CODESYS-Treiber

© 2025 PTC Inc. Alle Rechte vorbehalten.

Inhaltsverzeichnis

CODESYS-Treiber 1
Inhaltsverzeichnis 2
CODESYS-Treiber 6
Übersicht 6
Setup 6
Kanaleigenschaften – Allgemein 7
Tag-Zähler 8
Kanaleigenschaften – Schreiboptimierungen 9
Kanaleigenschaften – Erweitert 10
Kanaleigenschaften – Kommunikationsparameter 10
Geräteerkennungsvorgang 11
Geräteerkennungs-Einstellungen 12
Geräteeigenschaften – Allgemein 13
Betriebsmodus 14
Tag-Zähler 15
Geräteeigenschaften – Scan-Modus 15
Geräteeigenschaften – Automatische Herabstufung 16
Geräteeigenschaften – Tag-Generierung 17
Automatische Tag-Datenbankgenerierung 19
Geräteeigenschaften – Kommunikationsparameter 21
Geräteeigenschaften – Benutzeranmeldeinformationen 23
Geräteeigenschaften – Tag-Importeinstellungen 24
Geräteeigenschaften – Redundanz 26
Leistungsoptimierung 28
Datentypbeschreibung 29
Symbolische auf Tags basierende Adressierung 30
Tag-Umfang 30
Tag-Adressierung 30
Verschachtelte Strukturen 31
Arrays 32
Array-Elemente 32
Bit-Adressierung 32
Zugriffsberechtigungen 33
Adressieren unteilbarer Datentypen 33
Ereignisprotokollmeldungen 35

Tag-Generierung fehlgeschlagen, da die Gerätesymbole nicht geladen werden konnten. | Gerät = '<Kanalname.Gerätename>'. 35

Tag-Generierung fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte. | Gerät = '<Kanalname.Gerätename>'. 35

Tag-Generierung aufgrund eines unerwarteten Fehlers fehlgeschlagen. | Gerät = '<Kanalname.Gerätename>'. 36

Interner Fehler. Ein unerwarteter Fehler ist aufgetreten. Die SPS-Verbindung wird zurückgesetzt. | Transaktionsinformationen = '<Transaktions-Typ und Details>'. 36

Die Geräteermittlung ist aufgrund eines unerwarteten Fehlers fehlgeschlagen. 36

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Es kann keine Verbindung zum Gerät hergestellt werden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 37

Fehler beim Durchsuchen von Tags. 37

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein Fehler aufgetreten. Symbolliste konnte nicht vom Gerät oder aus der Datei abgerufen werden. 37

Beim Versuch, das Tag zu lesen, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, das Tag zu schreiben, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Es wurde ein nicht unterstützter Datentyp oder eine ungültige Adresse angegeben. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Datentyp = '<type>'. 38

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 39

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'. 39

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'. 39

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein interner Fehler aufgetreten. Die angegebene Konfiguration ist nicht gültig. 40

Der Durchsuchen-Pfad enthält ungültige Zeichen. 40

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'. 40

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'. 40

Durchsuchen nach Tags fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte. 41

Beim Versuch das Tag zu lesen ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag- 41

Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. .....

Beim Versuch das Tag zu schreiben ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 41

Der Durchsuchungspfad ist nicht vorhanden. 42

Durchsuchen nach Tags wurde aufgrund eines unbekannten Fehlers abgebrochen. 42

Die Durchsuchen-Anfrage wurde abgebrochen, weil der Treiber beendet wurde. 42

Datentyp für die angegebene Adresse wird nicht unterstützt. Für diesen Datenpunkt wird kein Tag generiert. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 43

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen. 43

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 43

Der aus dem Zeichenfolgen-Tag gelesene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 44

Der in das Zeichenfolgen-Tag geschriebene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 44

Geräte über Kanäle hinweg müssen eindeutige Adressen (IP-Adresse oder Host-Name) sowie Port-Kombinationen haben. | Adresse = = '<Nummer>', Port = '<Nummer>', Überlappendes Gerät = '<Gerät>'. 44

Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>'. 44

Der Dateipfad darf nicht leer sein. 44

Die Symboldatei war ungültig oder beschädigt. | Datei = '<Pfad zur Datei>'. 45

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Betriebssystem-Meldung>'. 45

Fehler beim Versuch, das Tag zu schreiben. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'. 45

Fehler beim Versuch, das Tag zu lesen. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'. 45

Ungültige Adresse. Gültige logische Adresse oder SPS-Namen angeben. 45

Geräte über Kanäle hinweg müssen eindeutige logische Adressen/SPS-Namen haben. | Adresse = '<Adresse>', Überlappendes Gerät = '<Gerät>'. 45

Einige der importierten Tags haben Tags mit demselben Namen in Ihrem Projekt ersetzt. 45

Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>', Grund = '<Fehlermeldung vom Betriebssystem>'. 46

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen. 46

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 46

Fehler beim Durchsuchen von Tags. | Grund = '<Zeichenfolge mit Erklärung für das Fehlschlagen 46

des Durchsuchens von Tags>'. .....
Tags generiert. | Tag-Anzahl = <Anzahl>. 47
Index **48**

CODESYS-Treiber

Hilfe-Version 1.054

INHALT

Übersicht
Was ist CODESYS-Treiber?

Kanal-Setup
Wie konfiguriere ich Kanäle für die Verwendung mit diesem Treiber?

Geräte-Setup
Wie konfiguriere ich ein bestimmtes Gerät für die Zusammenarbeit mit diesem Treiber?

Kommunikation optimieren
Wie erziele ich die beste Leistung mit CODESYS-Treiber?

Datentypbeschreibung
Welche Datentypen unterstützt dieser Treiber?

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

Übersicht

CODESYS-Treiber bietet eine zuverlässige Möglichkeit, CODESYS-konforme Controller mit Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP und zahlreichen benutzerdefinierten Anwendungen, zu verbinden.

Setup

Kommunikationsprotokoll

CODESYS V2.3 Ethernet vs. ARTI
CODESYS V3 Ethernet vs. ARTI
CODESYS Gateway für V2.3
CODESYS Gateway für V3
CODESYS Gateway für V3.5.18.0

Kanal- und Gerätegrenzwerte

Die von diesem Treiber unterstützte maximale Anzahl von Kanälen liegt bei 1024. Die maximale Anzahl von Geräten, die von diesem Treiber unterstützt werden, liegt bei 256 pro Kanal.

Tag-Datenbank-Erstellung

Die Funktion "Automatische Tag-Datenbankgenerierung" dieses Treibers wurde entworfen, um den Zeitaufwand bei der Einrichtung der OPC-Anwendung zu reduzieren. Dieser Treiber kann so konfiguriert werden, dass automatisch eine Liste von Server-Tags auf dem Server generiert wird, die gerätespezifischen Daten entsprechen. Die automatisch generierten OPC-Tags können dann vom OPC-Client aus durchsucht werden.

Geräte-Timeouts

Der Treiber unterstützt keine konfigurierbaren Zeitvorgabe-Parameter. Diese Parameter werden immer auf die folgenden Werte festgelegt:

- **Verbindungs-Timeout**: 20 Sekunden
- **Anforderungs-Timeout**: 10000 Millisekunden
- **Erneute Versuche**: 3 Versuche

Hinweis: Dieser Treiber ermöglicht nicht die Konfiguration des Anforderungs-Timeouts. Dadurch kann das Aktualisieren der Tag-Qualität durch ein Gerät viel Zeit in Anspruch nehmen, wenn die Konnektivität verloren gegangen ist.

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.

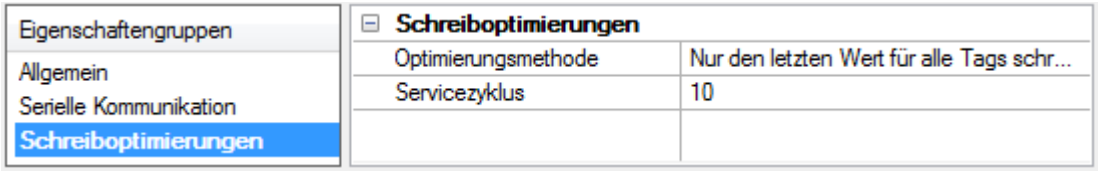
Weitere Informationen dazu finden Sie in der Serverhilfe unter "Kommunikationsdiagnosen".

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 – 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:

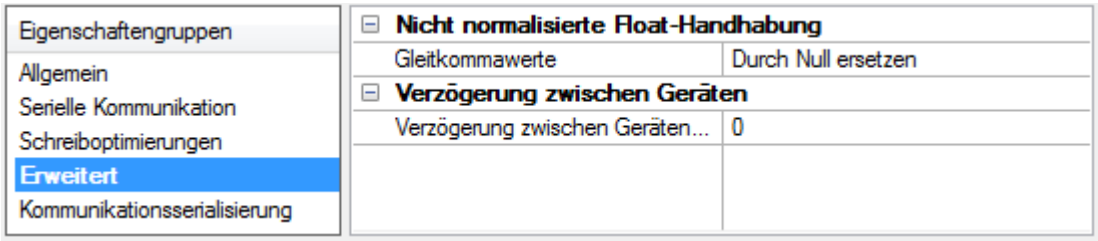
- **Alle Werte für alle Tags schreiben**: Mit dieser Option wird der Server gezwungen, für jeden Wert einen Schreibvorgang auf dem Controller zu versuchen. In diesem Modus sammelt der Server weiterhin Schreibanforderungen und fügt sie der internen Schreibwarteschlange des Servers hinzu. Der Server verarbeitet 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.

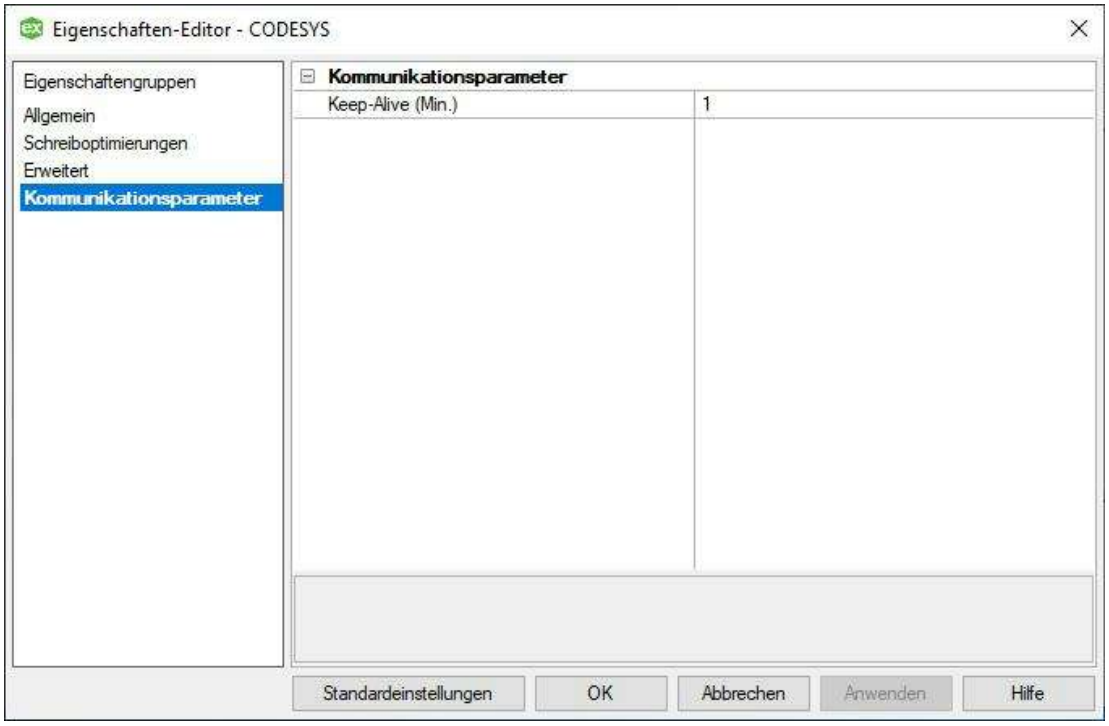
Weitere Informationen über die Gleitkommawerte finden Sie unter "Mit nicht normalisierten Gleitkommawerten arbeiten" in der Serverhilfe.

