

Allen-Bradley ControlLogix Ethernet ドライバー

© 2024 PTC Inc. All Rights Reserved.

目次

Allen-Bradley ControlLogix Ethernet ドライバー	1
目次	2
Allen-Bradley ControlLogix Ethernet ドライバー	10
概要	11
設定	11
チャンネルのプロパティ - 一般	15
タグ数	15
チャンネルのプロパティ - イーサネット通信	15
チャンネルのプロパティ - 書き込み最適化	16
チャンネルのプロパティ - 詳細	16
デバイスのプロパティ - 一般	17
動作モード	18
デバイスのプロパティ - スキャンモード	18
タグ数	19
デバイスのプロパティ - タイミング	19
デバイスのプロパティ - 自動格下げ	20
デバイスのプロパティ - タグ生成	20
デバイスのプロパティ - Logix 通信パラメータ	22
デバイスのプロパティ - Logix オプション	24
デバイスのプロパティ - Logix データベース設定	25
デバイスのプロパティ - ENI DF1/DH+/CN ゲートウェイ通信パラメータ	26
ブロック書き込み	27
デバイスのプロパティ - SLC 500 スロット構成	27
デバイスのプロパティ - 冗長	29
SLC 500 モジュール I/O 選択ガイド	30
Configuration API - Allen-Bradley ControlLogix Ethernet の例	33
列挙	34
デバイスモデル列挙	35
Configuration API - スロット構成	35
自動タグデータベース生成	37
タグ階層	38
コントローラからサーバーへの変換	40
自動タグデータベース生成の準備	41
パフォーマンスの最適化	42
通信の最適化	42
アプリケーションの最適化	43
パフォーマンス統計とチューニング	44
パフォーマンスチューニングの例	45
データ型の説明	56
デフォルトデータ型の条件	57
アドレスの説明	58

Logix のアドレス指定	58
MicroLogix のアドレス指定	59
SLC 500 固定 I/O のアドレス指定	62
SLC 500 モジュラー I/O のアドレス指定	62
PLC-5 シリーズのアドレス指定	63
Logix タグベースのアドレス指定	65
アドレスのフォーマット	66
タグの有効範囲	67
内部タグ	67
定義済みの用語タグ	68
アトミックデータ型のアドレス指定	68
構造体データ型のアドレス指定	70
STRING データ型のアドレス指定	71
Logix 配列データの順序	72
Logix の詳細なアドレス指定	73
BOOL の詳細なアドレス指定	73
SINT の詳細なアドレス指定	74
INT の詳細なアドレス指定	76
DINT の詳細なアドレス指定	79
LINT の詳細なアドレス指定	81
REAL の詳細なアドレス指定	82
USINT の詳細なアドレス指定	84
UINT の詳細なアドレス指定	86
UDINT の詳細なアドレス指定	87
ULINT の詳細なアドレス指定	89
LREAL の詳細なアドレス指定	90
詳細なアドレス指定: TIME32	92
詳細なアドレス指定: TIME	93
詳細なアドレス指定: LTIME	94
ファイル一覧	96
出力ファイル	96
入力ファイル	99
ステータスファイル	102
バイナリファイル	104
タイマーファイル	104
カウンタファイル	105
制御ファイル	105
整数ファイル	106
Float ファイル	108
ASCII ファイル	108
文字列ファイル	109
BCD ファイル	109
Long ファイル	110
MicroLogix PID ファイル	110

PID ファイル	111
MicroLogix メッセージファイル	113
メッセージファイル	114
ブロック転送ファイル	114
ファンクションファイル	116
高速カウンタファイル (HSC)	116
リアルタイムクロックファイル (RTC)	117
チャンネル 0 通信ステータスファイル (CS0)	118
チャンネル 1 通信ステータスファイル (CS1)	118
I/O モジュールステータスファイル (IOS)	119
エラーコード	120
カプセル化 エラーコード	120
CIP エラーコード	120
0x0001 拡張エラーコード	121
0x00C 拡張エラーコード	122
0x00FF 拡張エラーコード	122
イベントログメッセージ	123
デバイスからコントローラプロジェクトをアップロード中に次のエラーが発生しました。シンボリックプロトコルを使用します。	123
同期化中に無効または破損したコントローラプロジェクトが検出されました。まもなく同期化を再試行します。	123
同期化中にプロジェクトのダウンロードが検出されました。まもなく同期化を再試行します。	123
データベースエラー。参照タグのデータ型が不明です。エイリアスタグのデータ型をデフォルトに設定します。 参照タグ = '<タグ>'、エイリアスタグ = '<タグ>'、デフォルト データ型 = '<タイプ>'。	123
データベースエラー。タグインポートファイルでメンバーのデータ型が見つかりません。データ型をデフォルトに設定します。 メンバーのデータ型 = '<タイプ>'、UDT = '<タイプ>'、デフォルト データ型 = '<タイプ>'。	124
データベースエラー。タグインポートファイルでデータ型が見つかりません。タグは追加されません。 データ型 = '<タイプ>'、タグ名 = '<タグ>'。	124
データベースエラー。エイリアスタグの処理中にエラーが発生しました。タグは追加されませんでした。 エイリアスタグ = '<タグ>'。	124
データベースエラー。レジスタセッションの要求時にカプセル化エラーが発生しました。 カプセル化エラー = <コード>。	124
データベースエラー。レジスタセッションの要求時にフレーミングエラーが発生しました。	125
データベースエラー。フォワードオープンの要求時にカプセル化エラーが発生しました。 カプセル化エラー = <コード>。	125
データベースエラー。フォワードオープンの要求時にフレーミングエラーが発生しました。	125
データベースエラー。フォワードオープンの要求時にエラーが発生しました。 CIP エラー = <コード>、拡張エラー = <コード>。	125
データベースエラー。プロジェクト情報のアップロード中にカプセル化エラーが発生しました。 カプセル化エラー = <コード>。	125
データベースエラー。プロジェクト情報のアップロード中にエラーが発生しました。 CIP エラー = <コード>、拡張エラー = <コード>。	125
データベースエラー。プロジェクト情報のアップロード中にフレーミングエラーが発生しました。	126
データベースエラー。内部エラーが発生しました。	126
データベースエラー。プログラム情報のアップロード中にカプセル化エラーが発生しました。 プログラム名 = '<名前>'、カプセル化エラー = <コード>。	126
データベースエラー。プログラム情報のアップロード中にエラーが発生しました。 プログラム名 = '<名前>'、	127

CIP エラー = <コード>、拡張エラー = <コード>。	
データベースエラー。プログラム情報のアップロード中にフレーミングエラーが発生しました。 プログラム名 = '<名前>'。	127
データベースエラー。タグの CIP データ型を解決できません。デフォルトの型に設定します。 CIP データ型 = <タイプ>、タグ名 = '<タグ>'、デフォルトデータ型 = '<タイプ>'。	127
プロジェクト情報のアップロード中にカプセル化エラーが発生しました。 カプセル化エラー = <コード>。	128
プロジェクト情報のアップロード中にエラーが発生しました。 CIP エラー = <コード>、拡張エラー = <コード> >。	128
プロジェクト情報のアップロード中にフレーミングエラーが発生しました。	128
プログラム情報のアップロード中にカプセル化エラーが発生しました。 プログラム名 = '<名前>'、カプセル化エラー = <コード>。	129
プログラム情報のアップロード中にエラーが発生しました。 プログラム名 = '<名前>'、CIP エラー = <コード> >、拡張エラー = <コード>。	129
プログラム情報のアップロード中にフレーミングエラーが発生しました。 プログラム名 = '<名前>'。	129
コントローラプログラム情報のアップロード中にカプセル化エラーが発生しました。カプセル化エラー = <コード> >。	129
コントローラプログラム情報のアップロード中にエラーが発生しました。CIP エラー = <コード>、拡張エラー = <コード>。	129
コントローラプログラム情報のアップロード中にフレーミングエラーが発生しました。	129
プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。	129
データベースエラー。プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。	129
データベースエラー。フォワードオープンの要求に利用可能な接続はもうありません。	130
タグデータベースのインポート用のファイルを開くときにエラーが発生しました。 OS エラー = '<コード>'。	130
サポートされていないコントローラです。 ベンダー ID = <ID>、製品タイプ = <タイプ>、製品コード = <コード> >、製品名 = '<名前>'。	130
デバイスから受信したフレームにエラーが含まれています。	130
フレーミングエラーにより書き込み要求が失敗しました。 タグアドレス = '<アドレス>'。	130
フレーミングエラーによりタグの読み取り要求が失敗しました。 タグアドレス = '<アドレス>'。	130
フレーミングエラーによりブロック読み取り要求が失敗しました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。	131
フレーミングエラーによりブロック読み取り要求が失敗しました。 ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。	131
タグに書き込めません。 タグアドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。	131
タグを読み取れません。 タグアドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。	132
ブロックを読み取れません。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。	132
ブロックを読み取れません。 ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>'、CIP エラー = <コード>、拡張エラー = <コード>。	132
タグに書き込めません。コントローラタグのデータ型が不明です。 タグアドレス = '<アドレス>'、データ型 = <タイプ>。	132
タグを読み取れません。コントローラタグのデータ型が不明です。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'、データ型 = <タイプ>。	132
ブロックを読み取れません。コントローラタグのデータ型が不明です。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。	133
タグに書き込めません。データ型がサポートされていません。 タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。	133
タグを読み取れません。データ型がサポートされていません。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。	133
ブロックを読み取れません。データ型がサポートされていません。ブロックは非アクティブ化されました。 ブロ	134

クサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。	
タグに書き込めません。このタグには不正なデータ型です。 タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。	134
タグを読み取れません。このタグには不正なデータ型です。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。	134
ブロックを読み取れません。このブロックには不正なデータ型です。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。	134
タグに書き込めません。タグは複数要素の配列をサポートしません。 タグアドレス = '<アドレス>'。	135
タグを読み取れません。タグは複数要素の配列をサポートしません。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'。	135
ブロックを読み取れません。ブロックは複数要素の配列をサポートしません。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。	135
タグに書き込めません。ネイティブタグのサイズが不一致です。 タグアドレス = '<アドレス>'。	136
タグを読み取れません。ネイティブタグのサイズが不一致です。 タグアドレス = '<アドレス>'。	136
ブロックを読み取れません。ネイティブタグのサイズが一致しません。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。	136
ブロックを読み取れません。ネイティブタグのサイズが一致しません。 ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。	136
タグに書き込めません。 タグアドレス = '<アドレス>'。	137
タグを読み取れません。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'。	137
ブロックを読み取れません。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。	137
ブロックを読み取れません。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>'。	138
デバイスへの要求中にエラーが発生しました。 CIP エラー = <コード>、拡張エラー = <コード>。	138
デバイスへの要求中にカプセル化エラーが発生しました。 カプセル化エラー = <コード>。	138
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>'。	139
ブロックを読み取れません。受信したフレームにエラーが含まれています。 ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'。	139
デバイスからファンクションファイルを読み取れません。受信したフレームにエラーが含まれています。 ファンクションファイル = '<名前>'。	139
ブロックを読み取れません。タグは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	139
デバイスからファンクションファイルを読み取れません。タグは非アクティブ化されました。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	140
アドレスに書き込めません。受信したフレームにエラーが含まれています。 アドレス = '<アドレス>'。	140
ファンクションファイルに書き込めません。受信したフレームにエラーが含まれています。 ファンクションファイル = '<名前>'。	140
ブロックを読み取れません。 ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	140
ファンクションファイルを読み取れません。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	141
ブロックを読み取れません。タグは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	141
ファンクションファイルを読み取れません。タグは非アクティブ化されました。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。	141
アドレスに書き込めません。 アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	142
ファンクションファイルに書き込めません。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。	142

ブロックを読み取れません。 ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>。	143
ファンクションファイルを読み取れません。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。 ..	143
アドレスに書き込めません。 アドレス = '<アドレス>'、DF1 ステータス = <コード>。	143
ファンクションファイルに書き込めません。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。 ...	144
タグを読み取れません。内部メモリが無効です。 タグアドレス = '<アドレス>'。	144
タグを読み取れません。このタグには不正なデータ型です。 タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。	144
ブロックを読み取れません。内部メモリが無効です。タグは非アクティブ化されました。 タグアドレス = '<アドレス>'。	144
ブロックを読み取れません。内部メモリが無効です。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。	145
アドレスに書き込めません。内部メモリが無効です。 タグアドレス = '<アドレス>'。	145
ブロックを読み取れません。ブロックは非アクティブ化されました。 ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。	145
デバイスが応答していません。ローカルノードがエラーを返しました。 DF1 ステータス = <コード>。	145
ファンクションファイルに書き込めません。ローカルノードがエラーを返しました。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。	145
アドレスに書き込めません。ローカルノードがエラーを返しました。 ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。	146
タグで予期しないオフセットが見つかりました。タグはシンボリックプロトコルを使用します。 タグアドレス = '<アドレス>'。	146
タグで予期しないオフセットが見つかりました。 タグアドレス = '<アドレス>'。	146
タグで予期しないオフセット/スパンが見つかりました。 タグアドレス = '<アドレス>'。	146
プロジェクトのダウンロードが進行中であるかプロジェクトが存在しません。	146
プロジェクトのダウンロードが完了しました。	146
プロジェクトのオンライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。	146
プロジェクトのオフライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。	146
デバイスからコントローラプロジェクトをアップロード中に次のエラーが発生しました。シンボリックプロトコルを使用します。	147
デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。 カプセル化エラー = <コード>。	147
デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。 CIP エラー = <コード>、拡張エラー = <コード>。	147
デバイスの識別情報を取得できません。受信したフレームにエラーが含まれています。すべてのタグがシンボリックプロトコルを使用します。	147
要求された CIP 接続サイズはこのデバイスによってサポートされていません。自動的に最大サイズにフォールバックします。 要求されたサイズ = <数値> (バイト)、最大サイズ = <数値> (バイト)。	148
タグのインポートファイル名が無効です。ファイルパスは使用できません。	148
デバイスへの読み取り書き込み要求が中止しました。デバイスプロジェクトからの論理アドレスを更新しています。	148
デバイスへの読み取り書き込み要求が再開しました。デバイスからの論理アドレスの更新が完了しました。現在、論理アドレス指定を使用しています。	148
タグに書き込めません。書き込まれた値には構文エラーが含まれています。 タグアドレス = '<アドレス>'、必要なフォーマット = '<フォーマット>'。	148
タグに書き込めません。書き込まれた値は範囲外です。 タグアドレス = '<アドレス>'。	149
データベースステータス。非エイリアスタグをインポートしています。	149
データベースステータス。エイリアスタグをインポートしています。	149
データベースステータス。タグプロジェクトを構築しています。お待ちください。 タグプロジェクト数 = <数値>。	149

データベースエラー。最大文字長さを超えているため、タグ名が変更されました。 タグ名 = '<タグ>'、最大長さ = <数値>、新しいタグ名 = '<タグ>'。	149
データベースエラー。最大文字長さを超えているため、配列タグの名前が変更されました。 配列タグ = '<タグ>'、最大長さ = <数値>、新しい配列タグ = '<tags>'。	149
データベースエラー。プログラムグループの名前が最大文字長さを超えています。プログラムグループの名前が変更されました。 グループ名 = '<名前>'、最大長さ = <数値>、新しいグループ名 = '<名前>'。	149
データベースステータス。コントローラプロジェクトを読み込んでいます。	150
データベースステータス。 プログラムの数 = <数値>、データ型の数 = <数値>、インポートされたタグの数 = <数値>。	150
データベースステータス。OPC タグを生成しています。	150
メモリリソース量が低下しています。	150
不明なエラーが発生しました。	150
データベースステータス。L5X ファイルからタグをインポートしています。 スキーマリビジョン = '<値>'、ソフトウェアリビジョン = '<値>'。	150
詳細。 IP = '<アドレス>'、ベンダー ID = <ベンダー>、製品タイプ = <タイプ>、製品コード = <コード>、リビジョン = '<値>'、製品名 = '<名前>'、製品シリアル番号 = <数値>。	150
経過時間 = <数値> (秒)。	150
シンボリックデバイスの読み取り回数 = <数値>。	150
シンボリック配列ブロックデバイスの読み取り回数 = <数値>。	150
シンボリック配列ブロックキャッシュの読み取り回数 = <数値>。	150
シンボリックインスタンス非ブロックデバイスの読み取り回数 = <数値>。	150
シンボリックインスタンス非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。	151
シンボリックインスタンス非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。	151
シンボリックインスタンスブロックデバイスの読み取り回数 = <数値>。	151
シンボリックインスタンスブロックキャッシュの読み取り回数 = <数値>。	151
物理非ブロックデバイスの読み取り回数 = <数値>。	151
物理非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。	151
物理非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。	151
物理ブロックデバイスの読み取り回数 = <数値>。	151
物理ブロックキャッシュの読み取り回数 = <数値>。	151
読み取りタグ数 = <数値>。	151
送信パケット数 = <数値>。	151
受信パケット数 = <数値>。	151
初期化トランザクション数 = <数値>。	152
読み取り/書き込みトランザクション数 = <数値>。	152
1 秒あたり平均送信パケット数 = <数値>。	152
1 秒あたり平均受信パケット数 = <数値>。	152
1 秒あたり平均タグ読み取り回数 = <数値>。	152
1 トランザクションあたり平均タグ数 = <数値>。	152
-----	152
%s デバイス統計	152
デバイス平均ターンアラウンドタイム = <数値> (ミリ秒)	152
%s チャネル統計	152
ドライバー統計	152
デバイスタグのインポートが中断しました。	152
インポートファイル '%s' はパス '%s' に見つかりません。	153

