

Allen-Bradley ControlLogix サーバードライバー

© 2026 Kepware. All Rights Reserved.

目次

Allen-Bradley ControlLogix サーバードライバー	1
目次	2
Allen-Bradley ControlLogix サーバードライバー ヘルプへようこそ	4
概要	4
設定	5
チャンネルのプロパティ - 一般	5
タグ数	6
Channel Properties — Ethernet Communications	6
Channel Properties — Write Optimizations	6
Channel Properties — Advanced	7
Channel Properties — EtherNet/IP Module	8
デバイスのプロパティ - 一般	9
動作モード	10
タグ数	11
Device Properties — Scan Mode	11
デバイスのプロパティ - タグ生成	12
Device Properties — Controller Module	14
Device Properties — Native Tag Database	14
Tag Hierarchy	15
デバイスのプロパティ - オプション	18
ControlLogix クライアントデバイスの構成	18
Logix Config Object	20
Logix Config CSV Import	20
Defining New Logix Addresses	23
Data Types Description	24
アドレスの説明	25
アドレスのフォーマット	26
Tag Scope	27
詳細なアドレス指定	27
Error Codes	30
EtherNet/IP Encapsulation Error Codes	30
CIP Error Codes	30
0x01 Extended Error Codes	31
0xFF 拡張エラーコード	31
Event Log Messages	32

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>'。	32
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。	32
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。	32
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。	32
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>'。	33
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>'。	33
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。	33
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。	34
非送信請求 Logix サーバーを起動できませんでした。	34
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>'。	34
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>'。	34
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 = '<アドレス>'。	34
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>'。	35
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>'。	35
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>'。	35
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 512 KB に制限されています。 タグアドレス = '<アドレス>'。	35
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>。	36
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>。	36
CSV インポートに失敗しました。既存の Logix アドレスを削除できません。	36
CSV インポートに失敗しました。新規 Logix アドレスを作成できません。	36
フラグメント化された書き込みの処理中に予期しないフラグメントが検出されました。 Logix アドレス = <タグ>。	36
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>'。	36
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>'。	36
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。	37
索引	38

Allen-Bradley ControlLogix サーバードライバー ヘルプへようこそ

これは、Kepware Allen-Bradley ControlLogix サーバードライバー のユーザードキュメントです。このヘルプセンターは、最新の機能と情報を反映して定期的に更新されます。

概要

Allen-Bradley ControlLogix サーバードライバー 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

Logix 構成オブジェクト

このデバイスのオブジェクトを設定する方法

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 サーバードライバー 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

バージョン 1.038

© 2026 Kepware. All Rights Reserved.

概要

Allen-Bradley ControlLogix サーバードライバー 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。

● **注記:** Allen-Bradley ControlLogix サーバードライバー では現在のところチャンネル数が 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。

● 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 サーバードライバーは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)。このドライバーでサポートされているチャンネルの最大数は1です。このドライバーでサポートされているデバイスの最大数は、1つのチャンネルにつき16です。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、CIP データテーブル読み取り/書き込み MSG 命令を使用してドライバーと通信し、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。

チャンネルのプロパティ - 一般

このサーバーでは、複数の通信ドライバーを同時に使用することができます。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ンネルプロパティを指定するときに使用します。

● **関連項目:** プロパティ情報に関するAPIドキュメントについては、`/config/v1/doc/drivers` エンドポイントを参照してください。

プロパティグループ	<input type="checkbox"/> 識別	
一般	名前	
イーサネット通信	説明	
書き込み最適化	ドライバー	
詳細	<input type="checkbox"/> 診断	
プロトコル設定	診断取り込み	無効化
	<input type="checkbox"/> タグ数	
	静的タグ	1

識別

「名前」: このチャンネルのユーザー定義識別情報を指定します。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。チャンネルの作成にはこのプロパティが必要です。

● 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情報を指定します。

● 「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」: このチャンネル用のプロトコルドライバーを指定します。チャンネル作成時に選択されたデバイスドライバーを指定します。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。チャンネルの作成にはこのプロパティが必要です。

● 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。大規模なクライアントアプリケーションを開発した場合は、プロパティを変更しないようにしてください。オペレータがプロパティを変更したりサーバーの機能にアクセスしたりすることを防ぐため、適切なユーザー役割を使用し、権限を正しく管理する必要があります。

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: ドライバーまたはオペレーティングシステムが診断をサポートしていない場合、このプロパティは使用できません。

● 詳細については、サーバーのヘルプで「通信診断」と「統計タグ」を参照してください。

タグ数

「静的タグ」: デバイスレベルまたはチャンネルレベルで定義される静的タグの数を指定します。この情報は、トラブルシューティングと負荷分散を行う場合に役立ちます。

Channel Properties — Ethernet Communications

Ethernet Communication can be used to communicate with devices.

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		

Ethernet Settings

Network Adapter: Specify the network adapter to bind. When left blank or Default is selected, the operating system selects the default adapter.

Channel Properties — Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.

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		

Write Optimizations

Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

- **Write All Values for All Tags:** This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- **Write Only Latest Value for Non-Boolean Tags:** Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
 - **Note:** This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- **Write Only Latest Value for All Tags:** This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

Duty Cycle: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

● **Note:** It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

Channel Properties — Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.