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 – Kommunikationsparameter

Die Gruppe "Kommunikationsparameter" ist eine Sammlung von Eigenschaften, durch die Kommunikationsparameter konfiguriert werden, die für alle Geräte innerhalb des Kanals gelten.

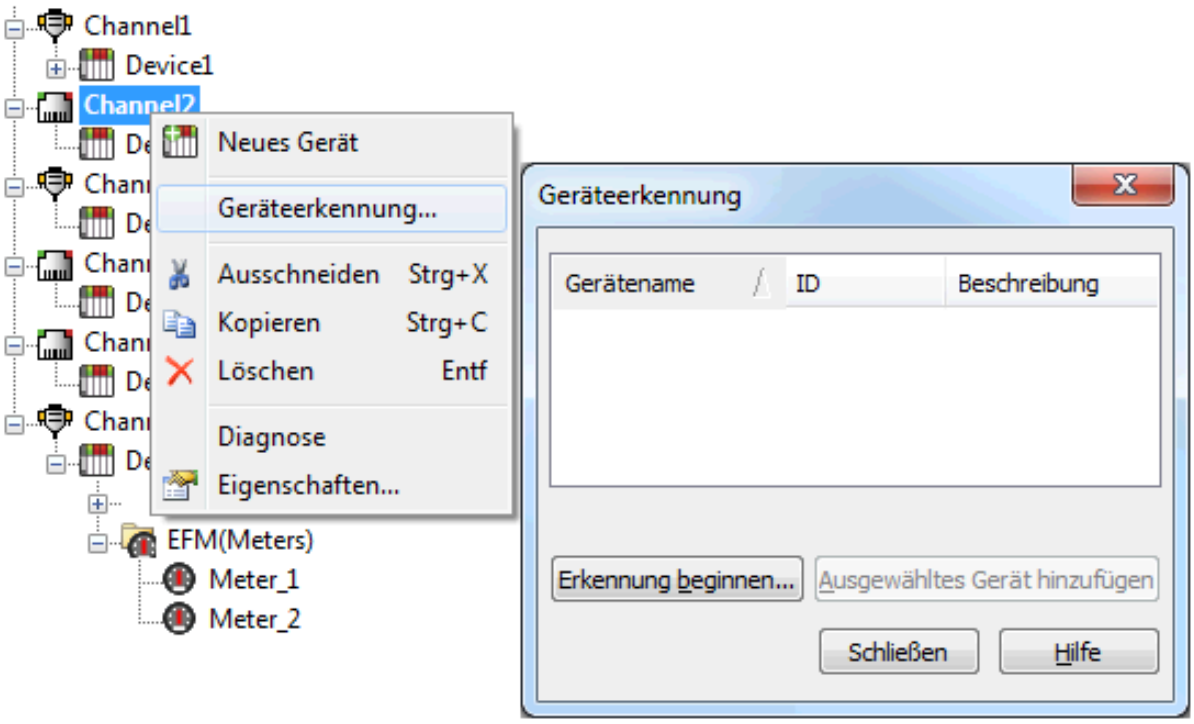


Kommunikationsparameter

Keep-Alive (Min.): Konfigurieren Sie wie viele Minuten der Treiber eine Verbindung mit dem Gerät aufrecht erhält, nachdem alle Client-Referenzen entfernt wurden. Wenn vor Ablauf des Timeouts keine Client-Referenzen hinzugefügt werden, wird die Verbindung geschlossen. Wenn eine Client-Referenz hinzugefügt wird, während der Zeitgeber noch aktiv ist, wird das Timeout abgebrochen, damit die Verbindung nicht geschlossen wird.

Geräteerkennungsvorgang

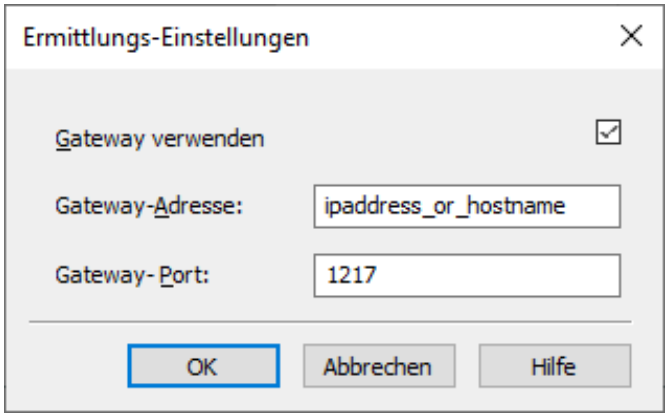
Die Geräteerkennung steht für Treiber zur Verfügung, die das Lokalisieren von Geräten im Netzwerk unterstützen. Sobald Geräte gefunden werden, können sie ggf. einem Kanal hinzugefügt werden. Die maximale Anzahl von Geräten, die auf einmal erkannt werden können, liegt bei 65535.



1. Wählen Sie den Kanal aus, in dem Geräte erkannt und hinzugefügt werden sollten.
2. Klicken Sie mit der rechten Maustaste auf den Kanalknoten, und wählen Sie **Geräteerkennung...** aus.
3. Geben Sie die Erkennungseigenschaften an, die treiberspezifisch sind, wie Adressbereich, Timeout, Erkennungsumfang.
4. Klicken Sie auf **OK**.
5. Erkannte Geräte füllen das Dialogfenster mit den folgenden Informationen/Überschriften: **Gerätename**, **ID**, **Beschreibung**.
6. Sollte ein erkanntes Gerät von Interesse sein, wählen Sie die gewünschten Geräte aus, und klicken Sie auf **Ausgewählte Geräte hinzufügen**.
7. Klicken Sie auf **Schließen**.

Geräteerkennungs-Einstellungen

Um Geräte zu erkennen, ist es erforderlich, mit dem Netzwerk, auf dem sich die Geräte befinden, zu kommunizieren.



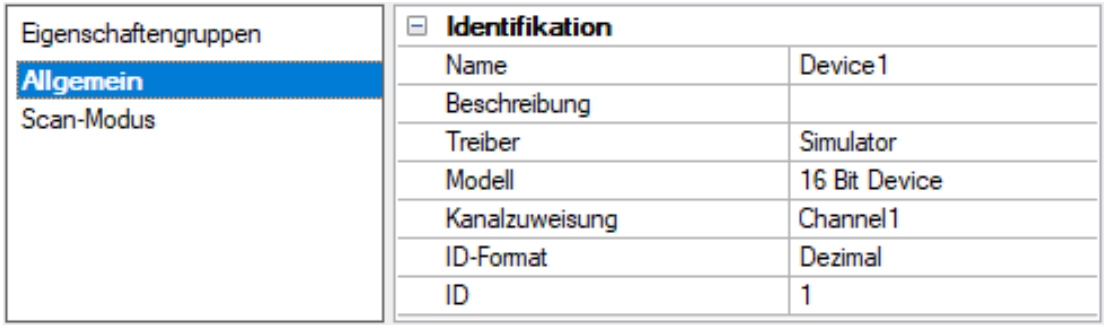
Gateway verwenden: Aktivieren Sie diese Option, wenn die Erkennung über einen Netzwerk-Gateway erfolgen kann.

Gateway-Adresse: Geben Sie die IP-Adresse oder den Hostnamen für den Gateway-Server an.

Gateway-Port: Geben Sie die Port-Nummer, über die Kontakt mit dem Gateway-Server hergestellt werden soll, ein.

Geräteeigenschaften – Allgemein

Ein Gerät stellt ein einzelnes Ziel in einem Kommunikationskanal dar. Wenn der Treiber mehrere Controller unterstützt, müssen Benutzer eine Geräte-ID für jeden Controller eingeben.



Identifikation

Name: Geben Sie den Namen des Geräts an. Es ist ein logischer, benutzerdefinierter Name, der bis zu 256 Zeichen lang sein und auf mehreren Kanälen verwendet werden kann.

Hinweis: Zwar sind beschreibende Namen allgemein eine gute Idee, doch haben einige OPC-Client-Anwendungen beim Durchsuchen des Tag-Raums des OPC-Servers möglicherweise ein eingeschränktes Anzeigefenster. Der Geräte- und Kanalname werden ebenfalls Teil der Informationen zum Durchsuchen der Hierarchiebaumstruktur. Innerhalb eines OPC-Clients würde die Kombination aus Kanalname und Gerätename als "ChannelName.DeviceName" angezeigt werden.
Weitere Informationen dazu finden Sie in der Serverhilfe unter "So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig".

Beschreibung: Geben Sie die benutzerdefinierten Informationen über dieses Gerät an.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

Kanalzuweisung: Geben Sie den benutzerdefinierten Namen des Kanals an, zu dem dieses Gerät derzeit gehört.

Treiber: Ausgewählter Protokolltreiber für dieses Gerät.

Modell: Geben Sie den Gerätetyp an, der dieser ID zugeordnet ist. Der Inhalt des Dropdown-Menüs hängt vom Typ des verwendeten Kommunikationstreibers ab. Modelle, die von einem Treiber nicht unterstützt werden, sind deaktiviert. Wenn der Kommunikationstreiber mehrere Gerätemodelle unterstützt, kann die Modellauswahl nur geändert werden, wenn keine Client-Anwendungen mit dem Gerät verbunden sind.
Hinweis: Wenn der Kommunikationstreiber mehrere Modelle unterstützt, sollten Benutzer versuchen, die Modellauswahl mit dem physischen Gerät abzugleichen. Wenn das Gerät im Dropdown-Menü nicht dargestellt wird, wählen Sie ein Modell aus, das dem Zielgerät am ehesten entspricht. Einige Treiber unterstützen die Modellauswahl "Offen", wodurch Benutzer kommunizieren können, ohne bestimmte Details des Zielgeräts zu kennen. *Weitere Informationen dazu finden Sie in der Dokumentation für den Treiber.*

ID: Geben Sie die treiberspezifische Station oder den treiberspezifischen Knoten für das Gerät an. Der Typ der eingegebenen ID hängt vom verwendeten Kommunikationstreiber ab. Für viele Kommunikationstreiber ist die ID ein numerischer Wert. Treiber, die eine numerische ID unterstützen, stellen Benutzern die Option zum Eingeben eines numerischen Werts bereit, dessen Format den Anforderungen der Anwendung oder der Charakteristik des ausgewählten Kommunikationstreibers entsprechend angepasst werden kann. Das Format wird standardmäßig durch den Treiber festgelegt. Zu den Optionen gehören "Dezimal", "Oktal" und "Hexadezimal".
Hinweis: Wenn der Treiber Ethernet-basiert ist oder eine unkonventionelle Station oder einen unkonventionellen Knotennamen unterstützt, kann die TCP/IP-Adresse des Geräts ggf. als Geräte-ID verwendet werden. TCP/IP-Adressen bestehen aus vier Werten, die durch Punkte getrennt sind, wobei jeder Wert im Bereich von 0 bis 255 liegt. Einige Geräte-IDs sind zeichenfolgenbasiert. Abhängig vom Treiber gibt es möglicherweise zusätzliche zu konfigurierende Eigenschaften innerhalb des ID-Felds.

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: Versetzen Sie das Gerät in den Simulationsmodus, oder beenden Sie den Modus. 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

Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

Hinweise:

1. Aktualisierungen werden erst nach dem Trennen von Clients und nach dem Wiederherstellen von deren Verbindung angewendet.
2. Das 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.
3. 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.
4. Wird ein Gerät simuliert, so werden Aktualisierungen möglicherweise nicht schneller als innerhalb einer Sekunde auf dem Client angezeigt.