コントローラプロジェクトの読み込み中にエラーが発生しました。	153
内部ドライバーエラーが発生しました。	153
同期化中に無効または破損したコントローラプロジェクトが検出されました。後でもう一度試してください。 ..	153
同期化中にプロジェクトのダウンロードが検出されました。後でもう一度試してください。	153
メモリリソース量が低下しています。	153
L5K ファイルが無効であるか破損しています。	153
不明なエラーが発生しました。	153
データベースエラー。PLC5/SLC/MicroLogix デバイスはこの機能をサポートしていません。	153
L5X ファイルが無効であるか破損しています。	153
インポートファイル '<空>' はパス '<空>' に見つかりません。	153
インポートファイル '%s' はパス '<空>' に見つかりません。	153
インポートファイル '<空>' はパス '%s' に見つかりません。	154
XML 要素がポストスキーマの検証に失敗しました。デバイスからのタグのインポートはこのモデルではサポートされていません。代替要素を使用してください。 XML 要素 = '{<名前空間><要素>', サポートしていないモデル = '<モデル>', 代替 XML 要素 = '{<名前空間><要素>'。	154
この値はこのモデルの XML 要素ではサポートされていません。新しい値に自動的に設定します。 値 = '<値>', XML 要素 = '{<名前空間><要素>', モデル = '<モデル>', 新しい値 = '<値>'。	154
付録	155
Allen-Bradley ControlLogix Ethernet チャンネルのプロパティ	155
Allen-Bradley ControlLogix Ethernet デバイスのプロパティ	155
Allen-Bradley ControlLogix Ethernet タグのプロパティ	158
Logix デバイス ID	159
CompactLogix 5300 イーサネットのデバイス ID	159
1761-NET-ENI の設定	160
Data Highway Plus ゲートウェイの設定	161
シリアルゲートウェイの設定	162
MicroLogix 1100 の設定	163
通信のルーティング	163
接続パスの指定	164
ルーティングの例	165
プロトコルモードの選択	168
コントローラプロジェクト内の変更の検出	169
SoftLogix 5800 の接続の注意事項	171
索引	172

Allen-Bradley ControlLogix Ethernet ドライバー

ヘルプバージョン 1.188

目次

概要

Allen-Bradley ControlLogix Ethernet ドライバー とは

通信のルーティング

リモートプロセッサまたはインタフェースモジュールとの通信方法

設定

このドライバーを使用するためにチャンネルとデバイスを構成する方法

APIを使用した設定

Configuration API を使用してチャンネルとデバイスを設定する方法

自動タグデータベース生成

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

パフォーマンスの最適化

Allen-Bradley ControlLogix Ethernet ドライバー から最高のパフォーマンスを得る方法

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 Ethernet デバイスでタグのアドレスを指定する方法

エラーコード

Allen-Bradley ControlLogix Ethernet 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

付録

Allen-Bradley ControlLogix Ethernet ドライバー に関連する補足情報の場所

概要

Allen-Bradley ControlLogix Ethernet ドライバー は Allen-Bradley ControlLogix Ethernet コントローラが HMI、SCADA、Historian、MES、ERP や多数のカスタムアプリケーションを含む OPC クライアントアプリケーションに接続するための簡単かつ信頼性の高い手段を提供します。

サポートされる Allen-Bradley コントローラ

ControlLogix® 5500 シリーズ

ControlLogix との通信は、EtherNet/IP 通信モジュール (イーサネット通信の場合) または 1761-NET-ENI モジュール (コントローラのシリアルポートを使用したイーサネット/シリアル間通信の場合) を介して確立できます。

CompactLogix™ 5300 シリーズ

CompactLogix とのイーサネット通信には、1769-L35E などの内蔵 EtherNet/IP ポートがあるプロセッサが必要です。これがない場合、CompactLogix との通信には、コントローラのシリアルポートを使用したイーサネット/シリアル間通信用 1761-NET-ENI モジュールが必要です。

FlexLogix 5400 シリーズ

FlexLogix との通信は、1788-ENBT ドーターカード (イーサネット通信の場合) または 1761-NET-ENI モジュール (コントローラのシリアルポートを使用したイーサネット/シリアル間通信の場合) を介して確立できます。

SoftLogix 5800

このドライバーは Allen-Bradley SoftLogix 5800 シリーズのコントローラをサポートしており、SoftLogix PC でイーサネットカードを必要とします。

Data Highway Plus ゲートウェイ

このドライバーは Data Highway Plus インタフェースを備えた PLC-5 シリーズおよび SLC 500 シリーズをサポートしています。これは DH+ ゲートウェイを介して確立され、前述のいずれかの PLC、EtherNet/IP 通信モジュール、および 1756-DHRIO インタフェースモジュール (どちらも ControlLogix ラック内) を必要とします。

ControlNet ゲートウェイ

このドライバーは PLC-5C シリーズをサポートしています。これは ControlNet ゲートウェイを介して確立され、前述の PLC、EtherNet/IP 通信モジュール、および 1756-CNBR/CNBR インタフェースモジュール (どちらも ControlLogix ラック内) を必要とします。

1761-NET-ENI

このドライバーは 1761-NET-ENI デバイスとの通信をサポートしています。ENI デバイスによって全二重 DF1 コントローラと Logix コントローラの両方にイーサネット/シリアル間インタフェースが提供されることで、デバイスのネットワークと通信の柔軟性が向上します。ENI デバイスと併用した場合、このドライバーは以下をサポートします。

- ControlLogix 5500 シリーズ*
- CompactLogix 5300 シリーズ*
- FlexLogix 5400 シリーズ*
- MicroLogix シリーズ
- SLC 500 固定 I/O プロセッサ
- SLC 500 モジュラー I/O シリーズ
- PLC-5 シリーズ

*これらのモデルでは 1761-NET-ENI シリーズ B 以上が必要です。

MicroLogix 1100

このドライバーは EtherNet/IP を使用した MicroLogix 1100 (チャンネル 1 イーサネット) との通信をサポートしています。

ControlLogix は Allen-Bradley Company, LLC. の登録商標です。
CompactLogix は Rockwell Automation, Inc. の商標です。
すべての商標はそれぞれの所有者に帰属します。

設定

チャンネルとデバイスの制限値

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 1024 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 1024 です。

サポートされるデバイス

デバイスファミリー	通信
ControlLogix 5550 / 5553 / 5555 / 5561 / 5562 / 5563 / 5564 / 5565 / 5571 / 5572 / 5573 / 5574 / 5575 / 5580 プロセッサ (GuardLogix モデルを含む)	1756-ENBT / ENET / EN2F / EN2T / EN2TR / EN3TR / EWEB / EN2TXT イーサネットモジュール経由 シリアルゲートウェイ経由 チャンネル 0 を使用した 1761-NET-ENI シリーズ B 以上 経由 (シリアル)
CompactLogix 5320 / 5323 / 5330 / 5331 / 5332 / 5335 / 5343 / 5345 / 5370 / 5380 / 5480 (GuardLogix モデルを含む)	サフィックス E が付いたプロセッサ上の内蔵 Ethernet/IP ポート* シリアルゲートウェイ経由 チャンネル 0 を使用した 1761-NET-ENI シリーズ B 以上 経由 (シリアル)
FlexLogix 5433 / 5434 プロセッサ	1788-ENBT イーサネットドーターカード 経由 シリアルゲートウェイ経由 チャンネル 0 を使用した 1761-NET-ENI シリーズ B 以上 経由 (シリアル)
SoftLogix 5810 / 5830 / 5860 プロセッサ	SoftLogix Ethernet / IP メッセージングモジュール 経由 シリアルゲートウェイ 経由
MicroLogix 1000 / 1200 / 1500	1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由
MicroLogix 1100 / 1400	MicroLogix 1100 / 1400 チャンネル 1 経由 (イーサネット) 1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由
SLC 500 固定 I/O プロセッサ	1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由
SLC 500 モジュラー I/O プロセッサ (SLC 5/01、SLC 5/02、SLC 5/03、SLC 5/04、SLC 5/05)	DH+ ゲートウェイ 経由** 1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由
PLC-5 シリーズ (PLC5/250 シリーズを除く)	DH+ ゲートウェイ 経由 1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由
PLC-5/20C、PLC-5/40C、PLC-5/80C	ControlNet ゲートウェイ 経由 1761-NET-ENI 経由 EtherNet/IP ゲートウェイ 経由

*たとえば、1769-L35E。

**このドライバーは、DH+ をサポートするか DH+ ネットワーク (KF2 インタフェースモジュールなど) にインタフェース接続可能なすべての SLC 500 シリーズ PLC をサポートしています。

ファームウェアのバージョン

デバイスファミリー	バージョン
ControlLogix 5550 (1756-L1)	11.035 - 13.034
ControlLogix 5553 (1756-L53)	11.028
ControlLogix 5555 (1756-L55)	11.032 - 16.004
ControlLogix 5561 (1756-L61)	12.031 - 20.011
ControlLogix 5562 (1756-L62)	12.031 - 20.011
ControlLogix 5563 (1756-L63)	11.026 - 20.011
ControlLogix 5564 (1756-L64)	16.003 - 20.011
ControlLogix 5565 (1756-L65)	16.003 - 20.011
ControlLogix 5571 (1756-L71)	20.011 - 35.011
ControlLogix 5572 (1756-L72)	19.011 - 35.011
ControlLogix 5573 (1756-L73)	18.012 - 35.011
ControlLogix 5574 (1756-L74)	19.011 - 35.011
ControlLogix 5575 (1756-L75)	18.012 - 35.011
ControlLogix 5580 (1756-L8)	28.011 - 35.011
CompactLogix 5370 (1769-L1)	20.011 - 35.011
CompactLogix 5370 (1769-L2)	20.011 - 35.011
CompactLogix 5370 (1769-L3)	20.011 - 35.011
CompactLogix 5320 (1769-L20)	11.027 - 13.018
CompactLogix 5323 (1769-L23)	17.005 - 20.011
CompactLogix 5330 (1769-L30)	11.027 - 13.018
CompactLogix 5331 (1769-L31)	16.022 - 20.011
CompactLogix 5332 (1769-L32)	16.022 - 20.011
CompactLogix 5335 (1769-L35)	16.022 - 20.011
CompactLogix 5343 (1768-L43)	15.007 - 20.011
CompactLogix 5345 (1768-L45)	16.024 - 20.011
CompactLogix 5380 (5069-L3)	28.011 - 35.011
CompactLogix 5480 (5069-L4)	33.011 - 35.011
FlexLogix 5433 (1794-L33)	11.025 - 13.033
FlexLogix 5434 (1794-L34)	11.025 - 16.002
SoftLogix 5800 (1789-L60)	16.000 - 20.001
ControlLogix、CompactLogix、および FlexLogix シリアル通信	1761-NET-ENI シリーズ B 以上またはシリアルゲートウェイ
MicroLogix 1100 (1763-L16AWA/BWA/BBB)	1.1

通信プロトコル

通信プロトコルは、TCP/IP を使用した EtherNet/IP (イーサネットを介した CIP) です。

Logix モデルとゲートウェイモデル

Logix モデルとゲートウェイモデルは以下をサポートしています。

- 接続メッセージング
- シンボリック読み取り
- シンボリック書き込み
- シンボルインスタンス読み取り (V21 以上)

- 物理 (DMA) 読み取り (V20 以下)
- シンボルインスタンス書き込み

ENI モデル

ENI モデルは非接続メッセージングをサポートしています。

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。

プロパティグループ	<input type="checkbox"/> 識別 名前 説明 ドライバー	
一般		
イーサネット通信		
書き込み最適化		
詳細	<input type="checkbox"/> 診断 診断取り込み 無効化	
プロトコル設定	<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:** ドライバーで診断機能がサポートされていない場合、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」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。

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	<input type="checkbox"/> イーサネット設定 ネットワークアダプタ デフォルト	
一般		
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。空白のままにするか、「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。

チャネルのプロパティ - 書き込み最適化

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をデバイスに遅延なく届ける必要があります。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することができます。

プロパティグループ	<input checked="" type="checkbox"/> 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。モーメンタリブッシュボタンなどの Boolean 操作で問題が発生することなく、HMI データの操作を最適化できます。
- ・「**すべてのタグの最新の値のみを書き込み**」: このオプションを選択した場合、2 つ目の最適化モードの理論がすべてのタグに適用されます。これはアプリケーションが最新の値だけをデバイスに送信する必要がある場合に特に役立ちます。このモードでは、現在書き込みキューに入っているタグを送信する前に更新することによって、すべての書き込みが最適化されます。これがデフォルトのモードです。

「**デューティサイクル**」: 読み取り操作に対する書き込み操作の比率を制御するときに使用します。この比率は必ず、読み取り 1 回につき書き込みが 1 から 10 回の間であることが基になっています。デューティサイクルはデフォルトで 10 に設定されており、1 回の読み取り操作につき 10 回の書き込みが行われます。アプリケーションが多数の連続書き込みを行っている場合でも、読み取りデータを処理する時間が確実に残っている必要があります。これを設定すると、書き込み操作が 1 回行われるたびに読み取り操作が 1 回行われるようになります。実行する書き込み操作がない場合、読み取りが連続処理されます。これにより、連続書き込みを行うアプリケーションが最適化され、データの送受信フローがよりバランスのとれたものとなります。

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

チャネルのプロパティ - 詳細

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

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。

● **注記**: 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。

デバイスのプロパティ - 一般

プロパティグループ	<input type="checkbox"/> 識別	
一般	名前	ENI: SLC 500 Modular I/O
スキャンモード	説明	
タイミング	ドライバー	Allen-Bradley ControlLogix Ethernet
自動格下げ	モデル	ENI: SLC 500 Modular I/O
タグ生成	チャンネル割り当て	Allen-Bradley ControlLogix Ethernet
Logix 通信パラメータ	ID	<255.255.255.2>
Logix オプション	<input type="checkbox"/> 動作モード	
Logix データベース設定	データコレクション	有効化
ENI DF1/DH+/CN ゲートウェイ...	シミュレーション	いいえ
スロット構成		
冗長		

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。

● **ヒント**: GuardLogix には、ControlLogix または CompactLogix を使用します ([「サポートされるデバイス」](#)を参照)。

「ID」: デバイスの一意のネットワークアドレスを、通常は <IP またはホスト名>,1, <ルーティングパス>, <スロット> というフォーマットで入力します。

● アドレス指定の規則はモデルとルーティングによって異なります。詳細については、[参考資料](#)でモデル固有のアドレス指定のトピックを参照してください。

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. 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。

デバイスのプロパティ - スキャンモード

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

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、最大スキャン速度として設定されている値を指定します。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、要求ポーリングのみ」: 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、OPC 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「デバイス要求ポーリング」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期更新」: 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、初期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。

タグ数

プロパティグループ	<input checked="" type="checkbox"/> 識別
一般	<input checked="" type="checkbox"/> 動作モード
スキャンモード	<input type="checkbox"/> タグ数
	静的タグ
	0

「静的タグ」: デバイスレベルまたはチャ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。

デバイスのプロパティ - タイミング

デバイスのタイミングのプロパティでは、エラー状態に対するデバイスの応答をアプリケーションのニーズに合わせて調整できます。多くの場合、最適なパフォーマンスを得るためにはこれらのプロパティを変更する必要があります。電氣的に発生するノイズ、モデムの遅延、物理的な接続不良などの要因が、通信ドライバーで発生するエラーやタイムアウトの数に影響します。タイミングのプロパティは、設定されているデバイスごとに異なります。

プロパティグループ	<input type="checkbox"/> 通信タイムアウト
一般	接続タイムアウト (秒)
スキャンモード	要求のタイムアウト (ミリ秒)
タイミング	タイムアウト前の試行回数
	3
	1000
	3

通信タイムアウト

「接続タイムアウト」: このプロパティ (イーサネットベースのドライバーで主に使用) は、リモートデバイスとのソケット接続を確立するために必要な時間を制御します。デバイスの接続時間は、同じデバイスへの通常の通信要求よりも長くかかることがよくあります。有効な範囲は 1 から 30 秒です。デフォルトは通常は 3 秒ですが、各ドライバーの特性によって異なる場合があります。この設定がドライバーでサポートされていない場合、無効になります。

● 注記: UDP 接続の特性により、UDP を介して通信する場合には接続タイムアウトの設定は適用されません。

「要求のタイムアウト」: すべてのドライバーがターゲットデバイスからの応答の完了を待機する時間を決定するために使用する間隔を指定します。有効な範囲は 50 から 9,999,999 ミリ秒 (167 分) です。デフォルトは通常は 1000 ミリ秒ですが、ドライバーによって異なる場合があります。ほとんどのシリアルドライバーのデフォルトのタイムアウトは 9600 ボー以上のもので、低いボーレートに基づきます。低いボーレートでドライバーを使用している場合、データの取得に必要な時間が増えることを補うため、タイムアウト時間を増やします。

