesamtanzahl der definierten statischen Tags auf dieser Ebene (Gerät oder Kanal) an. Diese Informationen können bei der Problembehandlung und beim Lastenausgleich hilfreich sein.

Kanaleigenschaften – Serielle Kommunikation

Eigenschaften für serielle Kommunikation stehen seriellen Treibern zur Verfügung und sind je nach Treiber, Verbindungstyp und ausgewählten Optionen unterschiedlich. Unten finden Sie eine Übermenge der möglichen Eigenschaften.
Klicken Sie auf, um zu einem der Abschnitte zu springen: **Verbindungstyp**, **Serielle Port-Einstellungen** oder **Ethernet-Einstellungen**und **Betriebsverhalten**.

Hinweise:

- Während des Online-Vollzeitbetriebs des Servers können diese Eigenschaften jederzeit geändert werden. Verwenden Sie die richtige Benutzerrollen- und Berechtigungsverwaltung, um zu verhindern, dass Operatoren Eigenschaften ändern oder auf Serverfunktionen zugreifen.

- Benutzer müssen die spezifischen Parameter definieren, die verwendet werden sollen. Je nach Treiber können Kanäle identische Kommunikationsparameter gemeinsam nutzen. Es kann lediglich eine gemeinsam genutzte serielle Verbindung für ein virtuelles Netzwerk konfiguriert werden *(siehe **Kanaleigenschaften – Serielle Kommunikation**)*.

Eigenschaftengruppen		
Allgemein	**Verbindungstyp**	
Serielle Kommunikation	Physisches Medium	COM-Port
Schreiboptimierungen	Gemeinsam genutzt	Nein
Erweitert	**Serielle Port-Einstellungen**	
Kommunikationsserialisierung	COM-ID	3
Verknüpfungseinstellungen	Baudrate	19200
	Daten-Bits	8
	Parität	Keine
	Stopp-Bits	1
	Flusssteuerung	Keine
	Betriebsverhalten	
	Bericht Kommunikationsfehler	Aktivieren

Verbindungstyp

Physisches Medium: Wählen Sie den Hardware-Gerätetyp für Datenkommunikation. Zu den Optionen gehören Modem, Ethernet-Kapselung, COM-Port und Keine. Die Standardeinstellung ist COM-Port.

1. **Keine**: Wählen Sie "Keine" aus, um anzugeben, dass keine physische Verbindung vorhanden ist. Dadurch wird der Abschnitt **Operation ohne Kommunikation** angezeigt.
2. **COM-Port**: Wählen Sie "COM-Port" aus, um den Abschnitt **Serielle Port-Einstellungen** anzuzeigen und zu konfigurieren.
3. **Modem**: Wählen Sie "Modem" aus, wenn für die Kommunikation Telefonleitungen verwendet werden. Dies wird im Abschnitt **Modemeinstellungen** konfiguriert.
4. **Ethernet-Kapselung**: Wählen Sie diese Option aus, wenn für die Kommunikation Ethernet-Kapselung verwendet wird. Dadurch wird der Abschnitt **Ethernet-Einstellungen** angezeigt.
5. **Gemeinsam genutzt**: Überprüfen Sie, ob für die Verbindung korrekt angegeben ist, dass die aktuelle Konfiguration mit einem anderen Kanal gemeinsam genutzt wird. Dies ist eine schreibgeschützte Eigenschaft.

Serielle Port-Einstellungen

COM-ID: Gibt die zu verwendende Kommunikations-ID bei der Kommunikation mit dem Kanal zugewiesenen Geräten an. Der gültige Bereich ist 1 bis 9991 bis 16. Die Standardeinstellung ist 1.

Baudrate: Geben Sie die Baudrate an, die zur Konfiguration des ausgewählten Kommunikationsports verwendet werden soll.

Daten-Bits: Geben Sie die Anzahl der Daten-Bits pro Datenwort an. Zu den Optionen gehören 5, 6, 7 oder 8.

Parität: Geben Sie den Paritätstyp für die Daten an. Zu den Optionen gehören "Ungerade", "Gerade" oder "Keine".

Stopp-Bits: Geben Sie die Anzahl der Stopp-Bits pro Datenwort an. Zu den Optionen gehören 1 oder 2.

Flusssteuerung: Wählen Sie aus, wie die RTS- und DTR-Steuerleitungen verwendet werden. Flusssteuerung ist für die Kommunikation mit einigen seriellen Geräten erforderlich. Es gibt folgende Optionen:

- **Keine**: Mit dieser Option werden keine Steuerleitungen umgeschaltet oder in den aktiven Zustand gebracht.
- **DTR**: Mit dieser Option wird die DTR-Leitung in den aktiven Zustand gebracht, wenn der Kommunikationsport geöffnet ist und es auch bleibt.
- **RTS**: Mit dieser Option wird angegeben, dass die RTS-Leitung hoch ist, wenn Byte für die Übertragung zur Verfügung stehen. Nachdem alle gepufferten Byte gesendet wurden, ist die RTS-Leitung niedrig. Dies wird normalerweise mit der RS232/RS485-Konverter-Hardware verwendet.
- **RTS, DTR**: Diese Option ist eine Kombination aus DTR und RTS.
- **RTS immer**: Mit dieser Option wird die RTS-Leitung in den aktiven Zustand gebracht, wenn der Kommunikationsport geöffnet ist und es auch bleibt.
- **RTS manuell**: Mit dieser Option wird die RTS-Leitung basierend auf den für RTS-Leitungssteuerung eingegebenen Zeitvorgaben-Eigenschaften in den aktiven Zustand gebracht. Sie steht nur zur Verfügung, wenn der Treiber manuelle RTS-Leitungssteuerung unterstützt (oder wenn die Eigenschaften gemeinsam benutzt werden und mindestens einer der Kanäle zu einem Treiber gehört, der diese Unterstützung bereitstellt). Durch "RTS manuell" wird die Eigenschaft **RTS-Leitungssteuerung** mit den folgenden Optionen hinzugefügt:
    - **Anstieg**: Geben Sie an, wie lang die RTS-Leitung vor der Datenübertragung ansteigt. Der gültige Bereich liegt zwischen 0 und 9999 Millisekunden. Die Standardeinstellung ist 10 Millisekunden.
    - **Abfall**: Geben Sie die Zeitdauer an, während der die RTS-Leitung nach der Datenübertragung hoch bleibt. Der gültige Bereich liegt zwischen 0 und 9999 Millisekunden. Die Standardeinstellung ist 10 Millisekunden.
    - **Abrufverzögerung**: Geben Sie die Zeit an, um die der Abruf für die Kommunikation verzögert ist. Der gültige Bereich liegt zwischen 0 und 9999. Die Standardeinstellung ist 10 Millisekunden.

Tipp: Bei Verwendung von doppeladrigen RS-485-Kabeln können "Echos" in den Kommunikationsleitungen auftreten. Da diese Kommunikation keine Echounterdrückung unterstützt, wird empfohlen, Echos zu deaktivieren oder einen RS-485-Konverter zu verwenden.

Betriebsverhalten

- **Kommunikationsfehler melden**: Aktivieren oder deaktivieren Sie die Berichterstellung über geringfügige Kommunikationsfehler. Wenn diese Option aktiviert ist, werden geringfügige Fehler beim Auftreten im Ereignisprotokoll angezeigt. Wenn diese Option deaktiviert ist, werden dieselben Fehler nicht angezeigt, selbst wenn es normale Anforderungsfehler sind. Die Standardeinstellung ist "Aktivieren".
- **Inaktive Verbindung schließen**: Wählen Sie diese Option, um die Verbindung zu schließen, wenn es keinerlei Tags mehr gibt, die von einem Client im Kanal referenziert werden. Die Standardeinstellung ist "Aktivieren".
- **Inaktivitätsdauer bis Schließen**: Geben Sie an, wie lang der Server warten soll, bis alle Tags vor dem Schließen des COM-Ports entfernt wurden. Der Standardwert ist 15 Sekunden.

Ethernet-Einstellungen

Hinweis: Nicht alle seriellen Treiber unterstützen Ethernet-Kapselung. Wird diese Gruppe nicht angezeigt, wird die Funktion nicht unterstützt.

Ethernet-Kapselung ermöglicht die Kommunikation mit seriellen Geräten, die im Ethernet-Netzwerk mit Terminalservern verbunden sind. Ein Terminalserver ist im Wesentlichen ein virtueller serieller Port, der TCP/IP-Meldungen im Ethernet-Netzwerk in serielle Daten konvertiert. Sobald die Meldung konvertiert wurde, können Benutzer Standardgeräte verbinden, die eine serielle Kommunikation mit dem Terminalserver unterstützen. Der serielle Port des Terminalservers muss richtig konfiguriert werden, um den Anforderungen des seriellen Geräts zu entsprechen, mit dem er verbunden ist. *Weitere Informationen dazu finden Sie in der Serverhilfe unter "Ethernet-Kapselung verwenden".*

- **Netzwerkadapter**: Geben Sie für Ethernet-Geräte in diesem Kanal einen zu bindenden Netzwerkadapter an. Wählen Sie einen Netzwerkadapter für die Bindung, oder lassen Sie die Standardeinstellung vom Betriebssystem auswählen.
  Bestimmte Treiber zeigen unter Umständen zusätzliche Eigenschaften für Ethernet-Kapselung an. Weitere Informationen dazu finden Sie unter ***Kanaleigenschaften – Ethernet-Kapselung****.*

Modemeinstellungen

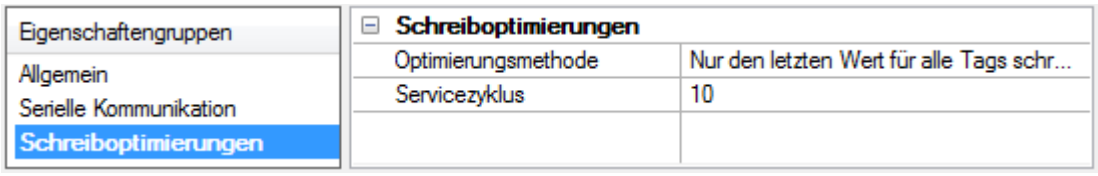
- **Modem**: Geben Sie das installierte Modem an, das für die Kommunikation verwendet werden soll.
- **Verbindungs-Timeout**: Diese Eigenschaft gibt an, wie lang auf das Herstellen von Verbindungen gewartet werden soll, bevor ein Lese- oder Schreibvorgang fehlschlägt. Der Standardwert ist 60 Sekunden.
- **Modemeigenschaften**: Konfigurieren Sie die Modem-Hardware. Durch Klicken auf diese Schaltfläche werden händlerspezifische Modemeigenschaften geöffnet.
- **Automatisches Wählen**: Ermöglicht das automatische Wählen von Einträgen im Telefonbuch. Die Standardeinstellung ist "Deaktivieren". *Weitere Informationen finden Sie unter "Modem Auto-Dial" in der Serverhilfe.*
- **Kommunikationsfehler melden**: Aktivieren oder deaktivieren Sie die Berichterstellung über geringfügige Kommunikationsfehler. Wenn diese Option aktiviert ist, werden geringfügige Fehler beim Auftreten im Ereignisprotokoll angezeigt. Wenn diese Option deaktiviert ist, werden dieselben Fehler nicht angezeigt, selbst wenn es normale Anforderungsfehler sind. Die Standardeinstellung ist "Aktivieren".
- **Inaktive Verbindung schließen**: Wählen Sie diese Option, um die Modemverbindung zu schließen, wenn es keinerlei Tags mehr gibt, die von einem Client im Kanal referenziert werden. Die Standardeinstellung ist "Aktivieren".
- **Inaktivitätsdauer bis Schließen**: Geben Sie an, wie lang der Server warten soll, bis alle Tags vor dem Schließen der Modemverbindung entfernt wurden. Der Standardwert ist 15 Sekunden.

Operation ohne Kommunikation

- **Leseverarbeitung**: Wählen Sie aus, welche Maßnahmen ergriffen werden sollen, wenn ein expliziter Gerätelesevorgang angefordert wird. Zu den Optionen gehören Ignorieren und Fehlgeschlagen. Bei Ignorieren geschieht nichts, bei Fehlgeschlagen wird das Fehlschlagen dem Client durch eine Aktualisierung angezeigt. Die Standardeinstellung ist Ignorieren.

