Simulator Driver

目次

Simulator Driver

ヘルプバージョン 1.036

目次

概要

Simulator DriverはSimulatorデバイスがHMI、SCADA、Historian、MES、ERPや多数のカスタムアプリケーションを含むOPCクライアントアプリケーションに接続するための信頼性の高い手段を提供します。外部デバイスが必要なく、OPCサーバーソフトウェア製品のテストのために提供されます。

設定

サポートされるデバイス

8 ビット
16 ビット

注記: 各デバイスは、1000 の可変長、読み取り/書き込み文字列の場所に加え、10000 のアドレス指定可能な読み取り/書き込みレジスタと定数の場所をサポートします。*詳細については、アドレスの説明を参照してください。*

チャネルとデバイスの最大数

サポートされているチャネルの最大数は 100 です。1 チャネルにつきサポートされているデバイスの最大数は 999 です。

デバイス ID

Simulator デバイスは、1 から 999 までの範囲でデバイス ID を割り当てることができます。同じチャネル内で異なるモデルタイプを使用している場合、一意のデバイス ID が必要です。

ライブデータシミュレーション

ドライバーは、レジスタのデータが整数データ型として読み取られるたびに、レジスタのデータを増加させることにより、ライブデータをシミュレーションします。PLC のようなデバイスの簡単なレジスタベースのメモリをシミュレーションすることに加え、Simulator Driverは 4 つの高度なシミュレーション関数もサポートします。これらの新しいシミュレーション関数には、RAMP、SINE、RANDOM、および USER 定義が含まれます。

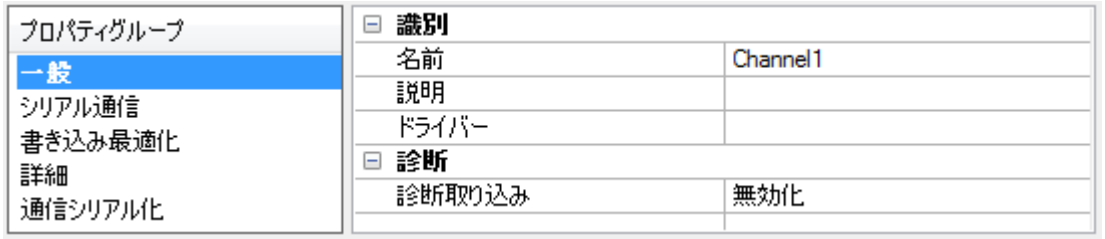
各新規シミュレーションタグは、プログラミング言語にある関数呼び出しのように構築されます。各関数を使用するには、必要なシミュレーションの効果を指定するために、適切なプロパティが適用されている必要があります。RAMP では、変更の速度、下限、上限、および増分値を設定するオプションがあります。SINE では、変更の速度、下限、上限、周波数、および位相を設定するオプションがあります。RANDOM では、変更の速度、下限および上限を設定するオプションがあります。ただし、最も創造的な新規シミュレーション関数は、USER 定義関数です。

USER 定義関数には、同様に変更の速度を指定するオプションがあります。RAMP、RANDOM、および SINE 関数のプリセットシミュレーション出力とは異なり、USER 定義関数は一連のデータを作成するために使用されます。上限または下限の代わりに、USER 定義関数ではコンマ区切りのアイテムが受け入れられます。アイテムのリストには、数値データまたは文字列のデータを使用できます。リストが確立されると、USER 定義関数は、指定された速度でリストの要素を循環します。USER 定義関数は、現実の出力および結果を反映するために使用する複雑なデモを作成できます。

詳細については、シミュレーション関数を参照してください。

チャネルのプロパティ - 一般

このサーバーは、複数の通信ドライバーの同時使用をサポートしています。サーバープロジェクトで使用される各プロトコルおよびドライバーをチャネルと呼びます。サーバープロジェクトは、同じ通信ドライバーまたは一意の通信ドライバーを使用する多数のチャネルから成ります。チャネルは、OPC リンクの基本的な構成要素として機能します。このグループは、識別属性や動作モードなどの一般的なチャネルプロパティを指定するときに使用します。



識別

「名前」: このチャネルのユーザー定義の識別情報。各サーバープロジェクトで、それぞれのチャネル名が一意でなければなりません。名前は最大 256 文字ですが、一部のクライアントアプリケーションでは OPC サーバーのタグ空間をブラウズする際の表示ウィンドウが制限されています。チャネル名は OPC ブラウザ情報の一部です。
予約済み文字の詳細については、サーバーのヘルプで「チャネル、デバイス、タグ、およびタググループに適切な名前を付ける方法」を参照してください。

「説明」: このチャネルに関するユーザー定義の情報。
「説明」などのこれらのプロパティの多くには、システムタグが関連付けられています。

「ドライバー」: このチャネルに選択されているプロトコル/ドライバー。このプロパティでは、チャネル作成時に選択されたデバイスドライバーが示されます。チャネルのプロパティではこの設定を変更することはできません。

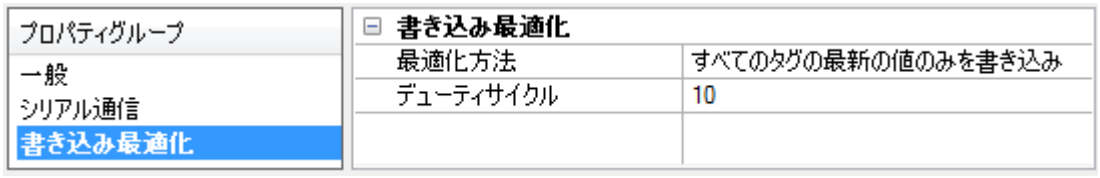
注記: サーバーがオンラインで常時稼働している場合、これらのプロパティをいつでも変更できます。これには、クライアントがデータをサーバーに登録できないようにチャネル名を変更することも含まれます。チャネル名を変更する前にクライアントがサーバーからアイテムをすでに取得している場合、それらのアイテムは影響を受けません。チャネル名が変更された後で、クライアントアプリケーションがそのアイテムを解放し、古いチャネル名を使用して再び取得しようとしても、そのアイテムは取得されません。このことを念頭において、大規模なクライアントアプリケーションを開発した後はプロパティに対する変更を行わないようにします。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。

