Simulator Driver

Inhaltsverzeichnis

Simulator Driver

Hilfeversion 1.036

INHALT

Übersicht

Simulator Driver bietet eine zuverlässige Möglichkeit, Simulator Geräte mit OPC-Client-Anwendungen, u.a. HMI, SCADA, Historian, MES, ERP sowie zahlreichen benutzerdefinierten Anwendungen, zu verbinden. Die Bereitstellung erfolgt zum Testen der OPC-Server-Software ohne dass dafür ein externes Gerät erforderlich ist.

Setup

Unterstützte Geräte

8-Bit
16-Bit

Hinweis: Jedes Gerät unterstützt bis zu 10000 adressierbare Lese-/Schreib-Register und konstante Speicherorte, zusätzlich zu 1000 Lese-/Schreib-Zeichenfolge-Speicherorten von variabler Länge. *Weitere Informationen finden Sie unter Adressbeschreibungen.*

Maximale Anzahl von Kanälen und Geräten

Die maximale Anzahl unterstützter Kanäle ist 100. Die maximale Anzahl von unterstützten Geräten pro Kanal liegt bei 999.

Geräte-ID

Simulator Geräten können Geräte-IDs im Bereich von 1 bis 999 zugewiesen werden. Werden andere Modelltypen im gleichen Kanal verwendet, sind eindeutige Geräte-IDs erforderlich.

Livedaten-Simulation

Der Treiber simuliert Livedaten durch Erhöhen von Registerdaten jedes Mal, wenn diese als Ganzzahlendatentyp gelesen werden. Zusätzlich zur Simulation eines einfachen registerbasierten Arbeitsspeichers eines SPS-ähnlichen Geräts, unterstützt Simulator Driver auch vier Simulationsfunktionen auf höchster Ebene. Diese neuen Simulationsfunktionen sind: RAMP, SINE, RANDOM und USER (benutzerdefiniert).

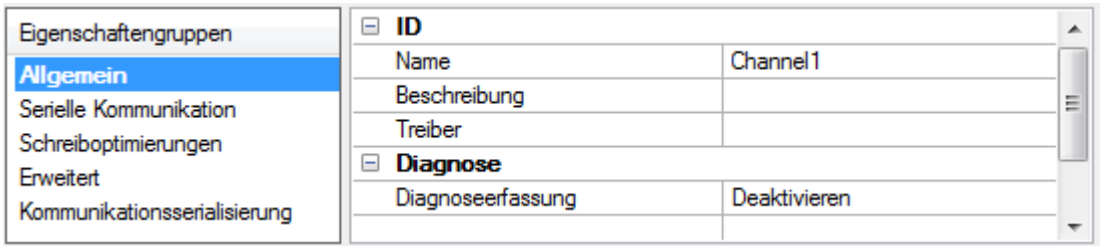
Jedes neue Simulations-Tag ist wie ein Funktionsaufruf in einer Programmiersprache strukturiert. Das Verwenden jeder Funktion erfordert, dass die entsprechenden Eigenschaften angewendet werden, um den gewünschten Simulationseffekt zu erzielen. Mit RAMP kann die Änderungsrate, der niedrige Grenzwert, der hohe Grenzwert sowie der Inkrementwert festgelegt werden. Mit SINE kann die Änderungsrate, der niedrige Grenzwert, der hohe Grenzwert, die Frequenz sowie die Phase festgelegt werden. Mit RANDOM kann die Änderungsrate, der niedrige Grenzwert sowie der hohe Grenzwert festgelegt werden. Die kreativste der neuen Simulationsfunktionen ist jedoch die benutzerdefinierte Funktion USER.

Die Funktion USER ermöglicht ebenfalls das Festlegen einer Änderungsrate. Im Gegensatz zu den voreingestellten Simulationsausgaben der Funktionen RAMP, RANDOM und SINE, wird die Funktion USER dazu verwendet, Datensequenzen zu erstellen. Anstatt hoher oder niedriger Grenzwerte akzeptiert die Funktion USER eine kommagetrennte Liste von Elementen. Die Liste von Elementen kann entweder numerische Daten oder Zeichenfolgendaten enthalten. Sobald eine Liste erstellt wurde, durchläuft die Funktion USER die Elemente in der Liste zur angegebenen Rate. Die Funktion USER kann komplexe Demos erstellen, die verwendet werden können, um praxisrelevante Ausgaben und Ergebnisse widerzuspiegeln.

Weitere Informationen finden Sie unter Simulationsfunktionen.

Kanaleigenschaften - Allgemein

Dieser Server unterstützt die Verwendung von gleichzeitigen Mehrfachkommunikationstreibern. 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.