Kanaleigenschaften – Schreiboptimierungen

Der Server muss sicherstellen, dass die von der Client-Anwendung geschriebenen Daten rechtzeitig auf das Gerät gelangen. In Anbetracht dieses Ziels stellt der Server Optimierungseigenschaften bereit, um die jeweiligen Anforderungen zu erfüllen bzw. die Reaktionsfähigkeit der Anwendungen zu verbessern.



Schreiboptimierungen

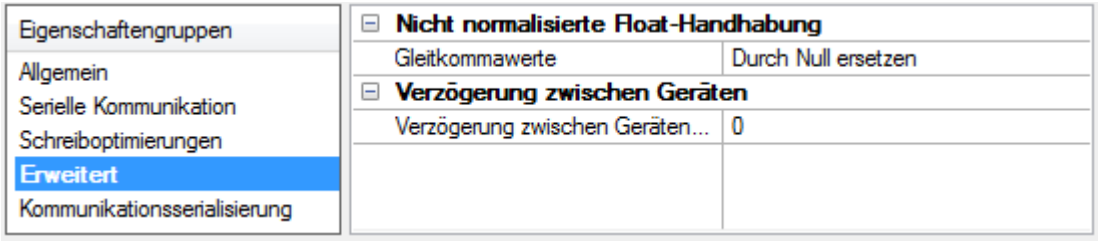
Optimierungsmethode: Mit dieser Option wird gesteuert, wie Schreibdaten an den zugrunde liegenden Kommunikationstreiber weitergeleitet werden. Die Optionen sind:

- **Alle Werte für alle Tags schreiben**: Mit dieser Option wird der Server gezwungen, für jeden Wert einen Schreibvorgang auf dem Controller zu versuchen. In diesem Modus sammelt der Server weiterhin Schreibanforderungen und fügt sie der internen Schreibwarteschlange des Servers hinzu. Der Server verarbeitet die Schreibwarteschlange und versucht, sie zu leeren, indem er so schnell wie möglich Daten auf das Gerät schreibt. In diesem Modus wird sichergestellt, dass alles, was von den Client-Anwendungen geschrieben wird, an das Zielgerät gesendet wird. Dieser Modus sollte ausgewählt werden, wenn die Reihenfolge des Schreibvorgangs oder der Inhalt des Schreibelements eindeutig auf dem Zielgerät zu finden sein muss.
- **Nur den letzten Wert für nicht boolesche Tags schreiben**: Viele aufeinander folgende Schreibvorgänge für denselben Wert können sich aufgrund der Zeit, die tatsächlich zum Senden der Daten auf das Gerät erforderlich ist, in der Schreibwarteschlange ansammeln. Wenn der Server einen Schreibwert aktualisiert, der bereits in die Schreibwarteschlange eingefügt wurde, sind weitaus weniger Schreibvorgänge erforderlich, um denselben Endausgabewert zu erhalten. Auf diese Weise sammeln sich keine zusätzlichen Schreibvorgänge in der Warteschlange des Servers an. Wenn der Benutzer den Schiebeschalter nicht mehr verschiebt, erreicht der Wert im Gerät praktisch in derselben Zeit den richtigen Wert. Dem Modus entsprechend wird jeder Wert, der kein boolescher Wert ist, in der internen Warteschlange des Servers aktualisiert und bei der nächstmöglichen Gelegenheit an das Gerät gesendet. Dies kann die Anwendungsleistung erheblich verbessern.
  Hinweis: Mit dieser Option wird nicht versucht, Schreibvorgänge in Boolesche Werte zu optimieren. Dadurch können Benutzer den HMI-Datenvorgang optimieren, ohne Probleme mit Booleschen Operationen (z.B. eine vorübergehende Schaltfläche) zu verursachen.
- **Nur den letzten Wert für alle Tags schreiben**: Mit dieser Option wird die hinter der zweiten Optimierungsmethode stehende Theorie auf alle Tags angewendet. Sie ist besonders nützlich, wenn die Anwendung nur den letzten Wert an das Gerät senden muss. In diesem Modus werden alle Schreibvorgänge optimiert, indem die derzeit in der Schreibwarteschlange befindlichen Tags vor dem Senden aktualisiert werden. Dies ist der Standardmodus.

Servicezyklus: Wird verwendet, um das Verhältnis von Schreib- und Lesevorgängen zu steuern. Das Verhältnis basiert immer auf einem Lesevorgang für jeden zehnten Schreibvorgang. Für den Servicezyklus wird standardmäßig 10 festgelegt. Dies bedeutet, dass 10 Schreibvorgänge für jeden Lesevorgang erfolgen. Zwar führt die Anwendung eine große Anzahl fortlaufender Schreibvorgänge durch, doch muss sichergestellt werden, dass es für Lesedaten weiterhin Verarbeitungszeit gibt. Die Einstellung 1 hat zur Folge, dass ein Lesevorgang für jeden Schreibvorgang erfolgt. Wenn es keine durchzuführenden Schreibvorgänge gibt, werden Lesevorgänge fortlaufend verarbeitet. Dies ermöglicht eine Optimierung für Anwendungen mit fortlaufenden Schreibvorgängen gegenüber einem ausbalancierteren Datenzufluss und -abfluss.
Hinweis: Es wird empfohlen, dass für die Anwendung die Kompatibilität mit den Verbesserungen zur Schreiboptimierung charakteristisch ist, bevor sie in einer Produktionsumgebung verwendet wird.

Kanaleigenschaften – Erweitert

Diese Gruppe wird verwendet, um erweiterte Kanaleigenschaften anzugeben. Nicht alle Treiber unterstützen alle Eigenschaften; so wird die Gruppe "Erweitert" für jene Geräte nicht angezeigt.



Nicht normalisierte Float-Handhabung: Ein nicht normalisierter Wert wird als "Unendlich", "Nichtzahlenwert (NaN)" oder als "Denormalisierte Zahl" definiert. Die Standardeinstellung ist Durch Null ersetzen. Für Treiber, die eine native Float-Handhabung aufweisen, wird standardmäßig unter Umständen "Nicht geändert" verwendet. Durch Behandlung nicht normalisierter Gleitkommazahlen können Benutzer festlegen, wie ein Treiber mit nicht normalisierten IEEE-754-Gleitkommadaten umgeht. Es folgen Beschreibungen der Optionen:

- **Durch Null ersetzen**: Diese Option ermöglicht es einem Treiber, nicht normalisierte IEEE-754-Gleitkommawerte durch Null zu ersetzen, bevor sie an Clients übertragen werden.
- **Nicht geändert**: Diese Option ermöglicht es einem Treiber, denormalisierte, normalisierte IEEE-754-Nichtzahlenwerte und unendliche IEEE-754-Werte ohne jegliche Konvertierung oder Änderungen an Clients zu übertragen.

Hinweis: Diese Eigenschaft ist deaktiviert, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise werden EFM-Daten nicht durch diese Einstellung beeinflusst.

Weitere Informationen über die Gleitkommawerte finden Sie unter "Mit nicht normalisierten Gleitkommawerten arbeiten" in der Serverhilfe.

Verzögerung zwischen Geräten: Geben Sie die Zeitdauer an, in der der Kommunikationskanal das Senden einer Anforderung an das nächste Gerät verzögert, nachdem Daten vom aktuellen Gerät in demselben Kanal empfangen wurden. Null (0) deaktiviert die Verzögerung.

Hinweis: Diese Eigenschaft ist nicht für alle Treiber, Modelle und abhängige Einstellungen verfügbar.

Kanaleigenschaften – Kommunikationsserialisierung

Die Multithreading-Architektur des Servers ermöglicht Kanälen die parallele Kommunikation mit Geräten. Zwar ist das effizient, doch kann die Kommunikation in Fällen mit physischen Netzwerkeinschränkungen (wie Ethernet-Funksignale) serialisiert werden. Kommunikationsserialisierung schränkt die Kommunikation auf einen Kanal gleichzeitig innerhalb eines virtuellen Netzwerks ein.

Der Begriff "virtuelles Netzwerk" beschreibt eine Sammlung von Kanälen und zugeordneten Geräten, die dieselbe Pipeline für die Kommunikation verwenden. Beispielsweise ist die Pipeline eines Ethernet-Radios das Client-Radio. Alle Kanäle mit demselben Client-Radio werden demselben virtuellen Netzwerk zugeordnet. Kanäle dürfen jeweils nacheinander im Roundrobin-Verfahren kommunizieren. Standardmäßig kann ein Kanal eine Transaktion verarbeiten, bevor die Kommunikation an einen anderen Kanal übergeben wird. Eine Transaktion kann einen oder mehrere Tags einschließen Wenn der steuernde Kanal ein Gerät enthält, das

nicht auf eine Anfrage antwortet, kann der Kanal die Steuerung erst bis zum Timeout der Transaktion freigeben. Dies hat Datenaktualisierungsverzögerungen für die anderen Kanäle im virtuellen Netzwerk zur Folge.

Eigenschaftengruppen
Allgemein
Serielle Kommunikation
Schreiboptimierungen
Erweitert
Kommunikationsserialisier...

Einstellungen auf Kanalebene	
Virtuelles Netzwerk	Keine
Transaktionen pro Zyklus	1
Globale Einstellungen	
Netzwerkmodus	Lastausgleich

Einstellungen auf Kanalebene

Virtuelles Netzwerk: Geben Sie den Kanalmodus der Kommunikationsserialisierung an. Zu den Optionen gehören "Keine" sowie "Netzwerk 1 – Netzwerk 500". Die Standardeinstellung ist "Keine". Es folgen Beschreibungen der Optionen:

- **Keine**: Mit dieser Option wird die Kommunikationsserialisierung für den Kanal deaktiviert.
- **Netzwerk 1 – Netzwerk 500**: Mit dieser Option wird das virtuelle Netzwerk angegeben, dem der Kanal zugewiesen wird.

Transaktionen pro Zyklus: Geben Sie die Anzahl einzelner blockierter/nicht blockierter Lese-/Schreibtransaktionen an, die auf dem Kanal vorkommen können. Wird einem Kanal die Möglichkeit zur Kommunikation gegeben, wird versucht, diese Anzahl von Transaktionen auszuführen. Der gültige Bereich liegt zwischen 1 und 99. Die Standardeinstellung ist 1.

Globale Einstellungen

Netzwerkmodus: Mit dieser Eigenschaft wird gesteuert, wie die Kanalkommunikation delegiert wird. Im Modus **Lastausgleich** wird jedem Kanal die Möglichkeit gegeben, nacheinander zu kommunizieren. Im Modus **Priorität** wird Kanälen die Möglichkeit gegeben, nach den folgenden Regeln (von der höchsten zur niedrigsten Priorität) zu kommunizieren:

1. Kanäle mit ausstehenden Schreibvorgängen haben den höchsten Vorrang.
2. Kanäle mit ausstehenden expliziten Lesevorgängen (durch interne Plugins oder externe Client-Schnittstellen) werden je nach Priorität des Lesevorgangs priorisiert.
3. Gescannte Lesevorgänge und andere periodische Ereignisse (treiberspezifisch).