「タイムアウト前の試行回数」: ドライバーが通信要求を発行する回数を指定します。この回数を超えると、要求が失敗してデバイスがエラー状態にあると見なされます。有効な範囲は 1 から 10 です。デフォルトは通常は 3 ですが、各ドライ

パーの特性によって異なる場合があります。アプリケーションに設定される試行回数は、通信環境に大きく依存します。このプロパティは、接続の試行と要求の試行の両方に適用されます。

タイミング

「**要求間遅延**」:ドライバーがターゲットデバイスに次の要求を送信するまでの待ち時間を指定します。デバイスに関連付けられているタグおよび1回の読み取りと書き込みの標準のポーリング間隔がこれによってオーバーライドされます。この遅延は、応答時間が長いデバイスを扱う際や、ネットワークの負荷が問題である場合に役立ちます。デバイスの遅延を設定すると、そのチャンネル上のその他すべてのデバイスとの通信に影響が生じます。可能な場合、要求間遅延を必要とするデバイスは別々のチャンネルに分けて配置することをお勧めします。その他の通信プロパティ(通信シリアル化など)によってこの遅延が延長されることがあります。有効な範囲は0から300,000ミリ秒ですが、一部のドライバーでは独自の設計の目的を果たすために最大値が制限されている場合があります。デフォルトは0であり、ターゲットデバイスへの要求間に遅延はありません。

● **注記**: すべてのドライバーで「要求間遅延」がサポートされているわけではありません。使用できない場合にはこの設定は表示されません。

タイミング	<input type="checkbox"/> タイミング	
自動格下げ	要求間遅延 (ミリ秒)	0

デバイスのプロパティ - 自動格下げ

自動格下げのプロパティを使用することで、デバイスが応答していない場合にそのデバイスを一時的にスキャン停止にできます。応答していないデバイスを一定期間オフラインにすることで、ドライバーは同じチャンネル上のほかのデバイスとの通信を引き続き最適化できます。停止期間が経過すると、ドライバーは応答していないデバイスとの通信を再試行します。デバイスが応答した場合はスキャンが開始され、応答しない場合はスキャン停止期間が再開します。

プロパティグループ	<input checked="" type="checkbox"/> 自動格下げ	
一般	エラー時に格下げ	有効化
スキャンモード	格下げまでのタイムアウト回数	3
タイミング	格下げ期間 (ミリ秒)	10000
自動格下げ	格下げ時に要求を破棄	無効化

「**エラー時に格下げ**」: 有効にした場合、デバイスは再び応答するまで自動的にスキャン停止になります。

● **ヒント**: システムタグ `_AutoDemoted` を使用して格下げ状態をモニターすることで、デバイスがいつスキャン停止になったかを把握できます。

「**格下げまでのタイムアウト回数**」: デバイスをスキャン停止にするまでに要求のタイムアウトと再試行のサイクルを何回繰り返すかを指定します。有効な範囲は1から30回の連続エラーです。デフォルトは3です。

「**格下げ期間**」: タイムアウト値に達したときにデバイスをスキャン停止にする期間を指定します。この期間中、そのデバイスには読み取り要求が送信されず、その読み取り要求に関連するすべてのデータの品質は不良に設定されます。この期間が経過すると、ドライバーはそのデバイスのスキャンを開始し、通信での再試行が可能になります。有効な範囲は100から3600000ミリ秒です。デフォルトは10000ミリ秒です。

「**格下げ時に要求を破棄**」: スキャン停止期間中に書き込み要求を試行するかどうかを選択します。格下げ期間中も書き込み要求を必ず送信するには、無効にします。書き込みを破棄するには有効にします。サーバーはクライアントから受信した書き込み要求をすべて自動的に破棄し、イベントログにメッセージを書き込みません。

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、サーバーのタグスペースに追加されたタグをプロジェクトとともに保存する必要があります。ユーザーは「**ツール**」|「**オプション**」メニューから、自動保存するようプロジェクトを設定できます。

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

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

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

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

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

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

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

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

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

デバイスのプロパティ - Logix 通信パラメータ

プロパティグループ	EtherNet/IP	
一般	TCP/IP ポート	44818
スキャンモード	CIP	
タイミグ	接続サイズ (バイト)	500
自動格下げ	不活動ウォッチドッグ (秒)	32
タグ生成	Logix	
Logix 通信パラメータ	配列ブロックサイズ	120
Logix オプション		
Logix データベース設定		
ENI DF1/DH+/CN ゲートウェイ...		
スロット構成		
冗長		

EtherNet/IP

「**TCP/IP ポート**」: このデバイスで使用するよう設定されている TCP/IP ポート番号を指定します。デフォルトは 44818 です。

CIP

「**接続サイズ**」: CIP 接続でデータの要求と応答に使用可能なバイト数を指定します。有効な範囲は 500 から 4000 バイトです。デフォルトは 500 バイトです。

● **注記**: ControlLogix 5500 および CompactLogix 5300 デバイスモデルだけがこの機能をサポートしています。500 バイトを超える接続サイズをサポートするには、デバイスはファームウェアバージョン 20 以降のコントローラおよびイーサネットブリッジ EN3x、EN2x、または EN5.x をサポートする必要があります。ENBT や ENET などの古いイーサネットモジュールはこの機能をサポートしていません。必要な要件を満たさないデバイスはデフォルト設定の 500 バイトに自動的にフォールバックします。ただし、通信失敗後の再試行では要求されたサイズが使用されます。

● 「**接続サイズ**」の値はシステムタグ "_CIPConnectionSizeRequested" を介して要求することもできます。詳細については、[内部タグ](#)を参照してください。

「**不活動ウォッチドッグ**」: 接続が(読み取り/書き込み可能なトランザクションがない)アイドル状態となりうる時間(秒数)を指定します。この時間を経過するとコントローラによって接続はクローズします。この値が大きいほど、接続のリソースがコントローラによって解放されるまでの時間が長くなり、値が小さいほど時間が短くなります。デフォルトは 32 秒です。

● **注記**: プロジェクトのアップロード中に CIP 接続のタイムアウトに関するエラーが頻繁に発生する場合、「不活動ウォッチドッグ」の値を増やしてください。それ以外の場合は、デフォルト値が推奨されます。

Logix

「**配列ブロックサイズ**」: このプロパティでは 1 回のトランザクションで読み取られる配列要素の最大数を指定します。この値は 30 から 3840 要素の範囲で調整できます。デフォルトは 120 要素です。

■ **ヒント:** Boolean 配列では、1 要素は 32 要素のビット配列と見なされます。ブロックサイズを 30 要素に設定すると 960 ビット要素として解釈され、3840 要素は 122880 ビット要素として解釈されます。

デバイスのプロパティ - Logix オプション

プロパティグループ	<input type="checkbox"/> プロトコルオプション	
一般	プロトコルモード	論理非ブロック
スキャンモード	オンライン編集後に同期化	はい
タイミング	オフライン編集後に同期化	はい
自動格下げ	LEN で文字列データを終了	有効化
タグ生成	<input type="checkbox"/> プロジェクトオプション	
Logix 通信パラメータ	デフォルトデータ型	デフォルト
Logix オプション	パフォーマンス統計	無効化
Logix データベース設定		
ENI DF1/DH+/CN ゲートウェイ...		
スロット構成		
冗長		

プロトコルオプション

「**プロトコルモード**」: コントローラから Logix タグデータを読み取る方法を「論理非ブロック」、「論理ブロック」、「シンボリック」の中から選択します。デフォルトは「論理非ブロック」です。このオプションは、クライアント/サーバータグ更新のパフォーマンス向上を図る上級ユーザーのみが変更するようにしてください。

● 詳細については、[プロトコルモードの選択](#)を参照してください。

● 注記: 「論理非ブロック」と「論理ブロック」はシリアルゲートウェイモデルでは使用できません。

「**オンライン編集後に同期化**」: 有効な場合、オンラインによるプロジェクト編集 (または RSLogix/Studio5000 からのプロジェクトのダウンロード) が検出されると、ドライバーはそのプロジェクトのイメージをコントローラプロジェクトのイメージと同期化します。このオプションによって、プロジェクト変更の際に不必要なエラーが発生しなくなります。これは選択されているプロトコルモードが「論理非ブロック」または「論理ブロック」の場合にのみ使用できます。デフォルトは「はい」です。

「**オフライン編集後に同期化**」: 有効な場合、オフラインによるプロジェクト編集 (または RSLogix/Studio5000 からのプロジェクトのダウンロード) が検出されると、ドライバーはその独自のプロジェクトのイメージをコントローラプロジェクトのイメージと同期化します。このオプションによって、プロジェクト変更の際に不必要なエラーが発生しなくなります。これは選択されているプロトコルが「論理非ブロック」または「論理ブロック」の場合にのみ使用できます。デフォルトは「はい」です。

● プロジェクトの変更との同期化に失敗した場合、間違ったネーティブタグアドレスとの間で読み書きが行われる可能性があります。

「**LEN で文字列データを終了**」: 有効な場合、ドライバーは STRING 構造体の DATA メンバーが読み取られると自動的に LEN メンバーを読み取ります。DATA 文字列は、最初に出現した Null 文字、LEN の値に相当する位置にある文字、または DATA の最大文字列長 (のいずれか最初に出現した位置) で終端されます。無効な場合、ドライバーは LEN メンバーの読み取りをバイパスし、最初に出現した Null 文字または DATA の最大文字列長 (のいずれか最初に出現した位置) で DATA 文字列を終端します。したがって、DATA が修正されることなく LEN が外部ソースによって短くなった場合でも、ドライバーはこの短い長さに従って DATA を終端しません。デフォルトは「Enable」です。

「プロジェクトオプション」

「**デフォルトデータ型**」: タグの追加、修正、インポート時にデフォルトのデータ型を選択した場合にクライアント/サーバータグに割り当てられるデータ型を選択します。デフォルトは「デフォルト」です。

● 詳細については、[デフォルトデータ型の条件](#)を参照してください。

「**パフォーマンス統計**」: Allen-Bradley ControlLogix Ethernet ドライバーには、ドライバーのパフォーマンスを測定するために通信に関する統計を収集する機能が備わっています。有効にした場合、このオプションが有効になります。ドライバーはクライアント/サーバータグの更新回数とタイプを追跡します。サーバーアプリケーションの再起動時に、サーバーのイベントログに結果が表示されます。デフォルトは「いいえ」です。

● 注記: 最適なパフォーマンスを得るようにプロジェクト構成を設計した後は、「パフォーマンス統計」を無効にすることをお勧めします。統計はシャットダウン時にイベントログに書き込まれるので、結果を表示するためにはサーバーを再起動する必要があります。

● 関連項目: [コントローラプロジェクト内の変更の検出](#)

デバイスのプロパティ - Logix データベース設定

プロパティグループ	データベースのインポート方法	
一般	データベースのインポート方法	デバイスから作成
スキャンモード	タグインポートファイル	*.15k
タイミング	タグの説明	有効化
自動格下げ	Logix データベースオプション	
タグ生成	名前の長さを制限	無効化
Logix 通信パラメータ	タグ階層	展開
Logix オプション	Logix データベースフィルタ	
Logix データベース設定	配列の制限を適用	無効化
ENI DF1/DH+/CN ゲートウェイ...	配列要素の上限	2000
スロット構成		
冗長		

「データベースのインポート方法」

「データベースのインポート方法」: タグデータベースに値を取り込む方法を選択します。

- 「デバイスから作成」: データアクセスに使用されているものと同じイーサネット接続を介してコントローラから直接タグを読み込みます。これは高速でほとんどのタグがインポートされますが、コントローラへのアクセスが必要であり、説明はインポートされません。インポートされないタグには Add-On Instruction (AOI) InOut プロパティなどがあります。
 - 注記: この機能はシリアルゲートウェイモデルでは使用できません。
- 「インポートファイルから作成」: 選択した RSLogix L5K/L5X ファイルからタグを読み込みます。コントローラへのアクセスは必要なく、説明がインポートされ、ユーザーはオフラインで作業できますが、このオプションは比較的速度が遅く、コントローラ内のすべてのタグはインポートされません。インポートされないタグには次のようなものがあります。
 - I/O タグ
 - Add-On Instruction (AOI) InOut プロパティ
 - ほかのプロパティのエイリアスである AOI プロパティ
 - 別の Equipment Phase または Program プロパティのエイリアスである Equipment Phase プロパティ
 - 別の Program または Equipment Phase プロパティのエイリアスである Program プロパティ
 - タイマー/カウンタ制御ビット

「タグインポートファイル」: ブラウズ (...) ボタンをクリックし、タグのインポート元である L5K/L5X ファイルを見つけて選択します。このファイルは自動タグデータベース生成でタグデータベースの作成が指定されている場合に使用されます。Global と Program を含むすべてのタグがインポートされ、それぞれのデータ型に従って展開されます。

「タグの説明」: 非構造体、非配列タグにタグの説明をインポートするには、「有効化」を選択します。必要な場合、タグには説明とともに、元のタグ名を示す長い名前が割り当てられます。

API を使用したタグ生成

CLX オフライン ATG の Config API プロパティは次のとおりです。

```
"controllogix_ethernet.DEVICE_DATABASE_IMPORT_METHOD": 1, "controllogix_ethernet.DEVICE_TAG_IMPORT_FILE": "myFile.15x", "controllogix_ethernet.DEVICE_DISPLAY_DESCRIPTIONS": true,
```

ここで、インポート方法の列挙は次のとおりです。
 デバイスから作成する場合は 0。
 または
 ファイルから作成する場合は 1。

Logix データベースオプション

「**名前の長さを制限**」: タグ名とグループ名を 31 文字に制限するには「有効化」に設定します。デフォルトは「無効化」です。

1. OPC サーババージョン 4.70 より前では、タグ名とグループ名の長さは 31 文字に制限されていました。現在の長さ制限である 256 文字では、Logix 40 文字制限で作成されている Logix タグ名をそのまま使用できます。
2. 以前のバージョンのサーバを使用して L5K/L5X インポートを介してタグがインポートされた場合、イベントログを調べるかサーバプロジェクトをスキャンして、文字制限によって切り捨てられたタグがないか確認してください。そのようなタグがある場合、サーバタグ名を維持するためにこのプロパティを有効にしてください。OPC クライアントタグ参照は影響を受けません。無効にした場合、それより長いタグ名が作成され、クリップされたタグを参照するクライアントがこの新しいタグ名を参照するためには変更が必要です。
3. 以前のバージョンの OPC サーバを使用して L5K/L5X インポートを介してタグがインポートされ、31 文字の制限によって切り捨てられたタグがない場合、このオプションを無効のままにしてください。
4. サーババージョン 4.70 以上を使用して L5K/L5X を介してタグがインポートされた場合も、このオプションを無効のままにしてください。

● **関連項目**: [コントローラからサーバへの名前の変換](#)

「**タグ階層**」: このプロパティではタグ階層のツリー構成を指定します。圧縮モードでは、自動タグ生成によって作成されたサーバタグは、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。展開モードでは、自動タグ生成によって作成されたサーバタグは、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論理グループを表すグループが作成されます。この機能を使用するには、「**タグ生成**」プロパティの「**サブグループを許可**」を有効にします。

● 作成されたグループの詳細については、[タグ階層](#) および [コントローラからサーバへの名前の変換](#) を参照してください。

「Logix データベースフィルタ」

「**配列の制限を適用**」: 配列要素の数を制限するには「有効化」を選択します。コントローラ内のタグは非常に大きな配列次元を使用して宣言されていることがあります。デフォルトでは、配列はタグ生成時に完全に展開されるので、大きな配列では時間がかかります。上限が課されている場合、各次元から指定された数の要素のみが生成されます。上限は配列次元のサイズが上限を超えた場合にのみ有効になります。デフォルトは「無効化」です。

「**配列要素の上限**」: 配列数の上限を指定します。デフォルトは 2000 です。

デバイスのプロパティ - ENI DF1/DH+/CN ゲートウェイ通信パラメータ

プロパティグループ	ENI DF1/DH+/CN ゲートウェイ通信パラメータ	
一般	TCP/IP ポート	44818
スキャンモード	要求サイズ (バイト)	232
タイミグ	ファンクションファイルのブロック書き込み...	無効化
自動格下げ		
タグ生成		
Logix 通信パラメータ		
Logix オプション		
Logix データベース設定		
ENI DF1/DH+/CN ゲートウ...		
スロット構成		
冗長		

「**TCP/IP ポート**」: リモートデバイスで使用するよう設定されているポート番号を指定します (1756-ENBT など)。デフォルトは 44818 です。

「**要求サイズ**」: パフォーマンスを微調整するため、デバイスから一度に要求可能なバイト数を選択します。オプションは 32、64、128、232 です。デフォルトは 232 バイトです。