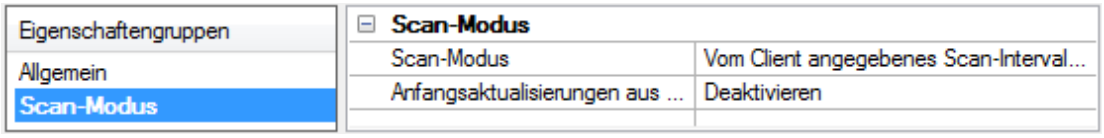
Der Simulationsmodus ist nur für Test- und Simulationszwecke. Es sollte niemals in einer Produktionsumgebung nie verwendet werden.

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.

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

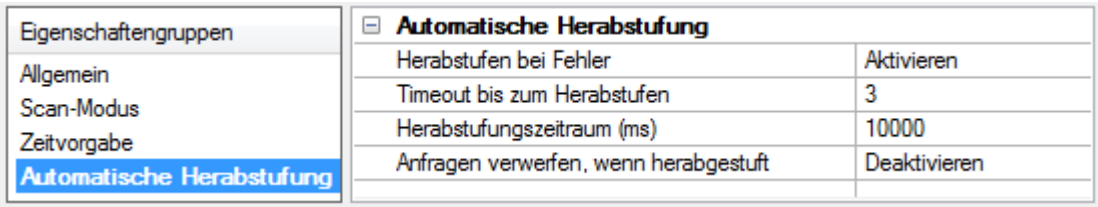
das Scan-Intervall verringert wird, werden die Änderungen erst wirksam, wenn alle Client-Anwendungen getrennt wurden.

- **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 – 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

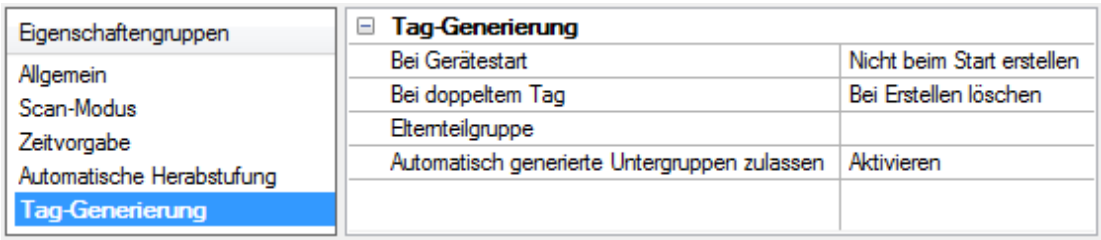
Mithilfe der Funktionen zur automatischen Tag-Datenbankgenerierung wird die Einrichtung einer Anwendung zu einem Plug-and-Play-Vorgang. Ausgewählte Kommunikationstreiber können so konfiguriert werden, dass automatisch eine Liste von Tags erstellt wird, die gerätespezifischen Daten entsprechen. Diese 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 Server 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.

Automatische Tag-Datenbankgenerierung

- Wenn Tags automatisch generiert werden, werden Sie mit Client-Zugriffsrechten erstellt, die den von der SPS bereitgestellten Zugriffsrechten entsprechen, soweit dies möglich ist.
- Tags, die nur Leseschutz haben, werden mit den Client-Zugriffsrechten für Lesen/Schreiben erstellt, da es keine Client-Zugriff-Einstellung für "Nur schreiben" auf dem Server gibt.
- Tags ohne Lese- oder Schreibberechtigungen werden nicht generiert, da kein entsprechendes Zugriffsrecht auf dem Server vorhanden ist.
- Gemäß dem CODESYS V2.3-Protokoll müssen Adressen in einer SPS-Programm-Variablenliste oder einer Liste mit globalen Variablen definiert werden. Bei der ersten Gruppe handelt es sich immer um den Speicherort der Variablen (Global oder der Name des SPS-Programms, in dem das Tag definiert ist).
- Arrays werden immer während der Tag-Generierung erweitert, d.h. für jedes Element im Array wird ein Tag erstellt. Für das Array selbst wird kein Tag generiert.
- Ist **Automatisch generierte Untergruppen zulassen** aktiviert, werden für jedes Segment einer Tag-Adresse Untergruppen erstellt. Die maximale Tiefe beträgt acht Gruppen. Der Tag-Name ist der Teil der Adresse, der nach dem Erstellen von Untergruppen übrig bleibt.
- Werden Array-Elemente generiert, vorausgesetzt der Grenzwert für die verschachtelte Gruppe wurde nicht erreicht, werden die Tags in einer Gruppe platziert, die nach dem Array, dem Sie angehören, benannt ist.

CODESYS V2.3-Beispiele

Tag-Adresse	Gruppen	Tag-Name
.myTag	Global	.myTag
PLC_PRG.myTag	PLC_PRG	.myTag
.myStruct.myArray[0]	Global.myStruct.myArray[x]	myArray[0]
PLC_PRG.myStruct.myArray[0]	PLC_PRG.myStruct.myArray[x]	myArray[0]
.group1...group7.myTag	Global.group1...group7	Group7_myTag

CODESYS V3-Beispiele

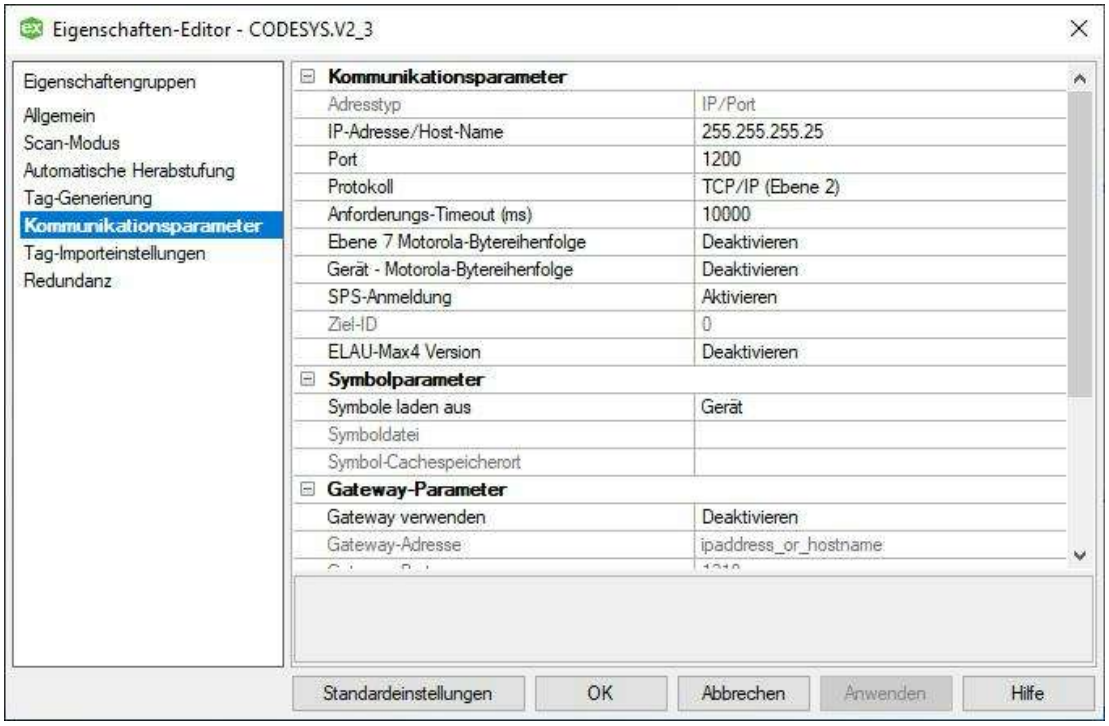
Tag-Adresse	Gruppen	Tag-Name
Application.GVL.myTag	Application.GVL	.myTag
Application.PLC_PRG.myTag	Application.PLC_PRG	.myTag
Application.GVL.myStruct.myArray[0]	Application.GVL.myStruct.myArray [x]	myArray[0]
Application.PLC_PRG.myStruct.myArray [0]	Application.PLC_PRG.-myStruct.myArray[x]	myArray[0]
Application.GVL.group1...group6.myTag	Application.GVL.group1...group6	Group6_myTag

Tipp: Um die Lesbarkeit zu verbessern, wurden Gruppen mithilfe von Punkten verkettet (ähnlich wie ein Benutzer über einen OPC DA/UA Client auf das Tag zugreifen würde).

Siehe auch: *Tag-Importeinstellungen*

Geräteeigenschaften – Kommunikationsparameter

Kommunikationsparameter sind Eigenschaften, die zum Herstellen einer Verbindung mit dem Gerät verwendet werden.



Kommunikationsparameter

Adresstyp: Geben Sie den Adresstyp für die Verbindung mit dem Gerät (nur V3-Modell) an.

Adresse: Geben Sie die logische Adresse oder den SPS-Namen für das Gerät (nur V3-Modell) an.

IP-Adresse/Host-Name: Geben Sie die IP-Adresse oder den Hostnamen für das Zielgerät an. Für V3-Geräte sind nur IP-Adressen gültig.

Port: Geben Sie die Ethernet/IP-Nummer auf dem Zielgerät an.

Protokoll: Geben Sie das in diesem Gerät verwendete Protokoll an.

Anforderungs-Timeout: Geben Sie an, wie lange der Treiber abschließend auf eine Antwort vom Zielgerät wartet. Der gültige Bereich liegt zwischen 4.000 und 9.999.999 Millisekunden (167,6667 Minuten). Die Standardeinstellung ist 10.000 Millisekunden. Gilt nur für V2.3 Modell.

Ebene 7 Motorola-Bytereihenfolge: Geben Sie an, ob Ebene 7 die Motorola-Byte-Reihenfolge (Big Endian) verwendet.

Gerät - Motorola-Bytereihenfolge: Geben Sie an, ob das Zielgerät die Motorola-Byte-Reihenfolge (Big Endian) verwendet. In den meisten Fällen wird der Wert auf den gleichen Wert wie die Ebene 7 Motorola-Bytereihenfolge festgelegt.

SPS-Anmeldung: Geben Sie an, ob der Treiber nach Herstellen der Verbindungen bei der SPS angemeldet bleiben soll. Diese Einstellung sollte deaktiviert werden, wenn die SPS nur eine einzelne Client-Verbindung unterstützt. In allen anderen Fällen sollte Sie aktiviert werden.

Ziel-ID: Geben Sie die ID an, wenn es sich um eine Unter-SPS handelt. SPS, welche die eine Weiterleitung von Kommunikationen durch eine andere SPS erfordern, werden Unter-SPS genannt. Die Ziel-ID enthält zusätzliche Adressinformationen, die für die Kommunikation mit Unter-SPS erforderlich sind. Erfolgt keine Kommunikation mit einer Unter-SPS, sollte diese Einstellung 0 sein.

ELAU-Max4 Version: Geben Sie die Hardware-Revision der Ziel-SPS an. Dieser Wert sollte deaktiviert werden, außer es handelt sich bei dem Gerät um ein ELAU Max 4 1100 oder 1200.

Die Kombination aus IP-Adresse und Port muss für alle Geräte eindeutig sein, auch wenn diese unterschiedliche Kanäle nutzen. Die meisten CODESYS-SPS unterstützen keine parallelen Kommunikationskanäle. Daher bringt das Zuweisen mehrerer Kanäle unter derselben CODESYS-SPS keine Leistungsverbesserung. Durch Kopieren und Einfügen eines Kanals auf dem Server wird derzeit ein Gerät mit derselben Kombination aus IP-Adresse und Port erstellt. Alle Geräte im Kanal, die kopiert wurden, müssen daher geändert werden, um Überlappungen von Kombinationen aus IP-Adresse und Port zu beheben. Wenn diese Kollisionen nicht behoben werden, kann dies zu nicht definiertem Verhalten führen.

Symbolparameter (nur V2.3)

Symbole laden aus: Geben Sie an, wo der Treiber nach Gerätesymbolen suchen soll.

- **Datei**: Lädt die Symboldatei wie in der Eigenschaft "Symboldatei" angegeben. Wenn diese Option ausgewählt ist, muss die Eigenschaft "Symboldatei" den vollständigen Dateinamen enthalten, einschließlich den Pfad zur Symboldatei sowie deren Erweiterung (.sdb).
- **Gerät**: Lädt die Symbole bei jedem Verbindungsaufbau oder erneutem Verbindungsversuch vom Gerät. Dies ist die Standardoption.
- **Gerät und Cache**: Lädt die Symbole aus dem lokalen Cacheverzeichnis, wie durch die Eigenschaft Symbol-Cachespeicherort angegeben. Wenn diese Option ausgewählt ist, muss die Eigenschaft Symbol-Cachespeicherort den vollständigen Verzeichnispfad zum lokalen Cachespeicherort angeben.