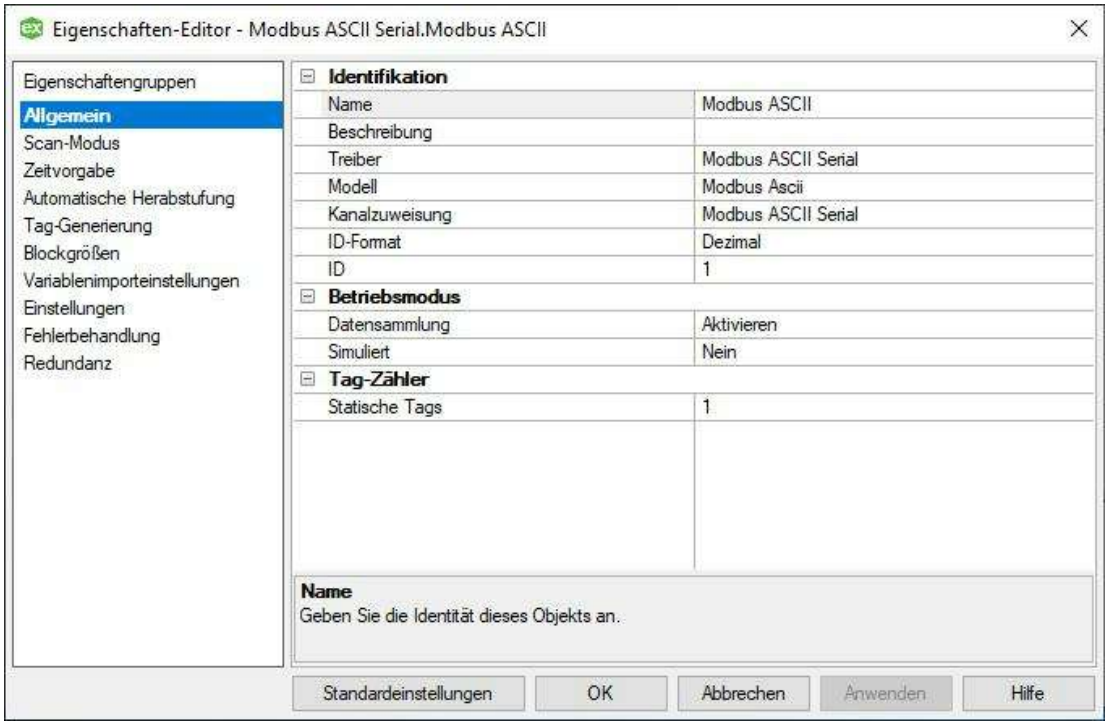
Die Standardeinstellung ist "Lastausgleich" und wirkt sich auf *alle* virtuellen Netzwerke und Kanäle aus.
Geräte, die sich auf unangeforderte Antworten verlassen, sollten nicht in ein virtuelles Netzwerk eingefügt werden. In Situationen, wo die Kommunikationen serialisiert werden muss, wird empfohlen, dass "Automatische Herabstufung" aktiviert wird.

Aufgrund von Unterschieden in der Art und Weise, wie Treiber Daten lesen und schreiben (wie z.B. einzelne blockierte oder nicht blockierte Transaktionen) muss die Eigenschaft "Transaktionen pro Zyklus" der Anwendung möglicherweise angepasst werden. Berücksichtigen Sie dabei die folgenden Faktoren:

- Wie viele Tags müssen von jedem Kanal gelesen werden?
- Wie oft werden Daten in jeden Kanal geschrieben?
- Verwendet der Kanal einen seriellen oder einen Ethernet-Treiber?

- Liest der Treiber Tags in separaten Anfragen, oder werden mehrere Tags in einem Block gelesen?
- Wurden die Zeitvorgabe-Eigenschaften des Geräts (wie z.B. Anforderungs-Timeout und Fehl-geschlagen nach *x* aufeinander folgenden Timeouts) für das Kommunikationsmedium des virtuellen Netzwerks optimiert?

Geräteeigenschaften – Allgemein



Identifikation

Name: Benutzerdefinierte ID dieses Geräts.

Beschreibung: Benutzerdefinierte Informationen über dieses Gerät.

Kanalzuweisung: Benutzerdefinierter Name des Kanals, zu dem dieses Gerät derzeit gehört.

Treiber: Ausgewählter Protokolltreiber für dieses Gerät.

Modell: Die jeweilige Version des Geräts.

ID-Format: Wählen Sie aus, wie die Geräte-ID formatiert wird. Zu den Optionen zählen Formatierungen als Dezimal-, Oktal- oder Hexadezimalzahl.

ID: Die eindeutige Gerätenummer. Seriellen Modbus-Geräten werden Geräte-IDs im Bereich 1 bis 247 zugewiesen.

Betriebsmodus

Datensammlung: Diese Eigenschaft steuert den aktiven Status des Geräts. Zwar sind Gerätekommunikationen standardmäßig aktiviert, doch kann diese Eigenschaft verwendet werden, um ein physisches Gerät zu deaktivieren. Kommunikationen werden nicht versucht, wenn ein Gerät deaktiviert ist. Vom Standpunkt eines Clients werden die Daten als ungültig markiert und Schreibvorgänge werden nicht akzeptiert. Diese Eigenschaft kann jederzeit durch diese Eigenschaft oder die System-Tags des Geräts geändert werden.

Simuliert: Diese Option versetzt das Gerät in den Simulationsmodus. In diesem Modus versucht der Treiber nicht, mit dem physischen Gerät zu kommunizieren, aber der Server gibt weiterhin gültige OPC-Daten zurück. Durch Auswählen von "Simuliert" wird die physische Kommunikation mit dem Gerät angehalten, OPC-Daten können jedoch als gültige Daten dem OPC-Client zurückgegeben werden. Im Simulationsmodus behandelt der Server alle Gerätedaten als reflektierend: was auch immer in das simulierte Gerät geschrieben wird, wird zurückgelesen, und jedes OPC-Element wird einzeln behandelt. Die Speicherzuordnung des Elements basiert auf der Gruppenaktualisierungsrate. Die Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

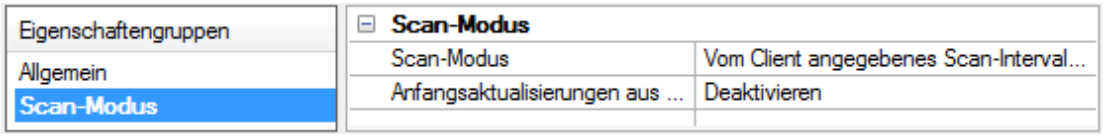
Hinweise:

1. Dieses System-Tag (_Simulated) ist schreibgeschützt und kann für den Laufzeitschutz nicht geschrieben werden. Das System-Tag ermöglicht es, dass diese Eigenschaft vom Client überwacht wird.
2. Im Simulationsmodus basiert die Speicherzuordnung des Elements auf Client-Aktualisierungsraten (Gruppenaktualisierungsrate für OPC-Clients oder Scan-Intervall für native und DDE-Schnittstellen). Das bedeutet, dass zwei Clients, die dasselbe Element mit unterschiedlichen Aktualisierungsraten referenzieren, verschiedene Daten zurückgeben.

Der Simulationsmodus ist nur für Test- und Simulationszwecke. Es sollte niemals in einer Produktionsumgebung nie verwendet werden.

Geräteeigenschaften – Scan-Modus

Der Scan-Modus gibt das vom abonnierten Client angeforderte Scan-Intervall für Tags an, die Gerätekommunikation erfordern. Synchrone und asynchrone Lese- und Schreibvorgänge des Geräts werden so bald wie möglich verarbeitet; unbeeinflusst von den Eigenschaften für den Scan-Modus.



Scan-Modus: Geben Sie an, wie Tags im Gerät für an abonnierende Clients gesendete Aktualisierungen gescannt werden. Es folgen Beschreibungen der Optionen:

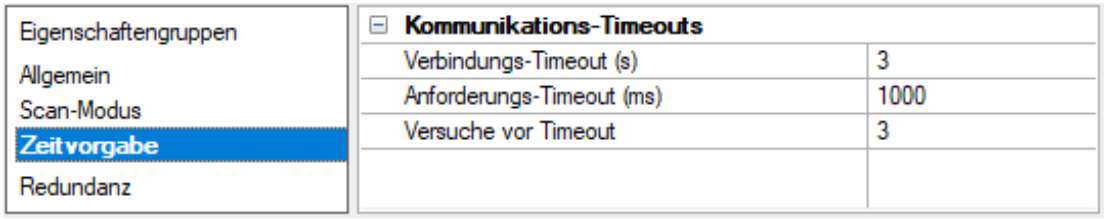
- **Vom Client angegebenes Scan-Intervall berücksichtigen**: Dieser Modus verwendet das vom Client angeforderte Scan-Intervall.
- **Datenanfrage nicht schneller als Scan-Intervall**: Dieser Modus gibt den Wert an, der als maximales Scan-Intervall festgelegt wurde. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
  Hinweis: Wenn der Server über einen aktiven Client und Elemente für das Gerät verfügt und der Wert für das Scan-Intervall erhöht wird, werden die Änderungen sofort wirksam. Wenn der Wert für das Scan-Intervall verringert wird, werden die Änderungen erst wirksam, wenn alle Client-Anwendungen getrennt wurden.

- **Alle Datenanfragen im Scan-Intervall**: Dieser Modus erzwingt, dass Tags im angegebenen Intervall nach abonnierten Clients gescannt werden. Der gültige Bereich liegt zwischen 10 und 99999990 Millisekunden. Die Standardeinstellung ist 1000 Millisekunden.
- **Nicht scannen, nur Abruf anfordern**: In diesem Modus werden Tags, die zum Gerät gehören, nicht periodisch abgerufen, und es wird auch kein Lesevorgang durchgeführt, um den Anfangswert eines Elements abzurufen, sobald es aktiv wird. Es liegt in der Verantwortung des OPC-Clients, Aktualisierungen abzurufen, entweder durch Schreiben in das _DemandPoll-Tag oder durch Ausgeben expliziter Lesevorgänge des Geräts für einzelne Elemente. *Weitere Informationen finden Sie unter "Geräte-Bedarfsabruf" in der Serverhilfe*.
- **Durch Tag angegebenes Scan-Intervall berücksichtigen**: Dieser Modus erzwingt das Scannen statischer Tags im Intervall, das in ihrer statischen Konfiguration Tag-Eigenschaften angegeben wurde. Dynamische Tags werden in dem vom Client angegebenen Scan-Intervall gescannt.

Anfangsaktualisierungen aus Cache: Wenn diese Option aktiviert ist, kann der Server die ersten Aktualisierungen für neu aktivierte Tag-Referenzen aus gespeicherten (Cache-)Daten zur Verfügung stellen. Cache-Aktualisierungen können nur bereitgestellt werden, wenn die neue Elementreferenz dieselben Eigenschaften für Adresse, Scan-Intervall, Datentyp, Client-Zugriff und Skalierung gemeinsam nutzt. Ein Lesevorgang des Geräts wird nur für die Anfangsaktualisierung für die erste Client-Referenz verwendet. Der Standardeinstellung ist "Deaktiviert"; immer wenn ein Client eine Tag-Referenz aktiviert, versucht der Server, den Anfangswert vom Gerät zu lesen.

Geräteeigenschaften – Zeitvorgabe

Mithilfe der Zeitvorgabe-Eigenschaften des Geräts kann die Antwort des Treibers auf Fehlerbedingungen so angepasst werden, dass sie den Anforderungen der Anwendung entspricht. In vielen Fällen erfordert die Umgebung für eine optimale Leistung Änderungen an diesen Eigenschaften. Faktoren wie elektrisch generiertes Rauschen, Modemverzögerungen und fehlerhafte physische Verbindungen können beeinflussen, wie viele Fehler oder Timeouts ein Kommunikationstreiber feststellt. Zeitvorgabe-Eigenschaften sind für jedes konfigurierte Gerät spezifisch.



Kommunikations-Timeouts

Verbindungs-Timeout: Mit dieser Eigenschaft (die in erster Linie von Ethernet-basierten Treibern verwendet wird) wird die Zeitdauer gesteuert, die zum Herstellen einer Socket-Verbindung mit einem Remote-Gerät erforderlich ist. Die Verbindungszeit des Gerät ist häufig länger als normale Kommunikationsanforderungen mit demselben Gerät. Der gültige Bereich liegt zwischen 1 und 30 Sekunden. Die Standardeinstellung ist normalerweise 3 Sekunden, kann jedoch abhängig vom jeweiligen Treiber unterschiedlich sein. Wenn diese Einstellung nicht vom Treiber unterstützt wird, ist sie deaktiviert.
Hinweis: Aufgrund der Art der UDP-Verbindungen ist die Einstellung für Verbindungs-Timeout nicht anwendbar, wenn die Kommunikation über UDP erfolgt.