診断

「診断取り込み」: このオプションが有効な場合、チャネルの診断情報が OPC アプリケーションに取り込まれます。サーバーの診断機能は最小限のオーバーヘッド処理を必要とするので、必要なときにだけ利用し、必要がないときには無効にしておくことをお勧めします。デフォルトでは無効になっています。
注記: ドライバーが診断をサポートしていない場合、このプロパティは無効になります。
詳細については、サーバーのヘルプで「通信診断」を参照してください。

チャネルのプロパティ - 書き込み最適化

OPC サーバーと同様に、デバイスへのデータの書き込みはアプリケーションの最も重要な要素です。サーバーは、クライアントアプリケーションから書き込まれたデータがデバイスに遅延なく届くようにします。このため、サーバーに用意されている最適化プロパティを使用して、特定のニーズを満たしたり、アプリケーションの応答性を高めたりできます。



書き込み最適化

「最適化方法」: 基礎となる通信ドライバーに書き込みデータをどのように渡すかを制御します。以下のオプションがあります。

- **「すべてのタグのすべての値を書き込み」**: このオプションを選択した場合、サーバーはすべての値をコントローラに書き込もうとします。このモードでは、サーバーは書き込み要求を絶えず収集し、サーバーの内部書き込みキューにこれらの要求を追加します。サーバーは書き込みキューを処理し、デバイスにできるだけ早くデータを書き込むことによって、このキューを空にしようとします。このモードでは、クライアントアプリケーションから書き込まれたすべてのデータがターゲットデバイスに送信されます。ターゲットデバイスで書き込み操作の順序または書き込みアイテムのコンテンツが一意に表示される必要がある場合、このモードを選択します。
- **「非 Boolean タグの最新の値のみを書き込み」**: デバイスにデータを実際に送信するのに時間がかかっているために、同じ値への多数の連続書き込みが書き込みキューに累積することがあります。書き込みキューにすでに置かれている書き込み値をサーバーが更新した場合、同じ最終出力値に達するまでに必要な書き込み回数ははるかに少なくなります。このようにして、サーバーのキューに余分な書き込みが累積することがなくなります。ユーザーがスライドスイッチを動かすのをやめると、ほぼ同時にデバイス内の値が正確な値になります。モード名からもわかるように、Boolean 値でない値はサーバーの内部書き込みキュー内で更新され、次の機会にデバイスに送信されます。これによってアプリケーションのパフォーマンスが大幅に向上します。
  注記: このオプションを選択した場合、Boolean 値への書き込みは最適化されません。モーメンタリプッシュボタンなどの Boolean 操作で問題が発生することなく、HMI データの操作を最適化できます。
- **「すべてのタグの最新の値のみを書き込み」**: このオプションを選択した場合、2 つ目の最適化モードの理論がすべてのタグに適用されます。これはアプリケーションが最新の値だけをデバイスに送信する必要がある場合に特に役立ちます。このモードでは、現在書き込みキューに入っているタグを送信する前に更新することによって、すべての書き込みが最適化されます。これがデフォルトのモードです。

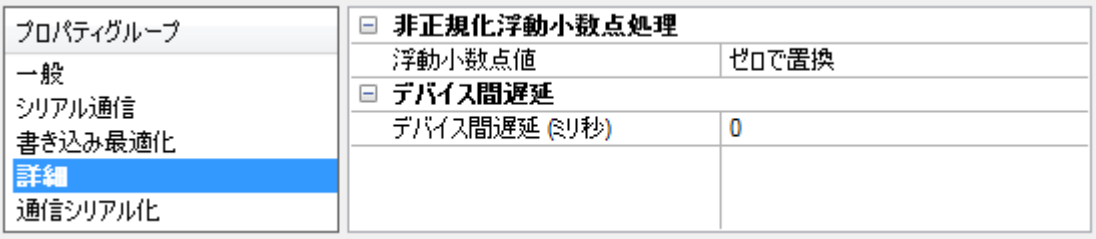
「デューティサイクル」: 読み取り操作に対する書き込み操作の比率を制御するときに使用します。この比率は必ず、読み取り 1 回につき書き込みが 1 から 10 回の間であることが基になっています。デューティサイクルはデフォルトで 10 に設

定されており、1 回の読み取り操作につき 10 回の書き込みが行われます。アプリケーションが多数の連続書き込みを行っている場合でも、読み取りデータを処理する時間が確実に残っている必要があります。これを設定すると、書き込み操作が 1 回行われるたびに読み取り操作が 1 回行われるようになります。実行する書き込み操作がない場合、読み取りが連続処理されます。これにより、連続書き込みを行うアプリケーションが最適化され、データの送受信フローがよりバランスのとれたものとなります。

注記: 本番環境で使用する前に、強化された書き込み最適化機能との互換性が維持されるようにアプリケーションのプロパティを設定することをお勧めします。

チャネルのプロパティ - 詳細

このグループは、チャネルの詳細プロパティを指定するときに使用します。すべてのドライバーがすべてのプロトコルをサポートしているわけではないので、サポートしていないデバイスには詳細グループが表示されません。



「非正規化浮動小数点処理」:「非正規化浮動小数点処理」では、ドライバーによる非正規化 IEEE-754 浮動小数点データの処理方法を指定できます。非正規化値は無限、非数 (NaN)、または非正規化数として定義されます。デフォルトは「ゼロで置換」です。ネイティブの浮動小数点処理が指定されているドライバーはデフォルトで「未修正」になります。オプションの説明は次のとおりです。

- **「ゼロで置換」**: このオプションを選択した場合、ドライバーが非正規化 IEEE-754 浮動小数点値をクライアントに転送する前にゼロで置き換えることができます。
- **「未修正」**: このオプションを選択した場合、ドライバーは IEEE-754 非正規化、正規化、非数、および無限の値を変換または変更せずにクライアントに転送できます。

