

Allen-Bradley ControlLogix Unsolicited ド ライバー

© 2018 PTC Inc. All Rights Reserved.

目次

Allen-Bradley ControlLogix Unsolicited ドライバー	1
目次	2
Allen-Bradley ControlLogix Unsolicited ドライバー	4
概要	5
設定	6
チャンネル設定	6
チャンネルのプロパティ - 一般	6
チャンネルのプロパティ - イーサネット通信	7
チャンネルのプロパティ - 書き込み最適化	7
チャンネルのプロパティ - 詳細	8
チャンネルのプロパティ - EtherNet/IP モジュール	9
マスターデバイスの構成	10
デバイスの設定	10
デバイスのプロパティ - 一般	10
デバイスプロパティ - スキャンモード	12
デバイスのプロパティ - タグ生成	12
デバイスのプロパティ - コントローラモジュール	14
デバイスのプロパティ - ネイティブタグデータベース	15
タグ階層	16
ネイティブタグデータベースの CSV のインポート	18
デバイスのプロパティ - オプション	19
データ型の説明	21
アドレスの説明	22
アドレスのフォーマット	22
タグの有効範囲	23
詳細なアドレス指定	24
エラーコード	27
EtherNet/IP カプセル化 エラーコード	27
CIP エラーコード	27
0x01 拡張エラーコード	27
0xFF 拡張エラーコード	28
イベントログメッセージ	29
ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。 OS エラー = '<エラー>'。	29
ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。一般読み取りエラー。	29
ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルエンコーディングはサポートされていません。	29
ネイティブタグデータベースのインポート中にエラーが発生しました。	29
ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が認識されません。 認識されないフィールド名 = '<フィールド名>'。	29
ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が重複しています。 重複して30	

いるフィールド名 = '<フィールド>'。	
ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが見つかりません。	30
ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが不完全です。	30
非送信請求 Logix サーバーを起動できませんでした。	30
タグアドレスが無効です。ネイティブタグはインポートされません。 無効なアドレス = '<アドレス>'。	31
2 つのチャンネルが同じネットワークアダプタ IP と TCP ポートを使用するよう構成されています。各チャンネルが一意のローカル IP とポートにバインドされている必要があります。 1 つ目のチャンネル = '<チャンネル>'、2 つ目のチャンネル = '<channel>'。	31
2 つのデバイスが EtherNet/IP モジュールからの同じパスを使用するよう構成されています。各チャンネルが EtherNet/IP モジュールからの一意のパスを持つ必要があります。 1 つ目のデバイス = '<アドレス>'、2 つ目のデバイス = '<address>'。	31
タグアドレスが無効です。重複するタグアドレスは許可されません。 無効なアドレス = '<アドレス>'。	31
メモリをタグに割り当てることができませんでした。 タグアドレス = '<アドレス>'。	31
ネイティブタグが無効です。個々のタグサイズは 128 KB に制限されています。 タグアドレス = '<アドレス>'。	32
ネイティブタグのインポート中にエラーが発生しました。データベースタグデータの合計サイズは 128 KB に制限されています。 タグアドレス = '<アドレス>'。	32
指定された TCP/IP ポートは範囲外です。デフォルトポートを使用します。 有効な範囲 = <数値> ~ <数値>、デフォルトポート = <数値>。	32
このチャンネル内の別のデバイスがローカル CPU としてすでに登録されています。 デバイス = <デバイス>。	32
ネイティブタグがインポートされました。 タグの数 = <数>、タグデータベースのパス = '<パス>'。	32
タグが生成されました。 タグの数 = <数>、タグ階層のモード = '<モード>'。	32
リソースが不足しているため自動タグ生成を実行できませんでした。	32
索引	33

Allen-Bradley ControlLogix Unsolicited ドライバー

ヘルプバージョン 1.022

目次

概要

Allen-Bradley ControlLogix Unsolicited ドライバーとは

チャンネル設定

チャンネルに一意の TCP/IP ポートを指定する方法

デバイスの設定

このドライバーを使用するためにデバイスを構成する方法

データ型の説明

このドライバーでサポートされるデータ型

アドレスの説明

Allen-Bradley ControlLogix デバイスでタグのアドレスを指定する方法

自動タグデータベース生成

Allen-Bradley ControlLogix Unsolicited ドライバー 用にタグを設定する方法

エラーコード

Allen-Bradley ControlLogix のエラーコード

イベントログメッセージ

ドライバーで生成されるメッセージ

概要

Allen-Bradley ControlLogix Unsolicited ドライバー は Allen-Bradley ControlLogix PLC が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。これは 1 つの EtherNet/IP モジュールと最大 16 個の ControlLogix CPU から成る ControlLogix 5000 シリーズのラックをシミュレートします。ControlLogix 5000 シリーズの PLC は、MSG ラダー命令を使用してこのドライバーに対して CIP データテーブル読み取り/書き込みを実行するよう設定できます。

● このドライバーと通信するように ControlLogix 5000 シリーズの PLC を設定する方法については、Rockwell/Allen Bradley のドキュメントを参照してください。

設定

Allen-Bradley ControlLogix Unsolicited ドライバー は 1 つの Ethernet/IP モジュールが格納されているシミュレーション対象の ControlLogix 5000 シリーズラックとして機能します。このラックには最大 16 個の ControlLogix CPU を格納でき、その 1 つは EtherNet/IP モジュールに内蔵され (ローカル)、最大 15 の各 CPU モジュールがあります (EtherNet/IP モジュールに対してリモート)。シミュレーション対象の EtherNet/IP モジュールにはいつでも最大 256 台のデバイスが接続できます。

サポートされるデバイス

CIP データテーブル読み取り/書き込み MSG 命令をサポートし、ファームウェアバージョン 16 以上で動作する、すべての ControlLogix 5000 シリーズ PLC。接続および非接続 CIP データテーブルの読み取り/書き込みがサポートされています。

通信プロトコル

EtherNet/IP

PLC 構成

ネットワーク上のデバイスは、ドライバーに CIP データテーブル読み取り/書き込み MSG 命令を送信し、返されたデータを処理するようプログラミングされている必要があります。

● MSG 命令の設定方法については、Allen-Bradley のドキュメント Programming Messages In a ControlLogix System を参照してください。

ソケット

最大 256 の受信接続が同時に提供されます。接続は発信元が閉じるまで開いたままとなります。

● 関連項目:

[チャンネル設定](#)
[デバイスの設定](#)

チャンネル設定

チャンネル設定には次のプロパティグループの設定が含まれています。

[一般](#)
[イーサネット通信](#)
[書き込み最適化](#)
[詳細](#)
[EtherNet/IP モジュール](#)

チャンネルのプロパティ - 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ンネルと呼びます。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ンネルから成ります。チャンネルは、OPC リnk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ンネルプロパティを指定するときに使用します。

プロパティグループ	識別	
一般	名前	Channel1
シリアル通信	説明	
書き込み最適化	ドライバー	
詳細	診断	
通信シリアル化	診断取り込み	無効化

識別

「名前」: このチャン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、それぞれのチャン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。チャンネル名は 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「チャンネル、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」: このチャン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」: このチャン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、チャン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。チャン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ンネル名を変更することも含まれます。チャン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。チャン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、古いチャン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、チャンネルの診断情報が 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。

チャンネルのプロパティ - イーサネット通信

イーサネット通信を使用してデバイスと通信できます。

プロパティグループ	☐ イーサネット設定	
一般	ネットワークアダプタ	デフォルト
イーサネット通信		

イーサネット設定

「ネットワークアダプタ」: バインドするネットワークアダプタを指定します。「デフォルト」を選択した場合、オペレーティングシステムはデフォルトのアダプタを選択します。

チャンネルのプロパティ - 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。

プロパティグループ	☐ 書き込み最適化	
一般	最適化方法	すべてのタグの最新の値のみを書き込み
シリアル通信	デューティサイクル	10
書き込み最適化		

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ています。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的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。

チャンネルのプロパティ - 詳細

このグループは、チャン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。

プロパティグループ	<input type="checkbox"/> 非正規化浮動小数点処理	
一般	浮動小数点値	ゼロで置換
シリアル通信	<input type="checkbox"/> デバイス間遅延	
書き込み最適化	デバイス間遅延 (ミリ秒)	0
詳細		
通信シリアル化		

「**非正規化浮動小数点処理**」: 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。「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。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。