Symboldatei: Diese Eigenschaft ist erforderlich, wenn für die Eigenschaft Symbole laden aus die Fileoption ausgewählt ist. Geben Sie den zu verwendenden vollständigen Dateinamen einschließlich des Pfads an, wenn die Symboldatei nicht auf dem Gerät gespeichert werden kann. Diese Symboldatei muss den auf dem Gerät gespeicherten Symbolen entsprechen. Wenn die Symbole nicht übereinstimmen, schlägt die Kommunikation mit dem Gerät fehl.

Symbol-Cachespeicherort: Diese Eigenschaft ist erforderlich, wenn für die Eigenschaft Symbole laden aus die Option Gerät und Cache ausgewählt ist. Geben Sie das lokale Verzeichnis an, in dem die Gerätesymbole zwischengespeichert werden. Für die Verwendung mit Geräten, die Symbole speichern. Ist die Symboldateigröße groß oder das Netzwerk weist eine schlechte Qualität und/oder Bandbreite auf, sollten Sie diese Eigenschaft verwenden, um den Aufwand bei einem Symbolkopiervorgang über das Netzwerk bei jedem Verbindungs- oder erneuten Verbindungsversuch zu vermeiden. Der Verzeichnispfad, falls angegeben, muss mit einem umgekehrten Schrägstrich (\) enden.

Tipps zur Leistung:

- Aktivieren Sie Gerät und Cache, um eine potenzielle Netzwerküberlastung aufgrund von Symbolen, die über das Netzwerk kopiert werden, zu vermeiden.

- Bei Verwendung der Einstellung Gerät und Cache:
  - Der Symbol-Cachespeicherort sollte für jedes Gerät eindeutig sein.
  - Durch Kopieren der Symboldateien an den gewünschten Ort während der Projektkonfiguration wird das Kopieren von Symbolen bei der ersten Verbindungsherstellung vermieden.

Gateway-Parameter

Gateway verwenden: Geben Sie an, ob ein Gateway für die Verbindung mit dem Gerät verwendet werden soll.

Gateway-Adresse: Geben Sie die IP-Adresse oder den Hostnamen für den Gateway an.

Gateway-Port: Geben Sie den Port für den Gateway an.

Gateway-Passwort: Geben Sie das Passwort für den Gateway an (nur V2.3-Modell).

Geräteverbindung über ein Gateway erfordert, dass das CODESYS-Gateway auf dem gleichen Host installiert ist wie der Server. Wenn das CODESYS-Gateway nicht auf dem gleichen Server installiert ist wie der Host, führt das Lesen eines Tags zu einer Meldung des Typs "Gerät antwortet nicht" im Ereignisprotokoll. Dies trifft nur auf Konfigurationen zu, die das V2.3-Ethernet-Modell enthalten.

Geräteverbindung über ein Gateway erfordert, dass die anfänglichen Kommunikationen serialisiert werden. Sind Geräte unter unterschiedlichen Kanälen konfiguriert, werden die Verbindungen nacheinander hergestellt. Nach der anfänglichen Verbindung, werden die Kommunikationen nicht länger serialisiert. Dies trifft nur auf Konfigurationen zu, die das V2.3-Ethernet-Modell enthalten.

Leistung

Tags pro Anforderung: Die maximale Anzahl von Tags, die in eine einzelne Anforderung eingeschlossen werden dürfen. Jede einzelne Anforderung ist mit einem gewissen Dienstaufwand verbunden. Im Allgemeinen ist es besser, mehrere Elemente in einer Anforderung zusammenzufassen statt jedes Element separat anzufordern. Eine hohe Anzahl von Elementen in einer Anforderung führt jedoch dazu, dass die Round-Trip-Zeit (Antwortzeit) länger wird. Je nach Anwendung kann es besser sein, lange Listen in mehrere kleine Anforderungen aufzuteilen.

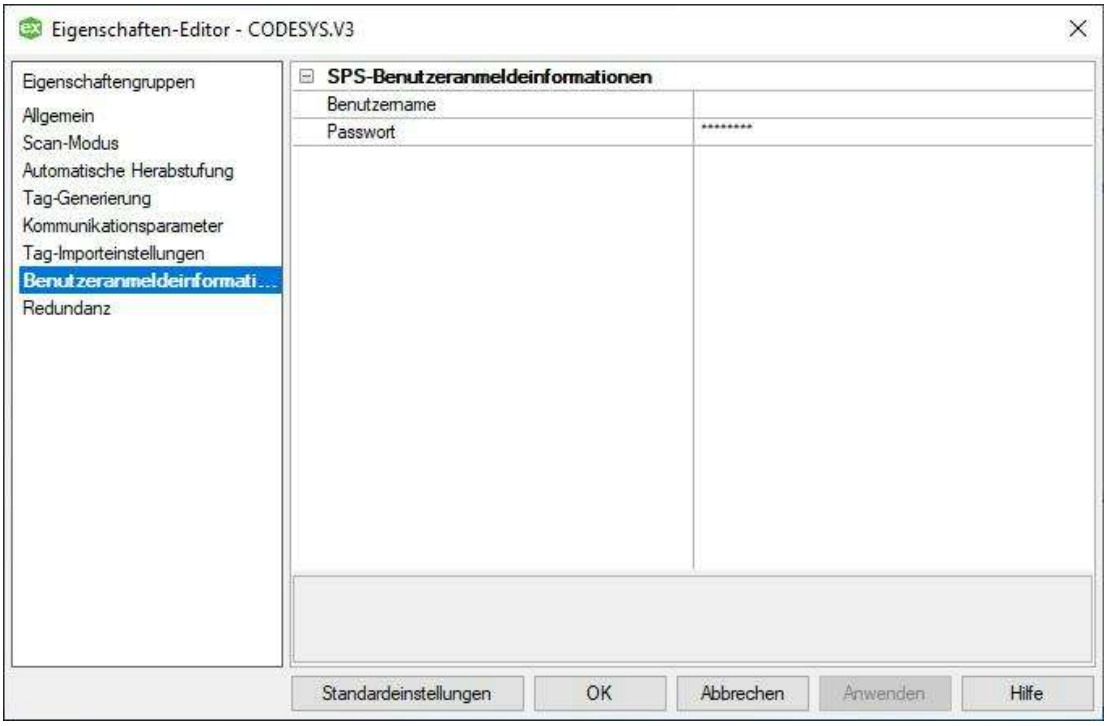
Hinweis: Die Anzahl der Schreibvorgänge pro Anforderung ist auf Grundlage des in den Kanaleinstellungen angegebenen **Servicezyklus** begrenzt.

Zeichenfolgen

Erweiterten ASCII-Zeichensatz zulassen: Geben Sie an, ob Zeichenfolgen mit erweiterten ASCII-Zeichen aktiviert oder deaktiviert werden sollen. Die Standardeinstellung ist deaktiviert (nur V2.3 Modell).

Geräteeigenschaften – Benutzeranmeldeinformationen

Bei CODESYS V3-Geräten kann optional eine Authentifizierung verlangt werden. Wenn eine Authentifizierung erforderlich ist, müssen zum Herstellen einer Verbindung Benutzername und Passwort mit den korrekten Einstellungen konfiguriert werden. Diese Einstellungen sind für V2.3-Geräte nicht verfügbar.

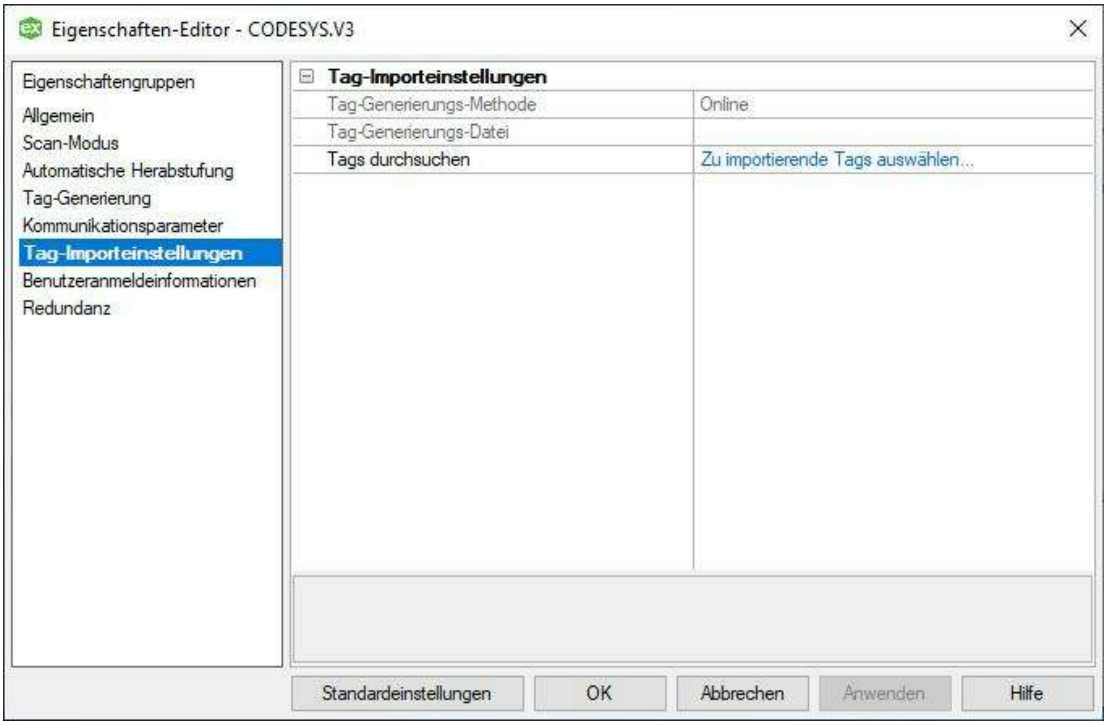


Benutzername: Geben Sie den Namen des zur Authentifizierung verwendeten Kontos an. Für Benutzernamen wird eine Länge von maximal 254 Zeichen unterstützt.

Passwort: Geben Sie das zum angegebenen Benutzernamen gehörige Passwort an. Für Passwörter wird eine Länge von maximal 251 Zeichen unterstützt.

Geräteeigenschaften – Tag-Importeinstellungen

Tag-Importeinstellungen sind für die Tag-Generierung verwendete Eigenschaften.



Tag-Importeinstellungen

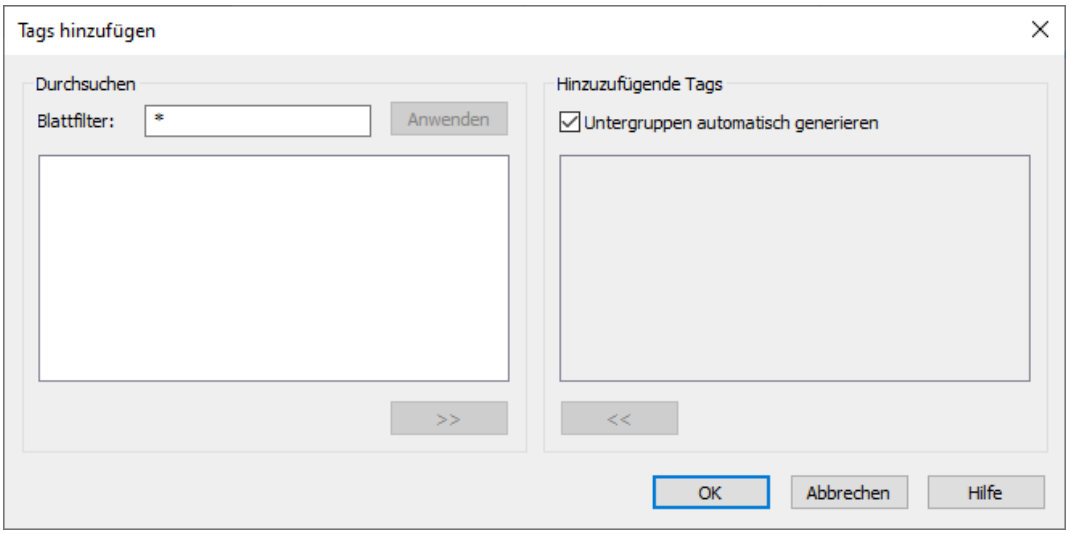
Tag-Generierungs-Methode: Tags können auf zwei unterschiedliche Weisen importiert werden. Wählen Sie **Online** aus, um Tags von einem Gerät in einem Netzwerk zu importieren, das für den Treiber zur Zeit des Imports sichtbar ist. Wählen Sie **Offline** aus, um Tags ohne eine Geräteverbindung zu importieren, indem Sie dem Server die entsprechenden Klartext-Symboldatei (.SYM) angeben, die beim Kompilieren des CODESYS-Projekts erstellt wird.