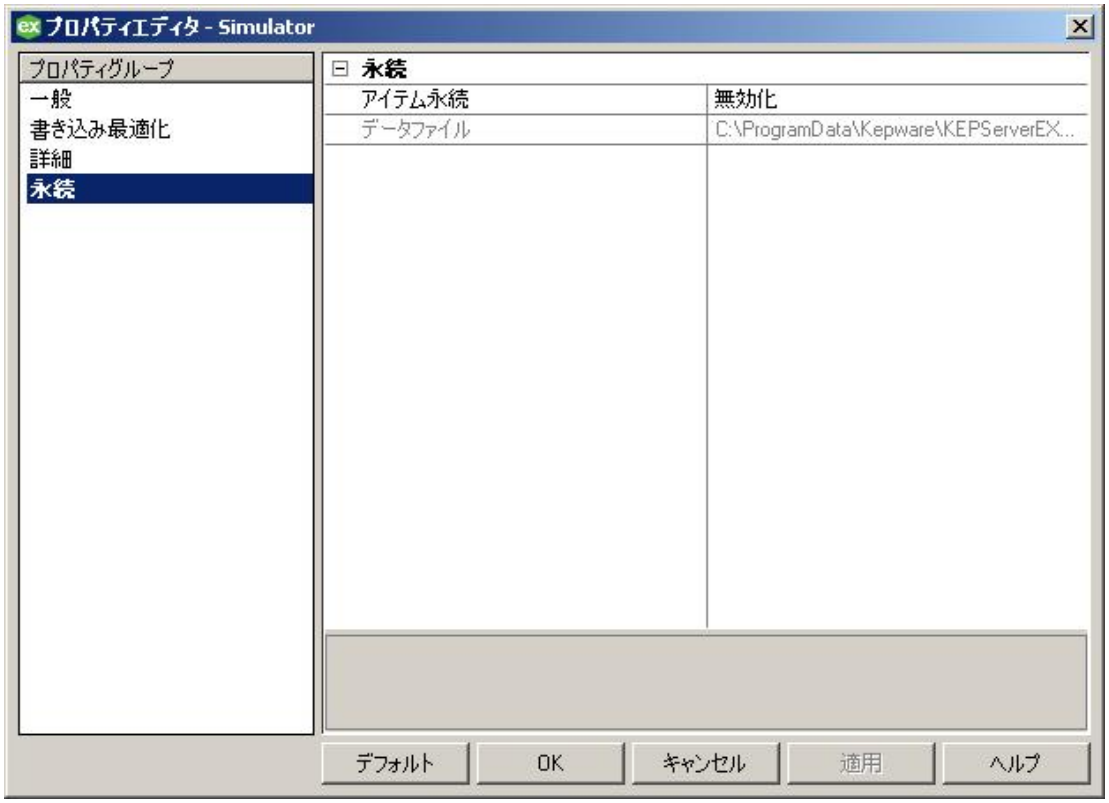
注記: ドライバーが浮動小数点値をサポートしていない場合や、表示されているオプションだけをサポートする場合、このプロパティは無効になります。チャネルの浮動小数点正規化の設定に従って、リアルタイムのドライバータグ (値や配列など) が浮動小数点正規化の対象となります。たとえば、EFM データはこの設定の影響を受けません。

浮動小数点値の詳細については、サーバーのヘルプで「非正規化浮動小数点値を使用する方法」を参照してください。

「デバイス間遅延」: 通信チャネルが同じチャネルの現在のデバイスからデータを受信した後、次のデバイスに新しい要求を送信するまで待機する時間を指定します。ゼロ (0) を指定すると遅延は無効になります。

注記: このプロパティは、一部のドライバー、モデル、および依存する設定では使用できません。

チャネルのプロパティ - 永続



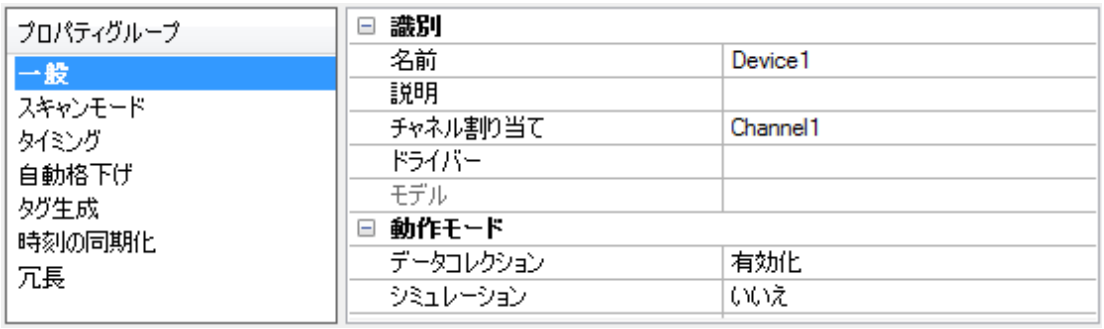
「アイテム永続」:「有効化」を選択して、デバイスが実行間で K (定数)、R (レジスタ)、および S (文字列) アドレスの値を維持するように設定します。シミュレーション関数のアイテム永続は現在サポートされていません。有効にした場合、チャネルに設定した各デバイスの K、R、および S アドレスすべてに保存した値は、サーバープロジェクトが閉じた際 (またはサーバーがシャットダウンした際) にファイルに保存されます。これらの値は、次回サーバープロジェクトが開いた際にファイルから回復します。この機能を使用すると、各チャネルに対して独立したファイルが使用されます。デフォルトでは無効に設定されています。

詳細については、シミュレーション関数を参照してください。

「データファイル」: このチャネル上のデバイスに対し、データの保存場所を指定します。完全修飾パスとファイル名が必要です。ドライバーによってそのパスにファイルとフォルダが作成されますが、ユーザーはこの機能を使用して、各 Simulator チャネルに対して一意のファイルを指定する必要があります。

デバイスのプロパティ - 一般

デバイスは、通信チャネル上の 1 つのターゲットを表します。ドライバーが複数のコントローラをサポートしている場合、ユーザーは各コントローラのデバイス ID を入力する必要があります。