Anforderungs-Timeout: Geben Sie ein von allen Treibern verwendetes Intervall an, um zu bestimmen, wie lange der Treiber abschließend auf eine Antwort vom Zielgerät wartet. Der gültige Bereich liegt zwischen 50 und 9999999 Millisekunden (167 Minuten). Die Standardeinstellung ist im Allgemeinen 1000 Millisekunden,

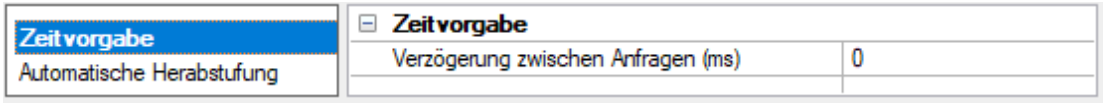
kann jedoch abhängig vom Treiber unterschiedlich sein. Das Standard-Timeout für die meisten seriellen Treiber basiert auf einer Baudrate von 9600 Baud oder besser. Wenn ein Treiber bei niedrigeren Baudraten verwendet wird, erhöhen Sie das Timeout, um die erhöhte Zeit auszugleichen, die zum Abrufen von Daten erforderlich ist.

Versuche vor Timeout: Geben Sie an, wie oft der Treiber eine Kommunikationsanforderung wiederholt, bevor er die Anforderung als fehlgeschlagen und das Gerät als fehlerhaft erachtet. Der gültige Bereich liegt zwischen 1 und 10. Die Standardeinstellung ist normalerweise 3, kann sich jedoch abhängig vom jeweiligen Treiber ändern. Die Anzahl der für eine Anwendung konfigurierten Wiederholungen hängt größtenteils von der Kommunikationsumgebung ab. Diese Eigenschaft trifft sowohl auf Verbindungsversuche als auch auf Anforderungsversuche zu.

Zeitvorgabe

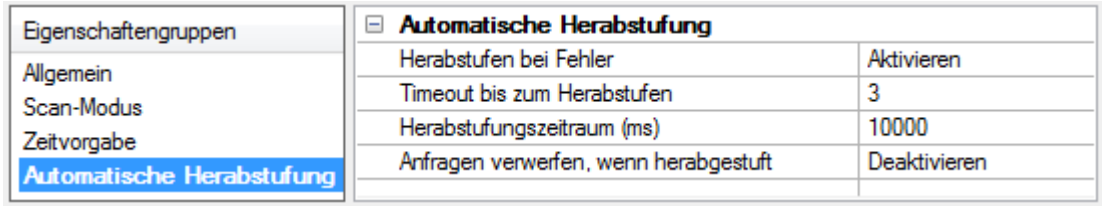
Verzögerung zwischen Anfragen: Geben Sie an, wie lange der Treiber wartet, bevor er die nächste Anforderung an das Zielgerät sendet. Sie setzt das dem Gerät zugewiesene normale Tag-Abfrageintervall sowie einmalige Lese- und Schreibvorgänge außer Kraft. Diese Verzögerung kann bei Geräten mit langsamen Durchlaufzeiten und in Situationen nützlich sein, in denen die Netzwerklast problematisch ist. Das Konfigurieren einer Verzögerung für ein Gerät wirkt sich auf die Kommunikation mit allen anderen Geräten im Kanal aus. Es wird empfohlen, dass Benutzer jedes Gerät trennen, das eine Verzögerung zwischen Anfragen für einen separaten Kanal erfordert (sofern möglich). Andere Kommunikationseigenschaften (wie z.B. Kommunikationsserialisierung) können diese Verzögerung verlängern. Der gültige Bereich liegt zwischen 0 und 300000 Millisekunden; jedoch können einige Treiber ggf. den maximalen Wert wegen einer Funktion ihrer spezifischen Konstruktion beschränken. Die Standardeinstellung ist 0. Dies weist darauf hin, dass es keine Verzögerung zwischen Anfragen mit dem Zielgerät gibt.

Hinweis: Nicht alle Treiber unterstützen Verzögerung zwischen Anfragen. Diese Einstellung wird nicht angezeigt, wenn sie nicht zur Verfügung steht.



Geräteeigenschaften – Automatische Herabstufung

Die Eigenschaften für automatische Herabstufung können ein Gerät vorübergehend in den Nicht-Scan-Modus versetzen, falls das Gerät nicht antwortet. Dadurch, dass ein nicht reagierendes Gerät für einen bestimmten Zeitraum offline gestellt wird, kann der Treiber weiterhin seine Kommunikation mit anderen Geräten in demselben Kanal optimieren. Nach Ablauf dieses Zeitraums versucht der Treiber die Kommunikation mit dem nicht reagierenden Gerät erneut. Wenn das Gerät reagiert, wird es wieder zum Scannen freigegeben. Andernfalls wird sein Nicht-Scan-Zeitraum erneut gestartet.



Herabstufen bei Fehler: Wird diese Option aktiviert, wird das Gerät automatisch in den Nicht-Scan-Modus versetzt, bis es wieder antwortet.

Tipp: Ermitteln Sie, wenn sich ein Gerät im Nicht-Scan-Modus befindet, indem Sie seinen herabgestuften Status mit dem _AutoDemoted-System-Tag überwachen.

Timeout bis zum Herabstufen: Legen Sie fest, wie viele aufeinander folgende Zyklen von Anforderungs-Timeouts und Wiederholungen vorkommen, bevor das Gerät in den Nicht-Scan-Modus versetzt wird. Der gültige Bereich ist 1 bis 30 aufeinander folgende Fehlschläge. Die Standardeinstellung ist 3.

Herabstufungszeitraum: Gibt an, wie lange das Gerät im Nicht-Scan-Modus sein sollte, wenn der Timeout-Wert erreicht wird. Während dieses Zeitraums werden keine Leseanforderungen an das Gerät gesendet, und für alle den Leseanforderungen zugeordneten Daten wird schlechte Qualität festgelegt. Wenn dieser Zeitraum abgelaufen ist, versetzt der Treiber das Gerät in den Scan-Modus und ermöglicht einen weiteren Kommunikationsversuch. Der gültige Bereich liegt zwischen 100 und 3600000 Millisekunden. Die Standardeinstellung ist 10000 Millisekunden.

Anfragen verwerfen, wenn herabgestuft: Durch Aktivieren dieser Option wird ausgewählt, ob Schreibanforderungen während des Nicht-Scan-Zeitraums versucht werden sollten. Deaktivieren Sie diese Option, damit Schreibanforderungen unabhängig vom Herabstufungszeitraum immer gesendet werden. Aktivieren Sie diese Option, um Schreibvorgänge zu verwerfen; auf dem Server schlägt jede von einem Client empfangene Schreibanforderung automatisch fehl, und es wird keine Meldung im Ereignisprotokoll angezeigt.

Geräteeigenschaften – Tag-Generierung

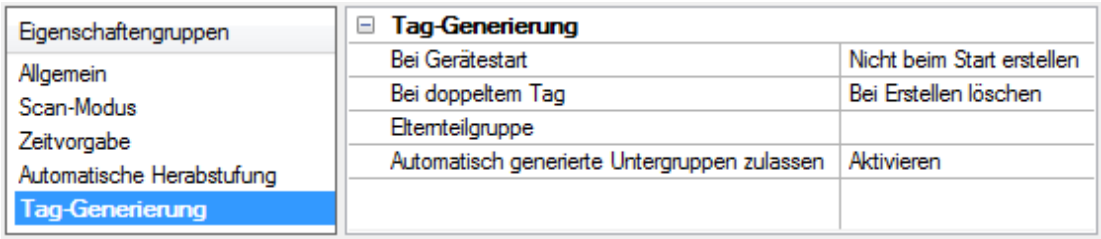
Mithilfe der Funktionen zur automatischen Tag-Datenbankgenerierung wird die Einrichtung einer Anwendung zu einem Plug-and-Play-Vorgang. Ausgewählte Kommunikationstreiber können so konfiguriert werden, dass automatisch eine Liste von Tags erstellt wird, die gerätespezifischen Daten entsprechen. Diese automatisch generierten Tags (die von der Art des unterstützenden Treibers abhängen) können von den Clients durchsucht werden.

Nicht alle Geräte und Treiber unterstützen die automatische Tag-Datenbankgenerierung und nicht alle unterstützen die gleichen Datentypen. Lesen Sie für weitere Informationen die Beschreibungen der Datentypen oder die Liste der unterstützten Datentypen für jeden Treiber.

Wenn das Zielgerät seine eigene lokale Tag-Datenbank unterstützt, liest der Treiber die Tag-Informationen des Geräts und verwendet die Daten zum Generieren von Tags innerhalb des Servers. Wenn das Gerät benannte Tags nicht nativ unterstützt, erstellt der Treiber eine Liste von auf treiberspezifischen Informationen basierenden Tags. Ein Beispiel dieser beiden Bedingungen sieht wie folgt aus:

1. Wenn ein Datenerfassungssystem seine eigene lokale Tag-Datenbank unterstützt, verwendet der Kommunikationstreiber die im Gerät gefundenen Tag-Namen, um die Tags des Servers zu erstellen.
2. Wenn ein Ethernet-E/A-System die Erkennung seiner eigenen verfügbaren E/A-Modultypen unterstützt, generiert der Kommunikationstreiber automatisch Tags auf dem Server, die auf den E/A-Modultypen im Ethernet-E/A-Rack basieren.

Hinweis: Der Betriebsmodus zur automatischen Tag-Datenbankgenerierung ist komplett konfigurierbar. *Weitere Informationen finden Sie in den nachfolgenden Eigenschaftsbeschreibungen.*



Bei Eigenschaftsänderung: Unterstützt das Gerät automatische Tag-Generierung, wenn bestimmte Eigenschaften geändert werden, so wird die Option **Bei Eigenschaftsänderung** angezeigt. Diese Option ist standardmäßig auf **Ja** festgelegt, kann jedoch auf **Nein** gesetzt werden, um zu steuern, wann eine Tag-Generierung stattfindet. In diesem Fall muss die Aktion **Tags erstellen** manuell aufgerufen werden, damit eine Tag-Generierung stattfindet.

Bei Gerätestart: Geben Sie an, wann OPC-Tags automatisch generiert werden. Es folgen Beschreibungen der Optionen:

- **Nicht beim Start erstellen**: Mit dieser Option wird verhindert, dass der Treiber irgendwelche OPC-Tags dem Tag-Raum des Servers hinzufügt. Dies ist die Standardeinstellung.
- **Immer beim Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Gerät für Tag-Informationen bewertet. Es werden auch jedes Mal, wenn der Server gestartet wird, Tags dem Tag-Raum des Servers hinzugefügt.
- **Beim ersten Start erstellen**: Das Auswählen dieser Option hat zur Folge, dass der Treiber das Zielgerät für Tag-Informationen bewertet, wenn das Projekt zum ersten Mal ausgeführt wird. Es werden bei Bedarf auch sämtliche OPC-Tags dem Tag-Raum des Servers hinzugefügt.

Hinweis: Wenn die Option zum automatischen Generieren von OPC-Tags ausgewählt wird, müssen sämtliche Tags, die dem Tag-Raum des Servers hinzugefügt werden, mit dem Projekt gespeichert werden. Benutzer können das Projekt konfigurieren, um automatisch über das Menü **Tools** | **Optionen** zu speichern.

Bei doppeltem Tag: Wenn die automatische Tag-Datenbankgenerierung aktiviert wird, muss der Server wissen, wie mit Tags, die er möglicherweise zuvor hinzugefügt hat, oder mit Tags, die nach dem Kommunikationstreiber seit ihrer ursprünglichen Erstellung hinzugefügt oder geändert wurden, zu verfahren ist. Mit dieser Einstellung wird gesteuert, wie der Server OPC-Tags behandelt, die automatisch generiert wurden und derzeit im Projekt vorhanden sind. Es wird auch verhindert, dass sich automatisch generierte Tags auf dem Server ansammeln.