「**Allow Function File Block Writes**」: ファンクションファイルは (PD および MG データファイルのように) 構造に基づくファイルであり、MicroLogix 1100、1200、および 1500 に固有のファイルです。該当するファンクションファイルでは、1 回の操作でデバイスにデータを書き込むことができます。デフォルトでは、ファンクションファイルのサブ要素 (ファンクションファイル構造内のフィールド) にデータが書き込まれると、そのタグでただちに書き込み操作が行われます。時 (HR)、分 (MIN)、

秒 (SEC) などのサブ要素が含まれている RTC ファイルなどのファイルでは、個別の書き込みは必ずしも許可されません。時刻のみに依存するこれらのサブ要素では、それぞれのサブ要素への書き込みの間に時間が経過するのを回避するため、1 回の操作ですべての値を書き込む必要があります。このため、これらのサブ要素をブロック書き込みするオプションがあります。デフォルトでは無効になっています。

● 詳細については、[ブロック書き込み](#)および[ファンクションファイル](#)を参照してください。

ブロック書き込み

ブロック書き込みでは、1 回の書き込み操作でファンクションファイル内の各読み取り/書き込みサブ要素の値がデバイスに書き込まれます。ブロック書き込みを実行する前に各サブ要素に書き込む必要はありません。影響を受けない (書き込まれない) サブ要素にはその現在の値が書き戻されます。たとえば、現在の (最後に読み取られた) 日時が 1/1/2001, 12:00.00, DOW = 3 であり、時刻を 1 時に変更した場合、デバイスには値 1/1/2001, 1:00.00, DOW = 3 が書き込まれます。詳細については、次の手順を参照してください。

1. 最初に、「デバイスのプロパティ」で「ENI DF1/DH+/CN ゲートウェイ通信パラメータ」を見つけます。
2. 「ファンクションファイルのブロック書き込みを許可」を有効にすることで、ブロック書き込みをサポートするファンクションファイルに対してブロック書き込みを利用するようドライバーに通知します。
3. 「OK」または「適用」をクリックします。
4. 対象のサブ要素タグに必要な値を書き込みます。サブ要素タグは書き込まれた値をただちにとります。
 - **注記:** ブロック書き込みモードでサブ要素に少なくとも 1 回書き込んだ後は、そのタグの値はコントローラからではなくドライバーの書き込みキャッシュから取得されます。ブロック書き込みが完了した後は、すべてのサブ要素タグの値がコントローラから取得されます。
5. 必要なすべてのサブ要素を書き込んだ後、これらの値をコントローラに送信するブロック書き込みを実行します。ブロック書き込みをインスタンス化するため、タグアドレス `RTC:<要素>_SET` を参照します。このタグの値を 'true' に設定した場合、現在の (最後に読み取られた) サブ要素と影響を受ける (書き込み先の) サブ要素に基づいてブロック書き込みが行われます。このタグを 'true' に設定するとただちに "false" に自動的にリセットされます。これがデフォルトの状態であり、操作は何も実行されません。

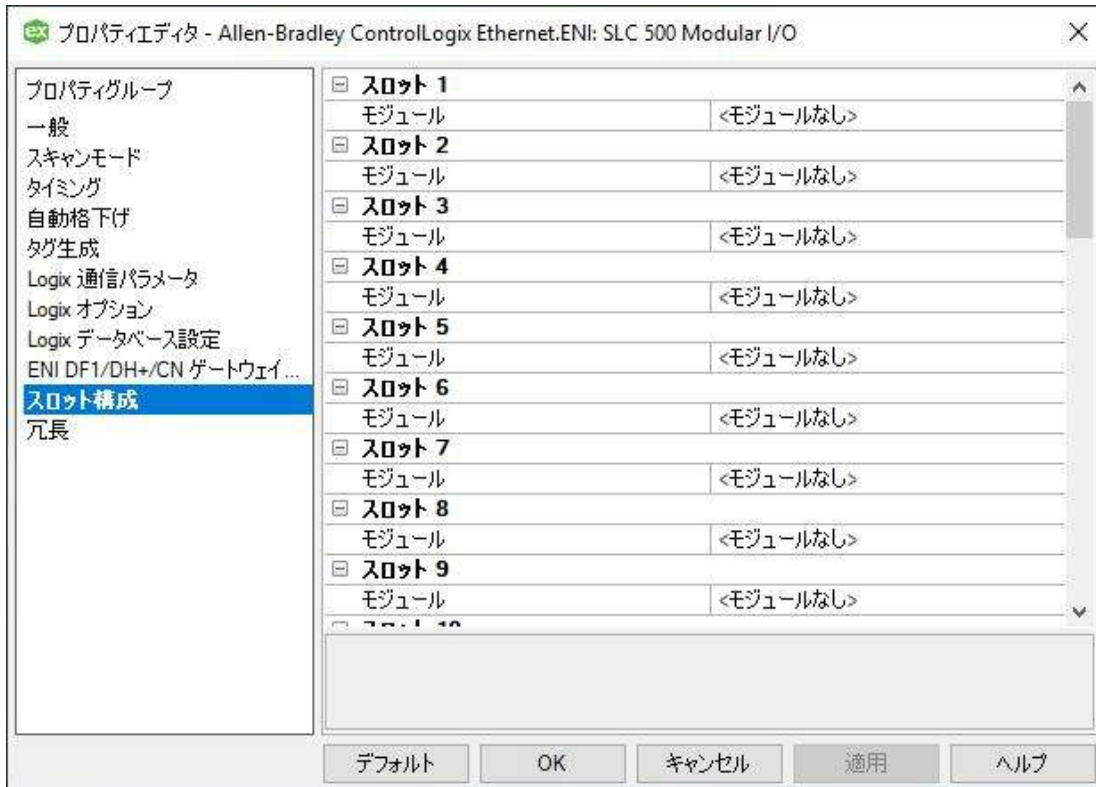
適用可能なファンクションファイル/サブ要素

RTC	
年	YR
月	MON
日	DAY
曜日	DOW
時	HR
分	MIN
秒	SEC

● **関連項目:** [ファンクションファイル一覧](#)

デバイスのプロパティ - SLC 500 スロット構成

I/O にアクセスするには、Allen-Bradley ControlLogix Ethernet ドライバー で使用するように SLC5/01/02/03/04/05 モデル (モジュラー I/O ラック) が構成されている必要があります。デバイスにつき最大 30 個のスロットを構成できます。



「**スロット n**」: 構成する物理スロット。プロパティを展開するにはプラスアイコンを使用します。

「**モジュール**」: スロットに格納するモジュールのタイプをドロップダウンリストから設定します。

● *Configuration API Service* を使用してスロットを構成するには、[Configuration API Service の例](#)を参照してください。

「**入力 Word**」: 選択したモジュールが必要な場合、このモジュールの「入力 Word」の最大数を入力します。

「**出力 Word**」: 選択したモジュールが必要な場合、このモジュールの「出力 Word」の最大数を入力します。

スロット構成を使用するには、以下の手順に従います。

1. モジュールのリストボックスで行をクリックすることで、構成するスロットを選択します。
2. モジュールを選択するには、使用可能なモジュールのドロップダウンリストでクリックします。
3. 必要に応じて、「入力 Word」と「出力 Word」を設定します。
4. スロット/モジュールを除去するには、使用可能なモジュールのドロップダウンリストから「モジュールなし」を選択します。
5. 完了後、「OK」をクリックします。

● **ヒント:**

- 使用可能なモジュールのリストに含まれていない I/O を構成するには、0000 ジェネリックモジュールを使用します。
- モジュールの選択肢は Allen Bradley APS ソフトウェアにおける場合と同じです。

● **注記:** 物理モジュールが格納されていない空きスロットがラックにあることが一般的です。モジュールが格納されていない各種スロットのデータに正しくアクセスするには、それより前のモジュールに正しい数の Word がマッピングされている必要があります。たとえば、スロット 3 の I/O のみが対象であるがスロット 1 と 2 に I/O モジュールが格納されている場合、このスロット構成グループからスロット 1、2、3 に正しいモジュールを選択する必要があります。

0000 ジェネリックモジュール

使用可能なモジュールのリストに示されていないモジュールの入力 Word 数と出力 Word 数をマッピングするにはジェネリックモジュールを使用します。ジェネリックモジュールを正しく使用するには、各モジュールに必要な入力 Word と出力 Word の数を把握しておく必要があります。

- Allen-Bradley I/O ユーザーマニュアルドキュメントで入力と出力の要件を確認し、クラス 1 とクラス 3 の動作ではそれらの要件が異なる場合があることを理解しておいてください。
- 各 I/O モジュールで使用可能な入力 Word と出力 Word の数については、[モジュラー I/O 選択ガイド](#)を参照してください。

デバイスのプロパティ - 冗長

プロパティグループ	冗長	
一般	セカンダリパス	
スキャンモード	動作モード	障害時に切り替え
タイミング	モニターアイテム	
冗長	モニター間隔 (秒)	300
	できるだけ速やかにプライマリに...	はい

冗長設定はメディアレベルの冗長プラグインで使用できます。

- 詳細については、Web サイトまたは[ユーザーマニュアル](#)を参照するか、営業担当者までお問い合わせください。

SLC 500 モジュール I/O 選択ガイド

以下の表には、スロット構成リスト内の各 I/O モジュールで使用可能な入力 Word 数と出力 Word 数が一覧されています。

モジュールID	モジュールタイプ	入力 Word 数	出力 Word 数
0	1746-I*8 すべての 8 ポイント 離散入力モジュール	1	0
1	1746-I*16 すべての 16 ポイント 離散入力モジュール	1	0
2	1746-I*32 すべての 32 ポイント 離散入力モジュール	2	0
3	1746-O*8 すべての 8 ポイント 離散出力モジュール	0	1
4	1746-O*16 すべての 16 ポイント 離散出力モジュール	0	1
5	1746-O*32 すべての 32 ポイント 離散出力モジュール	0	2
6	1746-IA4 4 入力 100 / 120 VAC	1	0
7	1746-IA8 8 入力 100/120 VAC	1	0
8	1746-IA16 16 入力 100/120 VAC	1	0
9	1746-IB8 8 入力 (シンク) 24 VDC	1	0
10	1746-IB16 16 入力 (シンク) 24 VDC	1	0
11	1746-IB32 32 入力 (シンク) 24 VDC	2	0
12	1746-IG16 16 入力 [TTL] (ソース) 5 VDC	1	0
13	1746-IM4 4 入力 200/240 VAC	1	0
14	1746-IM8 8 入力 200/240 VAC	1	0
15	1746-IM16 16 入力 200/240 VAC	1	0
16	1746-IN16 16 入力 24 VAC / VDC	1	0
17	1746-ITB16 16 入力 [高速] (シンク) 24 VDC	1	0
18	1746-ITV16 16 入力 [高速] (ソース) 24 VDC	1	0
19	1746-IV8 8 入力 (ソース) 24 VDC	1	0
20	1746-IV16 16 入力 (ソース) 24 VDC	1	0
21	1746-IV32 32 入力 (ソース) 24 VDC	2	0
22	1746-OA8 8 出力 (トライアック) 100/240 VAC	0	1
23	1746-OA16 16 出力 (トライアック) 100/240 VAC	0	1
24	1746-OB8 8 出力 [トランス] (ソース) 10/50 VDC	0	1
25	1746-OB16 16 出力 [トランス] (ソース) 10/50 VDC	0	1
26	1746-OB32 32 出力 [トランス] (ソース) 10/50 VDC	0	2
27	1746-OBP16 16 出力 [トランス 1 アンペア] (ソース) 24 VDC	0	1
28	1746-OV8 8 出力 [トランス] (シンク) 10/50 VDC	0	1
29	1746-OV16 16 出力 [トランス] (シンク) 10/50 VDC	0	1
30	1746-OV32 32 出力 [トランス] (シンク) 10/50 VDC	0	2
31	1746-OW4 4 出力 [リレー] VAC/VDC	0	1
32	1746-OW8 8 出力 [リレー] VAC/VDC	0	1
33	1746-OW16 16 出力 [リレー] VAC/VDC	0	1
34	1746-OX8 8 出力 [絶縁リレー] VAC/VDC	0	1
35	1746-OVP16 16 出力 [トランス 1 アンペア] (シンク) 24 VDC3	0	1
36	1746-IO4 2 入力 100/120 VAC 2 出力 [リレー] VAC/VDC3	1	1
37	1746-IO8 4 入力 100/120 VAC 4 出力 [リレー] VAC/VDC4	1	1
38	1746-IO12 6 入力 100/120 VAC 6 出力 [リレー] VAC/VDC	1	1
39	1746-NI4 4 チャンネルアナログ入力	4	0
40	1746-NIO4I アナログ組み合わせ 2 入力 2 電流出力	2	2

モジュールID	モジュールタイプ	入力 Word 数	出力 Word 数
41	1746-NIO4V アナログ組み合わせ 2 入力 2 電圧出力	2	2
42	1746-NO4I 4 チャンネルアナログ電流出力	0	4
43	1746-NO4V 4 チャンネルアナログ電圧出力	0	4
44	1746-NT4 4 チャンネル熱電対入力モジュール	8	8
45	1746-NR4 4 チャンネル測温抵抗体/抵抗入力モジュール	8	8
46	1746-HSCE 高速カウンタ/エンコーダ	8	1
47	1746-HS 1 軸モーションコントローラ	4	4
48	1746-OG16 16 出力 [TLL] (シンク) 5 VDC	0	1
49	1746-BAS 基本モジュール 500 5/01 構成	8	8
50	1746-BAS 基本モジュール 5/02 構成	8	8
51	1747-DCM 直接通信モジュール (1/4 ラック)	2	2
52	1747-DCM 直接通信モジュール (1/2 ラック)	4	4
53	1747-DCM 直接通信モジュール (3/4 ラック)	6	6
54	1747-DCM 直接通信モジュール (フルラック)	8	8
55	1747-SN リモート I/O スキャナー	32	32
56	1747-DSN 分散 I/O スキャナー 7 ブロック	8	8
57	1747-DSN 分散 I/O スキャナー 30 ブロック	32	32
58	1747-KE インタフェースモジュール、シリーズ A	1	0
59	1747-KE インタフェースモジュール、シリーズ B	8	8
60	1746-NI8 8 チャンネルアナログ入力、クラス 1	8	8
61	1746-NI8 8 チャンネルアナログ入力、クラス 3	16	12
62	0000 ジェネリックモジュール	-	-
63	1746-IC16 16 入力 (シンク) 48 VDC	1	0
64	1746-IH16 16 入力 [トランス] (シンク) 125 VDC	1	0
65	1746-OAP12 12 出力 [トライアック] 120/240 VDC	0	1
66	1746-OB6E1 6 出力 [トランス] (ソース) 24 VDC	0	1
67	1746-OB16E 16 出力 [トランス] (ソース) 保護あり	0	1
68	1746-OB32E 32 出力 [トランス] (ソース) 10/50 VDC	0	2
69	1746-OBP8 8 出力 [トランス 2 アンペア] (ソース) 24 VDC	0	1
70	1746-IO12DC 6 入力 12 VDC、6 出力 [リレー]	1	1
71	1746-INI4I アナログ 4 チャンネル絶縁電流入力	8	8
72	1746-INI4VI アナログ 4 チャンネル絶縁電圧/電流入力	8	8
73	1746-INO4I アナログ 4 チャンネル絶縁電流出力	8	8
74	1746-INO4VI アナログ 4 チャンネル絶縁電圧/電流出力	8	8
75	1746-INT4 4 チャンネル絶縁熱電対入力	8	8
76	1746-NT8 アナログ 8 チャンネル熱電対入力	8	8
77	1746-HSRV モーションコントロールモジュール	12	8
78	1746-HSTP1 ステップモーターモジュール	8	8
79	1747-MNET MNET ネットワーク通信モジュール	0	0
80	1746-QS 同期軸モジュール	32	32
81	1747-QV オープンループ速度制御	8	8
82	1747-RCIF ロボット制御インタフェースモジュール	32	32
83	1747-SCNR ControlNet SLC スキャナー	32	32
84	1747-SDN DeviceNet スキャナーモジュール	32	32
85	1394-SJT GMC ターボシステム	32	32

モジュールID	モジュールタイプ	入力 Word 数	出力 Word 数
86	1203-SM1 SCANport 通信モジュール - 基本	8	8
87	1203 SM1 SCANport 通信モジュール - 拡張	32	32
88	AMCI-1561 AMCI シリーズ 1561 リゾルバーモジュール	8	8

Configuration API - Allen-Bradley ControlLogix Ethernet の例

チャンネル定義、デバイス定義、列挙のリストについては、REST クライアントを使用して次のエンドポイントにアクセスするか、付録を参照してください。

チャンネル定義

エンドポイント (GET):

```
https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/channels
```

デバイス定義

エンドポイント (GET):

```
https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/devices
```

Allen-Bradley ControlLogix Ethernet チャンネルの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート>/config/v1/project/channels
```

ボディ:

```
{
  "common.ALLTYPES_NAME": "MyChannel",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet"
}
```

● **関連項目:** チャンネルプロパティのリストについては、[付録](#)を参照してください。

Allen-Bradley ControlLogix Ethernet デバイスの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート>/config/v1/project/channels/MyChannel/devices
```