ID

Name: Benutzerdefinierte ID dieses Kanals. 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.
Informationen über reservierte Zeichen finden Sie in der Serverhilfe unter „So benennen Sie Kanäle, Geräte, Tags und Tag-Gruppen richtig“.

Beschreibung: Benutzerdefinierte Informationen über diesen Kanal.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

Treiber: Ausgewähltes Protokoll/ausgewählter Treiber für diesen Kanal. Diese Eigenschaft gibt den Gerätetreiber an, der während der Kanalerstellung ausgewählt wurde. Es ist eine deaktivierte Einstellung in den Kanaleigenschaften.

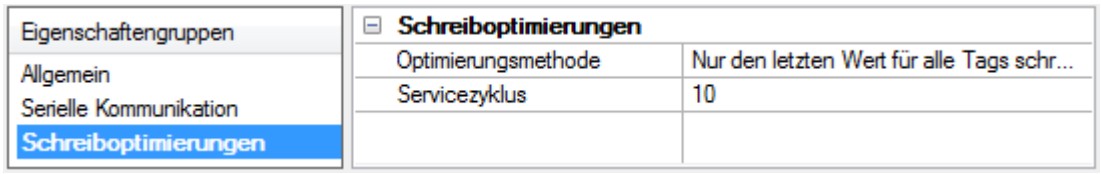
Hinweis: Beim Online-Vollzeitbetrieb des Servers können diese Eigenschaften jederzeit geändert werden. Dies schließt das Ändern des Kanalnamens ein, um zu verhindern, dass Clients Daten am Server registrieren. Wenn ein Client bereits ein Element vom Server abgerufen hat, bevor der Kanalname geändert 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. Unter Berücksichtigung dessen sollten keine Änderungen an den Eigenschaften erfolgen, sobald eine große Client-Anwendung entwickelt wurde. Verwenden Sie den Benutzermanager, um zu verhindern, dass Operatoren Eigenschaften ändern, und um Zugriffsrechte auf Serverfunktionen zu beschränken.

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.
Hinweis: Diese Eigenschaft ist deaktiviert, wenn der Treiber Diagnosen nicht unterstützt.
Weitere Informationen dazu finden Sie in der Serverhilfe unter „Kommunikationsdiagnosen“.

Kanaleigenschaften - Schreiboptimierungen

Wie bei jedem OPC-Server ist das Schreiben von Daten auf das Gerät unter Umständen der wichtigste Aspekt der Anwendung. Der Server soll sicherstellen, dass die von der Client-Anwendung geschriebenen Daten rechtzeitig auf das Gerät gelangen. In Anbetracht dieses Ziels stellt der Server Optimierungseigenschaften bereit, anhand derer die jeweiligen Anforderungen erfüllt oder die Reaktionsfähigkeit der Anwendungen verbessert werden können.



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.

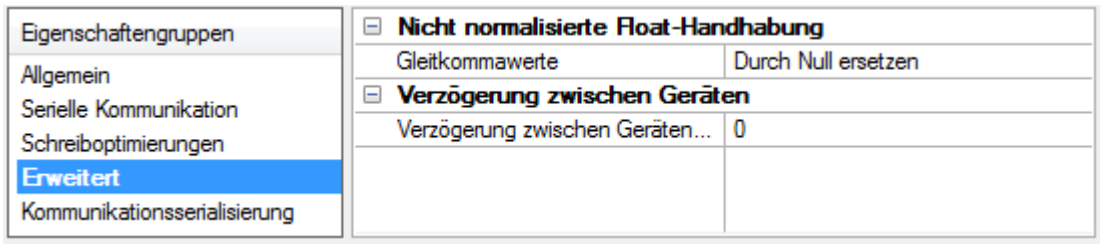
- **Nur den letzten Wert für nicht boolesche Tags schreiben**: Viele aufeinander folgende Schreibvorgänge für denselben Wert können sich aufgrund der Zeit, die tatsächlich zum Senden der Daten auf das Gerät erforderlich ist, in der Schreibwarteschlange ansammeln. Wenn der Server einen Schreibwert aktualisiert, 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: Durch nicht normalisierte Float-Handhabung können Benutzer festlegen, wie ein Treiber mit nicht normalisierten IEEE-754-Gleitkommadaten umgeht. 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. 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 senden.