識別

「名前」: このプロパティでは、デバイスの名前を指定します。これは最大 256 文字のユーザー定義の論理名であり、複数のチャネルで使用できます。

注記: わかりやすい名前にすることを一般的にはお勧めしますが、一部の OPC クライアントアプリケーションでは OPC サーバーのタグ空間をブラウズする際の表示ウィンドウが制限されています。デバイス名とチャネル名はブラウズツリー情報の一部にもなります。OPC クライアント内では、チャネル名とデバイス名の組み合わせが "<チャネル名>.<デバイス名>" として表示されます。
詳細については、サーバーのヘルプで「チャネル、デバイス、タグ、およびタググループに適切な名前を付ける方法」を参照してください。

「説明」: このデバイスに関するユーザー定義の情報。
「説明」などのこれらのプロパティの多くには、システムタグが関連付けられています。

「チャネル割り当て」: このデバイスが現在属しているチャネルのユーザー定義の名前。

「ドライバー」: このデバイスに設定されているプロトコルドライバー。このプロパティは、チャネル作成時に選択されたドライバーを示します。チャネルプロパティではこれは無効になっています。

「モデル」: このプロパティでは、この ID に関連付けられるデバイスのタイプを指定します。このドロップダウンメニューの内容は、使用されている通信ドライバーのタイプによって異なります。ドライバーによってサポートされていないモデルは無効になります。通信ドライバーが複数のデバイスモデルをサポートしている場合、デバイスにクライアントアプリケーションが 1 つも接続していない場合にのみモデル選択を変更できます。

注記: 通信ドライバーが複数のモデルをサポートしている場合、ユーザーは物理デバイスに合わせてモデルを選択する必要があります。このドロップダウンメニューにデバイスが表示されない場合、ターゲットデバイスに最も近いモデルを選択します。一部のドライバーは "オープン" と呼ばれるモデル選択をサポートしており、ユーザーはターゲットデバイスの詳細を知らなくても通信できます。詳細については、ドライバーのヘルプドキュメントを参照してください。

「ID」: このプロパティは、デバイスのステーション/ノード/アイデンティティ/アドレスを指定します。入力する ID のタイプは、使用されている通信ドライバーによって異なります。多くのドライバーでは、ID は数値です。数値 ID をサポートするドライバーでは、ユーザーは数値を入力でき、そのフォーマットはアプリケーションのニーズまたは選択した通信ドライバーの特性に合わせて変更できます。ID フォーマットには「10 進数」、「8 進数」、「16 進数」があります。ドライバーがイーサネットベースであるか、通常とは異なるステーションまたはノード名をサポートしている場合、デバイスの TCP/IP アドレスをデバイス ID として使用できます。TCP/IP アドレスはピリオドで区切った 4 つの値から成り、各値の範囲は 0 から 255 です。一部のデバイス ID は文字列ベースです。ドライバーによっては、ID フィールドで追加のプロパティを設定する必要があります。

動作モード

「データコレクション」: このプロパティでは、デバイスのアクティブな状態を制御します。デバイスの通信はデフォルトで有効になっていますが、このプロパティを使用して物理デバイスを無効にできます。デバイスが無効になっている場合、通信は試みられません。クライアントから見た場合、そのデータは無効としてマークされ、書き込み操作は許可されません。このプロパティは、このプロパティまたはデバイスのシステムタグを使用していつでも変更できます。

「シミュレーション」: このオプションは、デバイスをシミュレーションモードにします。このモードでは、ドライバーは物理デバイスとの通信を試みませんが、サーバーは引き続き有効な OPC データを返します。シミュレーションモードではデバイスとの物理的な通信は停止しますが、OPC データは有効なデータとして OPC クライアントに返されます。シミュレーションモードでは、サーバーはすべてのデバイスデータを自己反映的データとして扱います。つまり、シミュレーションモードのデバイスに書き込まれたデータはすべて再び読み取られ、各 OPC アイテムは個別に処理されます。アイテムのメモリマップはグループ更新レートに基づきます。(サーバーが再初期化された場合などに) サーバーがアイテムを除去した場合、そのデータは保存されません。デフォルトは「いいえ」です。

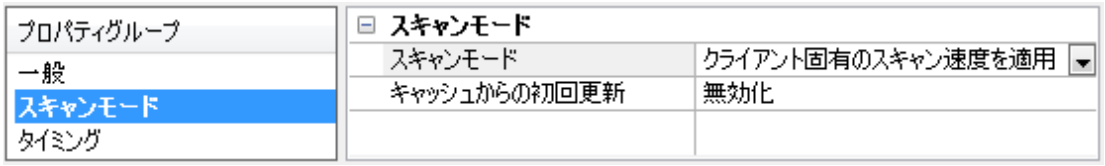
注記:

1. システムタグ (_Simulated) は読み取り専用であり、ランタイム保護のため、書き込みは禁止されています。このシステムタグを使用することで、このプロパティをクライアントからモニターできます。
2. シミュレーションモードでは、アイテムのメモリマップはクライアントの更新レート (OPC クライアントではグループ更新レート、ネイティブおよび DDE インタフェースではスキャン速度) に基づきます。つまり、異なる更新レートで同じアイテムを参照する 2 つのクライアントは異なるデータを返します。

シミュレーションモードはテストとシミュレーションのみを目的としています。本番環境では決して使用しないでください。

デバイスのプロパティ - スキャンモード

「スキャンモード」では、デバイスとの通信を必要とする、サブスクリプション済みクライアントが要求したタグのスキャン速度を指定します。同期および非同期デバイスの読み取りと書き込みは可能なかぎりただちに処理され、「スキャンモード」のプロパティの影響を受けません。