Tag-Generierungs-Datei: Geben Sie den Pfad und Dateinamen der zu importierenden Symboldatei an. Um die .SYM-Datei zu finden und auszuwählen, klicken Sie auf die Schaltfläche **Durchsuchen** (**...**). Diese Eigenschaft ist deaktiviert, wenn Online als Importmethode gewählt wird.
Tipp: Starten Sie nach Auswahl der Datei die Tag-Generierung mit dem Befehl Tags erstellen in der Gruppe **Tag-Generierung**.

Nach Tags durchsuchen: Klicken Sie auf **Zu importierende Tags auswählen...**, um das Dialogfenster Tag hinzufügen aufzurufen und zu Tags zu navigieren und diese auszuwählen.

Tag-Browser

Der Tag-Browser wird verwendet, um zu Tags auf dem aktuellen Gerät zu navigieren und diese auszuwählen und anschließend zu importieren.
Neben dem grafischen Tag-Browser steht auch ein Tag-Browser-Dienst zur Verfügung, der es Benutzern erlaubt, mithilfe des Konfigurations-API-Dienst ein Gerät nach Tags zu durchsuchen.
Weitere Informationen zur Verwendung des Konfigurations-API-Dienstes finden Sie in der Hilfe für den Server.



Durchsuchen: Gibt einen Hierarchiebaum an, der die Hierarchie des Geräte-Tags darstellt, einschließlich aller Tags und Tag-Gruppen auf dem Gerät. Es können mehrere Tags ausgewählt werden, es darf jedoch nur jeweils eine Tag-Gruppe ausgewählt werden.

Hinzuzufügende Tags: Eine Liste alle unter diesem Gerät in dem Projekt hinzuzufügenden Tags.

Blattfilter: Beschränkt die in der Struktur "Browsing" angezeigten Elemente nur auf diejenigen, die dem Filter entsprechen. Klicken Sie auf "Anwenden", um die Tag-Liste mit aktiviertem Filter zu aktualisieren.
Weitere Informationen zur Syntax finden Sie in der nachfolgenden Tabelle.

Untergruppen automatisch generieren: Steuert, ob der Server automatisch Gruppen für die durchsuchten Tags generiert.
Siehe auch: "Automatisch generierte Untergruppen zulassen" unter Tag-Generierung für weitere Informationen.

>>: Alle im Hierachiebaum ausgewählten Tags werden zur Liste **Hinzuzufügende Tags** hinzugefügt. Wird eine Tag-Gruppe ausgewählt, werden alle Tags unter diesem Zweig hinzugefügt.
Hinweis: Je nach Größe und Komplexität des Projekts kann das Hinzufügen einer Tag-Gruppe bis zu einer Stunde dauern. Ein Beispiel ist das Hinzufügen einer Gruppe, die viele verschachtelte Untergruppen enthält.

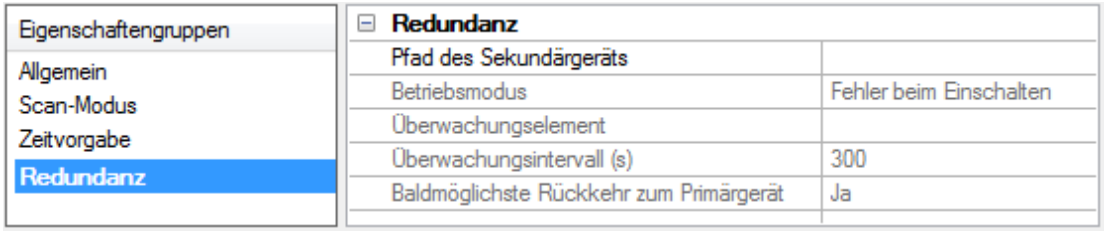
<<: Entfernt alle ausgewählten Tags aus der Liste **Hinzuzufügende Tags**.
Hinweis: Der Benutzer muss berechtigt sein, Tags hinzuzufügen, Tags zu löschen und Tag-Gruppen hinzuzufügen, um den Tag-Browser zu verwenden.

Filter-Syntax

Der Filter erlaubt es dem Browser die angezeigten Elemente auf solche zu beschränken, die einem Muster entsprechen. Der Filter unterstützt nur ASCII-Zeichen. Die unterstützte Syntax wird in der nachfolgenden Tabelle beschrieben.

Zeichen im Muster	Übereinstimmungen in Zeichenfolge
?	Jedes einzelne Zeichen Beispiel: ? stimmt mit "a", "!", "3", etc. überein. Es stimmt nicht mit "aa" oder "ok" überein, da diese Zeichenfolgen 2 Zeichen anstatt eines enthalten.
*	Null oder mehr Zeichen Beispiel: *server stimmt mit "server", "cloud server" und "OPC server" überein, nicht jedoch mit "OPC Server" oder "erver".
#	Jede einzelne Ziffer (0–9) Beispiel: # stimmt mit "0", "1" und "2" überein, nicht jedoch mit "a" oder "z".
[charlist]	Jedes einzelne Zeichen in einer Liste. Beispiel: [a-c] stimmt mit "a", "b" oder "c" überein, jedoch mit keinen anderen Zeichen, auch nicht mit "A", "B" und "C".
[!charlist]	Jedes einzelne Zeichen, das nicht in der Liste enthalten ist. Beispiel: [!a-c] stimmt mit "d", "e", "A" und "B" überein, nicht jedoch mit "a", "b" oder "c".

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.

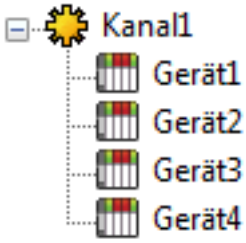
Leistungsoptimierung

Kommunikation optimieren

Bei jedem programmierbaren Controller gibt es Möglichkeiten, den Systemdurchsatz zu verbessern, und bei CODESYS-Treiber verhält es sich nicht anders. CODESYS-Treiber wurde zur Optimierung von Lese- und Schreibvorgängen entwickelt. Für Tags aller Datentypen werden Anforderungen in einer einzigen Transaktion zusammengefasst. Dies sorgt für eine drastische Verbesserung der Leistung gegenüber Einzel-Tag-Transaktionen. Die einzige Einschränkung ist die Anzahl an Tags, die in eine einzige Transaktion passen.

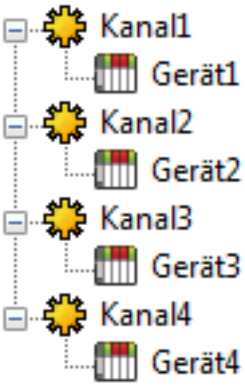
Anwendung optimieren

Zwar ist der Treiber schnell ist, doch gibt es eine Reihe von Richtlinien zur Optimierung der Anwendung und zum Erreichen der maximalen Leistung. Kommunikationsprotokolle wie CODESYS werden als Kanäle bezeichnet. Jeder in der Anwendung definierte Kanal stellt einen separaten Ausführungspfad im Server dar. Sobald ein Kanal festgelegt wurde, kann eine Reihe von Geräten unter diesem Kanal definiert werden. Jedes dieser Geräte stellt einen einzelnen CODESYS-Controller dar, von dem Daten gesammelt werden können. Zwar ermöglicht diese Methode zum Definieren der Anwendung ein hohes Leistungsniveau, doch kann damit nicht vollständig Nutzen aus dem Treiber oder Netzwerk gezogen werden. Ein Beispiel dafür, wie die Anwendung bei Konfiguration mit einem Kanal möglicherweise aussehen kann, wird im Folgenden gezeigt.



In diesem Beispiel wird jedes Gerät unter einem einzelnen Kanal angezeigt. In dieser Konfiguration muss sich der Treiber schnellstmöglich von einem Gerät zum nächsten bewegen, um Informationen in einem effektiven Intervall zu sammeln. Je mehr Geräte hinzugefügt oder je mehr Informationen von einem einzelnen Gerät angefordert werden, desto mehr leidet die Aktualisierungsrate insgesamt.

Wenn CODESYS-Treiber nur einen einzelnen Kanal festlegen könnte, wäre das Beispiel oben die einzige Option. Der Treiber kann bis zu 1024 Kanäle definieren. Durch Verwenden mehrerer Kanäle wird die Arbeitsbelastung bei der Datensammlung verteilt, indem mehrere Anfragen gleichzeitig an das Netzwerk gestellt werden. Ein Beispiel dafür, wie dieselbe Anwendung aussehen kann, wenn sie mit mehreren Kanälen konfiguriert wird, um die Leistung zu verbessern, wird im Folgenden gezeigt.



Jedes Gerät wird unter seinem eigenen Kanal definiert. In dieser Konfiguration wird ein einziger Ausführungspfad dediziert für das Sammeln von Daten von jedem Gerät eingesetzt. Wenn die Anwendung über 1024 oder weniger Geräte verfügt, kann sie wie hier gezeigt exakt optimiert werden.

Die Leistung wird verbessert, selbst wenn die Anwendung über mehr Geräte verfügt. Zwar sind weniger Geräte möglicherweise ideal, doch zieht die Anwendung weiterhin Nutzen aus zusätzlichen Kanälen. Das Verteilen der Gerätelast auf alle Kanäle hat zur Folge, dass sich der Server erneut von Gerät zu Gerät bewegt, und zwar mit weit weniger zu verarbeitenden Geräten auf einem einzigen Kanal.

Datentypbeschreibung

Datentyp	Beschreibung
Boolean	Einzelnes Bit
Byte	8-Bit-Wert ohne Vorzeichen
Char	8-Bit-Wert mit Vorzeichen
Word	16-Bit-Wert ohne Vorzeichen
Short	16-Bit-Wert mit Vorzeichen
DWord	32-Bit-Wert ohne Vorzeichen
Long	32-Bit-Wert mit Vorzeichen
QWord	64-Bit-Wert ohne Vorzeichen
Long Long	64-Bit-Wert mit Vorzeichen
Float	32-Bit-IEEE-Gleitkommawert
Double	64-Bit-IEEE-Gleitkommawert
Datum	64-bit IEEE-Datums- und Zeitwert
String	Mit Null beendetes Zeichen-Array

*Eine Beschreibung plattformspezifischer CODESYS-Daten finden Sie unter **CODESYS-Datentypen**.*

Nicht unterstützte Datentypen

Zu den nicht unterstützten Datentypen gehören LBCD und BCD.

***Siehe auch**: Adressbeschreibungen, Unteilbare Datentypen*

Symbolische auf Tags basierende Adressierung

CODESYS-Treiber verwendet eine symbolische Adressierungsstruktur für Tags. Diese Tags (im Allgemeinen als native Tags bezeichnet) unterscheiden sich von herkömmlichen SPS-Datenelementen dahingehend, dass der Tag-Name selbst die Adresse ist und keine physische oder logische Adresse.