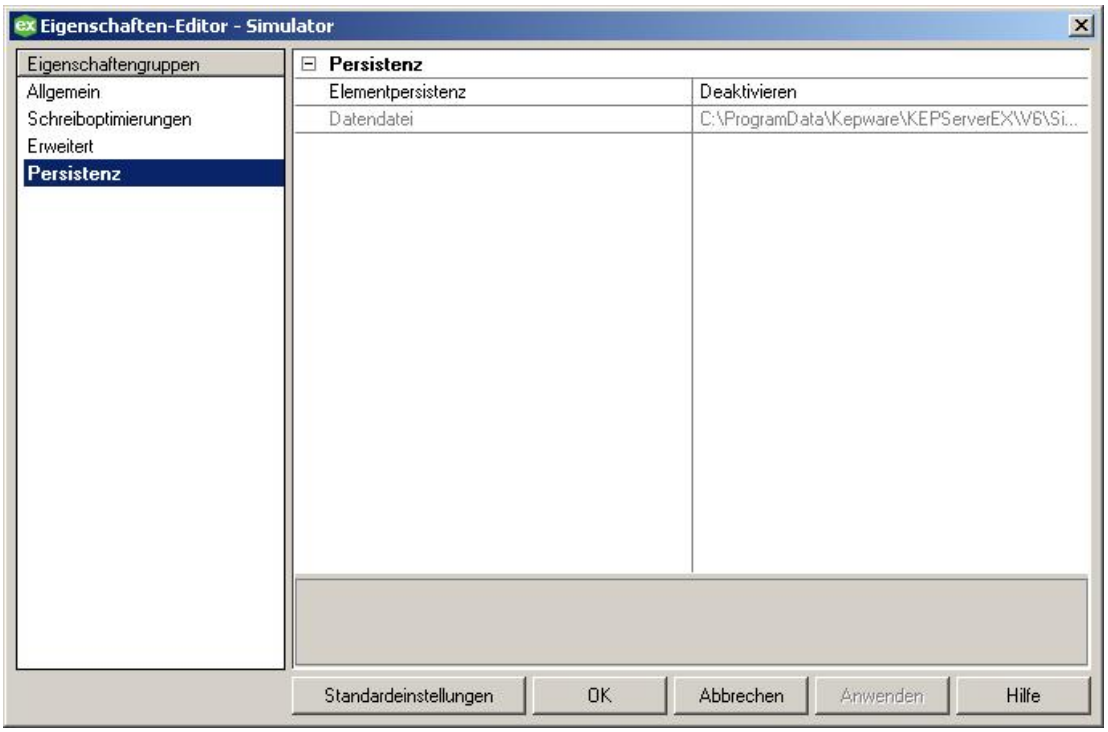
Hinweis: Diese Eigenschaft ist deaktiviert, wenn der Treiber keine Gleitkommawerte unterstützt, oder wenn er nur die angezeigte Option unterstützt. Gemäß der Float-Normalisierungseinstellung des Kanals unterliegen nur Echtzeit-Treiber-Tags (wie z.B. Werte und Arrays) der Float-Normalisierung. Beispielsweise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

Weitere Informationen über die Gleitkommawerte finden Sie unter "So arbeiten Sie mit nicht normalisierten Gleitkommawerten" 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 - Persistenz

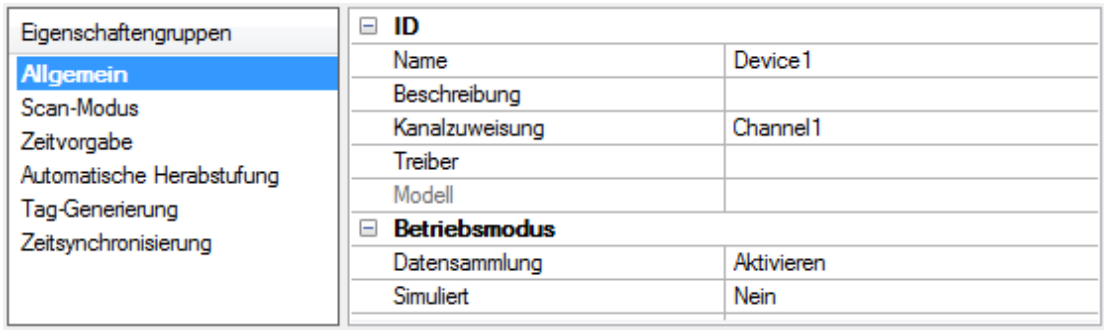


Elementpersistenz: Wählen Sie “Aktivieren“ aus, um das Gerät so zu konfigurieren, dass es seine Werte für K-, R- und S-Adressen (konstant, registrieren und Zeichenfolge) zwischen Ausführungen beibehält. Die Elementpersistenz für Simulationsfunktionen wird derzeit nicht unterstützt. Ist diese Funktion aktiviert, werden die in allen für K-, R- und S-Adressen gespeicherten Werte für jedes auf dem Kanal konfiguriertes Geräts in einer Datei gespeichert, wenn das Serverprojekt geschlossen oder der Server heruntergefahren wird. Diese Werte werden beim nächsten Öffnen des Serverprojekts aus der Datei wiederhergestellt. Für jeden Kanal, der diese Funktion verwendet, wird eine separate Datei verwendet. Die Standardeinstellung ist deaktiviert.
Weitere Informationen finden Sie unter Simulationsfunktionen.