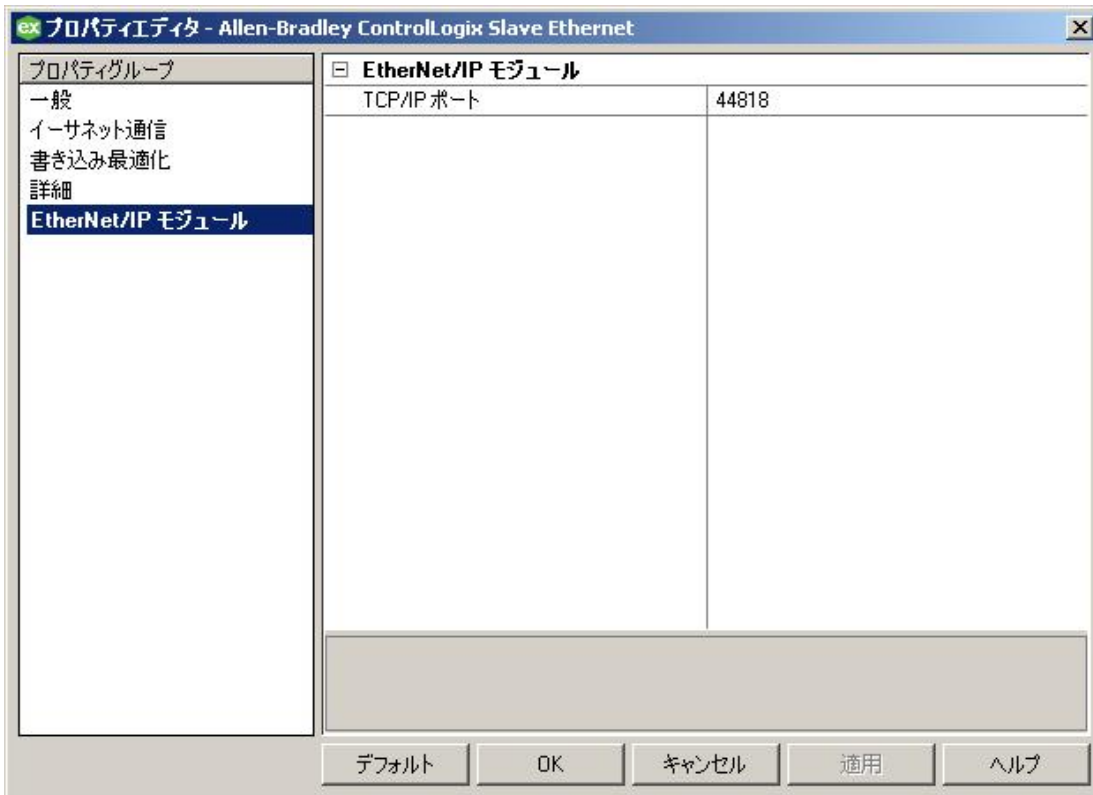
● **注記**: ドライバーが浮動小数点値をサポートしていない場合や、表示されているオプションだけをサポートする場合、このプロパティは無効になります。チャンネルの浮動小数点正規化の設定に従って、リアルタイムのドライバータグ (値や配列など) が浮動小数点正規化の対象となります。たとえば、EFM データはこの設定の影響を受けません。

● 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**」: 通信チャンネルが同じチャン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。

チャンネルのプロパティ - EtherNet/IP モジュール



「TCP/IP ポート」: イーサネット/IP モジュールに一意の通信チャネルを提供する TCP/IP または UDP ポートを指定します。有効な範囲は 1 から 65535 です。デフォルトは 44818 です。

- **注記:** Allen-Bradley ControlLogix Unsolicited ドライバー では現在のところチャネル数が 1 に制限されています。ネットワークアダプタとポートがホストマシン上の別のアプリケーションと競合している場合、ドライバーは受信 EtherNet/IP 接続を受け付けることができません。
- 詳細については、[イベントログメッセージ](#)を参照してください。

マスターデバイスの構成

Allen-Bradley ControlLogix PLC は MSG ラダー命令を使用してこのドライバーに CIP データテーブル読み取り/書き込みメッセージを発行するようプログラミングされている必要があります。このドライバーの構成を表す IP アドレス、スロット番号、オプションのポートから成るルーティングパスを使用する必要があります。MSG ラダー命令の詳細については、Rockwell/Allen-Bradley PLC のプログラミングドキュメントを参照してください。特定のスレーブデバイスに関連付けられているルーティングパスが「デバイスのプロパティ」の下の「コントローラモジュール」に示されています。詳細については、[デバイスのプロパティ - コントローラモジュール](#)を参照してください。

サポートされるサービス

フラグメント化されていない読み取り
フラグメント化された読み取り
フラグメント化されていない書き込み
フラグメント化された書き込み
読み取り修正/書き込み

● **注記:** ControlLogix の MSG ラダー命令は、要求のサイズに基づいて、フラグメントサービスと非フラグメントサービスのどちらを使用するかを自動的に決定します。これはユーザーが設定可能なオプションではありません。

サポートされる Logix のデータ型

BOOL
DWORD (BOOL 配列)
SINT
INT
DINT
LINT
REAL

エラーコード

Allen-Bradley ControlLogix Unsolicited ドライバー は受信した適切なフォーマットのメッセージすべてに回答します。要求を完了できない場合、ゼロ以外のエラーステータスとオプションの拡張エラーステータスが MESSAGE 構造体の ERR タグと EXERR タグに含まれている応答メッセージが返されます。これらのエラーを処理するラダープログラムを記述する必要があります。

● **注記:** マスターデバイスに返される可能性があるエラーコードの詳細については、[エラーコード](#)を参照してください。

● **注記:** このドライバーは上記の Logix アトミックデータ型リストについて CIP データテーブルの読み取り/書き込みをサポートしています。構造体データ型はサポートされていませんが、MSG ラダー命令を使用して、構造体データ型内の個々の Logix アトミックデータ型の読み取り/書き込みを行えます。たとえば、タグ "MyString @ STRING" をドライバーに書き込む必要がある場合、"MyString.DATA" と "MyString.LEN" それぞれに CIP データテーブルの読み取りを実行する必要があります。

デバイスの設定

デバイスの設定には次のプロパティグループの設定が含まれています。

[一般](#)

[スキャンモード](#)

[自動格下げ](#)

[コントローラモジュール](#)

[ネイティブタグデータベース](#)

[オプション](#)

デバイスのプロパティ - 一般



識別

「名前」: このデバイスのユーザー定義の識別情報。

「説明」: このデバイスに関するユーザー定義の情報。

「チャンネル割り当て」: このデバイスが現在属しているチャン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。

「モデル」: このデバイスのバージョン。

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。

- 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。

プロパティグループ	☐ スキャンモード	
一般	スキャンモード	クライアント固有のスキャン速度を適用 ▼
スキャンモード	キャッシュからの初回更新	無効化
タイミング		

「スキャンモード」: 購読している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。

●一部のデバイスやドライバーは自動タグデータベース生成のフル機能をサポートしていません。また、すべてのデバイスやドライバーが同じデータ型をサポートするわけではありません。詳細については、データ型の説明を参照するか、各ドライバーがサポートするデータ型のリストを参照してください。

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。

- 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。

プロパティグループ	☐ タグ生成	
一般	デバイス起動時	起動時に生成しない
スキャンモード	重複タグ	作成時に削除
タイミング	親グループ	
自動格下げ	自動生成されたサブグループを許可	有効化
タグ生成		

「**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削除します。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

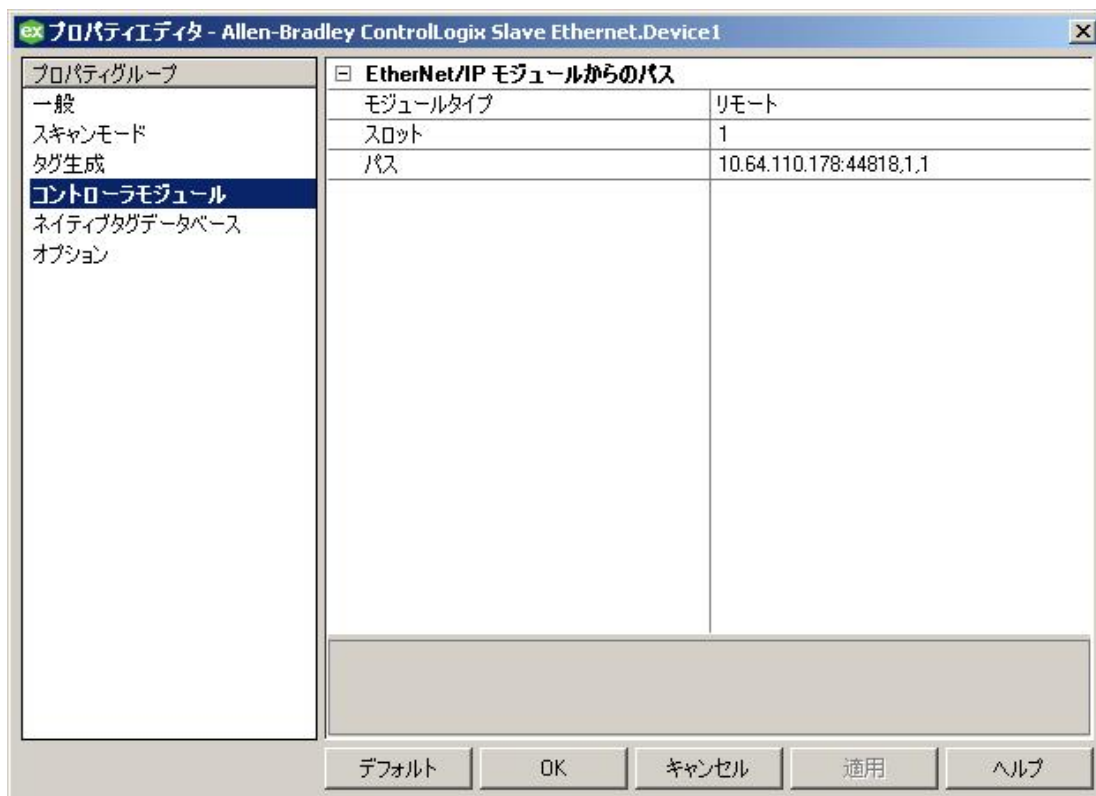
いで、デバイスのタグをフラットリスト内に生成します。サーバープロジェクトで、生成されたタグには名前としてアドレスの値が付きます。たとえば、生成プロセス中はタグ名は維持されません。