「スキャンモード」: 購読済みクライアントに送信される更新についてデバイス内のタグをどのようにスキャンするかを指定します。オプションの説明は次のとおりです。

- **「クライアント固有のスキャン速度を適用」**: このモードでは、クライアントによって要求されたスキャン速度を使用します。
- **「指定したスキャン速度以下でデータを要求」**: このモードでは、使用する最大スキャン速度を指定します。有効な範囲は 10 から 99999990 ミリ秒です。デフォルトは 1000 ミリ秒です。
  注記: サーバーにアクティブなクライアントがあり、デバイスのアイテム数とスキャン速度の値が増加している場合、変更はただちに有効になります。スキャン速度の値が減少している場合、すべてのクライアントアプリケーションが切断されるまで変更は有効になりません。
- **「すべてのデータを指定したスキャン速度で要求」**: このモードでは、指定した速度で購読済みクライアント用にタグがスキャンされます。有効な範囲は 10 から 99999990 ミリ秒です。デフォルトは 1000 ミリ秒です。
- **「スキャンしない、要求ポールのみ」**: このモードでは、デバイスに属するタグは定期的にポーリングされず、アクティブになった後はアイテムの初期値の読み取りは実行されません。更新のポーリングは、_DemandPoll タグに書き込むか、個々のアイテムについて明示的なデバイス読み取りを実行することによって、クライアントが行います。詳細については、サーバーのヘルプで「デバイス要求ポール」を参照してください。
- **「タグに指定のスキャン速度を適用」**: このモードでは、静的構成のタグプロパティで指定されている速度で静的タグがスキャンされます。動的タグはクライアントが指定したスキャン速度でスキャンされます。

「キャッシュからの初回更新」: このオプションを有効にした場合、サーバーは保存 (キャッシュ) されているデータから、新たにアクティブ化されたタグ参照の初回更新を行います。キャッシュからの更新は、新しいアイテム参照が同じアドレス、スキャン速度、データ型、クライアントアクセス、スケール設定のプロパティを共有している場合にのみ実行できます。1 つ目のクライアント参照についてのみ、初回更新にデバイス読み取りが使用されます。デフォルトでは無効になっており、クライアントがタグ参照をアクティブ化したときにはいつでも、サーバーがデバイスから初期値の読み取りを試みます。

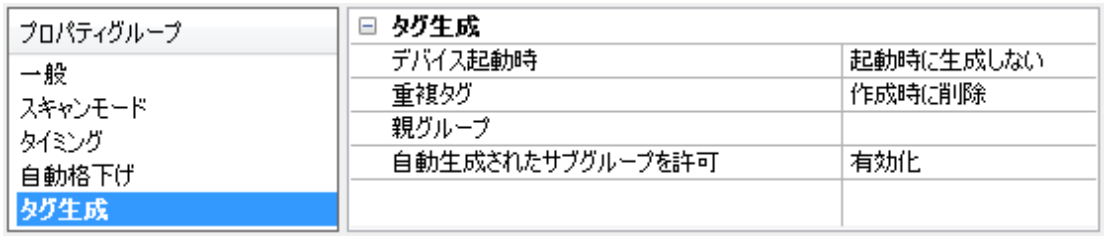
デバイスのプロパティ - タグ生成

自動タグデータベース生成機能によって、アプリケーションの設定がプラグアンドプレイ操作になります。デバイス固有のデータに対応するタグのリストを自動的に構築するよう通信ドライバーを設定できます。これらの自動生成されたタグ (サポートしているドライバーの特性によって異なる) をクライアントからブラウズできます。

ターゲットデバイスが独自のローカルタグデータベースをサポートしている場合、ドライバーはそのデバイスのタグ情報を読み取って、そのデータを使用してサーバー内にタグを生成します。デバイスが名前付きのタグをネイティブにサポートしていない場合、ドライバーはそのドライバー固有の情報に基づいてタグのリストを作成します。この 2 つの条件の例は次のとおりです。

1. データ取得システムが独自のローカルタグデータベースをサポートしている場合、通信ドライバーはデバイスで見つかったタグ名を使用してサーバーのタグを構築します。
2. イーサネット I/O システムが独自の使用可能な I/O モジュールタイプの検出をサポートしている場合、通信ドライバーはイーサネット I/O ラックにプラグイン接続している I/O モジュールのタイプに基づいてサーバー内にタグを自動的に生成します。

注記: 自動タグデータベース生成の動作モードを詳細に設定できます。詳細については、以下のプロパティの説明を参照してください。



「プロパティ変更時」: デバイスが、特定のプロパティが変更された際の自動タグ生成をサポートする場合、**「プロパティ変更時」**オプションが表示されます。これはデフォルトで**「はい」**に設定されていますが、**「いいえ」**に設定してタグ生成を実行する時期を制御できます。この場合、タグ生成を実行するには**「タグを作成」**操作を手動で呼び出す必要があります。

「デバイス起動時」: このプロパティでは、OPC タグを自動的に生成する場合を指定します。オプションの説明は次のとおりです。

- **「起動時に生成しない」**: このオプションを選択した場合、ドライバーは OPC タグをサーバーのタグ空間に追加しません。これはデフォルトの設定です。
- **「起動時に常に生成」**: このオプションを選択した場合、ドライバーはデバイスのタグ情報を評価します。さらに、サーバーが起動するたびに、サーバーのタグ空間にタグを追加します。
- **「最初の起動時に生成」**: このオプションを選択した場合、そのプロジェクトが初めて実行されたときに、ドライバーがデバイスのタグ情報を評価します。さらに、必要に応じて OPC タグをサーバーのタグ空間に追加します。

注記: OPC タグを自動生成するオプションを選択した場合、サーバーのタグ空間に追加されたタグをプロジェクトとともに保存する必要があります。ユーザーは**「ツール」**|**「オプション」**メニューから、自動保存するようプロジェクトを設定できます。

「重複タグ」: 自動タグデータベース生成が有効になっている場合、サーバーが以前に追加したタグや、通信ドライバーが最初に作成した後で追加または修正されたタグを、サーバーがどのように処理するかを設定する必要があります。この設定では、自動生成されてプロジェクト内に現在存在する OPC タグをサーバーがどのように処理するかを制御します。これによって、自動生成されたタグがサーバーに累積することもなくなります。

たとえば、**「起動時に常に生成」**に設定されているサーバーのラックで I/O モジュールを変更した場合、通信ドライバーが新しい I/O モジュールを検出するたびに新しいタグがサーバーに追加されます。古いタグが削除されなかった場合、多数の未使用タグがサーバーのタグ空間内に累積することがあります。以下のオプションがあります。