Datendatei: Geben Sie an, wo die Daten für Geräte auf diesem Kanal gespeichert werden sollen. Ein vollständig qualifizierter Pfad sowie ein Dateiname sind erforderlich. Der Treiber erstellt die Datei und Ordner unter ihrem Pfad, Benutzer müssen jedoch diese Funktion verwenden, um eine eindeutige Datei für jeden Simulator-Kanal anzugeben.

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: Diese Eigenschaft gibt 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: Benutzerdefinierte Informationen über dieses Gerät.
Viele dieser Eigenschaften, einschließlich der Beschreibung, verfügen über ein zugeordnetes System-Tag.

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

Treiber: Ausgewählter Protokolltreiber für dieses Gerät. Diese Eigenschaft gibt den während der Kanalerstellung ausgewählten Treiber an. Sie ist in den Kanaleigenschaften deaktiviert.

Modell: Diese Eigenschaft gibt den bestimmten Typ des Geräts an, das 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 Hilfedokumentation des Treibers.

ID: Diese Eigenschaft gibt die Station, den Knoten, die ID oder die Adresse des Geräts an. Der Typ der eingegebenen ID hängt vom verwendeten Kommunikationstreiber ab. Für viele Treiber 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 ID-Format kann Dezimal, Oktal oder Hexadezimal sein. 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: Diese Option versetzt das Gerät in den Simulationsmodus. In diesem Modus versucht der Treiber nicht, mit dem physischen Gerät zu kommunizieren, aber der Server gibt weiterhin gültige OPC-Daten zurück. Durch Auswählen von "Simuliert" wird die physische Kommunikation mit dem Gerät angehalten, OPC-Daten können jedoch als gültige Daten dem OPC-Client zurückgegeben werden. Im Simulationsmodus behandelt der Server alle Gerätedaten als reflektierend: was auch immer in das simulierte Gerät geschrieben wird, wird zurückgelesen, und jedes OPC-Element wird einzeln behandelt. Die Speicherzuordnung des Elementes basiert auf dem Gruppenaktualisierungsintervall. Die Daten werden nicht gespeichert, wenn der Server das Element entfernt (z.B., wenn der Server neu initialisiert wird). Die Standardeinstellung ist "Nein".

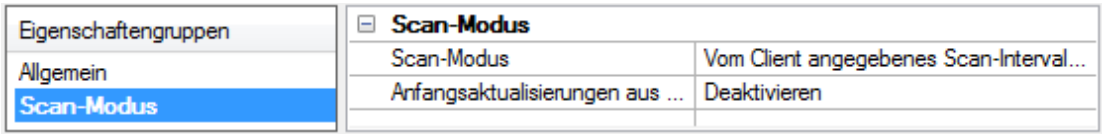
Hinweise:

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

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

Geräteeigenschaften - Scan-Modus

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



Scan-Modus: Gibt an, wie Tags im Gerät auf an abonnierte 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 das maximale Scan-Intervall an, das verwendet werden soll. 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 Clients, nach 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.

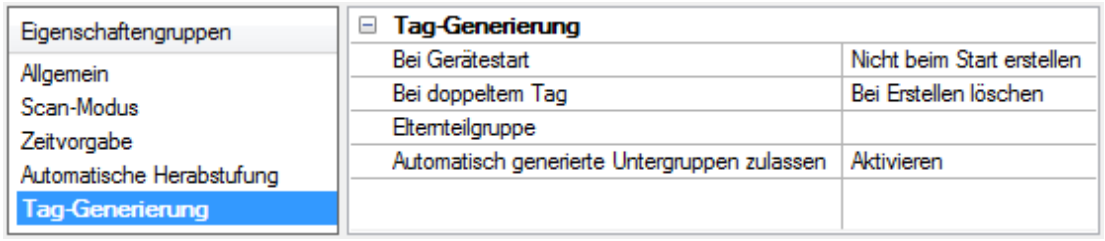
Geräteeigenschaften - Tag-Generierung

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.

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 dazu finden Sie in den Eigenschaftsbeschreibungen unten.