Tag-Adressregeln für Client/Server

CODESYS-Variablennamen entsprechen Tag-Adressen für Client/Server. CODESYS-Variablennamen (eingegeben über CODESYS SPS) folgen den ID-Regeln für IEC 61131-3. Tag-Adressen für Client/Server folgen denselben Regeln und sind nachfolgende aufgelistet:

- Müssen mit einem Buchstaben (A-Z, a-z) oder Unterstrich (_) beginnen.
- Dürfen nur Buchstaben und Unterstriche enthalten.
- Dabei wird die Groß-/Kleinschreibung nicht beachtet.

Tag-Namensregeln für Client/Server

Die Regeln für die Zuweisung des Tag-Namens im Server unterscheidet sich von der Adresszuweisung dahingehend, dass Tag-Namen nicht mit einem Unterstrich beginnen dürfen.

Tag-Umfang

Globale Tags

Bei globalen Tags handelt es sich um CODESYS-Variablen, die über globalen Umfang im Controller verfügen. Jedes Programm oder jede Aufgabe kann auf die globalen Tags zugreifen, welche die folgende Notation verwenden:

.<Tag-Name>
.<Strukturname>.<Tag-Name>

Hinweis: Strukturen können innerhalb anderer Strukturen verschachtelt werden. Ein Tag-Name kann somit mehrere Struktureinträge als Präfix haben (<Äußere Struktur>.<Innere Struktur>.<mein Tag>).

Programm-Tags

Programm-Tags sind mit globalen Tags identisch, jedoch mit der Ausnahme, dass der Umfang eines Programm-Tags lokal auf das Programm beschränkt ist, in dem es definiert wurde. Programm-Tags unterliegen denselben Adressierungsregeln und Beschränkungen wie globale Tags. Der einzige Unterschied ist, dass Programm-Tags als Präfix den Programmnamen haben:
<Programmname>.<Tag-Name>
ODER
<Programmname>.<Äußere Struktur>.<Innere Struktur>.<Tag-Name>

Beispiel:
"prog_1.tag_1": Programm "prog_1" enthält ein Tag mit Namen "tag_1".

Tag-Adressierung

CODESYS V2.3

Es gibt zwei Kategorien von Variablen, die von CODESYS unterstützt werden: global und privat. Jede dieser Kategorien hat ein leicht unterschiedliches Format. Die folgenden Tabellen enthalten Beispiele der unterschiedlichen CODESYS-Variablen sowie deren entsprechende Serveradressierungssyntax.

Adresstyp	Adresssyntax
Globale Variable	.MyTag
Globale Variable innerhalb von einer Struktur (struct*)	.MyStruct.MyTag
Globales Bit innerhalb der Variablen	.MyTag.[0]
Private Variable	Program.MyTag
Private Variable innerhalb von einer Struktur (struct*)	Program.MyStruct.MyTag
Privates Bit innerhalb einer Variablen	Program.MyTag.[0]

* Mehrere Ebenen von structs (bis zu acht Ebenen tief) werden unterstützt.

CODESYS V3

Es gibt zwei Kategorien von Variablen, die von CODESYS unterstützt werden: global und privat. Jede dieser Kategorien hat ein leicht unterschiedliches Format. Die folgenden Tabellen enthalten Beispiele der unterschiedlichen CODESYS-Variablen sowie deren entsprechende Serveradressierungssyntax.

Adresstyp	Adresssyntax
Globale Variable	Application.GVL.MyTag
Globale Variable innerhalb von einer Struktur (struct*)	Application.GVL.MyStruct.MyTag
Globales Bit innerhalb der Variablen	Application.GVL.MyTag.[0]
Private Variable	Application Program.MyTag
Private Variable innerhalb von einer Struktur (struct*)	Application Program.MyStruct.MyTag
Privates Bit innerhalb einer Variablen	Application Program.MyTag.[0]

* Mehrere Ebenen von structs (bis zu acht Ebenen tief) werden unterstützt.

Verschachtelte Strukturen

Symbolische Tags können tief in einer beliebigen Anzahl von Strukturen verschachtelt werden (.Struct1.Struct2.Struct3.Struct4.Struct5.Struct6.Struct7.Struct8.Struct9.tag). Damit die Tag-Hierarchie nicht durch verschachtelte Strukturen übermäßig komplex wird, hindert die automatische Tag-Generierung den Server daran, Tags mit mehr als acht Gruppen Tiefe zu generieren. Wenn die Hierarchie des Tags auf dem Gerät tiefer ist als acht Gruppen, werden die verbleibenden Gruppen im Tag-Namen platziert. Bei automatisch generierten Tags ist die erste Gruppe entweder GLOBAL oder der Name der jeweiligen Programmorganisationseinheit.

Hinweise:

1. Dies gilt nur, wenn die Eigenschaft Automatisch generierte Untergruppen zulassen auf "Aktiviert" festgelegt ist.
2. Die Symboldatei enthält Datenpunkte für Strukturen als Ganzes, die als Byte-Array dargestellt werden. Der Server unterstützt diese Tags nicht.

Arrays

CODESYS-Treiber unterstützt Arrays für alle Basistypen sowie für benutzerdefinierte Strukturen. Der Treiber unterstützt die Arrays, die in einer SPS definiert werden können, teilweise. Sowohl ein- als auch zweidimensionale Arrays werden für alle Typen unterstützt, ausgenommen Zeichenfolgen und Strukturen. Auf Zeichenfolgen und Strukturen muss mithilfe der Syntax **Array-Element** zugegriffen werden. Weder dreidimensionale Arrays noch verschachtelte Arrays können als Arrays gelesen werden. Auf diese muss Element für Element zugegriffen werden und zwar mithilfe der Syntax Array-Element.

Es ist erforderlich, die Größe aller Array-Bemaßungen anzugeben. Die Array-Größe auf dem Server muss mit der Array-Größe in der SPS übereinstimmen, um das Array zu lesen oder zu schreiben. *In der nachstehenden Tabelle finden Sie mehrere Beispiele.*

Adresstyp	Adresssyntax
Globales 1D-Array (10 Elemente)	.myArray{10}
Privates 1D-Array (100 Elemente)	PLC_PRG.myArray {100}
Globales 2D-Array (9 Elemente)	.myArray{3}{3}
Privates 2D-Array (25 Elemente)	PLC_PRG.myArray {5}{5}

Hinweis: Die maximale Anzahl der von einem Array unterstützten Elemente beträgt 65535.

Array-Elemente

Ist das Lesen eines gesamten Arrays nicht erforderlich oder wird nicht unterstützt, ist es möglich, direkt auf ein bestimmtes Array-Element zuzugreifen. Element-Zugriff ist für ein-, zwei- und dreidimensionale Arrays sowie für verschachtelte Arrays verfügbar. In der nachstehenden Tabelle finden Sie ein Beispiel für die Adresssyntax. Es ist wichtig zu beachten, dass auf Strukturen als Array-Elemente zugegriffen werden muss. Es muss jede Mitgliedsvariable einer Struktur verwendet werden, um auf diese zuzugreifen, da das Lesen einer ganzen Struktur als ein einzelnes Tag von OPC nicht unterstützt wird.

Adresstyp	Adresssyntax
Eindimensionales Array-Element	.myArray[0]
Zweidimensionales Array-Element	.myArray[1,7]
Dreidimensionales Array-Element	.myArray[2,4,5]
Verschachteltes Array-Element (5 Ebenen)	.myNestedArray[1][0][4][5][9]
Verschachteltes 3D-Array (2 Ebenen)	.my3DNestedArray[4,8,1][3,2,0]

Hinweis: Nicht mit Null indizierte Arrays, d.h. Arrays, deren erstes Element nicht mit einer Null beginnt, werden unterstützt. Beispiel: Verwenden Sie myArray[1] als Adresse für das erste mit 1 indizierte Element eines 1D-Arrays.

Bit-Adressierung

Der CODESYS-Treiber unterstützt Bit-Adressierung für die folgenden Controller-Datentypen: USINT, SINT, BYTE, UINT, INT, WORD, UDINT, DINT und DWORD. Bit-Adressierung ermöglicht den Zugriff auf ein einzelnes Bit innerhalb von 1-, 2- oder 4-Byte-Datentypen im Controller.

Beim Schreiben in ein Bit-Tag liest der Treiber den Inhalt des Datentyps im Controller, ändert den Wert des zu schreibenden Bits und schreibt den gesamten Inhalt des Datentyps zurück in den Controller. Dies wird auch als RMW-Verhalten bezeichnet (lesen/ändern/schreiben).

Adresstyp	Adresssyntax
Globale Bit-Variable	Application.GVL.MyTag.[0]
Globale Bit-Variable innerhalb einer Struktur	Application.GVL.MyStruct.MyTag.[1]
Private Bit-Variable	.Program.MyTag.[2]
Private Bit-Variable innerhalb einer Struktur	.Program.MyStruct.MyTag.[2]
Globale Bit-Variable in einem Array-Element	Application.GVL.MyArray[0].[5]

Hinweis: Der Bit-Wert in den Klammern für Bit-Adressen kann zwischen 0 bis 31 liegen.

Unionen

CODESYS-Treiber unterstützt die Union-Datenstruktur bei der Kommunikation mit einem V3-Gerät. Alle in der Union definierten Komponenten benutzen gemeinsam den gleichen Arbeitsspeicher.

Eine Wertänderung an einer Variablen in der Union wirkt sich auf alle anderen Variablen in der Union aus, unabhängig vom Datentyp.

Zugriffsberechtigungen

Jedes durch die Symboldatei eines Programms beschriebene Tag hat bestimmte Zugriffsberechtigungen: keine, Schreiben, Lesen oder beide (Lesen und Schreiben). Diesen Berechtigungen entnimmt der Server, was das SPS-Programm für die einzelnen Tags vorsieht, mit denen ein Client interagiert. Die Berechtigungen können jedoch vom Server überschrieben werden. Dies wird durch die Eigenschaft "Client-Zugriff" für jedes Tag unterstützt. Wenn ein Tag auf dem Server als schreibgeschützt konfiguriert ist, lässt der Server unabhängig von den vom Gerät gemeldeten Zugriffsberechtigungen nur Lesevorgänge von Clients zu. Wenn das Tag als "Lesen/Schreiben" konfiguriert ist, können Clients unabhängig von den vom Gerät gemeldeten Zugriffsberechtigungen Lese- und Schreibvorgänge im Tag ausführen.

Wenn ein Benutzer die automatische Tag-Generierung durchführt, werden die Tags anhand der vom Gerät gemeldeten Zugriffsberechtigungen mit dem richtigen Client-Zugriff konfiguriert. Es gibt zwei Fälle, in denen dies nicht geschieht.

Lesegeschützt: Als lesegeschützt gemeldete Tags werden als "Lesen/Schreiben" generiert und erhalten sowohl Lese- als auch Schreibberechtigung, da keine Einstellung für lesegeschützten Client-Zugriff vorhanden ist.

Keine: Mit der Berechtigung "Keine" gemeldete Tags werden weder durch die automatische Tag-Generierung noch durch die manuelle Tag-Generierung unterstützt, weil keine der vorhandenen Einstellungen für den Client-Zugriff diese Zugriffsberechtigung korrekt darstellt.

Adressieren unteilbarer Datentypen

Die nachstehende Tabelle zeigt die verfügbaren CODESYS-Datentypen sowie deren OPC-Entsprechungen.

Unteilbarer Datentyp (CODESYS-Typ)	OPC-Datentyp
BOOL	Boolean
USINT	Byte, Char

Unteilbarer Datentyp (CODESYS-Typ)	OPC-Datentyp
SINT	Byte, Char
BYTE	Byte, Char
UINT	Short, Word
INT	Short, Word
WORD	Short, Word
DWORD	DWord, Long
UDINT	DWord, Long
DINT	DWord, Long
ULINT	QWord, Long Long
LINT	QWord, Long Long
LWORD	QWord, Long Long
REAL	Float
LREAL	Double
TIME	DWord
TIME_OF_DAY	DWord
LTIME	QWord
DATUM	DWord, Date
DATE_AND_TIME	DWord, Date
STRING	String
WSTRING	String
ENUM	Short, Word
BIT	Boolean

Nicht unterstützte CODESYS-Datentypen

Der einzige nicht unterstützte Datentyp ist der Zeiger.