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		

Non-Normalized Float Handling: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

- **Replace with Zero:** This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified:** This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.

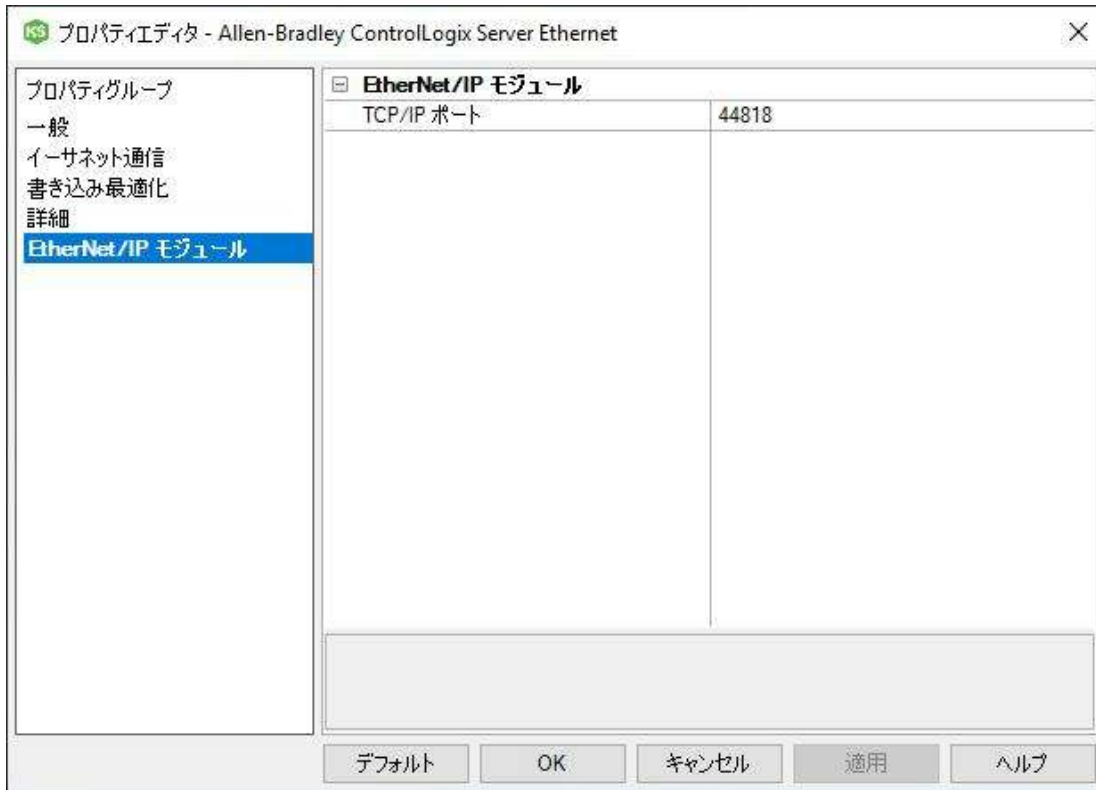
● **Note:** This property is disabled if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.

● *For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.*

Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

● **Note:** This property is not available for all drivers, models, and dependent settings.

Channel Properties — EtherNet/IP Module



TCP/IP Port: Specify the TCP/IP and UDP port that provides a unique communication channel to the EtherNet/IP module. The valid range is 1 to 65535. The default is 44818.

● **Note:** The Allen-Bradley ControlLogix サーバードライバー currently limits the number of channels to one. If the network adapter and port conflicts with another application on the host machine, the driver cannot accept inbound EtherNet/IP connections.

● For more information, refer to [Event Log Messages](#).

デバイスのプロパティ - 一般

デバイスは、通信チャンネル上の 1 つのターゲットを表します。ドライバ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。

● **関連項目:** プロパティ情報に関する API ドキュメントについては、`/config/v1/doc/drivers` エンドポイントを参照してください。

プロパティグループ	<div style="border: 1px solid gray; padding: 2px;"> <div style="background-color: #e0e0e0; padding: 2px;">識別</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">名前</td> <td colspan="2" style="padding: 2px;">Device1</td> </tr> <tr> <td style="padding: 2px;">説明</td> <td colspan="2" style="padding: 2px;"></td> </tr> <tr> <td style="padding: 2px;">ドライバー</td> <td colspan="2" style="padding: 2px;">Simulator</td> </tr> <tr> <td style="padding: 2px;">モデル</td> <td colspan="2" style="padding: 2px;">16 Bit Device</td> </tr> <tr> <td style="padding: 2px;">チャンネル割り当て</td> <td colspan="2" style="padding: 2px;">Channel1</td> </tr> <tr> <td style="padding: 2px;">ID フォーマット</td> <td colspan="2" style="padding: 2px;">10 進数</td> </tr> <tr> <td style="padding: 2px;">ID</td> <td colspan="2" style="padding: 2px;">1</td> </tr> </table> </div>		名前	Device1		説明			ドライバー	Simulator		モデル	16 Bit Device		チャンネル割り当て	Channel1		ID フォーマット	10 進数		ID	1	
名前	Device1																						
説明																							
ドライバー	Simulator																						
モデル	16 Bit Device																						
チャンネル割り当て	Channel1																						
ID フォーマット	10 進数																						
ID	1																						
<div style="background-color: #e0e0e0; padding: 2px;">一般</div>																							
スキャンモード																							