Bei Gerätestart
Mit dieser Eigenschaft wird festgelegt, wann OPC-Tags automatisch generiert werden. Es folgen Beschreibungen der Optionen:

- **Nicht beim Start erstellen**: Mit dieser Option wird verhindert, dass der Treiber dem Tag-Raum des Servers irgendwelche OPC-Tags 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.

Datentypbeschreibung

Jeder Adresse, auf die zugegriffen werden kann, muss ein Datentyp zugewiesen werden. Simulator Geräte unterstützen die folgenden Datentypen.

Datentyp	Beschreibung
BCD	Gepacktes 2-Byte-BCD Der Wertebereich liegt zwischen 0 und 9999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
Boolean	Einzelnes Bit
Byte	8-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 7 ist das High-Bit
Char	8-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 6 ist das High-Bit Bit 7 ist das Vorzeichen-Bit
Date	Gleitkomma-OLE-Automatisierungsdatum (Zuordnung zum Datentyp VARIANT VT_DATE)
Double*	64-Bit-Gleitkommawert Der Treiber interpretiert vier aufeinanderfolgende Register als Wert mit doppelter Genauigkeit, indem die letzten zwei Register als High-DWord und die ersten zwei Register als Low-DWord bewertet werden.
Double-Beispiel	Wenn Register 40001 als Double-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 64-Bit-Datentyps und Bit 15 des Registers 40004 ist Bit 63 des 64-Bit-Datentyps.
DWord	32-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 31 ist das High-Bit
Float*	32-Bit-Gleitkommawert Der Treiber interpretiert zwei aufeinanderfolgende Register als Wert mit einfacher Genauigkeit, indem das erste Register als Low-Wort und das zweite Register als High-Wort bewertet wird.
Float-Beispiel	Wenn Register 40001 als Float-Wert angegeben wird, ist Bit 0 des Registers 40001 Bit 0 des 32-Bit-Datentyps und Bit 15 des Registers 40002 ist Bit 31 des 32-Bit-Datentyps.
LBCD	Gepacktes 4-Byte-BCD Der Wertebereich liegt zwischen 0 und 99999999. Für Werte außerhalb dieses Bereichs ist das Verhalten nicht definiert.
Long	32-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 30 ist das High-Bit Bit 31 ist das Vorzeichen-Bit
LLong	64-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 62 ist das High-Bit Bit 63 ist das Vorzeichen-Bit
QWord	64-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 63 ist das High-Bit
Short	16-Bit-Wert mit Vorzeichen Bit 0 ist das Low-Bit Bit 14 ist das High-Bit Bit 15 ist das Vorzeichen-Bit
String	Mit Null beendetes ASCII-Zeichen-Array
Word	16-Bit-Wert ohne Vorzeichen Bit 0 ist das Low-Bit Bit 15 ist das High-Bit

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

Adressbeschreibungen

Simulator Driver unterstützt drei Arten von Adressen: R-Register, K-Register und S-Register. Die R- und K-Register sind numerische Daten. S-Register sind Zeichenfolgendatenpositionen von variabler Länge.

Die R-Register simulieren sich ändernde Daten durch das Erhöhen um 1 bei jedem Lesevorgang, wenn sie als folgende Typen referenziert werden: Char, Byte, Word, Short, BCD, Long, DWord, LLong, QWord oder LBCD. Arrays dieser Typen werden bei jedem Lesevorgang um 1 erhöht. Werden R-Register als Float oder Double referenziert, wird der Wert bei jedem Lesevorgang um 1.25 a erhöht. Arrays des Typs Float oder Double werden nicht erhöht, es sei denn, es befinden sich einzelne Tags für die Adressen im Array. Außerdem hat jeder Typ einen Bereich, über den das Erhöhen stattfindet. Für den Typ Float ist der Bereich 0 bis 32767. Für den Typ Double ist der Bereich 0 bis 65535.

Die K- und R-Register haben einen Anfangswert von Null. S-Register haben einen Anfangswert von 'String data S*n*', wobei *n* die Registernummer ist.

Adressbereich und Datentypspezifikationen sind je nach verwendetem Modell unterschiedlich. Simulator Driver unterstützt außerdem neue Simulationsfunktionen, zu denen RAMP, SINE, RANDOM und USER (benutzerdefiniert) zählen. Wenn Sie weitere Informationen benötigen, wählen Sie eine Verknüpfung aus der Liste unten aus.

8-Bit-Gerätedressen
16-Bit-Gerätedressen

8-Bit-Geräteadressen

Die Konfiguration des Speichers für das 8-Bit-Gerät wird als ein Byte-Positions-Block von 0 bis 9999 simuliert. Jedes Byte kann als ein Versatz vom Start des Blocks adressiert werden. Die Standard-Datentypen für jedes Format werden **fett** dargestellt.