Siehe auch: *Adressbeschreibungen*

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.

Tag-Generierung fehlgeschlagen, da die Gerätesymbole nicht geladen werden konnten. | Gerät = '<Kanalname.Gerätename>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die automatische Tag-Generierungs-Operation ist fehlgeschlagen, da der Server nicht auf die Symbolinformationen auf dem Gerät zugreifen konnte.

Mögliche Lösung:

1. Vergewissern Sie sich, dass das Gerät seine eigenen Symbolinformationen speichern kann. Einige Geräte verfügen nicht über diese Funktion und erfordern es, dass die Symbolinformationen manuell exportiert werden und der Server auf diese Informationen zugreift.
2. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.

Tag-Generierung fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte. | Gerät = '<Kanalname.Gerätename>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die automatische Tag-Generierungs-Operation ist fehlgeschlagen, da der Server keine Verbindung zu Gerät herstellen konnte.

Mögliche Lösung:

1. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.
2. Physische Verbindung zwischen Server und Gerät prüfen.
3. Wird für die Verbindung mit einem Gerät ein Gateway verwendet, überprüfen Sie, dass der CODESYS Gateway auf dem gleichen Host wie der Server installiert ist.

Tag-Generierung aufgrund eines unerwarteten Fehlers fehlgeschlagen. | Gerät = '<Kanalname.Gerätename>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die automatische Tag-Generierungs-Operation ist aufgrund unbekannter Ursache fehlgeschlagen.

Mögliche Lösung:

1. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.
2. Physische Verbindung zwischen Server und Gerät prüfen.
3. Sicherstellen, dass das Gerät ordnungsgemäß funktioniert.
4. Wenden Sie sich an den technischen Support.

Interner Fehler. Ein unerwarteter Fehler ist aufgetreten. Die SPS-Verbindung wird zurückgesetzt. | Transaktionsinformationen = '<Transaktions-Typ und Details>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Es ist ein unbekannter Fehler aufgetreten.

Mögliche Lösung:

Versuchen Sie die Operation erneut oder wenden Sie sich an den technischen Support.

Die Geräteermittlung ist aufgrund eines unerwarteten Fehlers fehlgeschlagen.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Geräteermittlungsvorgang ist aufgrund unbekannter Ursache fehlgeschlagen.

Mögliche Lösung:

1. Prüfen Sie die physische Verbindung zwischen dem Server bzw. Gateway (bei Verwendung) und den Geräten.
2. Stellen Sie sicher, dass die Geräte bzw. der Gateway (bei Verwendung) ordnungsgemäß funktionieren.
3. Wenden Sie sich an den technischen Support.

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Es kann keine Verbindung zum Gerät hergestellt werden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Der Server konnte keine Verbindung zum Gerät herstellen.

Mögliche Lösung:

1. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.
2. Physische Verbindung zwischen Server und Gerät prüfen.
3. Wird für die Verbindung mit einem Gerät ein Gateway verwendet, überprüfen Sie, dass der CODESYS Gateway auf dem gleichen Host wie der Server installiert ist.

Fehler beim Durchsuchen von Tags.

Fehlertyp:

Fehler

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein Fehler aufgetreten. Symbolliste konnte nicht vom Gerät oder aus der Datei abgerufen werden.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Der Server konnte nicht auf die Symbolinformationen auf dem Gerät zugreifen.
2. Die Symbole für das Gerät stimmen nicht mit der Symboldatei überein, die für die Gerät-Kommunikationsparameter angegebenen ist.

Mögliche Lösung:

1. Vergewissern Sie sich, dass das Gerät seine eigenen Symbolinformationen speichern kann. Einige Geräte verfügen nicht über diese Funktion und es muss eine Symboldatei in den Gerät-Kommunikationsparametern angegeben werden.
2. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.

Beim Versuch, das Tag zu lesen, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Das Lesen ist aufgrund unbekannter Ursache fehlgeschlagen.

Mögliche Lösung:

Wenden Sie sich an den technischen Support.

Beim Versuch, das Tag zu schreiben, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Es wurden ASCII-fremde Zeichen in das Zeichenfolgen-Tag geschrieben.
2. Das Schreiben ist aufgrund unbekannter Ursache fehlgeschlagen.

Mögliche Lösung:

1. Nur ASCII-Zeichen in das Zeichenfolgen-Tag schreiben.
2. Wenden Sie sich an den technischen Support.

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Es wurde ein nicht unterstützter Datentyp oder eine ungültige Adresse angegeben. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Datentyp = '<type>'.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Der angegebene Datentyp wird nicht unterstützt.
2. Die angegebene Adresse ist nicht gültig.

Mögliche Lösung:

Sicherstellen, dass der richtige Datentyp und die richtige Adresse angegeben wurden.

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Tag-Adresse wurde auf dem Gerät nicht gefunden.

Mögliche Lösung:

1. Vergewissern Sie sich, dass die richtige Adresse angegeben wurde.
2. Vergewissern Sie sich, dass die Adresse auf dem Gerät vorhanden ist.

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Tag-Adresse wurde auf dem Gerät nicht gefunden.

Mögliche Lösung:

1. Vergewissern Sie sich, dass die richtige Adresse angegeben wurde.
2. Vergewissern Sie sich, dass die Adresse auf dem Gerät vorhanden ist.

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Der angegeben Server-Datentyp ist nicht kompatibel mit dem Gerät-Datentyp.

Mögliche Lösung:

Server-Datentyp in einen Typ ändern, der mit dem Gerät-Datentyp für diese Adresse kompatibel ist.

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Der angegeben Server-Datentyp ist nicht kompatibel mit dem Gerät-Datentyp.

Mögliche Lösung:

Server-Datentyp in einen Typ ändern, der mit dem Gerät-Datentyp für diese Adresse kompatibel ist.

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein interner Fehler aufgetreten. Die angegebene Konfiguration ist nicht gültig.

Fehlertyp:

Fehler

Mögliche Ursache:

Die angegebene Konfiguration ist nicht gültig.

Mögliche Lösung:

Wenden Sie sich an den technischen Support.

Der Durchsuchen-Pfad enthält ungültige Zeichen.

Fehlertyp:

Fehler

Mögliche Ursache:

Der vom Benutzer angegebene Pfad enthält Zeichen, die keine ASCII-Zeichen sind.

Mögliche Lösung:

Alle ungültigen Zeichen entfernen.

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Array-Größe von Server und Gerät stimmt nicht überein.

Mögliche Lösung:

Geben Sie die gleiche Array-Größe für Server und Gerät an.

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Array-Größe von Server und Gerät stimmt nicht überein.

Mögliche Lösung:

Geben Sie die gleiche Array-Größe für Server und Gerät an.

Durchsuchen nach Tags fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Operation zum Durchsuchen nach Tags ist fehlgeschlagen, da der Server keine Verbindung zu Gerät herstellen konnte.

Mögliche Lösung:

1. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.
2. Physische Verbindung zwischen Server und Gerät prüfen.
3. Wird für die Verbindung mit einem Gerät ein Gateway verwendet, überprüfen Sie, dass der CODESYS Gateway auf dem gleichen Host wie der Server installiert ist.

Beim Versuch das Tag zu lesen ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen.

Fehlertyp:

Fehler

Mögliche Ursache:

Das entsprechende Tag auf dem Gerät hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet.

Mögliche Lösung:

Verringern Sie die Zeichenfolgenlänge des Tags auf dem Gerät auf einen Wert, der vom Server unterstützt wird.

Beim Versuch das Tag zu schreiben ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen.

Fehlertyp:

Fehler

Mögliche Ursache:

Das entsprechende Tag auf dem Gerät hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet.

Mögliche Lösung:

Verringern Sie die Zeichenfolgenlänge des Tags auf dem Gerät auf einen Wert, der vom Server unterstützt wird.

Der Durchsuchungspfad ist nicht vorhanden.

Fehlertyp:

Fehler

Mögliche Ursache:

Benutzer hat einen ungültigen Pfad angegeben.

Mögliche Lösung:

1. Vergewissern Sie sich, dass der Pfad für das Ziel-Gerät gültig ist.
2. Vergewissern Sie sich, dass Gerät richtig programmiert ist.

Durchsuchen nach Tags wurde aufgrund eines unbekannten Fehlers abgebrochen.

Fehlertyp:

Fehler

Mögliche Ursache:

1. Laufzeit wird heruntergefahren.
2. Es ist ein unbekannter Fehler aufgetreten.

Mögliche Lösung:

1. Durchsuchen nach Tags erneut versuchen.
2. Wenden Sie sich an den Technischen Support, wenn das Problem weiterhin auftritt.

Die Durchsuchen-Anfrage wurde abgebrochen, weil der Treiber beendet wurde.

Fehlertyp:

Fehler

Mögliche Ursache:

Die Lizenz oder der Evaluierungszeitraum für den Treiber ist abgelaufen.

Mögliche Lösung:

Überprüfen Sie, dass der Treiber ordnungsgemäß lizenziert ist.

Datentyp für die angegebene Adresse wird nicht unterstützt. Für diesen Datenpunkt wird kein Tag generiert. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Ursache hierfür ist ein nicht unterstützter Datentyp im SPS-Programm.

Mögliche Lösung:

Vergewissern Sie sich, dass es sich bei dem Tag um den richtigen Datentyp in der Programmierungs-Software handelt. Korrigieren Sie den Datentyp oder verwenden Sie einen anderen vom Server unterstützten Datentyp, andernfalls kann auf Daten, die der nicht unterstützte Typ enthält, nicht zugegriffen werden.

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Ursache hierfür ist eine Tag-Adresse im SPS-Programm, welche die vom Server unterstützte maximale Länge überschreitet.

Mögliche Lösung:

SPS-Programm neu strukturieren, sodass die Tag-Adresse unter der maximalen Länge liegt.

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Tag-Adresse ist falsch formatiert oder wird vom Server nicht unterstützt.

Mögliche Lösung:

1. Überprüfen Sie die Integrität der Symboldatei oder nehmen Sie eine Korrektur vor, bevor Sie es erneut versuchen.
2. Überprüfen oder korrigieren Sie die Tag-Adresse, bevor Sie es erneut versuchen.

Der aus dem Zeichenfolgen-Tag gelesene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen.

Fehlertyp:

Warnung

Mögliche Ursache:

Der aus dem Zeichenfolgen-Tag gelesene Wert war länger als die vom Gerät gemeldete Puffergröße.

Mögliche Lösung:

Server-Laufzeit neu starten.

Der in das Zeichenfolgen-Tag geschriebene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen.

Fehlertyp:

Warnung

Mögliche Ursache:

Der in das Zeichenfolgen-Tag geschriebene Wert war länger als der Puffer auf dem Gerät.

Mögliche Lösung:

1. Wert schreiben mit einer Länge, die dem Puffer auf dem Gerät entspricht oder darunter liegt.
2. Puffer auf dem Gerät so erhöhen, dass der Wert, welcher geschrieben wird, passt.

Geräte über Kanäle hinweg müssen eindeutige Adressen (IP-Adresse oder Host-Name) sowie Port-Kombinationen haben. | Adresse = = '<Nummer>', Port = '<Nummer>', Überlappendes Gerät = '<Gerät>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Kombination aus Adresse und Port-Nummer wurde bereits für ein anderes Gerät verwendet.

Mögliche Lösung:

Adresse und/oder Port für dieses Gerät ändern.

Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>'.

Fehlertyp:

Warnung

Der Dateipfad darf nicht leer sein.

Fehlertyp:

Warnung

Die Symboldatei war ungültig oder beschädigt. | Datei = '<Pfad zur Datei>'.

Fehlertyp:
Warnung

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

Fehlertyp:
Warnung

Fehler beim Versuch, das Tag zu schreiben. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'.

Fehlertyp:
Warnung

Fehler beim Versuch, das Tag zu lesen. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'.

Fehlertyp:
Warnung

Ungültige Adresse. Gültige logische Adresse oder SPS-Namen angeben.