● **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。

デバイスのプロパティ - コントローラモジュール



「モジュールタイプ」: デバイスがローカル (シミュレーション対象の EtherNet/IP モジュールの一部) かリモート (EtherNet/IP ルーティング用のスロット番号が必要) かを確認します。1つのローカル CPU と最大 15 個のリモート CPU が存在します。

- 「ローカル」: ローカルの場合、コントローラモジュールはシミュレーション対象の EtherNet/IP モジュールにローカルな CPU として扱われます。ローカルコントローラモジュールはチャンネルにつき 1 つだけ存在します。デフォルトで有効になっています。
- 「リモート」: リモートの場合、コントローラモジュールはシミュレーション対象の EtherNet/IP モジュールとは別の CPU として扱われます。チャンネルにつき最大 15 個のリモートコントローラモジュールが存在します。このオプションを有効にした場合、スロットも指定する必要があります。デフォルトでは無効になっています。

● **注記:**

1. ControlLogix 5000 シリーズのコントローラを表すように各スレーブデバイスを設定する必要があります。
2. 別のデバイスがすでにローカルとして設定されている場合、これは無効になりリモート (1) に設定されます。

「**スロット**」: このプロパティはコントローラモジュールへのルーティングパスの一部です。設定されているチャンネルデバイスで現在使用可能なスロットだけが含まれています。新しいスロットを指定した場合、以前のスロットは別のデバイスで使用可能になります。

「**パス**」: このプロパティはマスターデバイスからコントローラモジュールへのルーティングパスを表します。これは PLC でのマスターデバイスの構成時に使用されます。

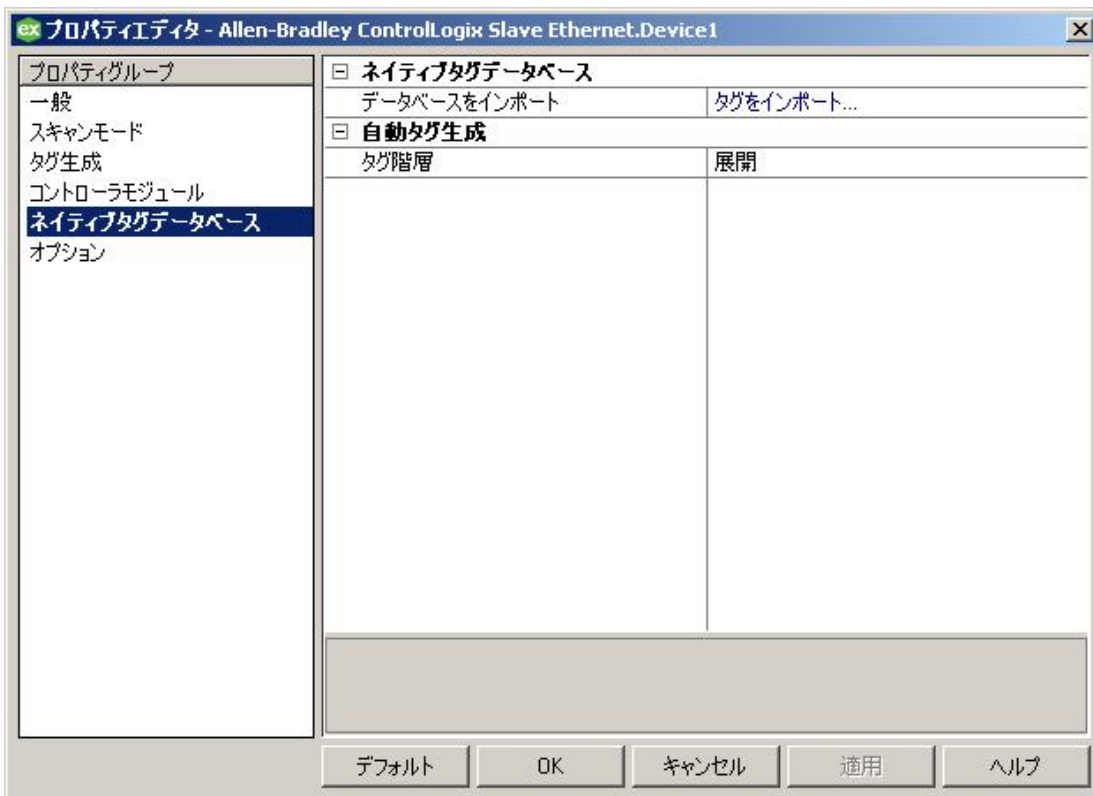
● マスターデバイスの構成の詳細については、[マスターデバイスの構成](#)を参照してください。

デバイスのプロパティ - ネイティブタグデータベース

ネイティブタグデータベースに対応するサーバータグのリストを自動生成するよう Allen-Bradley ControlLogix Unsolicited ドライバーを設定できます。ネイティブタグデータベースタグは定義済みの Logix アトミック型である必要がありますが、構造体型の一部であっても構いません。

ドライバーはネイティブタグデータベースで定義されている各アトミックタグにサーバータグを生成します。配列タイプの場合、配列の各要素にサーバータグが定義されます。配列タグではインポートされるタグの数およびサーバーで使用可能なタグの数が急速に増えることがあります。自動生成されたタグには必ず読み取り書き込みのクライアントアクセスが設定されます。

● 詳細については、[ネイティブタグデータベースの CSV のインポート](#)を参照してください。



ネイティブタグデータベース

「ネイティブタグデータベース」ダイアログでは、自動生成されるサーバータグに加え、シミュレーション対象のコントローラモジュールが表すデータを収集します。データベースはコンマ区切り値 (CSV) ファイルとしてインポートされます。

● 詳細については、[ネイティブタグデータベースの CSV のインポート](#)を参照してください。

「**データベースをインポート**」: 「[タグのインポート...](#)」をクリックし、ネイティブタグデータベースのインポートに使用する CSV ファイルを見つけて選択します。サーバーとのアクティブなクライアント接続がある場合にはネイティブタグをインポートできません。

● **注記**: ネイティブタグデータベースがインポートされた後は、サーバーのプロジェクトファイルを使用して管理されます。リモート配備では CSV ファイルは必要ありません。詳細については、「[自動タグデータベース生成](#)」を参照してください。

「自動タグ生成」

「タグ階層」: タグ表示の「ツリー」構成を「展開」または「圧縮」のどちらにするかを選択します。デフォルトは「展開」です。

- 「展開」: 生成されたクライアントタグは RSLogix と同様にグループ化され、(タグアドレス内のピリオドの後ろにある) セグメント、構造体、サブ構造体、配列ごとにグループが作成されます。
- 「圧縮」: 生成されたクライアントタグはタグのアドレス指定と同様にグループ化され、ピリオドの前のセグメントごとにグループが作成されます。

● インポートに必要なネイティブタグデータベースの CSV フォーマットの詳細については、[ネイティブタグデータベースの CSV のインポート](#)を参照してください。

タグ階層

自動生成されたサーバータグの階層は展開または圧縮のいずれかになります。デフォルトは「展開」モードです。

展開モード

展開モードでは、自動生成されたサーバータグは、RSLogix 5000 内のタグ階層と整合性のあるグループ/タグ階層に従います。ピリオドの前にある各セグメントにグループが作成され、グループは論理グループにも作成されます。作成されるグループは次のとおりです。