識別

「名前」: 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、複数のチャン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。デバイス名とチャン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、チャンネル名とデバイス名の組み合わせが "<チャンネル名>.<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**」: このデバイスに関するユーザー定義情報を指定します。

● 「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名を指定します。

「**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 ID に関連付けられてい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、ドライバーに関するマニュアルを参照してください。

「**ID**」: デバイスのドライバー固有のステーションまたはノードを指定します。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。多くの通信ドライバーでは、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。フォーマットはデフォルトではドライバーによって設定されます。オプションには「10 進数」、「8 進数」、「16 進数」があります。

● **注記:** 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

プロパティグループ	+ 識別	
一般	- 動作モード	
スキャンモード	データコレクション	無効化
自動格下げ	シミュレーション	いいえ
タグ生成	+ タグ数	

「**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**」: デバイスをシミュレーションモードに切り替えるかどうかを指定します。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

バー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。(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. クライアントが切断して再接続するまで、更新は適用されません。
2. 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。
3. 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。
4. デバイスをシミュレートしたときに、クライアントで更新が 1 秒未満で表示されない場合があります。

● 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。

タグ数

プロパティグループ	+ 識別	
一般	+ 動作モード	
スキャンモード	- タグ数	
	静的タグ	0

「静的タグ」: デバイスレベルまたはチャンネルレベルで定義される静的タグの数を指定します。この情報は、トラブルシューティングと負荷分散を行う場合に役立ちます。

Device Properties — Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.

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		

Scan Mode: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- **Respect Client-Specified Scan Rate:** This mode uses the scan rate requested by the client.
- **Request Data No Faster than Scan Rate:** This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
 - **Note:** When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- **Request All Data at Scan Rate:** This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.

- **Do Not Scan, Demand Poll Only:** This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the _DemandPoll tag or by issuing explicit device reads for individual items. *For more information, refer to "Device Demand Poll" in server help.*
- **Respect Tag-Specified Scan Rate:** This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

Initial Updates from Cache: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

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。

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」: OPC タグを自動的に生成するタイミングを指定します。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。

Device Properties — Controller Module



Module Type: Verify if the device is Local (part of the simulated EtherNet/IP Module) or Remote (which require a slot number for EtherNet/IP routing). There can be one Local CPU and up to fifteen Remote CPUs.

● **Note:** Each ControlLogix server device must be configured to represent a ControlLogix 5000 Series controller.

- **Local:** When Local, the Controller Module is treated as a CPU local to the simulated EtherNet/IP Module. There can only be one Local Controller Module per channel. The default is enabled.
- **Remote:** When Remote, the Controller Module is treated as a CPU separate from the simulated EtherNet/IP Module. There can be up to fifteen remote controller modules per channel. When this option is enabled, a slot must also be specified. The default is disabled.
 - **Note:** This is disabled and forced to Remote (1) if another device is already configured as Local.

Slot: This property is part of the routing path to the Controller Module. It only contains slots that are currently available for the channel / device being configured. When a new slot is specified, the previous slot is made available for use in another device.

Path: This property represents the routing path to the Controller Module from the ControlLogix client device. It should be used during ControlLogix Client Configuration in the PLC.

● For more information on configuring ControlLogix server devices, refer to [ControlLogix Client Device Configuration](#).