Fehlertyp:
Warnung

Geräte über Kanäle hinweg müssen eindeutige logische Adressen/SPS-Namen haben. | Adresse = '<Adresse>', Überlappendes Gerät = '<Gerät>'.

Fehlertyp:
Warnung

Einige der importierten Tags haben Tags mit demselben Namen in Ihrem Projekt ersetzt.

Fehlertyp:
Warnung

Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>', Grund = '<Fehlermeldung vom Betriebssystem>'.

Fehlertyp:

Warnung

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Ursache hierfür ist eine Tag-Adresse im SPS-Programm, welche die vom Server unterstützte maximale Länge überschreitet.

Mögliche Lösung:

SPS-Programm neu strukturieren, sodass die Tag-Adresse unter der maximalen Länge liegt.

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'.

Fehlertyp:

Warnung

Mögliche Ursache:

Die Tag-Adresse ist falsch formatiert oder wird vom Server nicht unterstützt.

Mögliche Lösung:

1. Überprüfen Sie die Integrität der Symboldatei oder nehmen Sie eine Korrektur vor, bevor Sie es erneut versuchen.
2. Überprüfen oder korrigieren Sie die Tag-Adresse, bevor Sie es erneut versuchen.

Fehler beim Durchsuchen von Tags. | Grund = '<Zeichenfolge mit Erklärung für das Fehlschlagen des Durchsuchens von Tags>'.

Fehlertyp:

Warnung

Mögliche Ursache:

1. Die Operation zum Durchsuchen nach Tags ist fehlgeschlagen, da der Server keine Verbindung zu Gerät herstellen konnte.
2. Angegebener Pfad ist ungültig.
3. Der Server konnte nicht auf die Symbolinformationen auf dem Gerät zugreifen.

4. Die Symbole für das Gerät stimmen nicht mit der Symboldatei überein, die für die Gerät-Kommunikationsparameter angegebenen ist.

Mögliche Lösung:

1. Sicherstellen, dass die Kommunikationsparameter des Geräts auf dem Server richtig konfiguriert sind.
2. Physische Verbindung zwischen Server und Gerät prüfen.
3. Wird für die Verbindung mit einem Gerät ein Gateway verwendet, überprüfen Sie, dass der CODESYS Gateway auf dem gleichen Host wie der Server installiert ist.
4. Vergewissern Sie sich, dass der Pfad für das Ziel-Gerät gültig ist.
5. Vergewissern Sie sich, dass Gerät richtig programmiert ist.
6. Vergewissern Sie sich, dass das Gerät seine eigenen Symbolinformationen speichern kann. Einige Geräte verfügen nicht über diese Funktion und es muss eine Symboldatei in den Gerät-Kommunikationsparametern angegeben werden.

Tags generiert. | Tag-Anzahl = <Anzahl>.

Fehlertyp:

Informationen

Index

A

Adresse 21

Adressieren unteilbarer Datentypen 33

Adressregeln 30

Alle Werte für alle Tags schreiben 9

Allgemein 13

Anfangsaktualisierungen aus Cache 16

Anforderungs-Timeout 21

Anfragen verwerfen, wenn herabgestuft 17

Array-Elemente 32

Arrays 32

Automatische Herabstufung 16

B

Bei doppeltem Tag 18

Bei Eigenschaftsänderung 17

Bei Gerätestart 17

Beim Versuch das Tag zu lesen ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 41

Beim Versuch das Tag zu schreiben ist ein Fehler aufgetreten. Die angegebene Tag-Adresse hat eine Zeichenfolgenlänge, die die vom Server unterstützte maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 41

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'. 39

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'. 40

Beim Versuch, das Tag zu lesen, ist ein Fehler aufgetreten. Es wurde ein nicht unterstützter Datentyp oder eine ungültige Adresse angegeben. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Datentyp = '<type>'. 38

Beim Versuch, das Tag zu lesen, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Der angegebene Server-Datentyp ist

nicht mit dem Datentyp des Geräts kompatibel. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Datentyp = '<Typ>', Gerät-Datentyp = '<Typ>'. 39

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die angegebene Tag-Adresse wurden auf dem Gerät nicht gefunden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 39

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Die Array-Größe von Server und Gerät muss übereinstimmen. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Server-Array-Größe = '<Länge>', Gerät-Array-Größe = '<Länge>'. 40

Beim Versuch, das Tag zu schreiben, ist ein Fehler aufgetreten. Es kann keine Verbindung zum Gerät hergestellt werden. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 37

Beim Versuch, das Tag zu schreiben, ist ein interner Fehler aufgetreten. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 38

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein Fehler aufgetreten. Symbolliste konnte nicht vom Gerät oder aus der Datei abgerufen werden. 37

Beim Versuch, eine Verbindung zum Gerät herzustellen, ist ein interner Fehler aufgetreten. Die angegebene Konfiguration ist nicht gültig. 40

Benutzeranmeldeinformationen 23

Benutzername 24

Betriebsmodus 14

BIT 34

Bit-Adressierung 32

BOOL 33

Boolean 29

Byte 29

BYTE 34

C

Cache 22

Char 29

Client-Zugriff 33

Client/Server 30

D

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen. 46

Das Tag konnte nicht in das Durchsuchen eingeschlossen werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 46

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse die maximale Länge überschreitet. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<1024>' Zeichen. 43

Das Tag konnte nicht zum Server hinzugefügt werden, da die Adresse nicht validiert werden konnte. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 43

DATE_AND_TIME 34

Datensammlung 14

Datentyp für die angegebene Adresse wird nicht unterstützt. Für diesen Datenpunkt wird kein Tag generiert. | Tag-Adresse = '<.mystruct.innerstruct.tag>'. 43

Datentypbeschreibung 29

Datum 29

DATUM 34

Der aus dem Zeichenfolgen-Tag gelesene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 44

Der Dateipfad darf nicht leer sein. 44

Der Durchsuchen-Pfad enthält ungültige Zeichen. 40

Der Durchsuchungspfad ist nicht vorhanden. 42

Der in das Zeichenfolgen-Tag geschriebene Wert wurde abgeschnitten. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Maximale Länge = '<Anzahl>' Zeichen. 44

Diagnose 8

Die Durchsuchen-Anfrage wurde abgebrochen, weil der Treiber beendet wurde. 42

Die Geräteermittlung ist aufgrund eines unerwarteten Fehlers fehlgeschlagen. 36

Die Symboldatei war ungültig oder beschädigt. | Datei = '<Pfad zur Datei>'. 45

DINT 34

Double 29

Durch Null ersetzen 10

Durch Tag angegebenes Scan-Intervall berücksichtigen 16

Durchsuchen nach Tags fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte. 41

Durchsuchen nach Tags wurde aufgrund eines unbekannten Fehlers abgebrochen. 42

DWord 29

DWORD 34

E

Ebene 7 21

Einige der importierten Tags haben Tags mit demselben Namen in Ihrem Projekt ersetzt. 45

ELAU-Max4 Version 22

Elternteilgruppe 18

ENUM 34

Ereignisprotokollmeldungen 35

Erstellen 19

Erweiterten ASCII-Zeichensatz zulassen 23

F

Fehler beim Durchsuchen von Tags. 37

Fehler beim Durchsuchen von Tags. | Grund = '<Zeichenfolge mit Erklärung für das Fehlschlagen des Durchsuchens von Tags>'. 46

Fehler beim Öffnen der Datei für Tag-Datenbank-Import. | Betriebssystemfehler = '<Betriebssystem-Meldung>'. 45

Fehler beim Versuch, das Tag zu lesen. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'. 45

Fehler beim Versuch, das Tag zu schreiben. Der Bit-Wert überschreitet die Größe des Controller-Datentyps. | Tag-Adresse = '<.mystruct.innerstruct.tag>', Bit-Wert = '<Bit-Speicherort>', Controller-Datentyp-Größe (Bytes) = '<Größe in Bytes>'. 45

Float 29

G

Gateway 23

Generieren 18

Gerät 22

Geräte über Kanäle hinweg müssen eindeutige Adressen (IP-Adresse oder Host-Name) sowie Port-Kombinationen haben. | Adresse = = '<Nummer>', Port = '<Nummer>', Überlappendes Gerät = '<Gerät>'. 44

Geräte über Kanäle hinweg müssen eindeutige logische Adressen/SPS-Namen haben. | Adresse = '<Adresse>', Überlappendes Gerät = '<Gerät>'. 45

Geräteeigenschaften – Allgemein 13

Geräteeigenschaften – Automatische Herabstufung 16

Geräteeigenschaften – Redundanz 26

Geräteeigenschaften – Tag-Generierung 17

Geräteerkennung 11

Geräteerkennungs-Einstellungen 12

Global 31

Globale Tags 30

Globales Bit 33

H

Herabstufen bei Fehler 16

Herabstufungszeitraum 17

Host-Name 21

I

ID 14
Identifikation 7, 13
Inhalt der Hilfe 6
INT 34
Interner Fehler. Ein unerwarteter Fehler ist aufgetreten. Die SPS-Verbindung wird zurückgesetzt. | Transaktionsinformationen = '<Transaktions-Typ und Details>'. 36

K

Kanaleigenschaften – Allgemein 7
Kanaleigenschaften – Erweitert 10
Kanaleigenschaften – Schreiboptimierungen 9
Kanalzuweisung 14
Keep-Alive-Anzahl 11
Keine 33
Kommunikationsparameter 10
Kommunikationsparameter, Bytereihenfolge, Byte, Reihenfolge, Motorola, Endian, Ebene 7 21

L

Leistungsoptimierung 28
Lesen 33
Lesen/Schreiben 33
LINT 34
Long 29
Long Long 29
Löschen 18
LREAL 34
LTIME 34
LWORD 34

M

Methode 25
Mit Vorzeichen 29
Modell 14

Motorola 21

N

Name 13
Nicht geändert 10
Nicht normalisierte Float-Handhabung 10
Nicht scannen, nur Abruf anfordern 16
Nicht unterstützt 29, 34
Nur den letzten Wert für alle Tags schreiben 9
Nur den letzten Wert für nicht boolesche Tags schreiben 9

O

Ohne Vorzeichen 29
Optimierungsmethode 9
OWord 29

P

Passwort 24
Port 21
Privat 31
Privates Bit 33
Programm-Tags 30
Protokoll 6, 21

R

REAL 34
Redundanz 26

S

Scan-Modus 15
Schreiben 33
Servicezyklus 9
Setup 6

Short 29
Simuliert 15
SINT 34
SPS 22
String 29
STRING 34
Symbol-Cachespeicherort 22
Symboldatei 22
Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>', Grund = '<Fehlermeldung vom Betriebssystem>'. 46
Symboldatei konnte nicht geöffnet werden. | Datei = '<Pfad zur Datei>'. 44
Symbole laden 22
Symbolisch 30

T

Tag-Adressierung 30
Tag-Generierung 17
Tag-Generierung aufgrund eines unerwarteten Fehlers fehlgeschlagen. | Gerät = '<Kanalname.Gerätename>'. 36
Tag-Generierung fehlgeschlagen, da die Gerätesymbole nicht geladen werden konnten. | Gerät = '<Kanalname.Gerätename>'. 35
Tag-Generierung fehlgeschlagen, da Kommunikation zum Gerät nicht hergestellt werden konnte. | Gerät = '<Kanalname.Gerätename>'. 35
Tag-Importeinstellungen 24
Tag-Umfang 30
Tag-Zähler 8, 15
Tags generiert. | Tag-Anzahl = <Anzahl>. 47
Tags pro Anforderung 23
TIME 34
TIME_OF_DAY 34
Timeout bis zum Herabstufen 16
Treiber 14

U

Überschreiben 18
Übersicht 6
UDINT 34

UINT 34
ULINT 34
Ungültige Adresse. Gültige logische Adresse oder SPS-Namen angeben. 45
Untergruppen zulassen 19
USINT 33

V

V2.3 23
V3 23
Verschachtelte Strukturen 31
Verzögerung zwischen Geräten 10

W

Word 29
WORD 34
WSTRING 34

Z

Ziel 22
Zugriffsberechtigungen 33