ボディ:

```
{
  "common.ALLTYPES_NAME": "MyDevice",
  "servermain.DEVICE_ID_STRING": "<IP アドレス>,0,1",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
  "servermain.DEVICE_MODEL": <モデル列挙>
}
```

<IP アドレス> は、デバイスの IP アドレスです。

● **注記:** servermain.DEVICE_ID_STRING の値のフォーマットは、servermain.DEVICE_MODEL に指定されているモデル列挙に応じて異なる場合があります。上記のデバイス ID の文字列のフォーマットは ControlLogix 5500 モデル向けのものです。

● **関連項目:** [デバイスモデル列挙](#)と[デバイスのプロパティ](#)を参照してください。

Allen-Bradley ControlLogix Ethernet タグの作成

エンドポイント (POST):

```
https://<ホスト名または IP>:<ポート>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

ボディ:

```
[
  {
    "common.ALLTYPES_NAME": "MyTag1",
```

```

    "servermain.TAG_ADDRESS": "My_Tag_Address"
  }
  {
    "common.ALLTYPES_NAME": "MyTag2",
    "servermain.TAG_ADDRESS": "My_Tag_Address"
  }
]

```

● **関連項目:** タグプロパティのリストについては、[付録](#)を参照してください。

● **Configuration API** を使用したタグとタググループの設定の詳細については、サーバーのヘルプおよびドライバー固有のヘルプを参照してください。

列挙

デバイスモデルなどの一部のプロパティには、列挙にマッピングされる値が含まれています。列挙とその値の有効なリストを確認するには、'content=property_definitions' を使用してデバイスのエンドポイントをクエリーするか、ドキュメント定義のエンドポイントをクエリーします。

たとえば、"MyChannel" というチャンネルの下にある "MyDevice" というデバイスのプロパティ定義を表示するには、GET リクエストを以下の URL に送信します。

```

https://<ホスト名または IP>:<ポート>/config/v1/project/channels/MyChannel/devices/MyDevice/?content=property_definitions

```

プロパティ定義は、チャンネル、タグなど、その他のオブジェクトでも使用することができます。

または、ドライバーのチャンネルとデバイスのプロパティ定義を Configuration API の設定内で有効にすると、以下の URL でそれらのプロパティ定義を表示することができます。

```

https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/<ドライバー名>/Channels

```

```

https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/<ドライバー名>/Devices

```

データ型列挙の例

ドキュメントエンドポイントでタグのデータ型のクエリーを実行すると、以下の列挙が表示されます。

```

{
  "Default": -1,
  "String": 0,
  "Boolean": 1,
  "Char": 2,
  "Byte": 3,
  "Short": 4,
  "Word": 5,
  "Long": 6,
  "DWord": 7,
  "Float": 8,
  "Double": 9,
  "BCD": 10,
  "LBCD": 11,
  "Date": 12,
  "LLong": 13,
  "QWord": 14,
  "String Array": 20,
  "Boolean Array": 21,
  "Char Array": 22,
  "Byte Array": 23,
  "Short Array": 24,
  "Word Array": 25,

```