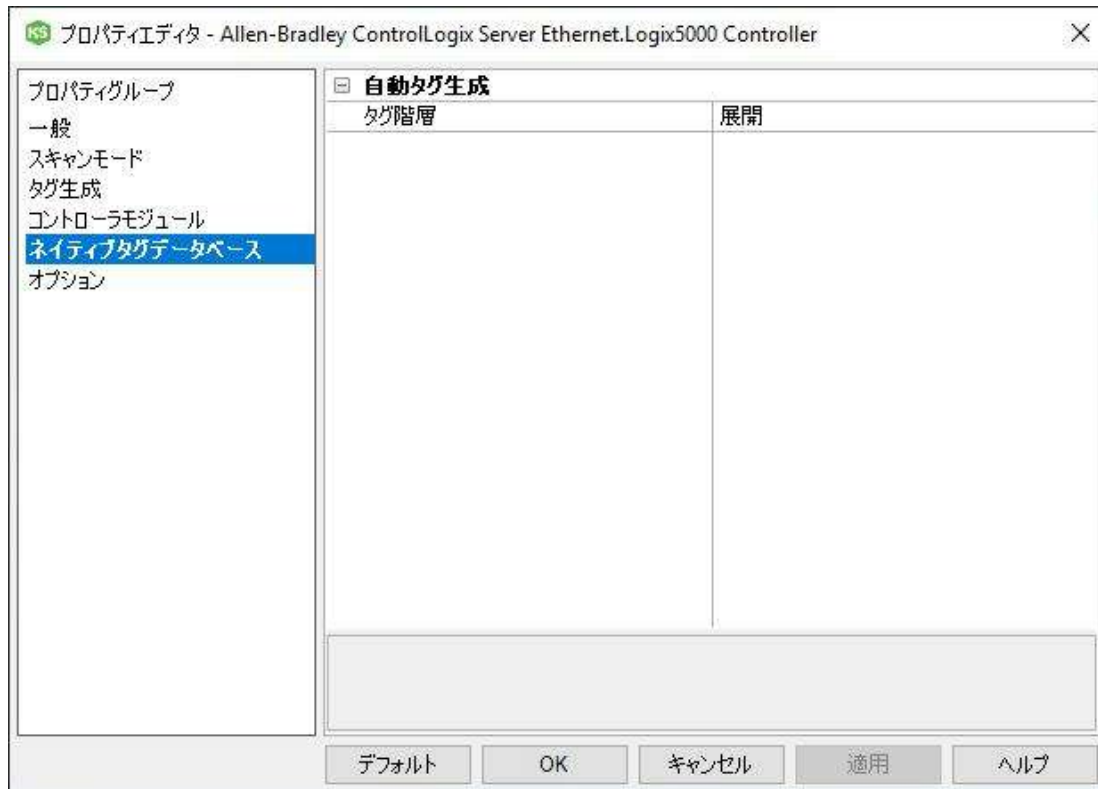
Device Properties — Native Tag Database

The Allen-Bradley ControlLogix サーバードライバー can be configured to automatically generate a list of server tags that correspond to the Native Tag Database. This is also referred to as the Logix Configuration (or Logix

Config). Logix Addresses in the Native Tag Database must be of pre-defined Logix Atomic Data Types, but may also be a part of a structured type.

The driver generates an OPC server tag for each Atomic Tag defined in the Native Tag Database. For array types, a server tag is defined for each element of the array. Array Tags can quickly increase the number of tags imported, as well as the number of tags available in the OPC server. Automatically generated tags are always configured with a client access of Read/Write.

● For more information, refer to [Logix Config](#).



Auto Tag Generation

● For more information, refer to [Device Properties — Tag Generation](#).

Tag Hierarchy: Select if the tag display "tree" organization should be Expanded or Condensed. The default is Expanded.

- **Expanded:** generated client tags are grouped similarly to RSLogix, with groups created for each segment (following a period in the tag address), structure, substructure, and array.
- **Condensed:** generated client tags are grouped similarly to the tag addressing, with groups created for each segment preceding a period.

● For more information on the Native Tag Database CSV format required for import, refer to [Native Tag Database CSV Import](#).

Tag Hierarchy

The automatically generated OPC server tags can follow one of two hierarchies: Expanded or Condensed. The default is Expanded Mode.

Expanded Mode

In Expanded Mode, the automatically generated server tags follow a Group/Tag hierarchy consistent with the tag hierarchy in RSLogix5000. Groups are created for each segment that precedes a period and are created in logical groupings. Groups created include the following:

- Global (Controller) Scope
- Structures and Substructures
- Arrays

● **Note:** Groups are not created for .bit addresses.

Basic Global Tags

Basic Global Tags (or non-structure, non-array tags) are placed under the Global group. Each Structure and Array Tag is provided with its own subgroup of the parent group. By organizing the data in this fashion, the OPC server's Tag View mimics RSLogix5000.

● **Note:** The name of the Structure/Array subgroup also provides a description. For example, an array "tag1 [1,6]" defined in the controller has a subgroup name "tag1[x,y]". In this example, x signifies that dimension 1 exists, and y signifies that dimension 2 exists. Furthermore, the tags within an array subgroup are the elements of that array unless explicitly limited. The tags within a structure subgroup are the structure members themselves. A structure that contains an array has an array subgroup of the structure group created as well.

Array Tags

A group is created for each array that contains the array's elements. Group names have the notation <array name>[x,y,z], where:

- **[x,y,z]:** 3 dimensional array.
- **[x,y]:** 2 dimensional array
- **[x]:** 1 dimensional array

● **Note:** Array Tags have the notation <tag element>_XXXX_YYYYY_ZZZZ. For example, element "tag1 [12,2,987]" has the tag name "tag1_12_2_987".

Simple Example

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

プロジェクト	Logix アドレス	Logix データ型	アクセスタイプ	説明
接続性				
チャンネル				
デバイス				
グローバル				
マイタグ				
メンバー				
	MEMBER1	DINT	読み取り専用	メンバー1
	MEMBER2	DINT	読み取り専用	メンバー2
	MEMBER3	DINT	読み取り専用	メンバー3
	MEMBER4	DINT	読み取り専用	メンバー4
	MEMBER5	DINT	読み取り専用	メンバー5
	MEMBER6	DINT	読み取り専用	メンバー6
	MEMBER7	DINT	読み取り専用	メンバー7
	MEMBER8	DINT	読み取り専用	メンバー8
	MEMBER9	DINT	読み取り専用	メンバー9

Complex Example

A Logix Address defined with the address "MyStructArray[0].MySubStruct.Data" would be represented in the following groups: "Global," "MYSTRUCTARRAY[x]," "MYSTRUCTARRAY[0]," and "MYSUBSTRUCT". The tag "DATA" would be in the last group. The static reference to "DATA" would be "Channel1.Device1.Global.MYSTRUCTARRAY[X].MYSTRUCTARRAY[0].MYSUBSTRUCT.DATA". The dynamic reference would be "Channel1.Device1. MyStructArray[0].MySubStruct.Data".