- **「作成時に削除」**: このオプションを選択した場合、新しいタグが追加される前に、以前にタグ空間に追加されたタグがすべて削除されます。これはデフォルトの設定です。
- **「必要に応じて上書き」**: このオプションを選択した場合、サーバーは通信ドライバーが新しいタグに置き換えているタグだけ除去します。上書きされていないタグはすべてサーバーのタグ空間に残ります。
- **「上書きしない」**: このオプションを選択した場合、サーバーは以前に生成されたタグやサーバーにすでに存在するタグを除去しません。通信ドライバーは完全に新しいタグだけを追加できます。
- **「上書きしない、エラーを記録」**: このオプションには上記のオプションと同じ効果がありますが、タグの上書きが発生した場合にはサーバーのイベントログにエラーメッセージも書き込まれます。

注記: OPC タグの除去は、通信ドライバーによって自動生成されたタグ、および生成されたタグと同じ名前を使用して追加されたタグに影響します。ドライバーによって自動生成されるタグと一致する可能性がある名前を使用してサーバーにタグを追加しないでください。

「親グループ」: このプロパティでは、自動生成されたタグに使用するグループを指定することで、自動生成されたタグと、手動で入力したタグを区別します。グループの名前は最大 256 文字です。この親グループは、自動生成されたすべてのタグが追加されるルートブランチとなります。

「自動生成されたサブグループを許可」: このプロパティでは、自動生成されたタグ用のサブグループをサーバーが自動的に作成するかどうかを制御します。これはデフォルトの設定です。無効になっている場合、サーバーはグループを作成しないで、デバイスのタグをフラットリスト内に生成します。サーバープロジェクトで、生成されたタグには名前としてアドレスの値が付きます。たとえば、生成プロセス中はタグ名は維持されません。

注記: サーバーがタグを生成しているときに、タグに既存のタグと同じ名前が割り当てられた場合、タグ名が重複しないようにするため、番号が自動的に 1 つ増分します。たとえば、生成プロセスによってすでに存在する "AI22" という名前のタグが作成された場合、代わりに "AI23" としてタグが作成されます。

「作成」: 自動生成 OPC タグの作成を開始します。**「タグを作成」**が有効な場合、デバイスの構成が修正されると、ドライバーはタグ変更の可能性についてデバイスを再評価します。システムタグからアクセスできるため、クライアントアプリケーションはタグデータベース作成を開始できます。

注記: 構成がプロジェクトをオフラインで編集する場合、**「タグを作成」**は無効になります。

データ型の説明

アクセス可能な各アドレスには、データ型が割り当てられている必要があります。Simulator デバイスは、次のデータ型をサポートしています。

データ型	説明
BCD	2 バイトパックされた BCD 値の範囲は 0-9999 です。この範囲外の値には動作が定義されていません。
Boolean	1 ビット
Byte	符号なし 8 ビット値 ビット 0 が下位ビット ビット 7 が上位ビット
Char	符号付き 8 ビット値 ビット 0 が下位ビット ビット 6 が上位ビット ビット 7 が符号ビット
Date	浮動小数点 OLE 自動化日付 (VARIANT VT_DATE データ型にマッピング)。
Double*	64 ビット浮動小数点値 ドライバーは最後の 2 つのレジスタを上位 DWord、最初の 2 つのレジスタを下位 DWord とすることで、連続する 4 つのレジスタを倍精度値として解釈します。
Double の例	レジスタ 40001 が Double として指定されている場合、レジスタ 40001 のビット 0 は 64 ビットデータ型のビット 0 になり、レジスタ 40004 のビット 15 は 64 ビットデータ型のビット 63 になります。
DWord	符号なし 32 ビット値 ビット 0 が下位ビット ビット 31 が上位ビット
Float*	32 ビット浮動小数点値 ドライバーは最後のレジスタを上位 Word、最初のレジスタを下位 Word とすることで、連続する 2 つのレジスタを単精度値として解釈します。
Float の例	レジスタ 40001 が Float として指定されている場合、レジスタ 40001 のビット 0 は 32 ビットデータ型のビット 0 になり、レジスタ 40002 のビット 15 は 32 ビットデータ型のビット 31 になります。
LBCD	4 バイトパックされた BCD 値の範囲は 0-99999999 です。この範囲外の値には動作が定義されていません。
Long	符号付き 32 ビット値 ビット 0 が下位ビット ビット 30 が上位ビット ビット 31 が符号ビット
LLong	符号付き 64 ビット値 ビット 0 が下位ビット ビット 62 が上位ビット ビット 63 が符号ビット
QWord	符号なし 64 ビット値 ビット 0 が下位ビット ビット 63 が上位ビット
Short	符号付き 16 ビット値 ビット 0 が下位ビット ビット 14 が上位ビット ビット 15 が符号ビット
文字列	Null 終端 ASCII 文字配列
Word	符号なし 16 ビット値 ビット 0 が下位ビット ビット 15 が上位ビット

*この説明は、64 ビットデータ型では最初の DWord を下位とし、32 ビットデータ型では最初の Word を下位とするデフォルト設定のデータ処理を前提としています。

アドレスの説明

Simulator Driverは、R レジスタ、K レジスタ、および S レジスタの 3 種類のアドレスをサポートします。R および K レジスタは数値データです。S レジスタは、長さが可変の文字列データの場所です。

R レジスタは、Char、Byte、Word、Short、BCD、Long、DWord、LLong、QWord、または LBCD のタイプとして参照される際に、各読み取りで 1 だけ増分することによって、変化するデータをシミュレーションします。これらのデータ型の配列は、各読み取りで 1 だけ増分します。R レジスタが Float または Double 型として参照される場合、値は各読み取りで 1.25 だけ増分します。Float または Double 型の配列は、配列内のアドレスに個々のタグがない限りは増分しません。さらに、各データ型には、増分が発生する範囲があります。Float 型では、範囲は 0 から 32767 です。Double 型では、範囲は 0 から 65535 です。

K および R レジスタには、ゼロの初期値があります。S レジスタには、'文字列データ S*n*' の初期値があります (*n* はレジスタ番号)。

アドレス範囲とデータ型の仕様は、使用中のモデルによって異なります。Simulator Driverは、RAMP、SINE、RANDOM および USER 定義などの新しいシミュレーション関数もサポートします。詳細については、以下のリストからリンクを選択してください。