Gerätetyp	Bereich	Datentyp	Zugriff
Register	R0000-R9999 R0000-R9998 R0000-R9996 R0000-R9992	**Byte**, Char Word, Short, BCD, DWord, Long, LBCD, Float, LLong, QWord, Double, Date, Boolean	Lesen/Schreiben
Konstanten	K0000-K9999 K0000-K9998 K0000-K9996 K0000-K9992	**Byte**, Char Word, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Lesen/Schreiben
Bits	R0000.0-R9999.7 K0000.0-K9999.7	**Boolean**	Lesen/Schreiben
Zeichenfolgen	S000-S999	**String**	Lesen/Schreiben

Hinweise:

1. Alle Datentypen, bis auf den booleschen Datentyp auf Bit-Ebene, unterstützen Array, indem der Adresse die Notation [r] oder [r][c] angehängt wird.
2. Die für den Datentyp angegebene Adresse muss Platz für den vollständigen Datentyp bieten. Dies bedeutet, Benutzer können nicht über das Ende des Datenbereich hinausschreiben.

Siehe auch: *Simulationsfunktionen*

16-Bit-Geräteadressen

Die Konfiguration des Speichers für das 16-Bit-Gerät wird als ein Wortpositions-Block von 0 bis 9999 simuliert. Jedes Wort kann als ein Versatz vom Start des Blocks adressiert werden. Die Standard-Datentypen für jedes Format werden **fett** dargestellt.

Gerätetyp	Bereich	Datentyp	Zugriff
Register	R0000-R9999 R0000-R9998 R0000-R9996	**Word**, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Lesen/Schreiben
Konstanten	K0000-K9999 K0000-K9998 K0000-K9996	**Word**, Short, BCD DWord, Long, LBCD, Float LLong, QWord, Double, Date, Boolean	Lesen/Schreiben
Bits	R0000.00-R9999.15 K0000.00-K9999.15	**Boolean**	Lesen/Schreiben
Zeichenfolgen	S000-S999	**String**	Lesen/Schreiben

Hinweise:

1. Alle Datentypen, bis auf den booleschen Datentyp auf Bit-Ebene, unterstützen Array, indem der Adresse die Notation [r] oder [r][c] angehängt wird.
2. Die für den Datentyp angegebene Adresse muss Platz für den vollständigen Datentyp bieten. Dies bedeutet, Benutzer können nicht über das Ende des Datenbereich hinausschreiben.

Siehe auch: Simulationsfunktionen

Simulation Functions

Simulationsfunktionen können verwendet werden, um OPC-Elemente zu erstellen, die praxisrelevante Datenquellen nachahmen. Auch wenn jede Simulationsfunktion andere Ausgaben bereitstellt, haben die Funktionen viele gemeinsame Eigenschaften wie **Rate**, **niedriger Grenzwert** und **hoher Grenzwert**. Die Rate (auch Änderungsrate genannt) wird verwendet, um anzugeben, wie oft der Simulationswert seinen Status ändert. Die Rate wird in Millisekunden mit einem Wertbereich von 10 bis 3600000 eingegeben. Diese Änderungsrate ist völlig unabhängig davon, wie oft die Client-Anwendung Daten liest. Wie bei einer SPS werden die Simulationsfunktionen immer im Hintergrund ausgeführt.

Der niedrige Grenzwert und der hohe Grenzwert

Die Eigenschaften für den niedrigen und hohen Grenzwert können verwendet werden, um den von der Simulationsfunktion generierten Bereich oder die generierte Datenspanne anzugeben. Durch die Verwendung der Eigenschaften für den niedrigen und hohen Grenzwert können Benutzer Simulationswerte erzeugen, die durch einfache Anpassung der Datenspanne versetzt werden. Für die Simulationsfunktionen, die Grenzwerte unterstützen, wird das Format, in dem der Bereich eingegeben wird, verwendet, um den Datentyp des Simulationswerts zu bestimmen. Beispiel: Wird ein RAMP-Tag mit einem niedrigen Grenzwert von 75 und einem hohen Grenzwert von 3000 eingegeben, ist das resultierende OPC-Element vom Datentyp Long. Wird dasselbe RAMP-Tag mit einem niedrigen Grenzwert von 75.123 und einem hohen Grenzwert von 3000.567 eingegeben, ist das resultierende OPC-Element vom Datentyp Float. In diesen beiden Beispielen war das Standarddatenformat entweder Float oder Long, aber einige der verfügbaren Datentypen können als Ausgabeformat für jede Simulationsfunktion ausgewählt werden. Der durch die niedrigen und hohen Grenzwerte angegebene Bereich muss jedoch innerhalb der gewünschten Datentypauswahl liegen.