```

"Long Array": 26,
"DWord Array": 27,
"Float Array": 28,
"Double Array": 29,
"BCD Array": 30,
"LBCD Array": 31,
"Date Array": 32,
"LLong Array": 33,
"QWord Array": 34
}

```

● **注記:** サポートされるデータ型は、プロトコルとドライバーによって異なります。

デバイスモデル列挙

デバイスモデルのプロパティには、以下に示す列挙にマッピングされる値が含まれています。以下の表は、参照用として記載されています。デバイスエンドポイントの詳細な情報と最新の情報については、以下の URL を参照してください。

<https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/Channels>

<https://<ホスト名または IP>:<ポート>/config/v1/doc/drivers/Allen-Bradley%20ControlLogix%20Ethernet/Devices>

列挙	デバイスモデル
0	ControlLogix 5500
1	CompactLogix 5300
2	FlexLogix 5400
3	SoftLogix 5800
4	DH+ ゲートウェイ: PLC-5
5	DH+ ゲートウェイ: SLC 5/04
6	ControlNet ゲートウェイ: PLC-5C
7	EIP ゲートウェイ: MicroLogix
8	EIP ゲートウェイ: SLC Fixed
9	EIP ゲートウェイ: SLC Modular
10	EIP ゲートウェイ: PLC-5
11	シリアルゲートウェイ: ControlLogix
12	シリアルゲートウェイ: CompactLogix
13	シリアルゲートウェイ: CompactLogix
14	シリアルゲートウェイ: FlexLogix
15	シリアルゲートウェイ: SoftLogix
16	ENI: ControlLogix 5500
17	ENI: FlexLogix 5400
18	ENI: MicroLogix
19	ENI: SLC 500 固定 I/O
20	ENI: SLC 500 モジュール I/O
21	ENI: PLC-5
22	MicroLogix 1100
23	MicroLogix 1400

Configuration API - スロット構成

- **注記:** ENI/DH+、ControlNet ゲートウェイ、および MicroLogix モデルでは自動タグデータベース生成はサポートされていません。これは ENI ControlLogix、CompactLogix、および FlexLogix モデルでのみサポートされています。

アトミックタグ -> 1 対 1 -> サーバータグ
 構造体タグ -> 1 対多 -> サーバータグ
 配列 -> 1 対多 -> サーバータグ

- データベース作成の設定の詳細については、サーバーのヘルプファイルを参照してください。

- **注記:** RSLogix5000 プログラミング環境で監視されたコントローラタグでは、「External Access」プロパティを「Read Only」または「Read/Write」に設定してタグを読み取る必要があります。自動的に生成されたタグでは、デフォルトで「External Access」が「None」に設定されている可能性があります。コントローラタグを読み取るには、RSLogix の Add-On Instruction パラメータで、必要に応じて External Access を再構成します。製造メーカーのドキュメントを参照してください。

タグ階層

自動タグ生成によって作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。この機能を使用するには、デバイスのプロパティの「サブグループを許可」を有効にします。

展開モード

展開モードでは、自動タグ生成によって作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。圧縮時と同様に、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が、論理グループ内にも作成されます。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)
- プログラム
- 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。

ルートレベルのグループ (または「親グループ」に指定されたサブグループレベルのグループ) には "Prgm_<プログラム名>" と "Global" があります。コントローラのプログラムそれぞれに独自の "Prgm_<プログラム名>" グループがあります。このドライバーは最初のグループレベルとしてこれを認識します。

基本グローバルタグ (非構造体、非配列タグ) は Global グループに配置されます。基本プログラムタグはそれぞれのプログラム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が、親グループ内の独自のサブグループに作成されます。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。

構造体/配列サブグループの名前から、その構造体/配列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、ここで 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。配列サブグループ内のタグはすべて、その配列の要素です (明示的な制限がないかぎり)。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。構造体に配列が含まれている場合、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。

複雑なプロジェクトでは、タグ階層に複数のグループレベルが必要です。自動タグ生成によって作成されるグループレベルの最大数は 7 です。これには「Add generated tags to the following group」で指定したグループは含まれません。7 つより多くのレベルが必要な場合、タグは 7 番目のグループに配置されます (階層が平坦になります)。

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

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

タグ名	アドレス
Member_00	TAG1_12_2_985
Member_01	TAG1_12_2_986
Member_02	TAG1_12_2_987
Member_03	TAG1_12_2_988
Member_04	TAG1_12_2_989
Member_05	TAG1_12_2_990
Member_06	TAG1_12_2_991
Member_07	TAG1_12_2_992
Member_08	TAG1_12_2_993
Member_09	TAG1_12_2_994

複雑な例

"Local:1:O.Slot[9].Data" というアドレスで Logix タグが定義されています。これは "Global" - "Local_1_O" - "Slot_x" - "Slot_09" というグループで表されます。最後のグループ内にはタグ "Data" があります。

"Data" への静的参照は "Channel1.Device1.Global.Local_1_O.Slot_x.Slot_09.Data" になります。"Data" への動的参照は "Channel1.Device1.Local:1:O.Slot[9].Data" になります。

圧縮モード

圧縮モードでは、自動タグ生成によって作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

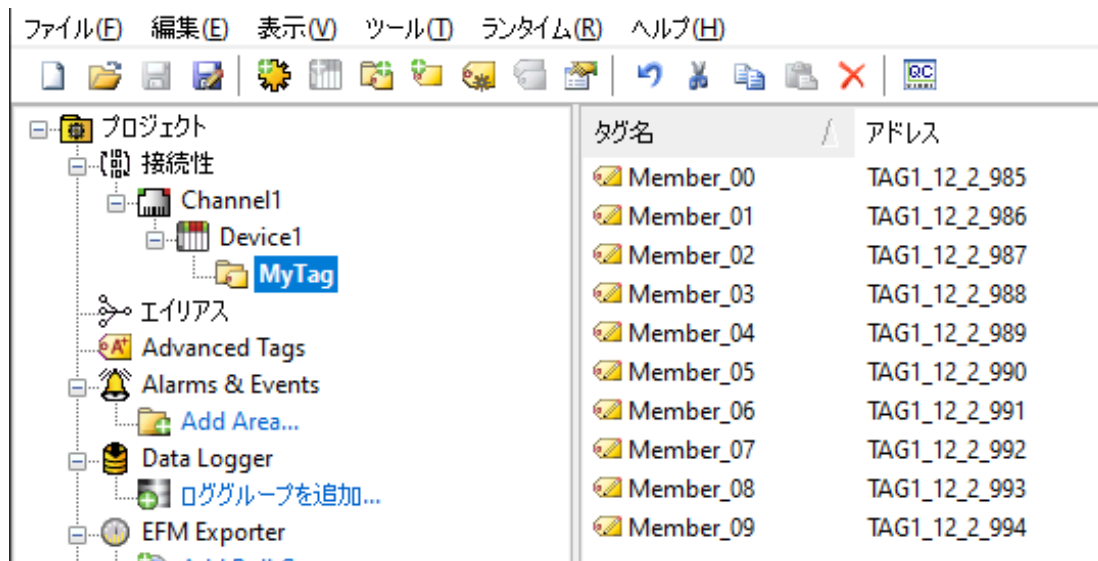
● **注記:** 配列と .bit アドレスにグループは作成されません。

複雑なプロジェクトでは、タグ階層に多数のグループレベルが必要であることは明らかです。自動タグ生成によって作成されるグループレベルの最大数は 7 です。これには「Add generated tags to the following group」で指定したグループは含まれません。7 つより多くのレベルが必要な場合、タグは 7 番目のグループに配置されます (階層が平坦になります)。

● **注記:** 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。詳細については、[コントローラからサーバーへの名前の変換](#)を参照してください。

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



複雑な例

"Local:1:O.Slot[9].Data" というアドレスで Logix タグが定義されています。これは "Local:1:O" -> "Slot[9]" というグループで表されます。最後のグループ内にはタグ "Data" があります。

"Data" への静的参照は "Channel1.Device1.Local:1:O.Slot[9].Data" になります。動的参照は "Channel1.Device1.Local:1:O.Slot[9].Data" になります。

● **注記:** オフラインモードで I/O モジュールタグを直接インポートすることはできません。エイリアスはインポート可能なので、RSLogix5000 で必要な I/O モジュールタグにエイリアスを作成することをお勧めします。

コントローラからサーバーへの名前の変換

先頭のアンダースコア

タグ名またはプログラム名の先頭のアンダースコア "_" は "U_" に置き換えられます。サーバーではアンダースコアで始まるタグ名やグループ名は使用できないので、この処理が必要となります。

長い名前 (OPC サーババージョン 4.64 以下)

古いバージョンの OPC サーバでは、Allen-Bradley ControlLogix Ethernet ドライバーのグループ名とタグ名は 31 文字に制限されていました。このため、コントローラプログラムまたはタグの名前が 31 文字を超えた場合、名前をクリップする必要がありました。OPC サーババージョン 4.70 以上では、文字数の上限は 256 文字であるため、この規則は適用されません。名前は次のようにクリップされます。

非配列

1. このタグの 5 桁の一意の ID を決める。
2. タグ名: ThisIsALongTagNameAndProbablyExceeds31
3. 31 文字でタグをクリップ: ThisIsALongTagNameAndProbablyEx
4. 一意の ID の場所を空ける: ThisIsALongTagNameAndProba#####
5. この ID を挿入: ThisIsALongTagNameAndProba00000

配列

1. この配列の 5 桁の一意の ID を決める。
2. 配列タグ名: ThisIsALongTagNameAndProbablyExceeds31_23_45_8
3. 要素の値を残して 31 文字になるようにタグをクリップ: ThisIsALongTagNameAndPr_23_45_8

4. 一意の ID の場所を空ける: ThisIsALongTagName#####_23_45_8
5. この ID を挿入: ThisIsALongTagName00001_23_45_8

長いプログラム名は長い非配列タグ名と同じ方法でクリップされます。タグ名またはプログラム名がクリップされるたびに、一意の ID が増分されます。クリップされた配列名の配列タグ名 (要素) には同じ一意の ID が付きます。これによって 100000 個の一意なタグ/プログラム名が用意されます。

● **注記:** 「名前の長さを制限」が有効化されている場合、256 文字の名前がサポートされていてもこの規則が適用されます。詳細については、「[Logix データベース設定](#)」を参照してください。

自動タグデータベース生成の準備

自動タグデータベース生成の使用方法については、以下の説明を参照してください。

オンライン

データベース作成プロセスの間は対象の Logix CPU へのすべての通信を停止することをお勧めします。

RSLogix5000

プロジェクトをオフラインに設定します。

サーバーで

1. タグが生成されるデバイスの「デバイスのプロパティ」を確認します。
2. 「Logix データベース設定」内の「データベースのインポート方法」で、「デバイスから作成」を選択します。
3. 「Logix データベースオプション」を必要に応じて変更し、「OK」をクリックします。
4. 「Logix データベースフィルタ」を必要に応じて変更し、「OK」をクリックします。
5. 「タグ生成」を選択し、「作成」の下で「[タグを作成](#)」という青色のリンクをクリックします。

● **注記:** 「Logix オプション」で、「プロトコルモード」を「シンボリック」、「デフォルトデータ型」を「デフォルト」に設定することで、コントローラで使用されているデータ型でタグがインポートされるようになります。

オフライン

Allen-Bradley ControlLogix Ethernet ドライバーでは、RSLogix5000 から生成された、L5K/L5X インポート/エクスポートファイルと呼ばれるファイルを使用して、タグデータベースが生成されます。

RSLogix5000

1. OPC サーバーに移すタグが含まれているプロジェクトを開きます。
2. 「ファイル」|「名前を付けて保存」をクリックします。
3. 「L5K/L5X Import/Export File」を選択し、名前を指定します。RSLogix によってこの L5K/L5X ファイルにプロジェクトのコンテンツがエクスポートされます。

OPC サーバー

1. タグが生成されるデバイスの「デバイスのプロパティ」を開きます。
2. 「Logix データベース設定」を選択し、「データベースのインポート方法」|「インポートファイルから作成」を選択します。
3. 以前に作成したファイルの場所を入力またはブラウズします。
4. 「Logix データベースオプション」を必要に応じて変更し、「OK」をクリックします。
5. 「Logix データベースフィルタ」を必要に応じて変更し、「OK」をクリックします。
6. 「タグ生成」を選択し、「作成」の下で「[タグを作成](#)」という青色のリンクをクリックします。

● **注記:** インポートされた定義済みのタグのデータ型は、ドライバーによってサポートされている最新バージョンに基づきます。詳細については、ファームウェアのバージョンを参照してください。

パフォーマンスの最適化

通信とアプリケーションを最適化する方法については、以下に示すいずれかのリンクを選択してください。

[通信の最適化](#)

[アプリケーションの最適化](#)

[パフォーマンス統計とチューニング](#)

[パフォーマンスチューニングの例](#)

通信の最適化

どのようなプログラマブルコントローラにも、パフォーマンスとシステム通信を向上させるさまざまな手段が備わっています。

プロトコルモード

プロトコルモードでは、コントローラから Logix タグデータへのアクセス方法を指定します。プロトコルモードにはシンボリック、論理非ブロック、論理ブロックの 3 つのタイプがあります。以下でこれについて説明します。

- **シンボリックモード:** パケット内の各クライアント/サーバータグアドレスはその ASCII 文字名によって表されます。
- **論理非ブロックモード:** 各クライアント/サーバータグは PLC 内のその論理メモリアドレスによって表されます。
- **論理ブロックモード:** Logix タグはひとまとまりのデータとしてアクセスされます。各クライアント/サーバータグ (MYTIMER.ACC など) には対応する Logix タグ (MYTIMER) があります。構造体と同様に、同じ Logix タグに多数のクライアント/サーバータグが属することができます。読み取りサイクルのたびに、Logix タグが読み取られ、ドライバーキャッシュ内でそのブロックが更新され、すべてのクライアント/サーバータグがこのキャッシュから更新されます。

Logix タグデータを収集して処理するには論理非ブロックモードが最も効率的なモードなので、一般的にはこのモードが推奨されます。後方互換性を確保するにはシンボリックモードが推奨され、UDT や定義済みの構造体 Logix タグへの参照がいくつか含まれているプロジェクトには論理非ブロックモードが推奨されます。論理ブロックモードは効率的ですが、間違えて使用した場合にはパフォーマンスが低下することもあります。各モードの利点と不利点の詳細については、[プロトコルモードの選択](#)を参照してください。

タグ分割のヒント

ユーザーは 1 つ以上のデバイスを論理ブロック用に指定し、1 つ以上のデバイスを論理非ブロック用に指定する必要があります。プロジェクト内のタグによって適したモードが異なることがよくあるため、これによってパフォーマンスが向上します。タグ分割を利用する場合、次の操作を行う必要があります。

1. アトミック Logix タグ (配列または非配列) を参照するサーバータグを論理非ブロックデバイスに割り当てます。
2. 構造体タグの 3 分の 1* 以下から構成される構造体 Logix タグを参照するサーバータグを論理非ブロックデバイスに割り当てます。たとえば、PID_ENHANCED Logix タグを参照するメンバータグが 55** 以下である場合、これらのタグすべてを論理非ブロックデバイスに割り当てる必要があります。
3. 構造体タグの 3 分の 1* 以上から構成される構造体 Logix タグを参照するサーバータグを論理ブロックデバイスに割り当てます。たとえば、PID_ENHANCED Logix タグを参照するメンバータグが 55** 以上である場合、これらのタグすべてを論理ブロックデバイスに割り当てる必要があります。

*3 分の 1 は厳密な制限ではなく、さまざまな分析に当てはまった数字です。

**構造体 PID_ENHANCED には 165 個のタグがあるため、その 3 分の 1 は 55 個になります。

接続サイズ

接続サイズを増やすことでデータ/パケットあたりの読み取り/書き込み要求の数が増え、これによってスループットが向上します。これによって CPU の負荷と応答時間も増えますが、パフォーマンスが大幅に向上します。接続サイズのプロパティは ControlLogix 5500 および CompactLogix 5300 デバイスモデルでのみ修正できます。詳細については、[Logix 通信パラメータ](#)を参照してください。

UDT サブ構造体のエイリアス

UDT に多数のサブ構造体が含まれ、サブ構造体のメンバーの 3 分の 1 以上がクライアントで参照される場合、以下の手順を参照してサブ構造体の読み取りを最適化してください。

1. RSLogix 5000 でサブ構造体のエイリアスを作成します。次に、残りの UDT サブ構造体を参照するサーバータグを論理ブロックデバイスに割り当てます。
2. 次に、残りの (サブ構造体ではない) UDT を参照するサーバータグを論理非ブロックデバイスに割り当てます。

システムオーバーヘッドタイムスライス

システムオーバーヘッドタイムスライス (SOTS) は、RSLogix 5000 で設定されている通信タスク (OPC ドライバー通信など) を実行するために割り当てられている時間の割合 (%) です。100% SOTS はコントローラタスク (ラダーロジックなど) の時間の割合 (%) です。デフォルトの SOTS は 10 % です。10 ミリ秒間のプログラムスキャンのうち、コントローラがドライバーの要求を処理する時間は 1 ミリ秒です (コントローラに連続タスクがある場合)。SOTS の値によってそのタスクの優先順位が決まります。コントローラタスクの優先順位が高い場合、SOTS を 30 % 未満に設定する必要があります。通信タスクの優先順位が高い場合、SOTS を 30 % 以上に設定する必要があります。通信のパフォーマンスと CPU 使用率の最適なバランスを実現するには、SOTS を 10 % から 40% に設定します。

複数要求パケット

Allen-Bradley ControlLogix Ethernet ドライバー は読み取りと書き込みを最適化するように設計されています。(要素を 1 つだけ要求する) 非配列、非文字列タグの場合、要求は 1 つのトランザクションにまとめられます。これによって単一タグのトランザクションよりもパフォーマンスが劇的に向上します。唯一、1 つのトランザクションに含めることができるデータバイトの数に制限があります。

● **重要:** シンボリックモードでは、タグの要求がそれ以上入らなくなるまで、要求パケットに各タグの ASCII 文字列値が挿入されます。最適なパフォーマンスを得るためには、タグ名のサイズを最小限に抑えてください。タグ名が小さいほど、1 つのトランザクションに含まれるタグの数が増え、すべてのタグを処理するために必要なトランザクションの数が減ります。

配列要素のブロック化 (シンボリックモードと論理非ブロックモードのみ)

アトミック配列要素の読み取りを最適化するには、配列を個別に読み取るのではなく 1 回の要求で配列のブロックを読み取ります。1 つのブロックで読み取る要素の数が多いほど、パフォーマンスが向上します。ほとんどの時間はトランザクションのオーバーヘッドと処理に割かれるので、できるだけ少ないトランザクションでできるだけ多くのタグをスキャンするようにします。これが配列要素ブロック化の要点です。

ブロックサイズは要素数として指定します。ブロックサイズを 120 要素として指定した場合、1 回の要求で最大 120 個の配列要素が読み取られます。ブロックの最大サイズは 3840 要素です。Boolean 配列は処理が異なり、プロトコルでは Boolean 配列は 32 ビット配列です。したがって、要素 0 を要求するということは、ビット 0 から 31 を要求することになります。説明が一貫したものになるように、Boolean 配列要素は 1 ビットと見なされます。つまり、(ブロックサイズを 3840 とした場合の) 要求可能な配列要素の最大数は 122880 BOOL、3840 SINT、3840 INT、3840 DINT、3840 REAL、3840 LINT、3840 UINT、3840 USINT、3840 ULINT、3840 LREAL、3840 TIME32、3840 TIME、および 3840 LTIME です。

Logix 通信パラメータで説明したように、ブロックサイズは調整可能であり、使用中のプロジェクトに基づいて選択する必要があります。たとえば、0 から 26 までの配列要素と配列要素 3839 が読み取り対象のタグである場合、ブロックサイズとして 3840 を使用するのはいくらも大きすぎるため、ドライバーのパフォーマンスが低下します。これは、0 から 3839 の要素のうち、重要であるのはその 28 個だけであるにもかかわらず、0 から 3839 のすべての要素が要求のたびに読み取られるためです。この場合、ブロックサイズを 30 にするのが妥当です。要素 0-26 は 1 回の要求で読み取られ、要素 3839 は次の要求で読み取られます。

文字列の最適化

論理アドレス指定モードでは、STRING.DATA に書き込むと STRING.LEN にも適切な長さ値が書き込まれます。

LEN で文字列データを終了

このドライバーでは、文字列タグは別個の文字データと長さ要素から成る構造体です。このため、ドライバーは 2 つのトランザクションで文字列タグを自動的に読み取ります。つまり、論理プロトコルモードで文字列の文字データ (DATA) を読み取り、シンボリックモードで文字列の長さ (LEN) を読み取ります。「LEN で文字列データを終了」オプションが無効になっている場合、文字列の文字データを読み取る 1 つのトランザクションだけが実行されます。その場合、文字列の長さを読み取るシンボリックモードの読み取りはバイパスされます。多数の文字列タグが含まれるプロジェクトでは、すべてのタグの読み取りに必要な時間をこれによって大幅に削減できます。

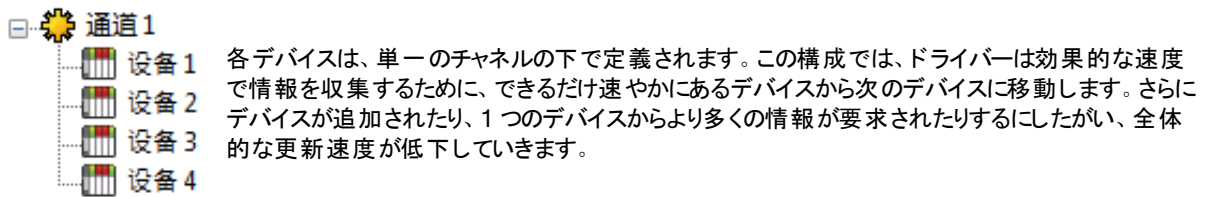
● 「LEN で文字列データを終了」オプションの詳細については、[Logix オプション](#)を参照してください。

アプリケーションの最適化

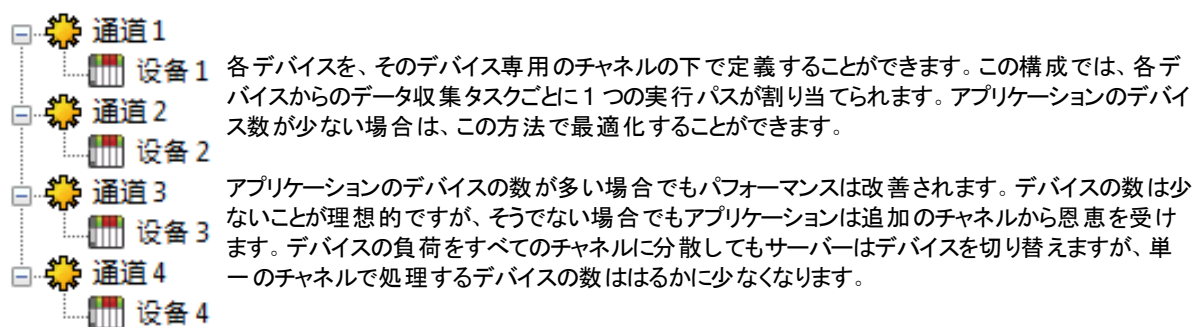
Allen-Bradley ControlLogix Ethernet ドライバー は、システム全体のパフォーマンスへの影響を最小限に抑えながら最大のパフォーマンスが得られるように設計されています。Allen-Bradley ControlLogix Ethernet ドライバー は高速ですが、このアプリケーションを最適化して最大のパフォーマンスを得るために参考となるいくつかのガイドラインがあります。

このサーバーでは、Allen-Bradley ControlLogix Ethernet などの通信プロトコルのことをチャネルと呼びます。アプリケーションで定義されている各チャネルは、サーバーでの個々の実行パスを表します。チャネルが定義された後、そのチャネルの下に一連のデバイスを定義する必要があります。これらのデバイスそれぞれが、データの収集元となる単一の Allen-

Bradley Logix CPU を表します。このアプローチに従ってアプリケーションを定義することで高いパフォーマンスが得られますが、Allen-Bradley ControlLogix Ethernet ドライバー やネットワークがフルに利用されるわけではありません。単一のチャンネルを使用して構成されているアプリケーションの表示例を次に示します。



ドライバーで定義できるチャンネルの数が1つだけの場合、上に示した例が唯一の方法になりますが、このドライバーでは、最大 1024 チャンネルを定義することができます。複数のチャンネルを使用して複数の要求をネットワークに同時に発行することで、データ収集のワークロードが分散されます。パフォーマンスを改善するために同じアプリケーションを複数のチャンネルを使用して構成した場合の例を次に示します。



パフォーマンス統計とチューニング

パフォーマンス統計機能では、アプリケーションのパフォーマンスに関するベンチマークと統計が提供されます。パフォーマンス統計は追加の処理層なので、サーバーのパフォーマンスに影響を与えることがあります。このため、デフォルトではオフになっています。パフォーマンス統計機能を有効にするには、「デバイスのプロパティ」にアクセスし、「Logix オプション」で「パフォーマンス統計」を有効にします。

パフォーマンス統計のタイプ

パフォーマンス統計は、デバイス、チャンネル、ドライバーという3つの領域において意味のある数値結果を提供します。タイプの説明は次のとおりです。

- **デバイス:** これらの統計では、個々のデバイスにおけるデータアクセスのパフォーマンスが提供されます。
- **チャンネル:** これらの統計では、「パフォーマンス統計」が有効になっているチャンネルの下にあるすべてのデバイスにおけるデータアクセスの平均パフォーマンスが提供されます。
- **ドライバー:** これらの統計では、「パフォーマンス統計」が有効になっている Allen-Bradley ControlLogix Ethernet ドライバーを使用しているすべてのデバイスにおけるデータアクセスの平均パフォーマンスが提供されます。

統計タイプの選択

必要な統計のタイプはアプリケーションによって異なります。一般に、ドライバー統計ではアプリケーションのパフォーマンスの実際の測定値が提供されるのに対し、チャンネル統計とデバイス統計はアプリケーションをチューニングする際に最も関連します。たとえば、10個のタグをデバイス A からデバイス B に移動するとデバイス A のパフォーマンスは向上するでしょうか？ デバイス A をチャンネル 1 からチャンネル 2 に移動するとチャンネル 1 のパフォーマンスは向上するでしょうか？ これらの疑問はデバイス統計とチャンネル統計を使用すべき状況の最も良い例です。

統計の検索

サーバー統計はシャットダウン時にサーバーのイベントログに出力されます。結果を表示するには、サーバーをシャットダウンしてから再起動します。

サーバー統計とパフォーマンス統計の違い

パフォーマンス統計では実行された読み取りのタイプの構成 (シンボリック、シンボルインスタンス、物理の構成比やデバイス読み取りとキャッシュ読み取りの構成比など) が示されますが、サーバー統計では読み取り全体での総合値が提供されます。

パフォーマンス向上のためのアプリケーションのチューニング

● デバイスとチャンネルの統計結果を向上させる方法については、以下の手順を参照してください。詳細については、[通信の最適化](#)を参照してください。

- アトミック Logix タグ (配列または非配列) を参照するサーバータグを論理非ブロックデバイスに割り当てます。
- 構造体タグの 3 分の 1 以下から構成される構造体 Logix タグを参照するサーバータグを論理非ブロックデバイスに割り当てます。
- 構造体タグの 3 分の 1 以上から構成される構造体 Logix タグを参照するサーバータグを論理ブロックデバイスに割り当てます。
- シンボリックモードを使用している場合、Logix 名の長さは最小限にしてください。
- できるだけ Logix 配列を使用する必要があります。
- ドライバー通信用に残すため、ラダーロジック/FBD 用のシステムオーバーヘッドタイムスライスに必要な量だけを割り当てます。
- 論理モードで多数の文字列タグを読み取るプロジェクトの場合、「デバイスのプロパティ」で「Logix オプション」の下にある「LEN で文字列データを終了」オプションを無効にします。

● ドライバー統計の結果を向上させる方法については、以下の手順を参照してください。詳細については、[アプリケーションの最適化](#)を参照してください。

- デバイスを複数のチャンネルに分散します。必要な場合を除き、1 つのチャンネルに複数のデバイスを配置しないでください。
- 負荷を複数のデバイスに均等に分散します。必要な場合を除き、1 つのデバイスに負荷をかけすぎないようにしてください。
- 異なるデバイス間で同じ Logix タグが参照されないようにしてください。

● **注記:** これらの一般規則はパフォーマンスの最適化に役立ちますが、最終的にはアプリケーションによって異なります。スキャンレートによって結果があいまいになることがあります。タグ要求が少ない場合、読み取り/書き込みトランザクションは次の要求が来る前に完了する可能性があります。その場合、論理ブロックと論理非ブロックでパフォーマンス統計の結果は同じになります。タグ要求が多い (タグが多いかスキャンレートが高い) 場合、トランザクションの完了に長い時間がかかることがあります。このような場合に、論理ブロックと論理非ブロックの長所と短所が明らかになります。パフォーマンス統計は最大のパフォーマンスが得られるようにアプリケーションをチューニングするのに役立ちます。例については、[パフォーマンスチューニングの例](#)を参照してください。

パフォーマンスチューニングの例

統計はすべてのアプリケーションに適用できます。以下の例では、パフォーマンスチューニングプロセスでクイッククライアントが使用されています。その目的は、プロジェクトで使用されているすべてのタグを高いスキャンレートで同時に読み取ることです。これは現実的ではありませんが、サーバー内のプロジェクトレイアウトについての正確なベンチマークを提供します (特定のデバイスに属するタグ、特定のチャンネルに属するデバイスなど)。

収集される統計は相対的なものです。ユーザーはサーバーのプロジェクトレイアウトから始めて、統計を収集してから、チューニングを行う必要があります。特定のレイアウトの結果を適切に評価するためには、複数のトライアルを使用することをお勧めします。最も効率的なレイアウトを特定した後は、そのサーバーが最適であるという安心感を持ってクライアントアプリケーションを構築できます。

● クイッククライアントを使用して取得したパフォーマンスの結果は、クライアントアプリケーションを使用して取得したパフォーマンスの結果とは等しくありません。いくつかの要因によって相違点が生じます。クライアントアプリケーションを使用したパフォーマンスチューニングの方がクイッククライアントよりも正確ですが、必要なチューニングはサーバープロジェクトだけでなくクライアントアプリケーションにも影響します。クイッククライアントを使用してアプリケーションをチューニングしてからクライアントアプリケーションを開発することをお勧めします。

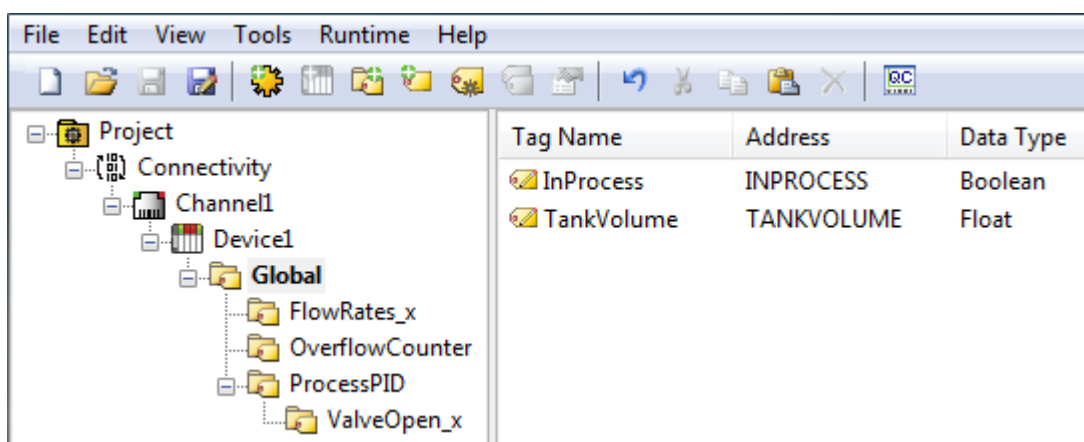
● **注記:** 以下で説明するチューニングプロセスでは、すべてのタグが高いスキャンレートで読み取られていることを前提としています。書き込みによってパフォーマンスは低下します。

1. 下に表示されているコントローラプロジェクトでは、以下があります。
 - 2 つのアトミック
 - 1 つのアトミック配列
 - 1 つの UDT
 - 1 つの UDT 配列
 - 1 つの定義済みデータ型

● **注記:** オーバーヘッドタイムスライス (OTS) = 10 %。

P	Tag Name	Alias For	Base Tag	Type
	InProcess			BOOL
<input type="checkbox"/>	+OverflowCounter			COUNTER
<input type="checkbox"/>	+ValveOpen			DINT[10]
<input type="checkbox"/>	+ProcessPID			PIDLoop
<input type="checkbox"/>	+FlowRates			ProcessVariable[2]
<input checked="" type="checkbox"/>	TankVolume			REAL
*				

2. このコントローラから自動タグデータベース生成を実行すると、サーバーによって以下のプロジェクトが生成されます。



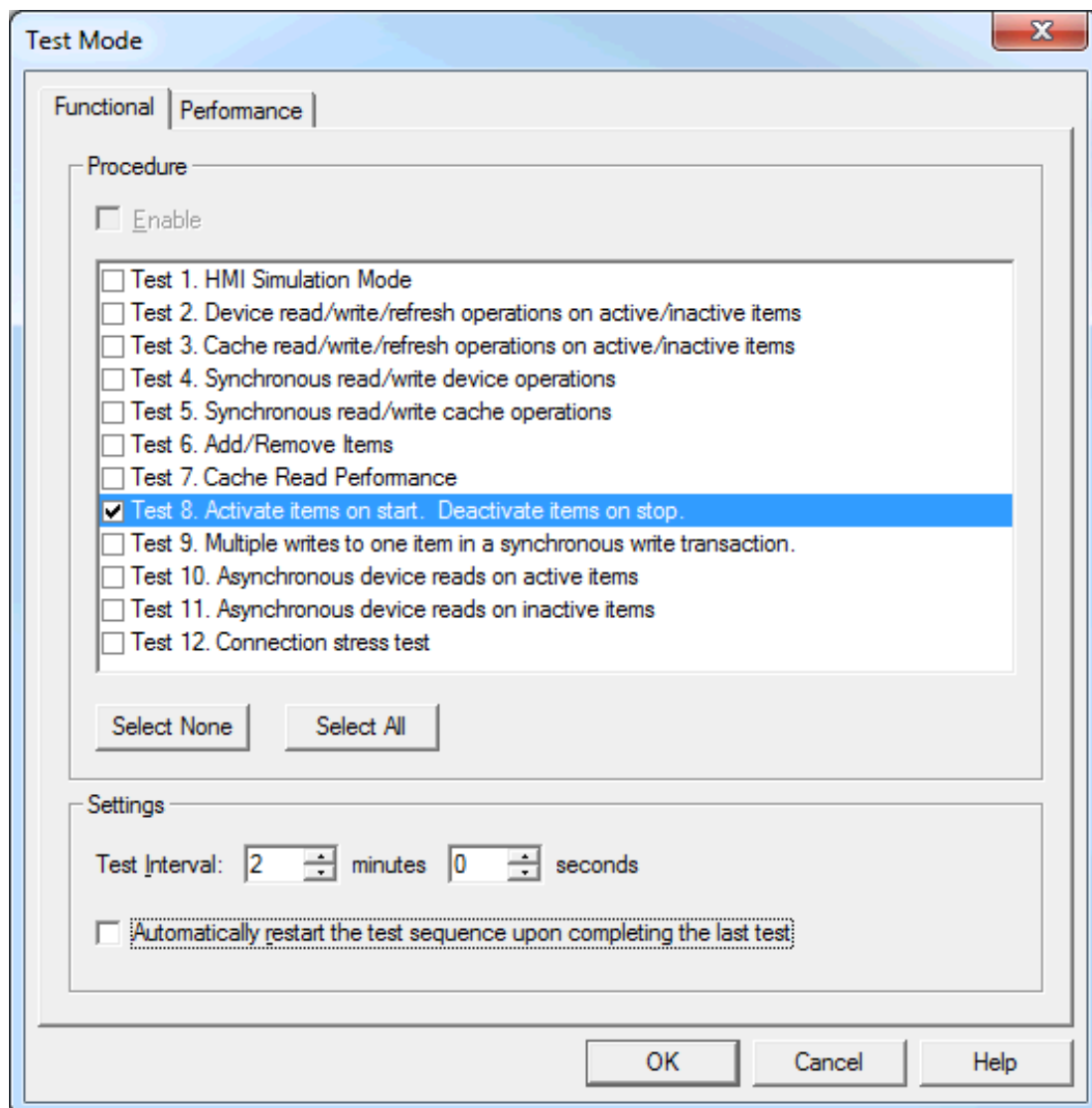
● **注記:** "グローバル" タググループには 130 個のタグが含まれています。

- タグ分割の利点を示すため、この例では一部のタグしか参照されていません。ProcessPID タグの 3 分の 1 以上、FlowRates タグの 3 分の 1 以下、およびその他すべてのタグが参照されています。このため、新しいタグの数は 105 です。
- クライアントでテストの準備を行います。このために、「クイッククライアント」アイコンをクリックして、サーバーアプリケーションからクイッククライアントを起動します。
- プロジェクトがロードされた後で、目的のタグが含まれているグループを除くすべてのグループを除去します。たとえば、統計タグとシステムタグは必要ありません。

● **注記:** 小さなプロジェクトでは、グループの「更新レート」を 0-10 ミリ秒に設定します。大きなプロジェクトでは、このレートを 10-50 ミリ秒に設定します。

- 「ツール」|「Test Mode」の順に選択します。
- 「Test 8. Activate items on start. Deactivate items on stop」を有効化してから、テスト期間を設定します。

● **注記:** このプロジェクトは比較的小さいので、この期間は 2 分に設定されています。大きなプロジェクトの場合、より正確な測定値を得るためにはこの期間を長くする必要があります。



8. 「ツール」|「Test Mode」に戻ってテストモードを無効にします。すべてのタグを無効化する必要があります。
9. タイムトライアルが開始できるようにクイッククライアントを切断します。
10. サーバーをシャットダウンします。
11. サーバーを起動し、デバイスのプロパティの「プロトコルモード」を「論理ブロック」に設定します。

12. 「パフォーマンス統計」を「はい」に設定します。

プロパティグループ	ENI DF1/DH+/CN ゲートウェイ通信パラメータ	
一般	TCP/IP ポート	44818
スキャンモード	要求サイズ (バイト)	232
タイミング	ファンクションファイルのブロック書き込み...	無効化
自動格下げ		
タグ生成		
Logix 通信パラメータ		
Logix オプション		
Logix データベース設定		
ENI DF1/DH+/CN ゲートウ...		
スロット構成		
冗長		

13. クイッククライアントを使用してサーバーに接続します。「ツール」|「Test Mode」の順に選択します。テストモードを有効にします。

● **注記:** データの読み取りが開始します。テスト期間が経過すると、すべてのタグが無効化され、ドライバーは統計収集を終了します。結果が表示可能になります。

14. クイッククライアントをサーバーから切断し、サーバーをシャットダウンします。
15. サーバーを再起動し、そのイベントログで統計をサーチします。以下の図には、デバイスに論理ブロックを使用した1回目のトライアルが表示されています。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5972.26