● For more information, refer to "Static Tags (User-Defined)" and "Dynamic Tags" in server help file.

Condensed Mode

In Condensed Mode, the automatically generated server tags follow a group / tag hierarchy consistent with the tag's address. Groups are created for each segment that precedes the period. Groups created include the following:

- Program Scope
- Structures and Substructures

Notes:

1. Groups are not created for arrays or .bit addresses.
2. Tag or structure member names that start with an underscore are converted to "U". This is required because the server does not support leading underscores in tag name fields.

Simple Example

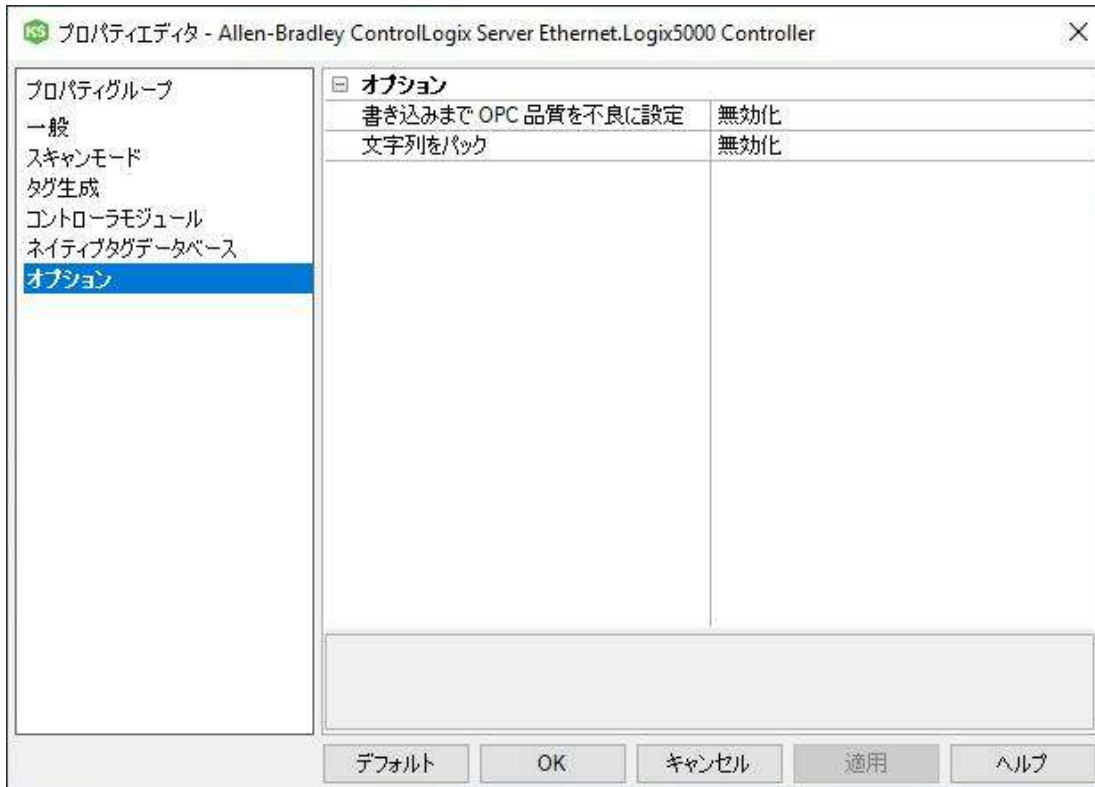
Name	Value	Force Mask	Style	Data Type
MyTag	{...}	{...}		MyDataType
MyTag.Member1	{...}	{...}	Decimal	DINT[10]
MyTag.Member1[0]	0		Decimal	DINT
MyTag.Member1[1]	0		Decimal	DINT
MyTag.Member1[2]	0		Decimal	DINT
MyTag.Member1[3]	0		Decimal	DINT

プロジェクト	Logix アドレス	Logix データ型	アクセスタイプ	説明
接続性	MEMBER1	DINT	読み取り専用	メンバー1
チャンネル	MEMBER2	DINT	読み取り専用	メンバー2
メンバー	MEMBER3	DINT	読み取り専用	メンバー3
	MEMBER4	DINT	読み取り専用	メンバー4
	MEMBER5	DINT	読み取り専用	メンバー5
	MEMBER6	DINT	読み取り専用	メンバー6
	MEMBER7	DINT	読み取り専用	メンバー7
	MEMBER8	DINT	読み取り専用	メンバー8
	MEMBER9	DINT	読み取り専用	メンバー9

Complex Example

A Logix Address defined with address "MyStructArray[0].MySubStruct.Data" would be represented in the following groups: "MYSTRUCTARRAY[0]" and "MYSUBSTRUCT". The tag "DATA" would be in the last group. The static reference to "DATA" would be "Channel1.Device1.MYSTRUCTARRAY[0].MYSUBSTRUCT.DATA" and the dynamic reference would be "Channel1.Device1.MyStructArray[0].MySubStruct.Data".