Im Gegensatz zur registerbasierten Adresse oben, sind die Simulationsfunktionen, die eingegeben werden können, nicht begrenzt. Jede Simulationsfunktion mit eindeutigen Eigenschaften wird als neuer Simulationswert angesehen, Beim Erstellen von Adressierungs-Simulationsfunktionen mit der Absicht, denselben Wert in mehreren Speicherorten in der Client-Anwendung zu lesen, ist es wichtig, dass die Simulationsfunktion jeweils auf die gleiche Weise eingegeben wird.

Simulationsfunktionen sind schreibgeschützte Objekte

Simulationsfunktionen sind schreibgeschützte Objekte. Sobald eine Client-Anwendung damit beginnt, Daten von einer Simulationsfunktion zu lesen, fährt die Funktion fort, Daten zu generieren, bis das Element von einer Client-Anwendung gelöscht wird. Wenn sie Gleitkommaeigenschaften eingeben, akzeptieren die Simulationsfunktionen die Eingabe von numerischen Werten in exponentieller Form nicht.

Funktionsdefinitionen

Funktion RAMP

RAMP(Rate, Low Limit, High Limit, Increment)

Die Rampenfunktion (RAMP) kann verwendet werden, um einen Wert zu erstellen, der durch einen numerischen Bereich hindurch erhöht oder verringert wird. Der niedrige und hohe Grenzwert sollten verwendet werden, um den gewünschten Bereich festzulegen. Die niedrigen oder hohen Grenzwerte können angepasst werden, um einen Versatz auf die generierten Daten anzuwenden. Der Inkrementwert kann positiv oder negativ sein. Ist der Inkrementwert positiv, ist die Rampe für den generierten Wert vom niedrigen zum hohen Grenzwert und zwar zum gewünschten Intervall. Ist der Inkrementwert negativ, ist die Rampe für den generierten Wert vom hohen zum niedrigen Grenzwert und zwar zum gewünschten Intervall. Die Werte für den niedrigen Grenzwert, den hohen Grenzwert und das Inkrement können entweder als Ganzzahlen oder im Gleitkommaformat angegeben werden.

Unterstützte Datentypen

Byte, Char, Word, Short, DWord, **Long**, Float, Double

Beispiele

RAMP(120, 35, 100, 4)
Es wird ein Wert erstellt, der sich von 35 auf 100 erhöht, inkrementiert um 4, alle 120 Millisekunden.

RAMP(300, 150.75, 200.50, -0.25)
Es wird ein Wert erstellt, der sich von 200.50 auf 150.75 verringert, verringert um 0.25, alle 300 Millisekunden.

Funktion RANDOM

RANDOM(Rate, Low Limit, High Limit)

Die Zufallsfunktion (RANDOM) kann verwendet werden, um ein Element zu erstellen, das sich zufällig innerhalb eines bestimmten numerischen Bereichs ändert. Der niedrige und hohe Grenzwert sollten verwendet werden, um den gewünschten Bereich festzulegen. Die niedrigen oder hohen Grenzwerte können angepasst werden, um einen Versatz auf die generierten Daten anzuwenden.

Unterstützte Datentypen

Byte, Char, Word, Short, DWord, **Long**, Float, Double

Beispiel

RANDOM(30, -20, 75)
Es wird ein Wert erstellt, der sich zufällig innerhalb des Bereichs -20 bis 75 ändert und zwar in einem Intervall von 30 Millisekunden.

Funktion SINE

SINE(Rate, Low Limit, High Limit, Frequency, Phase)

Die Sinusfunktion (SINE) kann verwendet werden, um ein Element zu erstellen, das einer sinusförmigen Wertänderung folgt. Der niedrige und hohe Grenzwert sollten verwendet werden, um den gewünschten Bereich festzulegen. Die niedrigen oder hohen Grenzwerte können angepasst werden, um einen Versatz auf die generierten Daten anzuwenden. Die Eigenschaft “Frequenz“ kann verwendet werden, um die gewünschte Wellenform in Hertz anzugeben. Die maximale effektive Frequenz liegt bei ungefähr 5 Hertz. Der gültige Bereich für die Eigenschaft “Frequenz“ ist 0.001 bis 5 Hertz. Die Eigenschaft “Phase“ kann verwendet werden, um die durch einen bestimmten Winkel generierte Sinusschwingung zu versetzen. Die Phase sollte mit einem Bereich von 0,0 bis 360.0 eingegeben werden. Die Eigenschaft “Rate“ spielt in diesem Fall eine Schlüsselrolle dabei, wie diese Simulationsfunktion funktioniert. Um eine gute sinusförmige Ausgabe von dieser Funktion zu erhalten, muss die Rate mindestens zweimal so schnell wie die gewünschte Frequenz sein. Beispiel: Wenn Benutzer eine Sinusschwingung von 5 Hertz wünschen, die sich mit einer Rate von ungefähr einer 200 Millisekunden ändert, sollte die Eigenschaft “Rate“ auf maximal 100 Millisekunden festgelegt werden. Es wird empfohlen, die die Rate auf entweder 10 oder 20 Millisekunden festzulegen, um die besten