8 ビットデバイスアドレス
16 ビットデバイスアドレス

8 ビットデバイスアドレス

8 ビットデバイスのメモリ構成は、0 から 9999 までの番号が付けられた Byte 位置のブロックとしてシミュレーションされます。各 Byte は、ブロックの始点からのオフセットとしてアドレス指定できます。各フォーマットのデフォルトのデータ型を**太字**で示しています。

デバイスタイプ	範囲	データ型	アクセス
レジスタ	R0000-R9999 R0000-R9998 R0000-R9996 R0000-R9992	**Byte**、Char Word、Short、BCD、 DWord、Long、LBCD、Float、 LLong、QWord、Double、 Date、Boolean	読み取り/書き込み
定数	K0000-K9999 K0000-K9998 K0000-K9996 K0000-K9992	**Byte**、Char Word、Short、BCD DWord、Long、LBCD、Float LLong、QWord、Double、 Date、Boolean	読み取り/書き込み
ビット	R0000.0-R9999.7 K0000.0-K9999.7	**Boolean**	読み取り/書き込み
文字列	S000-S999	**String**	読み取り/書き込み

注記:

1. ビットレベルの Boolean を除くすべてのデータ型は、[r] または [r][c] 表記をアドレスに追加することによって、配列をサポートします。
2. データ型に対して指定したアドレスは、フルサイズのデータ型を許可する必要があります。これは、ユーザーがデータの範囲外には書き込めないことを意味します。

関連項目: *シミュレーション関数*

16 ビットデバイスアドレス

16 ビットデバイスのメモリ構成は、0 から 9999 までの番号が付けられた Word 位置のブロックとしてシミュレーションされます。各 Word は、ブロックの始点からのオフセットとしてアドレス指定できます。各フォーマットのデフォルトのデータ型を**太字**で示しています。

デバイスタイプ	範囲	データ型	アクセス
レジスタ	R0000-R9999 R0000-R9998 R0000-R9996	**Word**、Short、BCD DWord、Long、LBCD、Float LLong、QWord、Double、 Date、Boolean	読み取り/書き込み
定数	K0000-K9999 K0000-K9998 K0000-K9996	**Word**、Short、BCD DWord、Long、LBCD、Float LLong、QWord、Double、 Date、Boolean	読み取り/書き込み
ビット	R0000.00-R9999.15 K0000.00-K9999.15	**Boolean**	読み取り/書き込み
文字列	S000-S999	**String**	読み取り/書き込み

注記:

1. ビットレベルの Boolean を除くすべてのデータ型は、[r] または [r][c] 表記をアドレスに追加することによって、配列をサポートします。
2. データ型に対して指定したアドレスは、フルサイズのデータ型を許可する必要があります。これは、ユーザーがデータの範囲外には書き込めないことを意味します。

関連項目: *シミュレーション関数*

シミュレーション関数

シミュレーション関数を使用して、現実のデータソースを大量に模倣する OPC アイテムを作成できます。各シミュレーション関数によって出力は異なりますが、**速度**、**下限**、および**上限**など多数の共通プロパティがあります。速度 (「変更の速度」とも呼ばれます) は、シミュレーション値の状態が変更される頻度の指定に使用されます。速度の値は、10 から 3600000 の範囲の値が秒単位で入力されます。この変更の速度は、クライアントアプリケーションがデータを読み取る頻度からは完全に独立しています。PLC のように、シミュレーション関数はバックグラウンドで常に実行されています。

下限と上限

下限と上限のプロパティは、シミュレーション関数によって生成されたデータの範囲またはスパンの指定に使用できます。下限と上限のプロパティを使用することによって、ユーザーは、データのスパンを調整するだけでオフセットのシミュレーション値を作成できます。制限をサポートするシミュレーション関数では、入力した範囲のフォーマットによって、シミュレーション値のデータ型が決定されます。たとえば、RAMP タグが下限 75 および上限 3000 で入力された場合、結果の OPC アイテムは Long データ型と見なされます。同じ RAMP タグが下限 75.123 および上限 3000.567 で入力された場合、結果の OPC アイテムは Float データ型と見なされます。これらの 2 つの例ではデフォルトのデータフォーマットは Float または Long のいずれかでしたが、任意のシミュレーション関数に対する出力フォーマットとして、使用可能な任意のデータ型を選択できます。ただし、下限と上限によって指定された範囲は、必要なデータ型の選択に一致する必要があります。

上記のレジスタベースのアドレスとは異なり、入力できるシミュレーション関数の数に制限はありません。一意のプロパティを持つシミュレーション関数は、新規のシミュレーション値と見なされます。したがって、クライアントアプリケーションの複数の場所で同じ値を読み取る目的でアドレス指定のシミュレーション関数を作成する際には、それぞれの場合についてシミュレーション関数を同じ方法で入力することが重要です。

シミュレーション関数は読み取り専用オブジェクト

シミュレーション関数は読み取り専用オブジェクトです。クライアントアプリケーションがシミュレーション関数からデータの読み取りを開始すると、クライアントアプリケーションによってアイテムが削除されるまで、関数はデータの生成を続行します。浮動小数点のプロパティを入力する際は、シミュレーション関数には指数形式での数値を入力できません。

関数の定義

RAMP 関数
RANDOM 関数
SINE 関数
USER 関数

RAMP 関数

RAMP(速度, 下限, 上限, 増分)

RAMP 関数を使用して、数値の範囲内で増分または減分する値を作成できます。必要な範囲を設定するには下限と上限を使用する必要があります。下限と上限を調整して、生成されたデータへのオフセットを適用できます。増分値は、正の値または負の値のどちらにも設定できます。増分値が正の値の場合、生成された値は、下限から上限へ指定した速度で傾斜します。増分値が負の値の場合、生成された値は、上限から下限へ指定した速度で傾斜します。下限、上限、および増分の値は、整数または浮動小数点の形式で入力できます。