- グローバル (コントローラ)
- 構造体とサブ構造体
- 配列

● **注記:** .bit アドレスにはグループは作成されません。

基本グローバルタグ

基本グローバルタグ (非構造体、非配列タグ) はグローバルグループに配置されます。構造体タグと配列タグそれぞれに、親グループ内に独自のサブグループが作成されます。この方法でデータを整理することによって、サーバーのタグビューは RSLogix5000 によく似たものになります。

● **注記:** 構造体/配列サブグループの名前からその内容を把握できます。たとえば、コントローラで定義されている配列 "tag1[1,6]" には "tag1[x,y]" という名前のサブグループがあります。この例では、x は次元 1 が存在することを示し、y は次元 2 が存在することを示しています。さらに、明示的な制限がない限り、配列サブグループ内のタグはその配列の要素です。構造体サブグループ内のタグは、それ自体が構造体のメンバーです。配列が含まれている構造体には、その構造体グループの配列サブグループも作成されます。

配列タグ

配列ごとに配列の要素を含むグループが作成されます。グループ名の表記は <配列名>[x,y,z] となり、ここで:

- [x,y,z]: 3 次元配列
- [x,y]: 2 次元配列
- [x]: 1 次元配列

● **注記:** 配列タグの表記は <タグ要素>_XXXXX_YYYYY_ZZZZZ となります。たとえば、要素 "tag1[12,2,987]" のタグ名は "tag1_12_2_987" になります。

簡単な例

Name	Value	Force Mask	Style	Data Type
[-] MyTag	{...}	{...}		MyDataType
[-] MyTag.Member1	{...}	{...}	Decimal	DINT[10]
[+] MyTag.Member1[0]	0		Decimal	DINT
[+] MyTag.Member1[1]	0		Decimal	DINT
[+] MyTag.Member1[2]	0		Decimal	DINT
[+] MyTag.Member1[3]	0		Decimal	DINT

	<table border="1"> <thead> <tr> <th>Tag Name</th> <th>Address</th> </tr> </thead> <tbody> <tr><td>MEMBER1_0</td><td>MYTAG.MEMBER1[0]</td></tr> <tr><td>MEMBER1_1</td><td>MYTAG.MEMBER1[1]</td></tr> <tr><td>MEMBER1_2</td><td>MYTAG.MEMBER1[2]</td></tr> <tr><td>MEMBER1_3</td><td>MYTAG.MEMBER1[3]</td></tr> <tr><td>MEMBER1_4</td><td>MYTAG.MEMBER1[4]</td></tr> <tr><td>MEMBER1_5</td><td>MYTAG.MEMBER1[5]</td></tr> <tr><td>MEMBER1_6</td><td>MYTAG.MEMBER1[6]</td></tr> <tr><td>MEMBER1_7</td><td>MYTAG.MEMBER1[7]</td></tr> <tr><td>MEMBER1_8</td><td>MYTAG.MEMBER1[8]</td></tr> <tr><td>MEMBER1_9</td><td>MYTAG.MEMBER1[9]</td></tr> </tbody> </table>	Tag Name	Address	MEMBER1_0	MYTAG.MEMBER1[0]	MEMBER1_1	MYTAG.MEMBER1[1]	MEMBER1_2	MYTAG.MEMBER1[2]	MEMBER1_3	MYTAG.MEMBER1[3]	MEMBER1_4	MYTAG.MEMBER1[4]	MEMBER1_5	MYTAG.MEMBER1[5]	MEMBER1_6	MYTAG.MEMBER1[6]	MEMBER1_7	MYTAG.MEMBER1[7]	MEMBER1_8	MYTAG.MEMBER1[8]	MEMBER1_9	MYTAG.MEMBER1[9]
Tag Name	Address																						
MEMBER1_0	MYTAG.MEMBER1[0]																						
MEMBER1_1	MYTAG.MEMBER1[1]																						
MEMBER1_2	MYTAG.MEMBER1[2]																						
MEMBER1_3	MYTAG.MEMBER1[3]																						
MEMBER1_4	MYTAG.MEMBER1[4]																						
MEMBER1_5	MYTAG.MEMBER1[5]																						
MEMBER1_6	MYTAG.MEMBER1[6]																						
MEMBER1_7	MYTAG.MEMBER1[7]																						
MEMBER1_8	MYTAG.MEMBER1[8]																						
MEMBER1_9	MYTAG.MEMBER1[9]																						

複雑な例

"MyStructArray[0].MySubStruct.Data" というアドレスで定義されている Logix タグは "Global"、"MYSTRUCTARRAY [x]"、"MYSTRUCTARRAY[0]"、および "MYSUBSTRUCT" の各グループで表されます。タグ "DATA" は最後のグループ に入ります。"DATA" への静的参照は "Channel1.Device1.Global.MYSTRUCTARRAY[X].MYSTRUCTARRAY [0].MYSUBSTRUCT.DATA" になります。動的参照は "Channel1.Device1.MyStructArray[0].MySubStruct.Data" になります。

● 詳細については、サーバーのヘルプファイルで「静的タグ(ユーザー定義)」および「動的タグ」を参照してください。

圧縮モード

圧縮モードでは、自動生成されたサーバータグは、タグのアドレスと整合性のあるグループ/タグ階層に従います。ピリオドの 前にある各セグメントにグループが作成されます。作成されるグループは次のとおりです。

- プログラム
- 構造体とサブ構造体

● 注記:

1. 配列と .bit アドレスにグループは作成されません。
2. アンダースコアで始まるタグ名と構造体メンバー名は "U" に変換されます。サーバーはタグ名フィールド内の先頭の アンダースコアをサポートしていないため、この処理が必要です。

簡単な例

Name	Value	Force Mask	Style	Data Type
MyTag	{...}	{...}		MyDataType
MyTag.Member1	{...}	{...}	Decimal	DINT[10]
MyTag.Member1[0]	0		Decimal	DINT
MyTag.Member1[1]	0		Decimal	DINT
MyTag.Member1[2]	0		Decimal	DINT
MyTag.Member1[3]	0		Decimal	DINT

	<table border="1"> <thead> <tr> <th>Tag Name</th> <th>Address</th> </tr> </thead> <tbody> <tr><td>MEMBER1_0</td><td>MYTAG.MEMBER1[0]</td></tr> <tr><td>MEMBER1_1</td><td>MYTAG.MEMBER1[1]</td></tr> <tr><td>MEMBER1_2</td><td>MYTAG.MEMBER1[2]</td></tr> <tr><td>MEMBER1_3</td><td>MYTAG.MEMBER1[3]</td></tr> <tr><td>MEMBER1_4</td><td>MYTAG.MEMBER1[4]</td></tr> <tr><td>MEMBER1_5</td><td>MYTAG.MEMBER1[5]</td></tr> <tr><td>MEMBER1_6</td><td>MYTAG.MEMBER1[6]</td></tr> <tr><td>MEMBER1_7</td><td>MYTAG.MEMBER1[7]</td></tr> <tr><td>MEMBER1_8</td><td>MYTAG.MEMBER1[8]</td></tr> <tr><td>MEMBER1_9</td><td>MYTAG.MEMBER1[9]</td></tr> </tbody> </table>	Tag Name	Address	MEMBER1_0	MYTAG.MEMBER1[0]	MEMBER1_1	MYTAG.MEMBER1[1]	MEMBER1_2	MYTAG.MEMBER1[2]	MEMBER1_3	MYTAG.MEMBER1[3]	MEMBER1_4	MYTAG.MEMBER1[4]	MEMBER1_5	MYTAG.MEMBER1[5]	MEMBER1_6	MYTAG.MEMBER1[6]	MEMBER1_7	MYTAG.MEMBER1[7]	MEMBER1_8	MYTAG.MEMBER1[8]	MEMBER1_9	MYTAG.MEMBER1[9]
Tag Name	Address																						
MEMBER1_0	MYTAG.MEMBER1[0]																						
MEMBER1_1	MYTAG.MEMBER1[1]																						
MEMBER1_2	MYTAG.MEMBER1[2]																						
MEMBER1_3	MYTAG.MEMBER1[3]																						
MEMBER1_4	MYTAG.MEMBER1[4]																						
MEMBER1_5	MYTAG.MEMBER1[5]																						
MEMBER1_6	MYTAG.MEMBER1[6]																						
MEMBER1_7	MYTAG.MEMBER1[7]																						
MEMBER1_8	MYTAG.MEMBER1[8]																						
MEMBER1_9	MYTAG.MEMBER1[9]																						