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。書き込み操作は、クライアントインタフェース (OPC など) または ControlLogix クライアント (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 サーバードライバー 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。

ControlLogix クライアントデバイスの構成

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 ControlLogix サーバードライバーに関連付けられているルーティングパスは、「デバイスプロパティ」の「コントローラモジュール」に表示されます。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

● サポートされている Logix タイプの詳細については、[アドレスの説明](#)を参照してください。

エラーコード

Allen-Bradley ControlLogix サーバードライバー 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。

● ControlLogix クライアントデバイスに返される可能性の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。

Logix Config Object

When a device is created, an object named Logix Config is created. Logix Config is also referred to as the Native Tag Database. This object defines the address space of the device, defining what addresses will be accepted for incoming Logix or OPC clients. The objects that represent this namespace are referred to as Logix Addresses or Native Tags throughout this help file. Logix Addresses may be created individually or imported in bulk from a CSV file.

Notes:

1. The total memory size of a Native Tag (Logix Address) must not exceed 128 kB.
2. The total memory size of a Native Tag Database (Logix Config) must not exceed 512 kB.

Logix Config CSV Import

The device address space may be configured by importing Logix Addresses from a CSV file. To import a CSV file, right click **Logix Config** and select **Import CSV**.

A CSV file specifies the Native Tags that each device can represent (its address space). It is used once for tag import and is not required for automatic tag database generation or remote deployment. When a CSV import is triggered by the user, all existing Logix Addresses in the Logix Config object will be deleted and replaced with the imported Logix Addresses. After the Logix Addresses have been imported, an ATG will be triggered creating static tags that represent the address space of the device. The following CSV header must be used for Native Tag import:

Logix Address	Logix DataType	External Access	Description
---------------	----------------	-----------------	-------------

Definitions and requirements for each column in the CSV file are discussed below.

Tip: A template Native Tag Database CSV file is included for reference in *<Server Installation Directory>/Drivers/controllogix_unsolicited_ethernet/import_template.csv*.

Importing Native Tags as SINT, INT, and DINT Arrays

Native Tags that are imported as SINT, INT, and DINT arrays also have a string tag defined that uses the number of elements of the corresponding array in the tag address. Examples are as follows:

- If a Native Tag called "MySINTarray @ SINT[100]" is imported, a Static Tag with the address "MYSINTARRAY / 100" and String data type is generated.
- If a Native Tag called "MyINTarray @ INT[100]" is imported, a Static Tag with the address "MYINTARRAY / 100" and String data type is generated.
- If a Native Tag called "MyDINTarray @ DINT[100]" is imported, a Static Tag with the address "MYDINTARRAY / 100" and String data type is generated.

Tip: To import RSLogix5000 pre-defined Strings, the two elements contained within the String types ("STRING.DATA" and "STRING.LEN") should be defined in the Native Tag Database CSV file before the import is performed.

Logix Address

Restrictions on the Logix Address are consistent with RSLogix5000 requirements, which correspond to the following IEC 1131-3 identifier rules:

- Must begin with an alphabetic character (A-Z, a-z) or an underscore.
- Can only contain alphanumeric characters and underscores.
- Can have as many as 40 characters in each segment.
- Cannot have consecutive underscores.
- Are not case sensitive.

Tags that do not have a unique Logix Address or meet the identifier requirements above fail to import, causing a message with the specified Logix Address to be posted to the OPC server's Event Log.

Logix DataType

The following pre-defined Logix Atomic Types are supported:

Logix Data Type	Supported Data Types
BOOL	Boolean
SINT	Char, Byte
INT	Short, Word
DINT	Long, DWord
LINT	Double, Date
REAL	Float

Array Syntax: When defining the data type, up to three dimensional arrays are allowed for most types. Array syntax is specified using brackets "[]" (for example a DataType of DINT[5,5] specifies a two-dimensional array with five elements in each dimension).

Special Rules: The following rules apply to array data types:

1. Boolean arrays are limited to a single dimension.
2. Boolean arrays must be DWORD-aligned; the dimension size must be a multiple of 32 bits.
3. The total memory size of a Native Tag (Logix Address) must not exceed 128 kB.
4. The total memory size of a Native Tag Database (Logix Config) must not exceed 512 kB.

● **Note:** Other pre-defined or user-defined complex (structured) data types are not supported. Structured data can be imported by qualifying the Logix Address down to the atomic type. For example, there exists a structured type called TIME, which is described as the following:

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

The structure can be broken down and the atomic member imported as "TIME.HOUR," "TIME.MIN," and "TIME.SEC" with the associated Logix data type, External Access, and Description following the CSV format outlined above. All unsupported Logix data type values specified in the CSV import are defaulted to DINT so that the import succeeds.