```

● **注記:** 以下の図には、チャンネルとドライバーに論理ブロックを使用した1回目のトライアルが表示されています。


```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 5972.26
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 40932
Physical Block Cache Reads = 661734
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 13644
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 716390, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5972.26

```

● **注記:** これは比較のための制御のセットです。

16. サーバーで、「**プロトコルモード**」を「**論理非ブロック**」に設定します。
17. クイッククライアントを使用してサーバーに接続します。「**ツール**」|「**Test Mode**」の順に選択してテストモードを有効にします。

● **注記:** データの読み取りが開始します。テスト期間が経過すると、すべてのタグが無効化され、ドライバーは統計収集を終了します。結果が表示可能になります。
18. クイッククライアントをサーバーから切断し、サーバーをシャットダウンします。
19. サーバーを再起動し、そのイベントログで統計をサーチします。以下の図には、デバイスに論理非ブロックを使用した2回目のトライアルが表示されています。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 3705.23

```

● **注記:** 以下の図には、チャンネルとドライバーに論理非ブロックを使用した2回目のトライアルが表示されていません。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 3705.23
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 8254
Physical Non Block, Array Block Cache Reads = 174419
Physical Non Block Device Reads = 261716
Symbolic, Array Block Device Reads = 2
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 63
Total tags read = 444454, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 3705.23

```

20. サーバーから、「**プロトコルモード**」を「**シンボリック**」に設定して、Allen-Bradley ControlLogix Ethernet ドライバーバージョン 4.6.0.xx より前でのパフォーマンスを確認します。
21. クイッククライアントを使用してサーバーに接続します。次に、「**ツール**」|「**Test Mode**」の順にクリックしてテストモードを有効にします。
 - **注記:** データの読み取りが開始します。テスト期間が経過すると、すべてのタグが無効化され、ドライバーは統計収集を終了します。結果が表示可能になります。
22. クイッククライアントをサーバーから切断し、サーバーをシャットダウンします。
23. サーバーを再起動し、そのイベントログで統計をサーチします。以下の図には、デバイスにシンボリックを使用した3回目のトライアルが示されています。

```

DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 777.442

```

以下の図には、チャンネルとドライバーにシンボリックを使用した3回目のトライアルが示されています。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 777.442
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 0
Symbolic, Array Block Device Reads = 1744
Symbolic, Array Block Cache Reads = 36613
Symbolic Device Reads = 54985
Total tags read = 93342, Elapsed Time = 120063 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 777.442

```

● **注記:** このアプリケーションには論理ブロックが最適であることがわかります。

チャンネル通信の最適化

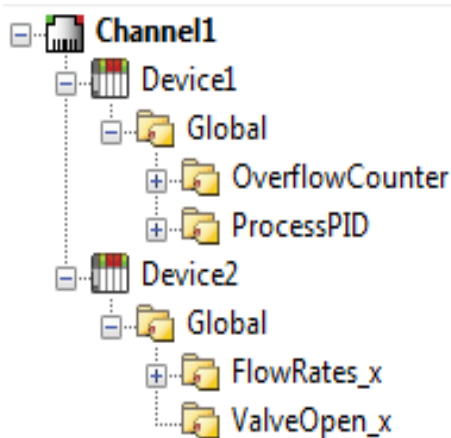
論理ブロックのタグを一方のデバイスに移動し、論理非ブロックタイプのタグをもう一方のデバイスに移動することによって、チャンネル通信を最適化できます。これをタグ分割と呼びます。

論理ブロック (Device1)

ProcessPID
OverflowCounter

論理非ブロック (Device2)

FlowRate
ValveOpen
InProcess
Tank Volume



1. ステップ 4 から 15 を繰り返します。ステップ 11 で、デバイス 1 を論理ブロック、デバイス 2 を論理非ブロックにします。
2. サーバーを起動し、サーバーのイベントログで統計をサーチします。以下の図には、デバイスにタグ分割を使用した 4 回目のトライアルが表示されています。

```
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 6933
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 76
Total tags read = 630979, Elapsed Time = 119782 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5268.43
DEVICE Channel1:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 27732
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 104044, Elapsed Time = 119969 ms
DEVICE Channel1:Device2 PERFORMANCE: AvgTagReadPerSec = 867.373
```

- **注記:** 以下の図には、チャンネルとドライバーにタグ分割を使用した 4 回目のトライアルが表示されています。

```
DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6126.77
Closing project C:\RDM\Support\ControlLogix Ethernet\CL_DEFAULT.opf
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 13866
Physical Block Cache Reads = 610104
Physical Non Block, Array Block Device Reads = 6933
Physical Non Block, Array Block Cache Reads = 69375
Physical Non Block Device Reads = 34665
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 735023, Elapsed Time = 119969 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 6126.77
```

● **注記:** 2つのデバイスは別々の統計カウンタで実行しているため、個々のデバイス統計は特別良い値ではありません。このテストの鍵となるのは、チャンネルとドライバーの統計 (6126) が、1チャンネル/1デバイスで論理ブロック (5972) または論理非ブロック (3705) を使用したものよりも優れているという点です。

アプリケーションの最適化

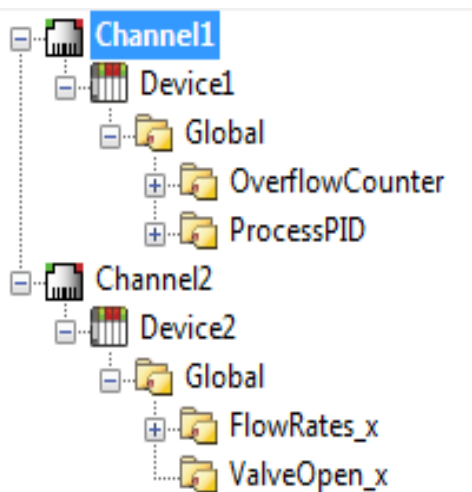
デバイス1を一方のチャンネル、デバイス2をもう一方のチャンネルに移動することによってアプリケーションを最適化できます。

論理ブロック (Channel1.Device1)

ProcessPID
OverflowCounter

論理非ブロック (Channel2.Device 2)

FlowRate
ValveOpen
InProcess
Tank Volume



1. ステップ4から15を繰り返します。ステップ11で、Channel1のDevice1を論理ブロック、Channel2のDevice2を論理非ブロックにします。
2. サーバーを起動し、サーバーのイベントログで統計をサーチします。以下の図には、Channel1のDevice1にLogixタグと複数のチャンネルを使用した5回目のトライアルが表示されています。

```
CHANNEL Channel1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
CHANNEL Channel1 PERFORMANCE: AvgTagReadPerSec = 5517.4
DEVICE Channel1:Device1 STATISTICS
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 0
Physical Non Block, Array Block Cache Reads = 0
Physical Non Block Device Reads = 7271
Symbolic, Array Block Device Reads = 0
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 80
Total tags read = 661741, Elapsed Time = 119968 ms
DEVICE Channel1:Device1 PERFORMANCE: AvgTagReadPerSec = 5517.4
```

● **注記:** 以下の図には、Channel2 の Device2 に Logix タグを使用した 4 回目のトライアルが表示されています。

```
CHANNEL Channel2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
CHANNEL Channel2 PERFORMANCE: AvgTagReadPerSec = 910.52
DEVICE Channel2:Device2 STATISTICS
Physical Block Device Reads = 0
Physical Block Cache Reads = 0
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 29120
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 4
Total tags read = 109205, Elapsed Time = 119968 ms
DEVICE Channel2:Device2 PERFORMANCE: AvgTagReadPerSec = 910.52
```

● **注記:** 以下の図には、ドライバーにタグ分割を使用した 4 回目のトライアルが表示されています。