複雑な例

"MyStructArray[0].MySubStruct.Data" というアドレスで定義されている Logix タグは "MYSTRUCTARRAY[0]" および "MYSUBSTRUCT" グループで表されます。タグ "DATA" は最後のグループに入ります。"DATA" への静的参照は "Channel1.Device1.MYSTRUCTARRAY[0].MYSUBSTRUCT.DATA" になり、動的参照は "Channel1.Device1.MyStructArray[0].MySubStruct.Data" になります。

ネイティブタグデータベースの CSV のインポート

CSV ファイルでは各デバイスが表すことができるネイティブタグが指定されています。これはタグインポートの際に 1 回使用され、自動タグデータベース生成やリモート配備では必要ありません。ネイティブタグのインポートでは次の CSV ヘッダーを使用する必要があります。

Logix アドレス	Logix データ型	外部アクセス	説明
------------	------------	--------	----

● **ヒント:** 参照用にネイティブタグデータベースのテンプレート CSV ファイルが <サーバーのインストールディレクトリ >/Drivers/controllogix_unsolicited_ethernet/import_template.csv に用意されています。

SINT、INT、DINT 配列としてのネイティブタグのインポート

SINT、INT、DINT 配列としてインポートされたネイティブタグには文字列タグも定義され、これにはタグアドレス内の対応する配列の要素数が使用されます。この例を次に示します。

- "MySINTarray @ SINT[100]" という名前のネイティブタグがインポートされた場合、アドレス "MYSINTARRAY / 100" を持つ String データ型の静的タグが生成されます。
- "MyINTarray @ INT[100]" という名前のネイティブタグがインポートされた場合、アドレス "MYINTARRAY / 100" を持つ String データ型の静的タグが生成されます。
- "MyDINTarray @ DINT[100]" という名前のネイティブタグがインポートされた場合、アドレス "MYDINTARRAY / 100" を持つ String データ型の静的タグが生成されます。

● **ヒント:** RSLogix5000 の定義済み文字列をインポートするには、インポートを実行する前に、String データ型に含まれている 2 つの要素 ("STRING.DATA" および "STRING.LEN") をネイティブタグデータベースの CSV ファイルで定義する必要があります。

Logix アドレス

Logix アドレスに関する制約は RSLogix5000 の要件と一致しており、これは次に示す IEC 1131-3 の識別子規則に対応しています。

- 先頭は英字 (A-Z、a-z) またはアンダースコアでなければなりません。
- 英数文字とアンダースコアのみを含むことができます。
- 各セグメントで最大 40 文字を使用できます。
- アンダースコアが連続してはなりません。
- 大文字と小文字は区別されません。

一意の Logix アドレスがないか上記の識別子の要件を満たさないタグはインポートに失敗し、Logix アドレスが示されたメッセージがサーバーのイベントログに書き込まれます。

Logix データ型

次に示す定義済みの Logix アトミックデータ型がサポートされています。

Logix データ型	サポートされるデータ型
BOOL	Boolean
SINT	Char、Byte
INT	Short、Word
DINT	Long、DWord
LINT	Double、Date
REAL	Float

● **注記:** その他の定義済みまたはユーザー定義の複合 (構造体) データ型はサポートされていません。構造体データ型は Logix アドレスを限定してアトミックデータ型にすることでインポートできます。たとえば、次のような TIME という名前の構造体があるとして。

```

TIME
{
  HOUR @ SINT
  MIN @ SINT
  SEC @ SINT
}

```

この構造体を分割し、関連する Logix データ型、外部アクセス、説明が前述の CSV フォーマットに従っているアトミックメンバーを "TIME.HOUR"、"TIME.MIN"、および "TIME.SEC" としてインポートできます。CSV インポートで指定されているサポートされない Logix データ型の値はすべて、インポートが成功するように、デフォルトで DINT になります。

外部アクセス

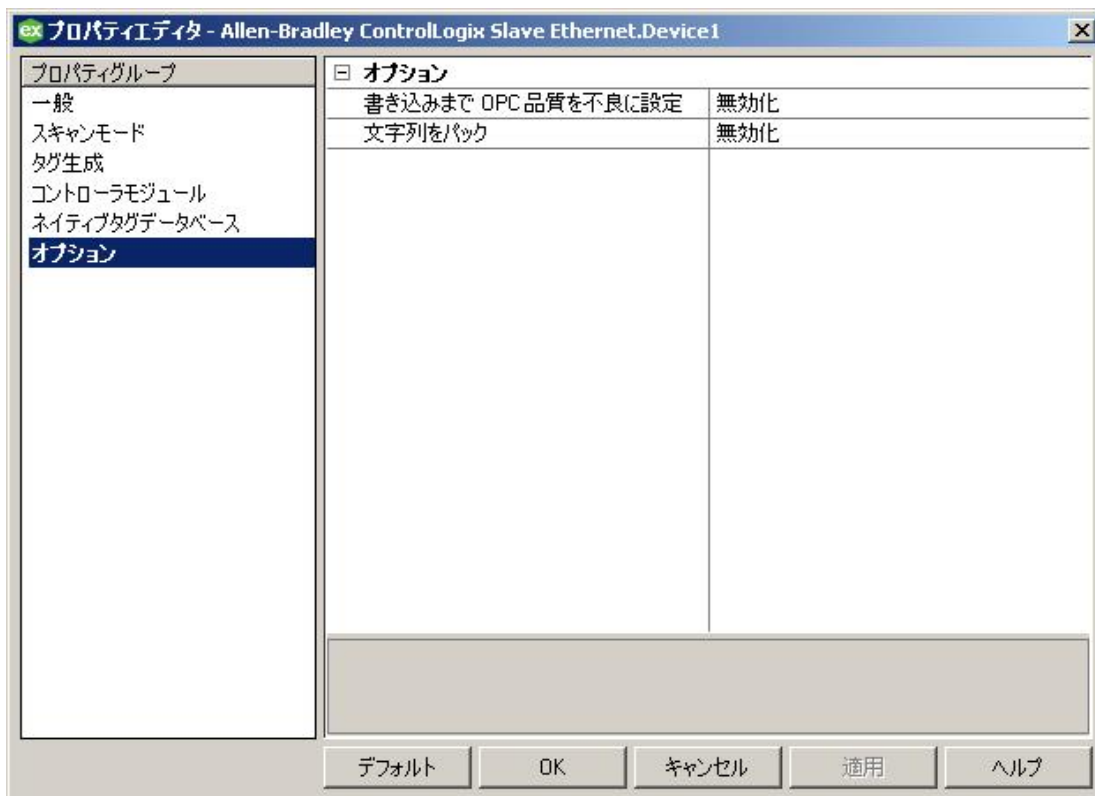
外部アクセスでは、マスターデバイスの読み取り/書き込み権限を指定します。このアクセス権はクライアントタグには適用されません。クライアントタグはデフォルトで必ず読み取り/書き込みのクライアントアクセス権を持ちます。次に示す外部アクセスタイプがサポートされています。指定されているその他の値はすべて読み取り/書き込みに設定されます。

- **R/W:** マスターデバイスはネイティブタグに対する読み取り/書き込みのアクセス許可を持ちます。
- **RO:** マスターデバイスは読み取り専用のアクセス許可を持ちます。書き込みはすべて失敗し、該当するエラー (CIP エラー 0x0F) が返ります。

説明

説明は自動タグデータベース生成中に使用され、64 文字に切り詰められます。「Description」フィールドが存在する必要がありますが、空白のままでも構いません。

デバイスのプロパティ - オプション



「書き込みまで OPC 品質を不良に設定」: ネイティブタグへの書き込みが発生するまでドライバーが品質として「不良」を返すようにするには、「有効化」を選択します。書き込みはクライアントインターフェース (OPC など) またはマスターデバイス (ControlLogix 5000 シリーズ PLC など) から発生する可能性があります。配列の 1 つのアイテムに対する書き込みが発生すると、配列全体が初期化され、品質として「良好」が返されます。デフォルトでは無効になっています。起動時に、Allen-Bradley ControlLogix Unsolicited ドライバーは整数/数値データ型の値をゼロ (0) に初期化し、文字列を空にします。デフォルトではクライアントは初回の更新で品質として「良好」を受信しますが、この動作はデバイスごとに修正可能です。

「**文字列をパック**」: 有効にした場合、文字列タグに表示される文字列には、配列要素のすべてのバイトがパックフォーマットで含まれます。無効にした場合、文字列タグに表示される文字列はパックされていないフォーマットになり、各要素の下位バイトのみが表示されます。

データ型の説明

データタイプ	説明
Boolean	1 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
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
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
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
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
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 ビット 0 が下位ビット ビット 31 が上位ビット
Double	64 ビット浮動小数点値 ビット 0 が下位ビット ビット 63 が上位ビット
文字列	通常は、Null 終端、Null パディング、または空白パディング ASCII 文字列。
日付	64 ビット浮動小数点値。