Access Type

The Access Type specifies the ControlLogix client devices' read / write privileges. This access does not apply to OPC client tags, which always have a client access of read / write by default (this setting is configurable using static OPC server tags). The following external access types are supported: all other values specified are set to read / write.

- **R/W**: ControlLogix client devices have Read / Write permissions to the Native Tag.
- **RO**: ControlLogix client devices have Read Only permissions. All write attempts fail with the appropriate error (CIP error 0x0F).

Description

Descriptions are used during automatic tag database generation, truncated to 64 characters. The Description field must be present, but may be left blank.

Defining New Logix Addresses

To define a new Logix Address, right-click **Logix Config** in the Tree View and select **New Logix Address**. Alternatively, select **Logix Config** in the Tree View, right-click the Detail View, then select **New Logix Address**.

Defining Logix Address Properties

プロパティグループ	<input type="checkbox"/> 識別 説明	
一般	<input type="checkbox"/> Logix アドレス定義 Logix アドレス Logix データ型 アクセスタイプ	
		DINT
		読み取り専用

Description: This property provides a brief summary of the object or its use.

Logix Address: This property specifies the address for the Logix Address. See [CSV Import](#) or [Logix Tag-Based Addressing](#) for more information about valid Logix addresses.

Logix DataType: This property specifies the data type for the Logix Address. See [CSV Import](#) for more information about valid Logix data types.

External Access: This property specifies whether the tag has Read-Only or Read / Write access. After creating a new Logix Address, it will appear in the tag list in the Detail View

Data Types Description

The following describes server data types supported by the Allen-Bradley ControlLogix サーバードライバー when defining static tags. For data types allowed when defining the address space of the device, see [Logix Tag-Based Addressing](#).

Data Types	Description
Boolean	Single bit
Char	Signed 8-bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Byte	Unsigned 8-bit value bit 0 is the low bit bit 7 is the high bit
Short	Signed 16-bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
Word	Unsigned 16-bit value bit 0 is the low bit bit 15 is the high bit
Long	Signed 32-bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
DWord	Unsigned 32-bit value bit 0 is the low bit bit 31 is the high bit
Float	32-bit floating point value bit 0 is the low bit bit 31 is the high bit
Double	64-bit floating point value bit 0 is the low bit bit 63 is the high bit
String	Typically null-terminated, null padded, or blank padded ASCII string.
Date	64-bit floating point value.

● For a description of Logix platform-specific data types, refer to [Address Descriptions](#).

アドレスの説明

Allen-Bradley ControlLogix サーバードライバー 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 を OPC サーバータグに割り当てることはできませんが、TIMER のアトミックメンバー (TIMER.EN や TIMER.ACC など) については、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

OPC クライアント/サーバータグアドレスの規則

Logix アドレスは、OPC クライアント/サーバータグのアドレスに相当します。Logix アドレス (RSLogix5000 から入力) は IEC 1131-3 の識別子の規則に従います。OPC 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

1. OPC 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、名前の先頭がアンダースコアであってはなりません。
2. タグが適切に検証されるためには、静的 OPC 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

OPC サーバーで静的にまたは OPC クライアントから動的に Logix アドレス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方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 OPC 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。
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。
String	<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。

Tag Scope

Global Tags

Global Tags are Logix Addresses that have global scope in the controller. Any program or task can access Global Tags; however, the number of ways a Global Tag can be referenced depends on its Logix data type and the address format being used.

Program Tags

Program Tags are identical to Global Tags except that their scope is local to the program in which it is defined. The driver does not currently support importing Native Tags with a program designation.

Structure Tag Addressing

Logix Structure Tags are tags with one or more member tags (which can be atomic or structured).
`<structure name> . <atomic-type tag>`

This implies that a substructure would be addressed as the following:

`<structure name> . <substructure name> . <atomic-type tag>`

Arrays of structures would be addressed as the following:

`<structure array name> [dim1, dim2, dim3] . <atomic-type tag>`

This implies that an array of substructures would be addressed as the following:

`<structure name> . <substructure array name> [dim1, dim2, dim3] . <atomic-type tag>`

● **Note:** The examples above display a few of the addressing possibilities that involve structures and are only provided as an introduction to structure addressing. *For more information, refer to Rockwell/Allen-Bradley documentation.*

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} <配列タグ名> {行数} {列数}	tag_1 {8} tag_1 {2} {4}	読み書きする要素の数は行数 × 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