Sinusschwingungsergebnisse zu erzielen. Das gültige Bereich für die Rate einer Sinusfunktion ist 10 bis 1000 Millisekunden.

Unterstützte Datentypen

Float, Double

Beispiel

SINE(10, -40, 40, 2, 0)
Es wird ein sinusförmiger Wert mit einer Frequenz von 2 Hertz erstellt, der im Bereich von -40 bis 40 liegt und keinen Phasenwechsel aufweist.

Funktion USER

USER(Rate, User Value1, User Value2, User Value3, ...User ValueN)

Die Benutzerfunktion (USER) bietet die größte Flexibilität beim Definieren, welche Datentypen die Simulationsfunktion zurückgibt. Im Gegensatz zu den anderen Funktionen, die innerhalb eines angegebenen Bereichs funktionieren, kann die Funktion USER verwendet werden, um einen Satz von numerischen Werten oder Zeichenfolgenwerten anzugeben, die zur angegebenen Rate durchlaufen werden. Die eingegebenen Werte werden verwendet, um den Datentyp dieses Elements zu bestimmen. Beispiel: Wenn ein Wert von 100.45 als einer der Benutzerwerte eingegeben wird, so besteht die Ausgabe des Simulationsobjekts aus Gleitkommadaten. Ist einer der Benutzerwerte “Hello World“, so besteht die Ausgabe des Simulationsobjekts aus Zeichenfolgendaten. Diese Standardauswahlen können durch Angabe des gewünschten Datentyps während der Definition des Elements überschrieben werden.

Hinweis: Werden Benutzerwerte als Zeichenfolgen eingegeben, muss dem Komma innerhalb eines Zeichenfolgenwerts ein umgekehrter Schrägstrich (\) vorangestellt werden.

Unterstützte Datentypen

Bool, Byte, Char, Word, Short, DWord, Long, Float, Double
Hinweis: Die in der Simulationsfunktion USER eingegebenen Werte werden verwendet, um den Standarddatentyp zu bestimmen.

Beispiele

USER(250, Hello, World, this, is, a, test)
Es wird eine Zeichenfolge erstellt, die mit einer Rate von 250 Millisekunden von einem Wort im Text der Sequenz zum nächsten wechselt.

USER(20, 1.25, 100.56, 200.11,75.1)
Es werden Gleitkommas erstellt, die mit einer Rate von 20 Millisekunden von einem Gleitkomma in der Sequenz zum nächsten wechseln.

USER(50, 1,1,0,1,0,1,0,0,1,1,1,0,0,0)
Es werden boolesche Werte generiert, die mit einer Rate von 50 Millisekunden von einem booleschen Status in der Sequenz zum nächsten wechseln. Hiermit können sehr komplexe Bitmuster erstellt werden.

USER(1000, In this case\, I needed to use a \, in my text)
Die umgekehrten Schrägstriche gefolgt von einem Komma (\,) in diesen Textfragmenten werden benötigt, um ein Komma in einen Textwert zu platzieren. Der Text für jeden Wert kann entweder aus einem Satz oder Satzfragment bestehen.

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 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).

Elementstatusdaten konnten nicht geladen werden. Grund: <Grund>.

Fehlertyp:
Warnung

Mögliche Ursache:

1. The driver could not load or save item state data for the specified reason.
2. Corrupt data files.
3. Inadequate disk space.
4. Invalid drive in path.
5. Deleted or renamed data files.

Mögliche Lösung:
Solution depends upon the reason given in the error message. In the case of file corruption or deletion, previous state data is lost.

Elementstatusdaten konnten nicht gespeichert werden. Grund: <Grund>.

Fehlertyp:
Warnung

Mögliche Ursache:

1. The driver could not load or save item state data for the specified reason.
2. Corrupt data files.
3. Inadequate disk space.
4. Invalid drive in path.
5. Deleted or renamed data files.

Mögliche Lösung:
Solution depends upon the reason given in the error message. In the case of file corruption or deletion, previous state data is lost.

Index

1

8

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

W