サポートされるデータ型

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

例

RAMP(120, 35, 100, 4)
この関数により、120 ミリ秒ごとに 4 ずつ増分しながら 35 から 100 まで増加する値が作成されます。

RAMP(300, 150.75, 200.50, -0.25)
この関数により、300 ミリ秒ごとに 0.25 ずつ減分しながら 200.50 から 150.75 まで減少する値が作成されます。

RANDOM 関数

RANDOM(速度, 下限, 上限)

RANDOM 関数を使用して、特定の数値の範囲内で不規則に変化するアイテムを作成できます。必要な範囲を設定するには下限と上限を使用する必要があります。下限と上限を調整して、生成されたデータへのオフセットを適用できます。

サポートされるデータ型

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

例

RANDOM(30, -20, 75)
これにより、30 ミリ秒の速度で -20 から 75 の間で不規則に変化する値が作成されます。

SINE 関数

SINE(速度, 下限, 上限, 周波数, 位相)

SINE 関数は、正弦関数の値の変更に従うアイテムの作成に使用できます。必要な範囲を設定するには下限と上限を使用する必要があります。下限と上限を調整して、生成されたデータへのオフセットを適用できます。周波数プロパティを使用して、必要な波形をヘルツ単位で指定できます。有効最大周波数は約 5 ヘルツです。周波数プロパティの有効な範囲は 0.001 ヘルツから 5 ヘルツです。位相プロパティを使用して、特定の角度によって生成された正弦波をオフセットできます。位相は、0.0 から 360.0 までの範囲内で入力する必要があります。この場合、速度プロパティは、このシミュレーション関数の動作方法において重要な役割を果たします。この関数から正弦関数の良好な出力を得るには、速度は必要な周波数より少なくとも 2 倍速くなければなりません。たとえば、ユーザーが、約 200 ミリ秒の速度で変化する 5 ヘルツの正弦波を要求している場合、速度プロパティは最大で 100 ミリ秒に設定する必要があります。正弦波で最大の結果を得るには、速度を 10 または 20 ミリ秒に設定することをお勧めします。SINE 関数の速度の有効な範囲は 10-1000 ミリ秒です。

サポートされるデータ型

Float、Double

例

SINE(10, -40, 40, 2, 0)
これにより、-40 から 40 の範囲内で位相シフトのない周波数 2 ヘルツの正弦関数の値が作成されます。

USER 関数

USER(速度, ユーザー値 1, ユーザー値 2, ユーザー値 3, ...ユーザー値 N)

USER 関数は、シミュレーション関数によって返されるデータ型の定義において、最も柔軟性が高い関数です。指定した範囲で動作するその他の関数とは異なり、USER 関数は、指定した速度で周期的に繰り返す一連の数値または文字列値を指定するために使用できます。入力した値は、このアイテムのデータ型の決定に使用されます。たとえば、ユーザー値の 1 つとして 100.45 の値が入力された場合、シミュレーションオブジェクトの出力は浮動小数点のデータと見なされま

す。入力されたユーザー値の 1 つが "Hello World" であった場合、シミュレーションオブジェクトの出力は文字列データと見なされます。これらのデフォルトの選択は、アイテムが定義されたときに目的のデータ型を指定することによってオーバーライドできます。

注記: ユーザー値を文字列として入力する際、その前にバックスラッシュ (円記号) "\," を付けることによって文字列値内にコンマを入力できます。

サポートされるデータ型

Bool、Byte、Char、Word、Short、DWord、Long、Float、Double

注記: USER シミュレーション関数に入力された値は、デフォルトのデータ型の決定に使用されます。

例

USER(250, Hello, World, this, is, a, test)

これにより、シーケンス内の 1 つの単語から次の単語に 250 ミリ秒の速度で代わる String データ型の値が作成されます。

USER(20, 1.25, 100.56, 200.11,75.1)

これにより、シーケンス内の 1 つの浮動小数点の値から次の浮動小数点の値に 20 ミリ秒の速度で代わる Float データ型の値が作成されます。

USER(50, 1,1,0,1,0,1,0,0,1,1,1,0,0,0)

これにより、シーケンス内の 1 つの Boolean 状態から次の Boolean 状態に 50 ミリ秒の速度で代わる Boolean 型の値が生成されます。これにより、非常に複雑なビットのパターンを作成できます。

USER(1000, In this case\, , I needed to use a \, in , my text)

これらの文字列内の "\," は、テキスト値内にコンマを配置するために必要でした。さらに、各値のテキストには、必要に応じて文または文の断片を指定できます。

イベントログメッセージ

次の情報は、メインユーザーインタフェースの「イベントログ」枠に記録されたメッセージに関するものです。「イベントログ」詳細ビューのフィルタと並べ替えについては、サーバーのヘルプを参照してください。サーバーのヘルプには共通メッセージが多数含まれているので、これらも参照してください。通常は、可能な場合、メッセージのタイプ (情報、警告) とトラブルシューティングに関する情報が提供されています。

アイテム状態データをロードできませんでした。理由: <理由>。

エラータイプ:
警告

考えられる原因:

1. 指定された理由により、ドライバーはアイテム状態データをロードできませんでした。
2. データファイルが破損しています。
3. ディスク容量が不十分です。
4. パス内のドライブが無効です。
5. データファイルが削除または名前変更されています。

解決策:
エラーメッセージ内の理由によって解決策が異なります。ファイルの破損または削除の場合、前の状態のデータは失われます。

アイテム状態データを保存できませんでした。理由: <理由>。

エラータイプ:
警告

考えられる原因:

1. 指定された理由により、ドライバーはアイテム状態データを保存できませんでした。
2. データファイルが破損しています。
3. ディスク容量が不十分です。
4. パス内のドライブが無効です。
5. データファイルが削除または名前変更されています。

解決策:
エラーメッセージ内の理由によって解決策が異なります。ファイルの破損または削除の場合、前の状態のデータは失われます。

索引

Q

R

S

U

W

あ

い

き

く

さ

し

す

た

ち

て

と

ふ

へ

も

れ

毘

梱

暠

任

凪

択

酳

暘

齏

襖

觚

璿

覦

詃

雰

擷

厭