要素	構文	例	注記
			のインデックスが0)。
オフセットがある 配列	<配列要素タグ> [オフセット] {列数} <配列要素タグ> [オフセット] {行数} {列数}	tag_1 [5] {8} tag_1 [5] {2} {4}	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)。
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 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]
```

Error Codes

The Allen-Bradley ControlLogix サーバードライバー may return the following error codes.

• For more information on a specific type of error code, select a link from the list below.

[EtherNet/IP Encapsulation Error Codes](#)

[CIP Error Codes](#)

[0x01 Extended Error Codes](#)

[0xFF Extended Error Codes](#)

EtherNet/IP Encapsulation Error Codes

The Allen-Bradley ControlLogix サーバードライバー may return the following error codes.

• **Note:** The error codes are in hexadecimal.

Error	Description
0001	Command not handled
0002	Memory not available for command
0003	Poorly formed or incomplete data
0064	Invalid Session ID
0065	Invalid length in header
0069	Requested protocol version not supported

CIP Error Codes

The error codes are in hexadecimal.

Error	Description
01	Connection failure*
02	Insufficient resources
03	Parameter value invalid
04	IOI could not be deciphered or tag does not exist
05	Unknown destination
06	Data requested would not fit in response packet
08	Unsupported service
0F	Permission denied
13	Insufficient command data / parameter specified to execute service
26	The number of IOI words specified does not match IOI word count
FF	General Error**

• ***See Also:** [0x01 Extended Error Codes](#)

• ****See Also:** [0xFF Extended Error Codes](#)

0x01 Extended Error Codes

The Allen-Bradley ControlLogix サーバードライバー may return the following extended errors for CIP error 0x01.

● **Note:** The error codes are in hexadecimal.

Error	Description
0x0205	Unconnected Send parameter error.
0x0312	Link address is not available.
0x0318	Link address to self is invalid.

0xFF 拡張エラーコード

Allen-Bradley ControlLogix サーバードライバー 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ました。
2107	データ型が無効であるかサポートされていません。

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

Tip: Messages that originate from a data source (such as third-party software, including databases) are presented through the Event Log. Troubleshooting steps should include researching those messages online and in vendor documentation.

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 などがあります。

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>'。

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 = '<アドレス>'。

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 512 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、512 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:

512 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

CSV インポートに失敗しました。既存の Logix アドレスを削除できません。

エラータイプ:

警告

CSV インポートに失敗しました。新規 Logix アドレスを作成できません。

エラータイプ:

警告

**フラグメント化された書き込みの処理中に予期しないフラグメントが検出されました。
| Logix アドレス = <タグ>。**

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 Extended Error Codes 31

0xFF 拡張エラーコード 31

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>'。 34

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 = '<アドレス>'。 34

B

BOOL 25

Boolean 24

Byte 24

C

Channel Properties — Advanced 7

Channel Properties — Ethernet Communications 6

Channel Properties — Write Optimizations 7

Char 24

CIP Error Codes 30

Condensed 15

Controller Module 14

ControlLogix クライアントデバイスの構成 18

CSV file 20

CSV インポートに失敗しました。既存の Logix アドレスを削除できません。 36

CSV インポートに失敗しました。新規 Logix アドレスを作成できません。 36

D

Data Types Description 24
Date 24
Defining New Logix Addresses 23
DINT 25
Do Not Scan, Demand Poll Only 12
Double 24
Duty Cycle 7
DWord 24

E

Error Codes 30
Ethernet Settings 6
EtherNet/IP Encapsulation Error Codes 30
EtherNet/IP Module 8
Event Log Messages 32
Expanded 15
External Access 23

F

Float 24

I

ID 10
Initial Updates from Cache 12
INT 25
Inter-Device Delay 8

L

LINT 25
Local 14
Logix Address 23

Logix Config CSV Import 20
Logix Config Object 20
Logix DataType 23
Logix タグベースのアドレス指定 25
Logix 配列データの順序 28
Long 24

M

Module Type 14

N

Native Tag Database 14
Network Adapter 6
Non-Normalized Float Handling 8

O

Optimization Method 7

P

Path 14

R

REAL 25
Remote 14
Replace with Zero 8
Respect Tag-Specified Scan Rate 12

S

Scan Mode 11
Short 24
SINT 25
Slot 14

String 24, 26

T

Tag Hierarchy 15

Tag Scope 27

TCP/IP Port 9

U

Unmodified 8

W

Word 24

Write All Values for All Tags 7

Write Only Latest Value for All Tags 7

Write Only Latest Value for Non-Boolean Tags 7

あ

アドレスのフォーマット 26

アドレスの説明 25

お

オプション 18

オフセットがある配列 26

オフセットがない配列 26

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>。 36

さ

サブグループを許可 13

し

シミュレーション 11

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>'。 34

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>'。 35

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>'。 36

タグ数 6, 11

タグ生成 12

ち

チャンネルのプロパティ - 一般 5

チャンネル割り当て 10

て

データコレクション 10

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 - 一般 9

デバイス起動時 12

と

ドライバー 10

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>'。 36

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>'。 35

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。 32

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。

33

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。 34

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。

32

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>'。 32

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。 32

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>'。 33

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>'。 33

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512 KB に制限されています。| タグアドレス = '<アドレス>'。 35

ひ

ビット 26

ふ

フラグメント化された書き込みの処理中に予期しないフラグメントが検出されました。| Logix アドレス = <タグ>。 36
プロパティ変更時 12

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>'。 35

も

モデル 10

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。 37

漢字

一般 9

概要 4

作成 13

削除 13

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>。 36

識別 5, 9

重複タグ 13

書き込みまで OPC 品質を不良に設定 18

詳細なアドレス指定 27

上書き 13

親グループ 13

診断 6

生成 13

設定 5

動作モード 10

配列要素 26

非送信請求 Logix サーバーを起動できませんでした。 34

標準 26

文字列をパック 18

名前 9