● Logix プラットフォーム固有のデータ型については、[アドレスの説明](#)を参照してください。

アドレスの説明

Allen-Bradley ControlLogix Unsolicited ドライバー ではシンボリックタグベースのアドレス指定がサポートされます。

Logix タグベースのアドレス指定

このドライバーでは、一般的に Logix タグまたはネイティブタグと呼ばれるタグ (シンボルベースのアドレス指定構造体) が使用されます (これは Rockwell Automation の統合アーキテクチャに適合)。これらのタグは従来の PLC データアイテムとは異なり、物理アドレスや論理アドレスではなくタグ名がアドレスになります。

このドライバーでは、コントローラのアトミックデータ型 BOOL、SINT、INT、DINT、LINT、および REAL にアクセスできます。定義済みの一部のデータ型は構造体ですが、これらは最終的にはそのアトミックデータ型に基づきます。このため、構造体のすべての非構造体 (アトミック) メンバーにアクセスできます。たとえば、TIMER はサーバータグに割り当てることができませんが、TIMER のアトミックメンバー (TIMER.EN、TIMER.ACC など) はタグに割り当てることができます。構造体メンバーが構造体自体である場合、サブ構造体のアトミックメンバーにアクセスするには両方の構造体を展開する必要があります。これはユーザー定義とモジュール定義の型でより一般的であり、定義済みの型では必要はありません。

アトミックデータ型	説明	範囲
BOOL	1 ビット値	VT_BOOL 0, 1
SINT	符号付き 8 ビット値	VT_I1 -128 から 127
INT	符号付き 16 ビット値	VT_I2 -32,768 から 32,767
DINT	符号付き 32 ビット値	VT_I4 -2,147,483,648 から 2,147,483,647
LINT	符号付き 64 ビット値	VT_R8 -9.22337E18 から 9.22336E18
REAL	32 ビット IEEE 浮動小数点	VT_R4 1.1755 E-38 から 3.403E38 0 -3.403E-38 から -1.1755

クライアント/サーバータグアドレスの規則

Logix タグ名はクライアント/サーバータグのアドレスに相当します。Logix タグ名 (RSLogix5000 から入力) は IEC 1131-3 の識別子の規則に従います。クライアント/サーバータグアドレスはこれと同じ規則に従います。以下に示します。

- 先頭は英字 (A-Z、a-z) またはアンダースコアでなければなりません。
- 英数文字とアンダースコアのみを含むことができます。
- 各セグメントで最大 40 文字を使用できます。
- アンダースコアが連続してはなりません。
- 大文字と小文字は区別されません。

● 注記

- サーバーでのタグ名の割り当てはアドレスの割り当てとは異なり、名前がアンダースコアであってはなりません。
- タグが適切に検証されるためには、静的クライアントタグを表すネイティブタグがネイティブタグデータベースに存在する必要があります。

アドレスのフォーマット

サーバーで静的にまたはクライアントから動的に Logix タグのアドレスを指定するにはいくつかの方法があります。選択するフォーマットはタグのタイプと使用法によって異なります。たとえば、SINT 型タグ内のビットにアクセスする場合にはビットフォーマットを使用します。

● アドレスのフォーマットと構文の詳細については、以下の表を参照してください。

● 注記: 配列と文字列以外のすべてのフォーマットが RSLogix5000 にネイティブです。アトミックデータ型を参照する場合、RSLogix 5000 のタグ名をコピーしてサーバーのタグアドレス/パラメータに貼り付けることができます。これは対応するネイティブタグがネイティブタグデータベース内に存在するかぎり有効です。

フォーマット	構文	例	注記
標準	<Logix タグ名>	tag_1	タグが配列であってはなりません。

フォーマット	構文	例	注記
配列要素	<Logix 配列タグ名> [次元 1, 次元 2, 次元 3]	tag_1 [2, 58, 547] tag_1 [0, 3]	次元の範囲 = 1 から 3。要素の範囲 = 0 から 65535。
オフセットがない配列*	<Logix 配列タグ名> {列数} <Logix 配列タグ名> {行数} {列数}	tag_1 {8} tag_1 {2} {4}	次元の範囲 = 1 から 2。 要素の範囲 = 1 から 65535。 読み書きする要素の数は行数 x 列数です。行数が指定されていない場合、行数はデフォルトで 1 になります。 配列はゼロオフセットで開始します (すべての次元で配列のインデックスが 0)。
オフセットがある配列*	<Logix 配列要素タグ> {列数} <Logix 配列要素タグ> {行数} {列数}	tag_1 [2, 3] {10} tag_1 [2, 3] {2} {5}	配列は配列要素タグで次元ごとに指定されているオフセットで開始します。この配列では必ず最大の次元がカバーされます。したがって、"tag_1[2,3]{10}" では要素 tag_1[2,3] -> tag_1[2,13] の配列が生成されます。
ビット	<Logix タグ名> . ビット <Logix タグ名> . [ビット]	tag_1.0 tag_1.[0]	ビット範囲 = 0 から 31。 タグが配列である場合、必ず BOOL 配列になります。BOOL 配列でなければタグが配列になることはできません。
文字列	<Logix タグ名> .Data/<最大文字列長>	tag_1.Data/4	長さの範囲 = 1 から 65535。 この文字列との間で読み取り書き込み可能な最大文字数。

*このフォーマットでは複数の要素が要求されることがあるため、配列データが渡される順序は Logix 配列タグの次元によって異なります。たとえば、行数 x 列数 = 4 でコントローラタグが 3X3 要素の配列である場合、"array_tag [0,0]"、"array_tag [0,1]"、"array_tag [0,2]"、"array_tag [1,0]" の順序で要素が参照されます。コントローラタグが 2X10 要素の配列であった場合には結果が異なります。

● 1 次元、2 次元、および 3 次元配列で要素がどのように参照されるかについては、[Logix 配列データの順序](#)を参照してください。

タグの有効範囲

グローバルタグ

グローバルタグは、コントローラでの有効範囲がグローバルである Logix タグです。どのようなプログラムまたはタスクでもグローバルタグにアクセスできますが、グローバルタグを参照可能な手段の数は、その Logix データ型および使用されているアドレスフォーマットによって異なります。

プログラムタグ

プログラムタグはグローバルタグとよく似ていますが、プログラムタグの範囲はそれが定義されているプログラムにローカルである点が異なります。このドライバーは現在のところ、プログラムが指定されているネイティブタグのインポートはサポートしていません。

構造体タグのアドレス指定

Logix 構造体タグは、(アトムックまたは構造体型である) 1 つ以上のメンバータグを持つタグです。
<構造体名> . <アトムック型のタグ>

ここから、サブ構造体は次のようにアドレス指定されます。
<構造体名> . <サブ構造体名> . <アトムック型のタグ> <サブ構造体名> . <アトムック型のタグ>

構造体の配列は次のようにアドレス指定されます。
<構造体の配列名> [次元 1, 次元 2, 次元 3] . <アトムック型のタグ>

ここから、サブ構造体の配列は次のようにアドレス指定されます。
<構造体名> . <サブ構造体名> . <アトムック型のタグ> <サブ構造体の配列名> [次元 1, 次元 2, 次元 3] . <アトムック型のタグ>

● **注記:** 上記の例は構造体に関連するアドレス指定方法のうちの一部にすぎません。これらは構造体のアドレス指定について概要を示すために挙げられています。詳細については、Rockwell/Allen-Bradley のドキュメントを参照してください。

詳細なアドレス指定

シンボリックタグアドレスに含めることができるシンボリックアドレス指定のためのオプションがいくつか用意されています。ビットおよび配列アドレス指定の構文ではデータ型に次のような制約が設けられています。

- ビット構文の場合、インデックスがデータ型のビットサイズを超えてはなりません。たとえば、"MyDint @ Dint" はネイティブタグとしてインポートされます。ビットインデックスが 31 を超えてはなりません (DINT は符号付き 32 ビット値なので)。
- 配列構文の場合、配列のオフセットと要素数の和が、関連付けられているネイティブタグ内の要素数を超えてはなりません。たとえば、"MyDintArray @ DINT[10]" はネイティブタグとしてインポートされます。アドレス "MYDINTARRAY[0] {5}" および "MYDINTARRAY[4] {5}" を持つ静的タグは有効です。これらの配列にはネイティブタグの最初の 5 つの要素と最後の 5 つの要素がそれぞれ含まれているためです。アドレス "MYDINTARRAY[5]{10}" を持つ静的タグは無効です。このタグはオフセット 5 で始まる 10 個の DINT を要求していますが、このネイティブタグ配列はそこまで大きくないためです。