Beispiel: Wenn ein Benutzer die E/A-Module im Rack mit dem für **Immer beim Start erstellen** konfigurierten Server ändert, würden neue Tags jedes Mal dem Server hinzugefügt werden, wenn der Kommunikationstreiber ein neues E/A-Modul erkannt hat. Wenn die alten Tags nicht entfernt wurden, könnten sich viele unbenutzte Tags im Tag-Raum des Servers ansammeln. Die Optionen sind:

- **Bei Erstellen löschen**: Mit dieser Option werden sämtliche Tags gelöscht, die zuvor dem Tag-Raum hinzugefügt wurden, bevor sämtliche neuen Tags hinzugefügt werden. Dies ist die Standardeinstellung.
- **Nach Bedarf überschreiben**: Mit dieser Option wird der Server angewiesen, nur die Tags zu entfernen, die der Kommunikationstreiber durch neue Tags ersetzt. Sämtliche Tags, die nicht überschrieben werden, bleiben im Tag-Raum des Servers.
- **Nicht überschreiben**: Mit dieser Option wird verhindert, dass der Server sämtliche Tags entfernt, die zuvor generiert wurden oder bereits auf dem Server vorhanden waren. Der Kommunikationstreiber kann nur Tags hinzufügen, die völlig neu sind.
- **Nicht überschreiben, Fehler protokollieren**: Diese Option hat denselben Effekt wie die vorherige Option und sendet auch eine Fehlermeldung an das Ereignisprotokoll des Servers, wenn eine Tag-Überschreibung stattgefunden hätte.

Hinweis: Das Entfernen von OPC-Tags wirkt sich auf Tags, die automatisch vom Kommunikationstreiber generiert wurden, sowie auf sämtliche Tags aus, die unter Verwendung von Namen, die generierten Tags entsprechen, hinzugefügt wurden. Benutzer sollten es vermeiden, Tags dem Ser-

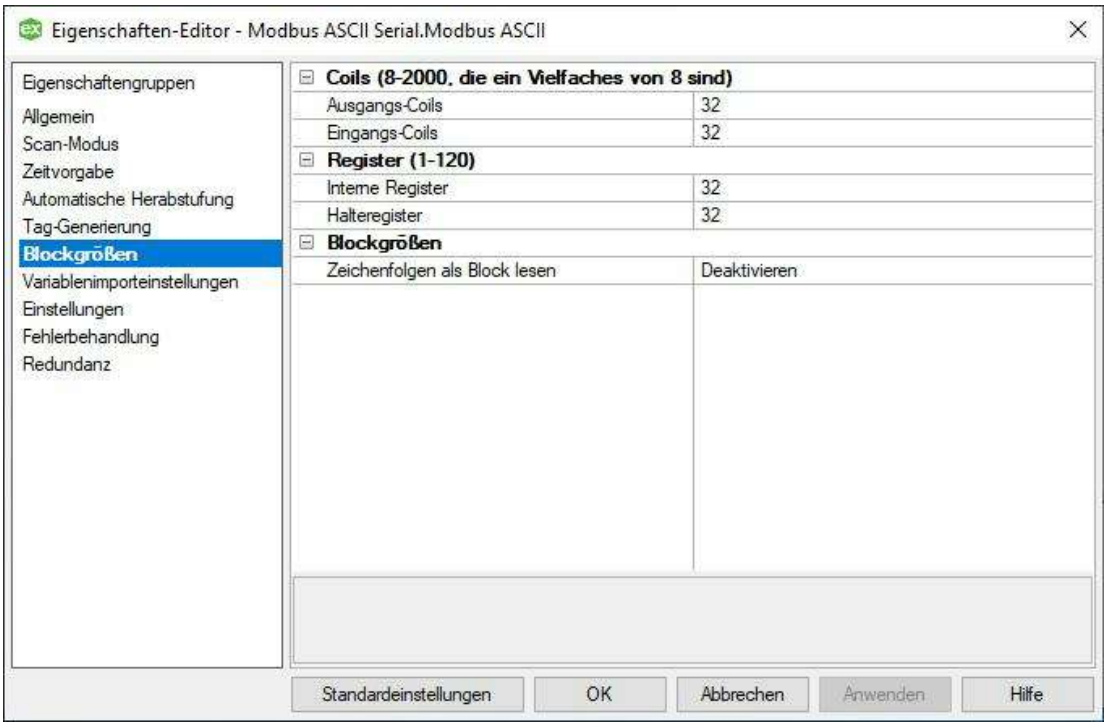
ver unter Verwendung von Namen hinzuzufügen, die möglicherweise den Tags entsprechen, die automatisch vom Treiber generiert werden.

Elternteilgruppe: Mit dieser Eigenschaft wird verhindert, dass sich automatisch generierte Tags mit Tags vermischen, die manuell eingegeben wurden, indem eine Gruppe festgelegt wurde, die für automatisch generierte Tags verwendet werden soll. Der Name der Gruppe kann bis zu 256 Zeichen lang sein. Diese Elternteilgruppe stellt einen Stammzweig bereit, dem alle automatisch generierten Tags hinzugefügt werden.

Automatisch generierte Untergruppen zulassen: Mit dieser Eigenschaft wird gesteuert, ob der Server automatisch Untergruppen für die automatisch generierten Tags erstellt. Dies ist die Standardeinstellung. Wenn diese Option deaktiviert ist, generiert der Server die Tags des Geräts in einer unstrukturierten Liste ohne jede Gruppierung. Im Serverprojekt werden die resultierenden Tags mit dem Adresswert benannt. Beispielsweise werden die Tag-Namen während des Generierungsprozesses nicht beibehalten.
Hinweis: Wenn beim Generieren von Tags durch den Server einem Tag derselbe Name wie einem bestehenden Tag zugewiesen wird, erhöht das System automatisch auf die nächste höchste Nummer, sodass der Tag-Name nicht dupliziert wird. Beispiel: Wenn der Generierungsprozess das Tag "AI22" erstellt, das bereits existiert, wird stattdessen das Tag als "AI23" erstellt.

Erstellen: Initiiert die Erstellung automatisch generierter OPC-Tags. Wenn die Konfiguration des Geräts geändert wurde, wird der Treiber durch die Option **Tags erstellen** gezwungen, das Gerät erneut auf mögliche Tag-Änderungen zu bewerten. Ihre Fähigkeit, über die System-Tags aufgerufen zu werden, ermöglicht einer Client-Anwendung das Initiieren der Tag-Datenbankerstellung.
Hinweis: **Tags erstellen** ist deaktiviert, wenn die Konfiguration ein Projekt offline bearbeitet.

Geräteeigenschaften – Blockgrößen



Coils

Ausgangs-Coils Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Punkte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, wenn Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts gelesen werden müssen.

Eingangs-Coils Coils können von 8 bis 2000 Punkten (Bit) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Punkte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, wenn Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts gelesen werden müssen.

Register

Interne Register Register können von 1 bis 100 Positionen (Wörtern) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Registerwerte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, wenn Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts gelesen werden müssen.

Halteregister Register können von 1 bis 100 Positionen (Wörtern) auf einmal gelesen werden. Eine höhere Blockgröße bedeutet, dass mehr Registerwerte vom Gerät bei einer einzelnen Anfrage gelesen werden. Die Blockgröße kann verringert werden, wenn Daten aus nicht zusammenhängenden Positionen innerhalb des Geräts gelesen werden müssen.

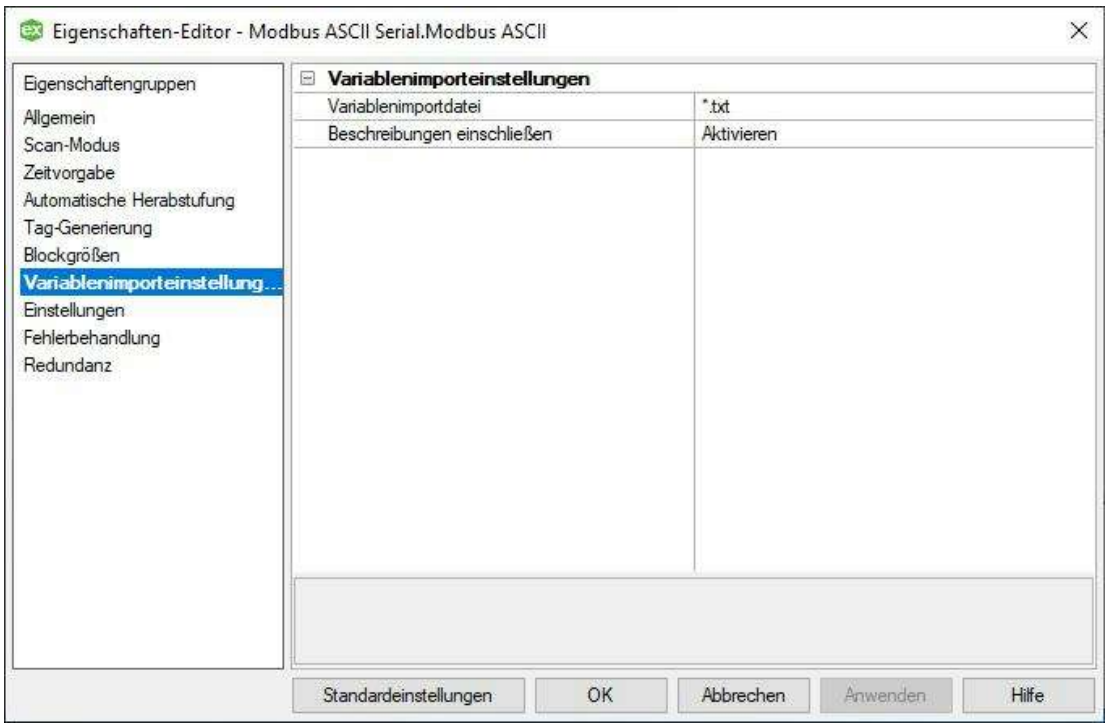
Achtung: Wenn die Blockgrößen auf einen Wert über 120 festgelegt werden und ein 32- oder 64-Bit-Datentyp für Tags verwendet wird, kann ein Fehler auftreten. Um das Auftreten von Fehlern zu verhindern, verringern Sie den Wert für die Blockgröße auf 120.

Blockgrößen

Zeichenfolgen als Block lesen: Aktivieren Sie diese Option, um Zeichenfolgen-Tags, die normalerweise einzeln gelesen werden, als Block zu lesen. Wenn diese Option aktiviert wird, werden Zeichenfolgen-Tags je nach ausgewählter Blockgröße gruppiert. Block-Lesevorgänge können nur für Zeichenfolgen-Tags des Modbus-Modells ausgeführt werden.

Geräteeigenschaften – Variablenimporteinstellungen

Weitere Informationen zu CSV-Dateien für Modbus-Treiber finden Sie unter **CSV-Dateien für Modbus-Treiber erstellen**.

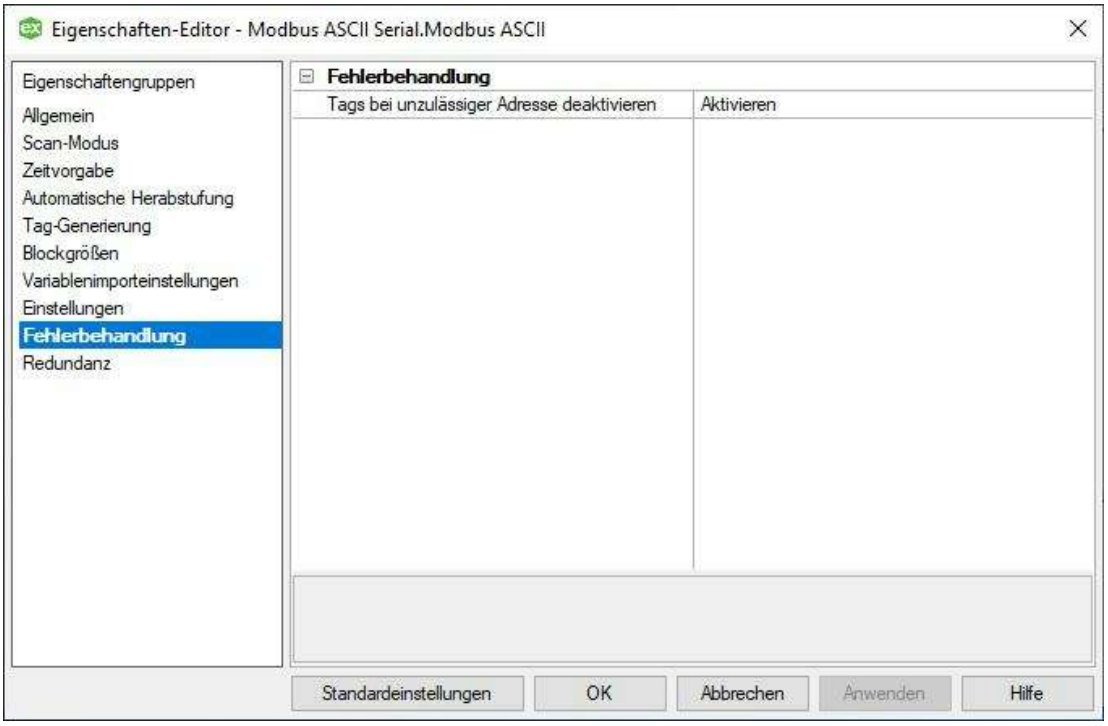


Variablenimportdatei: Gibt den genauen Speicherort und Namen der durch Semikola getrennten Textdatei an, die der Treiber für die automatische Tag-Datenbankgenerierung verwenden soll. Variablenimportdateien können aus vielen Anwendungen erstellt werden.