詳細については、以下の表を参照してください。

要素	構文	例	注記
標準	<タグ名>	tag_1	該当なし。
オフセットがない配列	<配列タグ名> {列数}	tag_1 {8}	読み書きする要素の数は行数 x 列数です。行数が指定されていない場合、行数はデフォルトで 1 になります。配列の 1 つ以上の要素がアドレス指定されている必要があります。 配列はゼロオフセットで開始します (すべての次元で配列のインデックスが 0)。
	<配列タグ名> {行数} {列数}	tag_1 {2} {4}	
オフセットがある配列	<配列要素タグ> [オフセット] {列数}	tag_1 [5] {8}	読み書きする要素の数は行数 x 列数です。行数が指定されていない場合、行数はデフォルトで 1 になります。配列の 1 つ以上の要素がアドレス指定されている必要があります。 配列はゼロオフセットで開始します (すべての次元で配列のインデックスが 0)。
	<配列要素タグ> [オフセット] {行数} {列数}	tag_1 [5] {2} {4}	
ビット	<タグ名> . ビット <タグ名> . [ビット]	tag_1 . 0 tag_1 . [0]	該当なし。 該当なし。
文字列	<タグ名> / <要素数>	tag_1 / 4	要素数は 1 以上でなければなりません。読み書きされる文字の数は「文字列をパック」プロパティによって異なります。*

*有効にした場合、文字数は要素数 x 要素サイズになります (INT 配列の要素が 4 つの場合は 8 文字)。無効にした場合、文字数は要素数と等しくなります (INT 配列の要素が 4 つの場合は 4 文字)。

● 詳細については、[オプション](#)を参照してください。

Logix 配列データの順序

ネイティブタグは 3 次元配列までサポートするため、Logix 配列データのオーダリングは 2 次元 OPC 配列にマッピングされます。

1 次元配列 - array [dim1]

1 次元配列データはコントローラとの間で昇順でやり取りされます。

```
for (dim1 = 0; dim1 < dim1_max; dim1++)
```

例: 3 要素の配列

```
array [0]  
array [1]  
array [2]
```

2 次元配列 - array [次元 1, 次元 2]

2 次元配列データはコントローラとの間で昇順でやり取りされます。

```
for (dim1 = 0; dim1 < dim1_max; dim1++)  
for (dim2 = 0; dim2 < dim2_max; dim2++)
```

例: 3x3 要素の配列

```
array [0, 0]  
array [0, 1]  
array [0, 2]  
array [1, 0]  
array [1, 1]  
array [1, 2]  
array [2, 0]  
array [2, 1]  
array [2, 2]
```

3 次元配列 - array [次元 1, 次元 2, 次元 3]

3 次元配列データはコントローラとの間で昇順でやり取りされます。

```
for (dim1 = 0; dim1 < dim1_max; dim1++)  
for (dim2 = 0; dim2 < dim2_max; dim2++)  
for (dim3 = 0; dim3 < dim3_max; dim3++)
```

例: 3x3x3 要素の配列

```
array [0, 0, 0]  
array [0, 0, 1]  
array [0, 0, 2]  
array [0, 1, 0]  
array [0, 1, 1]  
array [0, 1, 2]  
array [0, 2, 0]  
array [0, 2, 1]  
array [0, 2, 2]  
array [1, 0, 0]  
array [1, 0, 1]  
array [1, 0, 2]  
array [1, 1, 0]  
array [1, 1, 1]  
array [1, 1, 2]  
array [1, 2, 0]  
array [1, 2, 1]  
array [1, 2, 2]  
array [2, 0, 0]  
array [2, 0, 1]  
array [2, 0, 2]  
array [2, 1, 0]  
array [2, 1, 1]  
array [2, 1, 2]
```

```
array [2, 2, 0]  
array [2, 2, 1]  
array [2, 2, 2]
```

エラーコード

Allen-Bradley ControlLogix Unsolicited ドライバー では次のエラーコードが返されることがあります。

● 特定のタイプのエラーコードの詳細については、以下のリストからリンクを選択してください。

[EtherNet/IP カプセル化 エラーコード](#)

[CIP エラーコード](#)

[0x01 拡張エラーコード](#)

[0xFF 拡張エラーコード](#)

EtherNet/IP カプセル化 エラーコード

Allen-Bradley ControlLogix Unsolicited ドライバー では次のエラーコードが返されることがあります。

● 注記: エラーコードは 16 進数で表示されます。

エラー	説明
0001	コマンドが処理されませんでした。
0002	コマンド用のメモリがありません。
0003	データの形式が不適切であるか不完全です。
0064	セッション ID が無効です。
0065	ヘッダーの長さが無効です。
0069	要求されたプロトコルバージョンはサポートされていません。

CIP エラーコード

エラーコードは 16 進数で表示されます。

エラー	説明
01	接続エラー*
02	リソースが不足しています
03	無効なパラメータ値
04	IOI を解読できなかったタグが存在しません
05	宛先が不明です
06	要求されたデータは応答パケットに収まりません
08	サポートされていないサービスです
0F	Permission denied
13	サービスを実行するには指定されたコマンドデータ/パラメータでは不十分です
26	指定された IOI Word の数は IOI Word 数と一致しません
FF	一般的なエラー**

● *関連項目: [0x01 拡張エラーコード](#)

● **関連項目: [0xFF 拡張エラーコード](#)

0x01 拡張エラーコード

Allen-Bradley ControlLogix Unsolicited ドライバー では CIP エラー 0x01 について次の拡張エラーが返されることがあります。

● 注記: エラーコードは 16 進数で表示されます。

エラー	説明
0x0205	未接続送信パラメータエラー。
0x0312	リンクアドレスを使用できません。

エラー	説明
0x0318	自己へのリンクアドレスは無効です。

0xFF 拡張エラーコード

Allen-Bradley ControlLogix Unsolicited ドライバー では CIP エラー 0xFF について次の拡張エラーが返されることがあります。

●注記: エラーコードは 16 進数で表示されます。

エラー	説明
2104	アドレスが範囲外です。
2105	データオブジェクトの末尾以降にアクセスしようとした。
2107	データ型が無効であるかサポートされていません。

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ーフェースの「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。

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。| OS エラー = '<エラー>'。

エラータイプ:

エラー

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。一般読み取りエラー。

エラータイプ:

エラー

考えられる原因:

ファイルが見つからないか、破損しているか、フォーマットが不正です。

解決策:

タグデータベースファイルを見つけてヘッダー行、フォーマット、場所を確認してから、もう一度試してください。

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルエンコーディングはサポートされていません。

エラータイプ:

エラー

考えられる原因:

インポート中の CSV ファイルでサポートされていないファイルエンコーディングが使用されています。

解決策:

ANSI または UTF-8 エンコーディング方法を使用するよう CSV ファイルを更新してください。

ネイティブタグデータベースのインポート中にエラーが発生しました。

エラータイプ:

エラー

考えられる原因:

予期しないエラーが発生しました。

解決策:

インポート中の CSV ファイルが適切なフォーマットであることを確認してください。

ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が認識されません。| 認識されないフィールド名 = '<フィールド名>'。

エラータイプ:

エラー

考えられる原因:

CSV ファイルで定義されているフィールドが、ネイティブタグデータベースのインポートでサポートされていません。

解決策:

CSV ファイル内に意図しないフィールドがないことを確認してください。

● 注記:

サポートされているフィールド名には "Logix Address"、"Logix DataType"、"External Access"、"Description" があります。

ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が重複しています。| 重複しているフィールド名 = '<フィールド>'。

エラータイプ:

エラー

考えられる原因:

インポート中の CSV ファイルにフィールド名が同じである複数の定義が含まれています。

解決策:

CSV ファイル内の重複しているフィールドを除去するか修正してください。サポートされているフィールド名には Logix Address、Logix Data Type、External Access、Description があります。

ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが見つかりません。

エラータイプ:

エラー

考えられる原因:

インポート中の CSV ファイルにヘッダーが含まれていません。

解決策:

CSV ファイル内に欠落したフィールドがないことを確認し、ヘッダー行を確認して修正してください。

● 注記:

サポートされているフィールド名には Logix Address、Logix Data Type、External Access、Description があります。

ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが不完全です。

エラータイプ:

エラー

考えられる原因:

インポート中の CSV ファイルに完全で有効なヘッダーが含まれていません。

解決策:

CSV ファイル内に欠落したフィールドがないことを確認し、ヘッダー行を確認して修正してください。

● 注記:

サポートされているフィールド名には Logix Address、Logix Data Type、External Access、Description があります。

非送信請求 Logix サーバーを起動できませんでした。

エラータイプ:

エラー

考えられる原因:

ドライバーは指定された IP/ポートでバインドおよび受信待機できませんでした。

解決策:

示されたポート (TCP または UDP) が別のアプリケーションによって使用されていないことを確認し、競合をすべて解放してください。

タグアドレスが無効です。ネイティブタグはインポートされません。| 無効なアドレス = '<アドレス>'。

エラータイプ:

警告

2 つのチャンネルが同じネットワークアダプタ IP と TCP ポートを使用するよう構成されています。各チャンネルが一意的なローカル IP とポートにバインドされている必要があります。| 1 つ目のチャンネル = '<チャンネル>'、2 つ目のチャンネル = '<channel>'。

エラータイプ:

警告

2 つのデバイスが EtherNet/IP モジュールからの同じパスを使用するよう構成されています。各チャンネルが EtherNet/IP モジュールからの一意のパスを持つ必要があります。| 1 つ目のデバイス = '<アドレス>'、2 つ目のデバイス = '<address>'。

エラータイプ:

警告

考えられる原因:

このデバイスに設定されている CPU タイプとスロット番号はすでに使用されています。

解決策:

1. 別の CPU タイプ (ローカルまたはリモート) を選択してください。
2. デバイスがリモートコントローラモジュールとして構成されている場合、別のスロット番号を選択してください。

タグアドレスが無効です。重複するタグアドレスは許可されません。| 無効なアドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

インポート中の CSV ファイルに Logix アドレスが同じである 1 つ以上のタグが含まれています。

解決策:

インポート中のネイティブタグデータベース CSV ファイルで、重複する Logix アドレスが含まれているネイティブタグを除去してください。

メモリをタグに割り当てることができませんでした。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

タグの構築に必要なリソースを割り当てることができませんでした。タグはプロジェクトに追加されませんでした。

解決策:

使用していないアプリケーションを終了する、仮想メモリの量を増やすなどをした後でもう一度試してください。

ネイティブタグが無効です。個々のタグサイズは 128 KB に制限されています。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

インポート中の CSV ファイルの大きな配列などに、表示するために 128 KB を超えるメモリを必要とする単一のタグ定義が含まれています。128 KB を超えるメモリを必要とするネイティブタグは定義できません。

解決策:

配列のサイズを小さくするか、インポート対象の CSV ファイルからタグを除去してください。

ネイティブタグのインポート中にエラーが発生しました。データベースタグデータの合計サイズは 128 KB に制限されています。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

インポート中の CSV ファイルの大きな配列などに、128 KB を超えるメモリを必要とするタグ定義が含まれています。

解決策:

128 KB を超えるメモリが必要な場合、新しいデバイスを作成し、タグデータベースを複数のデバイスに分割してください。

指定された TCP/IP ポートは範囲外です。デフォルトポートを使用します。| 有効な範囲 = <数値> ~ <数値>、デフォルトポート = <数値>。

エラータイプ:

警告

考えられる原因:

TCP/IP ポート 0 を指定するプロジェクトがロードされましたが、このオプションはサポートされていません。

解決策:

デフォルトポート (44818) を使用するか、有効な範囲内 (1 ~ 65535) のポートを選択してください。

このチャンネル内の別のデバイスがローカル CPU としてすでに登録されています。| デバイス = <デバイス>。

エラータイプ:

警告

ネイティブタグがインポートされました。| タグの数 = <数>、タグデータベースのパス = '<パス>'。

エラータイプ:

情報

タグが生成されました。| タグの数 = <数>、タグ階層のモード = '<モード>'。

エラータイプ:

情報

リソースが不足しているため自動タグ生成を実行できませんでした。

エラータイプ:

情報

索引

0

0x01 拡張エラーコード 27

0xFF 拡張エラーコード 28

2

2つのチャンネルが同じネットワークアダプタ IP と TCP ポートを使用するよう構成されています。各チャンネルが一意のローカル IP とポートにバインドされている必要があります。| 1 つ目のチャンネル = '<チャンネル>', 2 つ目のチャンネル = '<channel>'。 31

2つのデバイスが EtherNet/IP モジュールからの同じパスを使用するよう構成されています。各チャンネルが EtherNet/IP モジュールからの一意のパスを持つ必要があります。| 1 つ目のデバイス = '<アドレス>', 2 つ目のデバイス = '<address>'。 31

B

Boolean 21

Byte 21

C

Char 21

CIP エラーコード 27

CSV 15

D

Date 21

Double 21

DWord 21

E

EtherNet/IP カプセル化エラーコード 27

F

Float 21

I

IEEE-754 浮動小数点 8

L

Long 21

S

Short 21

T

TCP/IP ポート 9

W

Word 21

あ

アドレスのフォーマット 22

アドレスの説明 22

い

イーサネット/IP モジュール 9

イベントログメッセージ 29

え

エラーコード 27

お

オプション 19

き

キャッシュからの初回更新 12

く

クライアント固有のスキャン速度を適用 12

こ

このチャンネル内の別のデバイスがローカル CPU としてすでに登録されています。| デバイス = <デバイス>。 32

コントローラモジュール 14

コンマ区切り値 15

さ

サブグループを許可 13

し

シミュレーション 11

す

スキャンしない、要求ポールのみ 12

スキャンモード 12

すべてのタグのすべての値を書き込み 7

すべてのタグの最新の値のみを書き込み 8

すべてのデータを指定したスキャン速度で要求 12

スロット 15

た

タグアドレスが無効です。ネイティブタグはインポートされません。| 無効なアドレス = '<アドレス>'。 31

タグアドレスが無効です。重複するタグアドレスは許可されません。| 無効なアドレス = '<アドレス>'。 31

タグが生成されました。| タグの数 = <数>、タグ階層のモード = '<モード>'。 32

タグに指定のスキャン速度を適用 12

タグのインポート... 15

タグの有効範囲 23

タグ階層 16

タグ生成 12

ち

チャンネルのプロパティ-イーサネット通信 7

チャンネルのプロパティ-一般 6

チャンネルのプロパティ-書き込み最適化 7

チャンネルのプロパティ-詳細 8

チャンネル割り当て 11

チャンネル設定 6

て

データコレクション 11

データベースをインポート 15

データ型の説明 21

デバイスのプロパティ-タグ生成 12

デバイスの設定 10

デバイス起動時 13

デューティサイクル 8

と

ドライバー 7, 11

ね

ネイティブタグがインポートされました。| タグの数 = <数>、タグデータベースのパス = '<パス>'。 32

ネイティブタグが無効です。個々のタグサイズは 128 KB に制限されています。| タグアドレス = '<アドレス>'。 32

ネイティブタグデータベース 15

ネイティブタグデータベースの CSV のインポート 18

ネイティブタグデータベースのインポート中にエラーが発生しました。 29

ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが見つかりません。
30

ネイティブタグデータベースのインポート中にエラーが発生しました。タグフィールドの識別レコードが不完全です。 30

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルエンコーディングはサポートされていません。
29

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。| OS エラー = '<エラー>'。 29

ネイティブタグデータベースのインポート中にエラーが発生しました。ファイルを開くことができません。一般読み取りエラー。 29

ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が重複しています。| 重複している
フィールド名 = '<フィールド>'。 30

ネイティブタグデータベースのインポート中にエラーが発生しました。フィールド名が認識されません。| 認識されない

フィールド名 = '<フィールド名>'。 29
ネイティブタグのインポート中にエラーが発生しました。データベースタグデータの合計サイズは 128 KB に制限されています。| タグアドレス = '<アドレス>'。 32
ネットワークアダプタ 7

は

パス 15

ふ

プロパティ変更時 13

へ

ヘルプの目次 4

ま

マスターデバイスの構成 10

め

メモリをタグに割り当てることができませんでした。| タグアドレス = '<アドレス>'。 31

も

モジュールタイプ 14

モデル 11

り

リソースが不足しているため自動タグ生成を実行できませんでした。 32

リモート 14

ろ

ローカル 14

曬

圧縮 16

梱

概要 5

髡

最適化方法 7

任

作成 14

凵

削除 13

摺

指定された TCP/IP ポートは範囲外です。デフォルトポートを使用します。| 有効な範囲 = <数値> ~ <数値>、デフォルトポート = <数値>。 32

指定したスキャン速度以下でデータを要求 12

諸

識別 10

醜

重複タグ 13

陽

書き込みまで OPC 品質を不良に設定 19

書き込み最適化 7

訓

詳細なアドレス指定 24

讖

上書き 13

楨

親グループ 13

觚

診断 7

璿

生成 13

覘

設定 6

寵

展開 16

雾

非 Boolean タグの最新の値のみを書き込み 8

非正規化浮動小数点処理 8

非送信請求 Logix サーバーを起動できませんでした。 30

擷

文字列 21

文字列をパック 20