Beschreibungen einschließen: Bei Aktivierung werden Tag-Beschreibungen importiert, sofern in der Datei vorhanden. Die Standardeinstellung ist aktiviert.

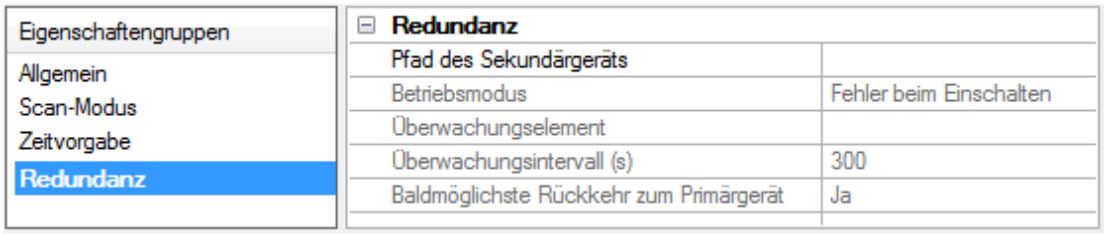
Weitere Informationen zum Konfigurieren der Funktion zur automatischen Tag-Datenbankgenerierung und zum Erstellen einer Variablenimportdatei finden Sie unter ***Automatische Tag-Datenbankgenerierung***.

Geräteeigenschaften – Fehlerbehandlung



Tags in unzulässigen Adressen deaktivieren: Wenn ein Gerät Modbus-Ausnahmencode 2 (unzulässige Adresse) oder 3 (unzulässige Daten, z.B. Anzahl von Punkten) als Antwort auf einen Lesevorgang dieses Blocks zurückgibt, kann der Treiber das Abrufen des Blocks mit den Fehlern anhalten oder weiterhin versuchen, den Blocks abzurufen. Wählen Sie "Aktivieren" aus, um das Abrufen bei einem Fehler aufgrund einer unzulässigen Adresse anzuhalten. Wählen Sie "Deaktivieren" aus, um das Abrufen dieses Datenblocks fortzusetzen. Ein Neustart des Servers ist zum Aktivieren eines deaktivierten Blocks nicht erforderlich. Die Standardeinstellung ist aktiviert.

Geräteeigenschaften – Redundanz



Redundanz steht mit dem Plugin für Redundanz auf Medienebene zur Verfügung.

*Weitere Informationen dazu erhalten Sie auf der Website, von einem Vertriebsrepräsentanten oder im **Benutzerhandbuch**.*

Automatische Tag-Datenbankgenerierung

Modbus ASCII Serial-Treiber nutzt die Funktion zur automatischen Tag-Datenbankgenerierung. Das ermöglicht es Treibern, automatisch Tags zu erstellen, die auf vom Kontaktplan des Geräts verwendete Datenpunkte zugreifen. Obwohl es manchmal möglich ist, ein Gerät nach Informationen abzufragen, die zur Erstellung einer Tag-Datenbank benötigt werden, muss dieser Treiber stattdessen eine Variablenimportdatei verwenden. Variablenimportdateien können mit Geräteprogrammieranwendungen wie Concept und ProWORX generiert werden.

Variablenimportdatei erstellen

Die Importdatei muss im durch Semikola getrennten TXT-Format vorliegen, dem Standard-Exportdateiformat der Geräteprogrammieranwendung Concept. Mit der Geräteprogrammieranwendung ProWORX können ebenfalls Variablendaten in diesem Format exportiert werden.

Spezifische Informationen zum Erstellen der Variablenimportdatei aus Concept und ProWORX finden Sie in der englischen Dokumentation "Technical Note 'Creating CSV Files for Modbus Drivers'" (Technical Note zum Erstellen von CSV-Dateien für Modbus-Treiber).

Serverkonfiguration

Die Funktion zur automatischen Tag-Datenbankgenerierung kann so angepasst werden, dass sie den Anforderungen der Anwendung entspricht. Diese Informationen können während des Schritts "Database Creation" des Gerätassistenten angegeben werden oder später durch Auswählen des Geräts und anschließendes Auswählen von **Eigenschaften** | **Tag-Generierung**.

Weitere Informationen dazu finden Sie in der Hilfedokumentation zum Server.

Dieser Treiber erfordert spezielle Eigenschaften zusätzlich zu den grundlegenden Einstellungen, die für alle Treiber gelten, die die automatische Tag-Datenbankgenerierung unterstützen. Zu diesen speziellen Eigenschaften gehören Name und Speicherort der Variablenimportdatei. Diese Informationen können während des Schritts "Variablenimporteinstellungen" des Gerätassistenten angegeben werden oder später durch Auswählen des Geräts und anschließendes Auswählen von **Eigenschaften** | **Variablenimporteinstellungen**.

Weitere Informationen finden Sie unter ***Variablenimporteinstellungen****.*

Operation

Je nach Konfiguration wird die Tag-Generierung automatisch gestartet, sobald das Serverprojekt gestartet wird, oder sie muss zu einem anderen Zeitpunkt manuell initiiert werden. Das Ereignisprotokoll zeigt an, wann der Tag-Generierungsprozess gestartet wurde, ob während der Verarbeitung der Variablenimportdatei Fehler aufgetreten sind und wann der Prozess abgeschlossen wurde.

Datentypbeschreibung

Datentyp	Beschreibung
Boolean	Einzelnes Bit
Word	16-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 15 ist das High-Bit
Short	16-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 14 ist das High-Bit Bit 15 ist das Vorzeichen-Bit
DWord	32-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 31 ist das High-Bit
Long	32-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 30 ist das High-Bit Bit 31 ist das Vorzeichen-Bit
BCD	Gepacktes 2-Byte-BCD Der Wertebereich liegt zwischen 0 und 9999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
LBCD	Gepacktes 4-Byte-BCD Der Wertebereich liegt zwischen 0 und 99999999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
String	Mit Null beendete ASCII-Zeichenfolge Wird im Modbus-Modell unterstützt, schließt eine Auswahl der Byte-Reihenfolgen Hi-Lo und Lo-Hi ein.
Double*	64-Bit-Gleitkommawert Der Treiber interpretiert vier aufeinanderfolgende Register als Wert mit doppelter Genauigkeit, indem die letzten zwei Register als High-DWord und die ersten zwei Register als Low-DWord bewertet werden.
Double-Beispiel	Wenn Register 40001 als Double-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 64-Bit-Datentyps und Bit 15 des Registers 40004 ist Bit 63 des 64-Bit-Datentyps.
Float*	32-Bit-Gleitkommawert Der Treiber interpretiert zwei aufeinanderfolgende Register als Wert mit einfacher Genauigkeit, indem das erste Register als Low-Wort und das zweite Register als High-Wort bewertet wird.
Float-Beispiel	Wenn Register 40001 als Float-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 32-Bit-Datentyps und Bit 15 des Registers 40002 ist Bit 31 des 32-Bit-Datentyps.

*Bei den obenstehenden Beschreibungen werden die Standardeinstellungen angenommen, d.h., dass für 64-Bit-Datentypen die Datenbehandlung "Erstes DWord 'Low'" verwendet wird und für 32-Bit-Datentypen die Datenbehandlung "Erstes Wort 'Low'".

Adressbeschreibungen

Adressspezifikationen sind je nach verwendetem Modell unterschiedlich. Wählen Sie einen Link von der folgenden Liste aus, um bestimmte Adressinformationen für das entsprechende Modell zu erhalten.

Modbus-ASCII-Adressierung
Mengenumwerter-Adressierung
Flussautomatisierungs-Adressierung

Modbus-ASCII-Adressierung

5-stellige Adressierung vs. 6-stellige Adressierung

In der Modbus-Adressierung gibt die erste Stelle der Adresse die primäre Tabelle an. Die verbleibenden Stellen stellen das Datenelement des Geräts dar. Der Höchstwert ist eine 2-Byte-Ganzzahl ohne Vorzeichen (65.535). Sechs Stellen sind erforderlich, um die gesamte Adresstabelle und das Element darzustellen. Deshalb werden Adressen, die im Handbuch des Geräts als 0xxxx, 1xxxx, 3xxxx oder 4xxxx angegeben sind, mit einer zusätzlichen Null aufgefüllt, sobald sie auf das Adressfeld eines Modbus-Tags angewendet werden.

Primäre Tabelle	Beschreibung
0	Ausgangs-Coils
1	Eingangs-Coils
3	Interne Register
4	Halteregister

Modbus-ASCII-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt.

*Hinweise und Einschränkungen finden Sie unter **Gepackte Coil-Tags**, **Zeichenfolgenunterstützung** und **Array-Unterstützung**.*

Adresse	Bereich	Datentyp	Zugriff	Funktionscodes*
Ausgangs-Coils	000001–065536 000001#1–065521#16	**Boolean** Wort (gepacktes Coil-Tag)	Lesen/Schreiben	01, 05, 15**
Eingangs-Coils	100001–165536 100001#1–165521#16	**Boolean** Wort (gepacktes Coil-Tag)	Schreibgeschützt	02**
Interne Register	300001–365536 300001–365535 300001–365533 3xxxxx.0/1–3xxxxx.15/16***	**Word**, Short, BCD Float, DWord, Long, LBCD Double Boolean	Schreibgeschützt	04
Interne Register	300001.2H–365536.240H	**String**	Schreibgeschützt	04

Adresse	Bereich	Datentyp	Zugriff	Funktionscodes*
als Zeichenfolge mit Byte-Reihenfolge Hi-Lo	.Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.			
Interne Register als Zeichenfolge mit Byte-Reihenfolge Lo-Hi	300001.2L–365536.240L .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Schreibgeschützt	04
Halteregister	400001–465536 400001–465535 400001–465533 4xxxxx.0/1–4xxxxx.15/16 ***	**Word**, Short, BCD Float, DWord, Long, LBCD Double Boolean	Lesen/Schreiben	03, 06, 16 03, 06, 16, 22
Halteregister als Zeichenfolge mit Byte-Reihenfolge Hi-Lo	400001.240H-465536.2H .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben	03,·16
Halteregister als Zeichenfolge mit Byte-Reihenfolge Lo-Hi	400001.2L–465536.240L .Bit ist die Zeichenfolgenlänge, Bereich 2 bis 240 Byte.	**String**	Lesen/Schreiben	03,·16

*Die unterstützten Funktionscodes werden als Dezimalzahlen angezeigt. *Weitere Informationen finden Sie unter* ***Beschreibung von Funktionscodes****.*
Weitere Informationen finden Sie unter* ***Gepackte Coil-Tags*.*
Weitere Informationen finden Sie im Unterthema "Nullbasierte Bit-Adressierung verwenden" unter ***Einstellungen****.*

Lesegeschützter Zugriff

Alle Lesen/Schreiben-Adressen können als lesegeschützt festgelegt werden, indem das Präfix "W" vor die Adresse gesetzt wird, z.B. "W40001". Dadurch wird verhindert, dass der Treiber das Register an der angegebenen Adresse liest. Versuche durch den Client, ein lesegeschütztes Tag zu lesen, führen dazu, dass der letzte erfolgreich in die angegebene Adresse geschriebene Wert abgerufen wird. Wenn keine erfolgreichen Schreibvorgänge stattgefunden haben, empfängt der Client 0 bzw. NULL für numerische bzw. Zeichenfolgenwerte als Anfangswert.

Achtung: Wenn die Client-Zugriffsrechte für lesegeschützte Tags auf "Lesegeschützt" festgelegt werden, führt dies dazu, dass Schreibvorgänge in diesen Tags fehlschlagen und der Client immer 0 bzw. NULL für numerische bzw. Zeichenfolgenwerte empfängt.

Gepackte Coil-Tags

Der gepackte Coil-Adresstyp ermöglicht den Zugriff auf mehrere aufeinanderfolgende Coils als analogen Wert. Diese Funktion ist nur für das Modbus-ASCII-Modell verfügbar. Der einzige gültige Datentyp ist Word. Die Syntax lautet folgendermaßen.

Ausgangs-Coils: 0xxxxx#nn Word Lesen/Schreiben
Eingangs-Coils: 1xxxxx#nn Word schreibgeschützt

Wobei xxxxx für die Adresse der ersten Coil steht und nn für die Anzahl der Coils, die in einen analogen Wert gepackt werden sollen (1–16).

Die Bit-Reihenfolge ist so, dass die Startadresse das niedrigstwertige Bit (LSB, Least Significant Bit) des analogen Werts ist.

Zeichenfolgenunterstützung

Das Modbus-Modell unterstützt das Lesen und Schreiben im Halteregisterspeicher als ASCII-Zeichenfolge. Bei Verwendung von Halteregistern für Zeichenfolgendaten enthält jedes Register zwei Byte ASCII-Daten. Die Reihenfolge der ASCII-Daten innerhalb eines gegebenen Registers kann beim Definieren der Zeichenfolge ausgewählt werden. Die Länge der Zeichenfolge kann zwischen 2 und 240 Byte liegen und wird statt einer Bit-Nummer eingegeben. Die Länge muss als gerade Zahl eingegeben werden. Geben Sie die Byte-Reihenfolge an, indem Sie entweder ein "H" oder ein "L" an die Adresse anhängen.

Weitere Informationen zum Ausführen eines Block-Lesevorgangs für Zeichenfolgen-Tags des Modbus-Modells finden Sie unter ***Blockgrößen****.*

Zeichenfolgenbeispiele

1. Um eine Zeichenfolge zu adressieren, die bei 40200 beginnt sowie eine Länge von 100 Byte und die Byte-Reihenfolge Hi-Lo aufweist, geben Sie "40200.100H" ein.
2. Um eine Zeichenfolge zu adressieren, die bei 40500 beginnt sowie eine Länge von 78 Byte und die Byte-Reihenfolge Lo-Hi aufweist, geben Sie "40500.78L" ein.

Hinweis: Die Zeichenfolgenlänge kann durch die maximale Größe der Schreibanforderung begrenzt werden, die das Gerät zulässt. Wenn bei Verwendung eines Zeichenfolgen-Tags die Fehlermeldung "In Adresse <Adresse> auf dem Gerät <Gerät> kann nicht geschrieben werden: Gerät hat mit Ausnahmecode 3 geantwortet." empfangen wird, wurde die Länge der Zeichenfolge vom Gerät abgelehnt. Kürzen Sie nach Möglichkeit die Zeichenfolge.

Normale Adressbeispiele

1. Die 255. Ausgangs-Coil wird bei dezimaler Adressierung als "0255" adressiert.
2. In der Dokumentation werden Modbus-Adressen manchmal mit Funktionscode und Position angegeben. Funktionscode 3, Position 2000 wird beispielsweise als "42000" adressiert. (Die vorangestellte "4" stellt Halteregister oder Funktionscode 3 dar.)
3. In der Dokumentation werden Modbus-Adressen manchmal mit Funktionscode und Position angegeben. Funktionscode 5, Position 100 wird beispielsweise als "0100" adressiert. (Die vorangestellte "0" stellt Ausgangs-Coils oder den Funktionscode 5 dar.) Durch Schreiben von 1 oder 0 an dieser Adresse wird die Coil eingestellt bzw. zurückgesetzt.

Array-Unterstützung

Arrays werden für interne und Halteregister-Positionen für alle Datentypen außer Boolean und String unterstützt. Arrays werden auch für Eingangs- und Ausgangs-Coils unterstützt (Boolean-Datentypen). Es gibt zwei Methoden, um ein Array zu adressieren. In den Beispielen werden Halteregister-Speicherorte verwendet.

4xxxx [Zeilen] [Spalten]

4xxxx [Spalten]: Bei dieser Methode wird angenommen, dass "Zeilen" gleich 1 ist.

Bei Arrays darf das Produkt aus "Zeilen" und "Spalten" nicht die Blockgröße überschreiten, die dem Gerät für den Register- bzw. Coil-Typ zugewiesen wurde. Bei Register-Arrays von 32-Bit-Datentypen darf das Produkt aus "Zeilen" und "Spalten", multipliziert mit 2, nicht die Blockgröße überschreiten.

Beschreibung von Funktionscodes

Dezimalwert	Hexadezimalwert	Beschreibung
01	0x01	Read Coil Status
02	0x02	Read Input Status
03	0x03	Read Holding Registers
04	0x04	Read Internal Registers
05	0x05	Force Single Coil
06	0x06	Preset Single Register
15	0x0F	Force Multiple Coils
16	0x10	Preset Multiple Registers
22	0x16	Masked Write Register

Mengenumwerter-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt.

Adresse	Bereich	Datentyp	Zugriff
Ausgangs-Coils	000001–065536	Boolean	Lesen/Schreiben
Eingangs-Coils	100001–165536	Boolean	Schreibgeschützt
Interne Register	300001–365536 300001–365535	**Word**, Short, BCD Float, DWord, Long, LBCD	Schreibgeschützt
Halteregister	400001–465536 400001–465535	**Word**, Short, BCD*, Float, DWord, Long, LBCD	Lesen/Schreiben
Mengenumwerter-Register	405000–406800 407000–407800	**Long**, DWord, LBCD **Float**, Long, DWord	Lesen/Schreiben

*****Die Adressbereiche 405000 bis 406800 und 407000 bis 407800 sind 32-Bit-Register. Adressen im Bereich 405000 bis 406800 verwenden den Standard-Datentyp Long. Adressen im Bereich 407000 bis 407800 verwenden den Standard-Datentyp Float. Da diese Adressregister 32-Bit-Register sind, sind nur die Datentypen Float, DWord, Long und LBCD zulässig. Arrays sind für diese speziellen Adressbereiche nicht zulässig.

Arrays

Arrays werden für interne und Halteregister-Speicherorte unterstützt, und zwar für alle Datentypen außer Boolean. Es gibt zwei Methoden, um ein Array zu adressieren. In den Beispielen werden Halteregister-Speicherorte verwendet.

4xxxx [Zeilen] [Spalten]

4xxxx [Spalten]: Bei dieser Methode wird angenommen, dass "Zeilen" gleich 1 ist.

Das Produkt aus "rows" und "cols" darf nicht die Blockgröße überschreiten, die dem Gerät für den Registertyp zugewiesen wurde. Bei Arrays von 32-Bit-Datentypen darf das Produkt aus "rows" und "cols", multipliziert mit 2, nicht die Blockgröße überschreiten.

Flussautomatisierungs-Adressierung

Die Standard-Datentypen für dynamisch definierte Tags werden **fett** dargestellt.

Adresse	Bereich	Datentyp	Zugriff
Mengenumwerter-Register	40001–465535	**Float**	Lesen/Schreiben

Der Flussautomatisierungs-Mengenumwerter behandelt alle Daten als 32-Bit-Gleitkommawert. Alle Adressen im Halteregisterbereich des Geräts werden als 32-Bit-Gleitkommazahlen gelesen. Eine vollständige Speicherzuordnung der Flussautomatisierungs-Steuerung finden Sie im englischen Flussautomatisierungs-Handbuch im Abschnitt zu benutzerdefinierten Berichten.

Ereignisprotokollmeldungen

Die folgenden Informationen betreffen Meldungen, die im Fensterbereich Ereignisprotokoll in der Hauptbenutzeroberfläche angezeigt werden. Informationen zum Filtern und Sortieren der Detailansicht Ereignisprotokoll finden Sie in der OPC-Serverhilfe. In der Serverhilfe sind viele allgemeine Meldungen enthalten, die also auch gesucht werden sollten. Im Allgemeinen werden die Art der Meldung (Information, Warnung) sowie Fehlerbehebungsinformationen bereitgestellt (sofern möglich).

Tipp: Meldungen, die aus einer Datenquelle stammen (z.B. Drittanbieter-Software, einschließlich Datenbanken), werden über das Ereignisprotokoll dargestellt. Die Schritte der Problembehandlung sollten eine Recherche zu diesen Meldungen im Internet und in der Händlerdokumentation beinhalten.

Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>.

Fehlertyp:

Fehler

Mögliche Ursache:

Es wurde versucht, einen nicht vorhandenen Speicherplatz auf dem angegebenen Gerät zu referenzieren.

Mögliche Lösung:

Überprüfen Sie die Tags, die Adressen im angegebenen Bereich auf dem Gerät zugewiesen sind, und entfernen Sie solche, die ungültige Speicherorte referenzieren.

Ungültiges Array. | Array-Bereich = <Start> bis <Ende>.

Fehlertyp:

Fehler

Mögliche Ursache:

Das definierte Array von Adressen reicht über das Ende des Adressraums hinaus.

Mögliche Lösung:

Überprüfen Sie die Größe des Gerätespeichers, und definieren Sie die Array-Länge entsprechend um.

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'.

Fehlertyp:

Fehler

Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. | Empfangene Länge = <Anzahl> (Byte), erwartete Länge = <Anzahl> (Byte).

Fehlertyp:

Warnung

Blockanfrage auf Gerät hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Kommunikation mit dem Gerät war erfolgreich, das Gerät hat jedoch ein Problem gemeldet.

Mögliche Lösung:

Weitere Informationen zum angegebenen Fehlercode erhalten Sie in der Dokumentation zum Gerät.

Siehe auch:

Modbus-Ausnahmecodes

In Adresse auf Gerät kann nicht geschrieben werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Kommunikation mit dem Gerät war erfolgreich, das Gerät hat jedoch ein Problem gemeldet.

Mögliche Lösung:

Weitere Informationen zum angegebenen Fehlercode erhalten Sie in der Dokumentation zum Gerät.

Siehe auch:

Modbus-Ausnahmecodes

Von Adresse auf Gerät kann nicht gelesen werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>.

Fehlertyp:

Warnung

Mögliche Ursache:

Kommunikation mit dem Gerät war erfolgreich, das Gerät hat jedoch ein Problem gemeldet.

Mögliche Lösung:

Weitere Informationen zum angegebenen Fehlercode erhalten Sie in der Dokumentation zum Gerät.

Siehe auch:

Modbus-Ausnahmecodes

Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen.

Fehlertyp:

Warnung

Mögliche Ursache:

Der Treiber konnte nicht den erforderlichen Speicherplatz für die Verarbeitung der Variablenimportdatei zuordnen.

Mögliche Lösung:

Fahren Sie nicht benötigte Anwendungen herunter, und versuchen Sie es erneut.

Beim Tag-Import ist eine Dateiausnahme aufgetreten.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Variablenimportdatei konnte nicht gelesen werden.

Mögliche Lösung:

Korrigieren Sie die Variablenimportdatei, oder regenerieren Sie sie.

Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Anzahl>.

Fehlertyp:

Warnung

Mögliche Ursache:

Das angegebene Feld in der Variablenimportdatei konnte nicht geparst werden, da es länger als erwartet oder ungültig ist.

Mögliche Lösung:

Bearbeiten Sie die Variablenimportdatei, um das Feld zu korrigieren.

Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Tag-Beschreibung im angegebenen Datensatz ist zu lang.

Mögliche Lösung:

Die Beschreibung wird entsprechend gekürzt. Damit dieser Fehler nicht mehr auftritt, bearbeiten Sie die Variablenimportdatei, um die Beschreibung zu kürzen.

Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der in der Variablenimportdatei gefundene Tag-Name enthält ungültige Zeichen.

Mögliche Lösung:

Ein gültiger Name wurde basierend auf der Variablenimportdatei erstellt. Um diesen Fehler zu vermeiden und die Konsistenz der Namen zu wahren, ändern Sie den Namen der exportierten Variablen.

Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Der in der Variablenimportdatei angegebene Datentyp gehört nicht zu den Typen, die von diesem Treiber unterstützt werden.

Mögliche Lösung:

Ändern Sie den in der Importdatei angegebenen Datentyp in einen unterstützten Typ. Wenn die Variable für eine Struktur verwendet wird, bearbeiten Sie die Datei, um jedes für die Struktur erforderliche Tag zu definieren, oder konfigurieren Sie die erforderlichen Tags im Server.

Siehe auch:

Variablen werden aus Concept exportiert

Tag-Datenbank wird importiert. | Quelldatei = '<Pfad>'.

Fehlertyp:

Informationen

Modbus-Ausnahmecodes

Folgende Daten stammen aus der englischen Dokumentation "Modbus Application Protocol Specifications" (Spezifikationen für das Modbus-Anwendungsprotokoll).

Code dezimal/hexadezimal	Name	Bedeutung
01/0x01	ILLEGAL FUNCTION	Der in der Abfrage erhaltene Funktionscode ist keine zulässige Aktion für den Server. Das kann daran liegen, dass der Funktionscode nur auf neuere Geräte anwendbar ist und in der ausgewählten Einheit nicht implementiert wurde. Es könnte auch anzeigen, dass der Server sich im falschen Status befindet, um eine Anfrage dieses Typs zu verarbeiten, z.B. weil er nicht konfiguriert ist und aufgefordert wird, Registerwerte zurückzugeben.
02/0x02	ILLEGAL DATA ADDRESS	Die in der Abfrage erhaltene Datenadresse ist keine zulässige Adresse für den Server. Insbesondere ist die Kombination aus Referenznummer und Übertragungslänge ungültig. Für einen Controller mit 100 Registern ist eine Anfrage mit Offset 96 und Länge 4 erfolgreich. Eine Anfrage mit Offset 96 und Länge 5 generiert Ausnahme 02.
03/0x03	ILLEGAL DATA VALUE	Ein im Abfrage-Datenfeld enthaltener Wert ist kein zulässiger Wert für den Server. Dies deutet darauf hin, dass ein Fehler in der Struktur des Rests einer komplexen Anfrage vorliegt, z.B. eine falsche implizierte Länge. Es bedeutet insbesondere nicht, dass ein Datenelement, das zur Speicherung in einem Register eingereicht wurde, einen Wert außerhalb der Erwartung des Anwendungsprogrammes hat, da das Modbus-Protokoll die Bedeutung bestimmter Werte für bestimmte Register nicht kennt.
04/0x04	SERVER DEVICE FAILURE	Ein nicht wiederherstellbarer Fehler ist aufgetreten, während der Server versucht hat, die angeforderte Aktion auszuführen.
05/0x05	ACKNOWLEDGE	Der Server hat die Anfrage akzeptiert und verarbeitet sie, aber dies wird viel Zeit in Anspruch nehmen. Diese Antwort wird zurückgegeben, um einen Timeout-Fehler im Client zu verhindern. Der Client kann als Nächstes eine Meldung ausgeben, dass das Abrufprogramm abgeschlossen ist, um zu ermitteln, ob Verarbeitung abgeschlossen ist.
06/0x06	SERVER DEVICE BUSY	Der Server ist mit der Verarbeitung eines lang dauernden Programmbefehls beschäftigt. Der Client muss die Nachricht später erneut senden, wenn der Server frei ist.
07/0x07	NEGATIVE ACKNOWLEDGE	Der Server kann die in der Abfrage erhaltene Programmfunktion nicht ausführen. Dieser Code wird für eine erfolglose Programmieranfrage mit Funktionscode 13 oder 14 (dezimal) zurückgegeben. Der

Code dezimal/hexadezimal	Name	Bedeutung
		Client muss Diagnose- oder Fehlerinformationen vom Server anfordern.
08/0x08	MEMORY PARITY ERROR	Der Server hat versucht, Erweiterungsspeicher zu lesen, aber dabei einen Paritätsfehler im Arbeitsspeicher gefunden. Der Client kann die Anfrage erneut versuchen, aber möglicherweise muss das Servergerät gewartet werden.
10/0x0A	GATEWAY PATH UNAVAILABLE	Die spezielle Verwendung in Verbindung mit Gateways deutet darauf hin, dass das Gateway keinen internen Kommunikationspfad vom Eingangskanal zum Ausgangskanal zuordnen konnte, um die Anfrage zu verarbeiten. Das bedeutet normalerweise, dass das Gateway falsch konfiguriert oder überlastet ist.
11/0x0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Die spezielle Verwendung in Verbindung mit Gateways deutet darauf hin, dass keine Antwort vom Zielgerät empfangen wurde. Das bedeutet normalerweise, dass das Gerät im Netzwerk nicht vorhanden ist.

Hinweis: Für diesen Treiber werden die Begriffe "Server" und "unangefordert" synonym verwendet.

Fehlermaskendefinitionen

B = Hardwareunterbrechung festgestellt
F = Framing-Fehler
E = E/A-Fehler
O = Zeichenpufferüberlauf
R = RX-Pufferüberlauf
P = Erhaltener Byte-Paritätsfehler
T = TX-Puffer voll

Index

5

5-stellige Adressierung 27

6

6-stellige Adressierung 27

A

Abfall 9
Abrufverzögerung 9
Adressbeschreibungen 27
Alle Werte für alle Tags schreiben 11
Allgemein 14
Als Block lesen 22
Anfangsaktualisierungen aus Cache 16
Anforderungs-Timeout 17
Anfragen verwerfen, wenn herabgestuft 18
Anstieg 9
Array-Unterstützung 29
Arrays 30
ASCII-Protokoll 5
Ausgangs-Coils 27, 30
Ausnahmecode 2 23
Ausnahmecode 3 23
Automatische Herabstufung 17
Automatische Tag-Datenbankgenerierung 24
Automatisches Wählen 10

B

Baudrate 6, 8
BCD 25
Bei doppeltem Tag 19
Bei Eigenschaftsänderung 19

Bei Gerätestart 19
Beim Tag-Import ist eine Dateiausnahme aufgetreten. 34
Beschreibung für Datensatz in Importdatei abgeschnitten. | Datensatznummer = <Nummer>. 34
Beschreibung von Funktionscodes 30
Beschreibungen einschließen 22
Betriebsverhalten 9
Blockanfrage auf Gerät hat mit Ausnahme geantwortet. | Blockbereich = <Adresse> bis <Adresse>, Ausnahme = <Code>. 33
Blockgrößen 21
Boolean 25

C

Coils 21
COM-ID 8
COM-Port 8
Concept 24

D

Daniels 5
Daten-Bits 6, 8
Datensammlung 15
Datentypbeschreibung 25
Deaktivieren 23
Dezimalformat 30
Diagnose 7
Double 25
DTR 9
Durch Null ersetzen 12
Durch Tag angegebenes Scan-Intervall berücksichtigen 16
DWord 25

E

E/A-Fehler 37
Eingangs-Coils 27, 30
Einstellungen auf Kanalebene 13

Elliot 5

Elternteilgruppe 20

Empfangene Blocklänge stimmt nicht mit erwarteter Länge überein. | Empfangene Länge = <Anzahl> (Byte), erwartete Länge = <Anzahl> (Byte). 32

Ereignisprotokollmeldungen 32

Erstellen 20

Ethernet-Einstellungen 9

Ethernet-Kapselung 5, 8

F

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Fehler>'. 32

Fehler beim Parsen von Datensatz in Importdatei. | Datensatznummer = <Nummer>, Feld = <Anzahl>. 34

Fehlerbehandlung 22

Fehlermaskendefinitionen 37

Float 25

Flussautomatisierungs-Adressierung 31

Flusssteuerung 6, 9

Force Multiple Coils 30

Force Single Coil 30

Framing 37

G

Generieren 19

Gepackte Coil-Tags 28

Geräteeigenschaften – Automatische Herabstufung 17

Geräteeigenschaften – Redundanz 23

Geräteeigenschaften – Tag-Generierung 18

Geräteeigenschaften – Zeitvorgabe 16

Geteilt 8

Globale Einstellungen 13

H

Halteregister 27, 30

Hardwareunterbrechung 37

Herabstufen bei Fehler 17

Herabstufungszeitraum 18
Hexadezimalwert 30

I

ID 14
ID-Format 14
Identifikation 6
Import 22
Importierter Tag-Name ist ungültig und wurde geändert. | Tag-Name = '<Tag>', geänderter Tag-Name = '<Tag>'. 34
In Adresse auf Gerät kann nicht geschrieben werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 33
Inaktive Verbindung schließen 9-10
Inaktivitätsdauer bis Schließen 9-10
Interne Register 27, 30

K

Kanaleigenschaften – Allgemein 6
Kanaleigenschaften – Erweitert 12
Kanaleigenschaften – Kommunikationsserialisierung 12
Kanaleigenschaften – Schreiboptimierungen 11
Kanaleigenschaften – Serielle Kommunikation 7
Kanalzuweisung 14
Keine 8
Kommunikations-Timeouts 16
Kommunikationsfehler melden 9-10

L

Lastausgleich 13
LBCD 25
Lesegeschützter Zugriff 28
Leseverarbeitung 10
Long 25
Löschen 19

M

Masked Write Register 30
Mengenumwerter-Adressierung 30
Mengenumwerter-Register 30
Modbus-ASCII-Adressierung 27
Modbus-Ausnahmecodes 36
Modell 14
Modem 8, 10
Modemeinstellungen 10

N

Netzwerk 5
Netzwerk 1 – Netzwerk 500 13
Netzwerkadapter 10
Netzwerksmodus 13
Nicht geändert 12
Nicht normalisierte Float-Handhabung 12
Nicht scannen, nur Abruf anfordern 16
Nur den letzten Wert für alle Tags schreiben 11
Nur den letzten Wert für nicht boolesche Tags schreiben 11

O

Omni 5
OPC-Client 5
Operation ohne Kommunikation 10
Optimierungsmethode 11

P

Parität 6, 8, 37
Physisches Medium 8
Preset Multiple Registers 30
Preset Single Register 30
Priorität 13
Protokoll 5

ProWORX 24
ProWORX-Geräteprogrammieranwendung 24

R

Read Coil Status 30
Read Holding Registers 30
Read Input Status 30
Read Internal Registers 30
Redundanz 23
Register 21
RS-485 9
RS232 6
RS485 6
RTS 9
RX-Pufferüberlauf 37

S

Scan-Modus 15
Serialisierung 6
Serielle Kommunikation 7
Serielle Port-Einstellungen 8
Servicezyklus 11
Setup 5
Short 25
Simuliert 15
Stopp-Bits 6, 9
String 25

T

Tag-Datenbank wird importiert. | Quelldatei = '<Pfad>'. 35
Tag-Generierung 18
Tag-Import wegen zu wenig Speicherressourcen fehlgeschlagen. 33
Tag-Zähler 7
Tag konnte nicht importiert werden, da der Datentyp nicht unterstützt wird. | Tag-Name = '<Tag>', nicht unterstützter Datentyp = '<Typ>'. 35

Timeout bis zum Herabstufen 18
Transaktionen pro Zyklus 13
Treiber 14
TX-Puffer voll 37

U

Überlauf 37
Überschreiben 19
Übersicht 5
Ungültige Adresse im Block. | Blockbereich = <Adresse> bis <Adresse>. 32
Ungültiges Array. | Array-Bereich = <Start> bis <Ende>. 32
Untergruppen zulassen 20
Unterstützte Geräte 5
Unzulässige Adresse 23

V

Variablenimportdatei 22, 24
Variablenimporteinstellungen 22, 24
Verbindungs-Timeout 10, 16
Verbindungstyp 8
Versuche vor Timeout 17
Verzögerung zwischen Geräten 12
Virtuelles Netzwerk 13
Von Adresse auf Gerät kann nicht gelesen werden. Gerät hat mit Ausnahme geantwortet. | Adresse = '<Adresse>', Ausnahme = <Code>. 33

W

Word 25

Z

Zeichenfolgenunterstützung 29
Zeitvorgabe 16