```

DRIVER STATISTICS (ALL CHANNELS)
Physical Block Device Reads = 14542
Physical Block Cache Reads = 639848
Physical Non Block, Array Block Device Reads = 7280
Physical Non Block, Array Block Cache Reads = 72800
Physical Non Block Device Reads = 36391
Symbolic, Array Block Device Reads = 1
Symbolic, Array Block Cache Reads = 0
Symbolic Device Reads = 84
Total tags read = 770946, Elapsed Time = 119968 ms
DRIVER PERFORMANCE: AvgTagReadPerSec = 6426.26

```

結果

サーバープロジェクトのレイアウト	ドライバーのパフォーマンス (読み取り回数/秒)	シンボリックを基準 とした改善
1 チャネル 1 デバイス (論理ブロック)	5972	768 %
1 チャネル 1 デバイス (論理非ブロック)	3705	476 %
1 チャネル 1 デバイス (シンボリック)	777	該当なし
1 チャネル 複数デバイス (タグ分割)	6126	788 %
複数チャネル 複数デバイス (タグ分割)	6426	827%

結論

1 つのコントローラでのデフォルトの動作である、1 つのチャネルと 1 つのデバイスを使用してこのプロジェクトを開始しました。このコントローラからこのチャネルのデバイスにすべてのタグをインポートしました。次に、3 つのプロトコルモードすべてをテストし、そのうちのどれが最も高いパフォーマンスを実現するかを調べました。この場合、論理ブロックプロトコルが最適でした。最適なプロトコルは実行中のアプリケーションによって異なります。パフォーマンスが重要である場合、そのアプリケーションに最適なプロトコルモードを調べるには、論理ブロックと論理非ブロックのトライアルは実行するだけの価値があります。シンボリックプロトコルは、ほかの 2 つのプロトコルモードのパフォーマンスの水準に達することはないので、必要ありません。ここでは、これは一例として示されています。

通信の最適化で概要を説明しているヒントに従って、通信を最適化する措置を講じました。最も注目する点として、タグ分割を使用して、論理ブロックが指定されているデバイスに論理ブロックタイプのタグを配置し、論理非ブロックが指定されているデバイスに論理非ブロックタイプのタグを配置しました。さらに、両方のデバイスは同じチャネル上にあります。結果では、1 つのデバイスで論理ブロックを使用したものより改善が見られました。これは、一部のタグではどちらか一方のプロトコルモードの方が適しているためです。たとえば、COUNTER 全体の読み取りでは論理非ブロックよりも論理ブロックの方に利点があり、これは COUNTER を個々のメンバーとして読み取るよりもブロックとして読み取ったほうがはるかに速いためです。

それぞれのデバイスを独自のチャネルに配置することによってアプリケーションを最適化するという措置も講じました。前のトライアルで作成したデバイスを使用して、論理ブロックデバイスを一方のチャネル上に配置し、論理非ブロックデバイスももう一方のチャネル上に配置しました。結果では、前のトライアルで使用した 1 チャネル/複数デバイスのシナリオより改善が見られました。ここから、チャネルあたりのデバイスの数を最小にして必要な数だけチャネルを使用することでパフォーマンスが改善するという予想が裏打ちされました。

これら 3 つの最適化手法を使用した結果、このプロジェクトはバージョン 4.6.0.xx より前の Allen-Bradley ControlLogix Ethernet ドライバーと比較してパフォーマンスが 827 % 改善しました。タグ分割と複数チャネルではパフォーマンスが 107 % 改善しました。パフォーマンスはプロジェクトが大きいほど顕著な改善が見られます。

データ型の説明

データ型	説明
Boolean	1 ビット
Byte	符号なし 8 ビット値
Char	符号付き 8 ビット値
Word	符号なし 16 ビット値
Short	符号付き 16 ビット値
DWord	符号なし 32 ビット値
Long	符号付き 32 ビット値
BCD	2 バイトパックされた BCD、4 桁の 10 進数
LBCD	4 バイトパックされた BCD、8 桁の 10 進数
Float	32 ビット IEEE 浮動小数点
Double	64 ビット IEEE 浮動小数点
Date	64 ビットの日付/時刻
String	Null 終端文字配列

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

● ファームウェア V30 の Boolean 配列の詳細については、PTC Web サイトにアクセスし、eSupport にログインしてアーティクル cs332995 を検索してください。

デフォルトデータ型の条件

次のいずれかの条件が満たされる場合、クライアント/サーバータグにはデフォルトのデータ型が割り当てられます。

1. クライアントでネイティブのデータ型が割り当てられた動的タグが作成された場合。
2. サーバーでデフォルトのデータ型が割り当てられた静的タグが作成された場合。
3. オフライン自動タグ生成時に、L5K/L5X ファイルで UDT メンバーとエイリアスタグに不明なデータ型が見つかった場合。
4. オフライン自動タグ生成時に、L5K/L5X で次のタイプのエイリアスが見つかった場合。
 - a. エイリアスのエイリアス。
 - b. Word/DWord 内のビットではない I/O モジュールタグのエイリアス。たとえば、タグ "AliasTag" が I/O モジュールタグ "Local:5:C.ProgToFaultEn" @ BOOL を参照する場合、"AliasTag" のデータ型は解決できないので、デフォルトのデータ型が割り当てられます。これに対し、"AliasTag" が I/O モジュールタグ "Local:5:C.Ch0Config.RangeType.0" @ BOOL を参照する場合、Word/DWord 内ビットとして定義する . (ドット) ビットによりデータ型を解決できます。Word/DWord 内ビット I/O モジュールタグのエイリアスには Boolean データ型が自動的に割り当てられます。

● 注記:

1. 「デフォルト」を選択すると、クライアントがタグに動的にアクセスしていてアイテムにデータ型を明示的に割り当てない場合、ドライバーはコントローラから Logix タグのデータ型を取得します。たとえば、コントローラ内には REAL データ型の "MyTag" というタグが存在するとします。これに対応するクライアントアイテムは "Channel1.Device1.MyTag" として指定され、データ型は割り当てられていません。サーバーでデフォルトのデータ型として「デフォルト」が選択されている場合、ドライバーはコントローラから "MyTag" を読み取って、応答でこれが REAL データ型であることを特定し、クライアントには Float データ型を返します。
2. I/O モジュールタグの大部分は Word/DWord 内ビットタグではないので、.ACD プロジェクトで最も多く使用されているデータ型にデフォルトのデータ型を設定することをお勧めします。たとえば、エイリアス I/O モジュールタグの 75 % が INT 型のタグである場合、デフォルトのデータ型を INT に設定します。

アドレスの説明

アドレスの様子は使用されているモデルによって異なります。対象のモデルのアドレス情報については、以下の表を参照してください。

モデル	Output	Input	Status	Binary	Timer	Counter	Control	Integer	Float	ASCII	String	BCD	Long	PID	Message	Block Transfer	Function
MicroLogix	X	X	X	X	X	X	X	X	X	X	X	X		X	X		
PLC5	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	
SLC5/05	X	X	X	X	X	X	X	X	X	X	X						

関連項目:

[Logix のアドレス指定](#)

[MicroLogix のアドレス指定](#)

[PLC-5 シリーズのアドレス指定](#)

[SLC 500 モジュール I/O のアドレス指定](#)

プロトコルクラス	モデル	ヘルプリンク
Logix-Ethernet	ControlLogix 5500 Ethernet、CompactLogix 5300 Ethernet、FlexLogix 5400 Ethernet、SoftLogix 5800	Logix のアドレス指定
DH+ ゲートウェイ	DH+ ゲートウェイ: PLC-5 DH+ ゲートウェイ: SLC 5/04	PLC-5 シリーズのアドレス指定 SLC 500 モジュール I/O のアドレス指定
ControlNet ゲートウェイ	ControlNet ゲートウェイ: PLC-5C	PLC-5 シリーズのアドレス指定
1761-NET-ENI	ENI: ControlLogix 5500 ENI: CompactLogix 5300 ENI: FlexLogix 5400 ENI: MicroLogix ENI: SLC 500 固定 I/O ENI: SLC 500 モジュール I/O ENI: PLC-5	Logix のアドレス指定 MicroLogix のアドレス指定 SLC 500 固定 I/O のアドレス指定 SLC 500 モジュール I/O のアドレス指定 PLC-5 シリーズのアドレス指定
MicroLogix 1100 Ethernet	MicroLogix 1100	MicroLogix のアドレス指定
MicroLogix 1400 Ethernet	MicroLogix 1400	MicroLogix のアドレス指定

● コントローラの定義済みデータ型の詳細については、各デバイスのドキュメントを参照してください。

Logix のアドレス指定

● これらのモデルのタグベースのアドレス指定および Allen-Bradley ControlLogix Ethernet ドライバー との関係の詳細については、[Logix タグベースのアドレス指定](#) を参照してください。

イーサネットでの ControlLogix 5500 のアドレス指定

ControlLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

ENI での ControlLogix 5500 のアドレス指定

ControlLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

シリアルゲートウェイでの ControlLogix 5500 のアドレス指定

ControlLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

イーサネットでの CompactLogix 5300 のアドレス指定

CompactLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

ENI での CompactLogix 5300 のアドレス指定

CompactLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

シリアルゲートウェイでの CompactLogix 5300 のアドレス指定

CompactLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

イーサネットでの FlexLogix 5400 のアドレス指定

FlexLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

ENI での FlexLogix 5400 のアドレス指定

FlexLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

シリアルゲートウェイでの FlexLogix 5400 のアドレス指定

FlexLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

SoftLogix 5800 のアドレス指定

SoftLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

シリアルゲートウェイでの SoftLogix 5800 のアドレス指定

SoftLogix は Logix ファミリーのメンバーであり、Rockwell Automation の統合アーキテクチャを構成しています。つまり、タグまたはシンボルベースのアドレス指定構造が使用されます。Logix 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自体がアドレスになります。

MicroLogix のアドレス指定

イーサネット/IP ゲートウェイでの MicroLogix のアドレス指定

使用可能なアドレスの実際数は PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)

[入力ファイル](#)

[ステータスファイル](#)

[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[Long ファイル](#)
[MicroLogix PID ファイル](#)
[MicroLogix メッセージファイル](#)

ファンクションファイルの詳細については、以下のリストからリンクを選択してください。

[高速カウンタファイル \(HSC\)](#)
[リアルタイムクロックファイル \(RTC\)](#)
[チャンネル 0 通信ステータスファイル \(CS0\)](#)
[チャンネル 1 通信ステータスファイル \(CS1\)](#)
[I/O モジュールステータスファイル \(IOS\)](#)

ENI での MicroLogix のアドレス指定

使用可能なアドレスの実際数は PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[Long ファイル](#)
[MicroLogix PID ファイル](#)
[MicroLogix メッセージファイル](#)

ファンクションファイルの詳細については、以下のリストからリンクを選択してください。

[高速カウンタファイル \(HSC\)](#)
[リアルタイムクロックファイル \(RTC\)](#)
[チャンネル 0 通信ステータスファイル \(CS0\)](#)
[チャンネル 1 通信ステータスファイル \(CS1\)](#)
[I/O モジュールステータスファイル \(IOS\)](#)

MicroLogix 1100 のアドレス指定

使用可能なアドレスの実際数は PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[文字列ファイル](#)
[Long ファイル](#)
[MicroLogix PID ファイル](#)
[MicroLogix メッセージファイル](#)

ファンクションファイルの詳細については、以下のリストからリンクを選択してください。

[高速カウンタファイル\(HSC\)](#)
[リアルタイムクロックファイル\(RTC\)](#)
[チャンネル0 通信ステータスファイル\(CS0\)](#)
[チャンネル1 通信ステータスファイル\(CS1\)](#)
[I/O モジュールステータスファイル\(IOS\)](#)

MicroLogix 1400 のアドレス指定

使用可能なアドレスの実際の数は PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[Long ファイル](#)
[MicroLogix PID ファイル](#)
[MicroLogix メッセージファイル](#)

ファンクションファイルの詳細については、以下のリストからリンクを選択してください。

[高速カウンタファイル\(HSC\)](#)
[リアルタイムクロックファイル\(RTC\)](#)
[チャンネル0 通信ステータスファイル\(CS0\)](#)
[チャンネル1 通信ステータスファイル\(CS1\)](#)
[I/O モジュールステータスファイル\(IOS\)](#)

SLC 500 固定 I/O のアドレス指定

イーサネット/IP ゲートウェイでの SLC 500 固定 I/O のアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)

ENI での SLC 500 固定 I/O のアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)

SLC 500 モジュール I/O のアドレス指定

DH+ での SLC 500 モジュール I/O のアドレス指定

使用可能なアドレスの実際の数には PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)

イーサネット/IP ゲートウェイでの SLC 500 モジュール I/O のアドレス指定

使用可能なアドレスの実際の数には PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバーが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)

[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)

ENI での SLC 500 モジュール I/O のアドレス指定

使用可能なアドレスの実際の数には PLC のモデルによって異なります。将来のモデルでの柔軟性を最大限に確保するために一定の範囲が空けられています。ドライバが実行時にデバイスにアドレスが存在しないことを検出した場合、エラーメッセージを送信し、そのスキャンリストからタグを除去します。ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)

PLC-5 シリーズのアドレス指定

ControlNet での PLC-5 シリーズのアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[BCD ファイル](#)
[PID ファイル](#)
[メッセージファイル](#)
[ブロック転送ファイル](#)

DH+ での PLC-5 シリーズのアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)

[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[BCD ファイル](#)
[PID ファイル](#)
[メッセージファイル](#)
[ブロック転送ファイル](#)

イーサネット/IP ゲートウェイでの PLC-5 シリーズのアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[BCD ファイル](#)
[PID ファイル](#)
[メッセージファイル](#)
[ブロック転送ファイル](#)

ENI での PLC-5 シリーズのアドレス指定

ファイル固有のアドレス指定の詳細については、以下のリストからリンクを選択してください。

[出力ファイル](#)
[入力ファイル](#)
[ステータスファイル](#)
[バイナリファイル](#)
[タイマーファイル](#)
[カウンタファイル](#)
[制御ファイル](#)
[整数ファイル](#)
[Float ファイル](#)
[ASCII ファイル](#)
[文字列ファイル](#)
[BCD ファイル](#)
[PID ファイル](#)
[メッセージファイル](#)
[ブロック転送ファイル](#)

Logix タグベースのアドレス指定

Rockwell Automation の統合アーキテクチャでは、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自体がアドレスになります。

● **注記:** このヘルプファイル全体として、特に指定がないかぎり、Logix タグはグローバルであるものとします。

Allen-Bradley ControlLogix Ethernet ドライバー では、BOOL、SINT、INT、DINT、LINT、REAL、LREAL、USINT、UINT、UDINT、ULINT、TIME32、TIME、および LTIME のコントローラのアトミックデータ型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_UI1	-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_I8	-9,223,372,036,854,775,808 から 9,223,372,036,854,775,807
USINT	符号なし 8 ビット値	VT_UI1	0 から 255
UINT	符号なし 16 ビット値	VT_UI2	0 から 65535
UDINT	符号なし 32 ビット値	VT_UI4	0 から 4294967295
ULINT	符号なし 64 ビット値	VT_I8	0 から 18,446,744,073,709,551,615
REAL	32 ビット IEEE 浮動小数点	VT_R4	±1.1754943508222875E-38 から ±3.4028234663852886E+38 (正規化) 0 ±1.4012984643248170E-45 から ±1.1754942106924411E-38 (非正規化)
LREAL	符号付き 64 ビット値	VT_R8	±2.2250738585072014E-308 から ±1.7976931348623157E+308 (正規化) 0、 ±4.9406564584124654E-324 から ±2.2250738585072010E-308 (非正規化)
TIME32	文字列としての経過時間 (マイクロ秒)	VT_BSTR	T32#-35m_47s_483ms_647us から T32#35m_47s_483ms_647us
	符号付き 32 ビット値としてのマイクロ秒単位の経過時間	VT_I4	-2147483647 から 2147483647
TIME	文字列としての経過時間 (マイクロ秒)	VT_BSTR	T#-31d_12h_59m_59s_999ms_999us から T#31d_12h_59m_59s_999ms_999us
	符号付き 64 ビット値としてのマイクロ秒単位の経過時間	VT_I8	-2725199999999 から 2725199999999
LTIME	文字列としての経過時間 (ナノ秒)	VT_BSTR	LT#-31d_12h_59m_59s_999ms_999us_999ns から LT#31d_12h_59m_59s_999ms_999us_999ns
	符号付き 64 ビット値としてのナノ秒単位の経過時間	VT_I8	-2725199999999999 から 2725199999999999

● **関連項目:** [Logix の詳細なアドレス指定](#)

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。
- 最大 40 文字まで使用できます。
- 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。

クライアント/サーバータグ名の規則

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。

● **注記:** 最適なパフォーマンスを得るためには、Logix タグ名のサイズを最小限に抑えてください。名前が小さいほど、1 つのトランザクションに含めることができる要求の数が増えます。

● シンボリックモードのユーザーはクライアント/サーバータグアドレスを 400 文字以下にしてください。たとえば、tagarray [1,2,4].somestruct.substruct_array[3].basetag.[4] の長さは 57 文字です。パケットには 500 データバイトしか格納できないので、パケットに追加する必要がある付加バイトによって文字そのものに使用可能なスペースが大幅に減る可能性があります。アドレスを 400 文字以下にすることで、タグ要求は完全かつ有効なままとなります。

● 関連項目: [パフォーマンスの最適化](#)

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。使用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については、以下の表を参照してください。

● **注記:** 配列と String 型を除くフォーマットはすべて RSLogix5000 ネイティブです。したがって、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フィールドに貼り付けることでそのタグ名が有効になります。

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> [offset1,offset2] {列数} <Logix 配列要素タグ> [offset1,offset2] {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。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せん
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><最大文字列長>	tag_1.Data/4 Stringtag_1.Data/82 SINTarraytag_1/16	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、[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。プログラムタグはグローバルタグと同じアドレス指定の規則と制限に従いますが、先頭に次の表記が付きまます。

Program: <プログラム名>

たとえば、プログラム "prog_1" 内の Logix タグ "tag_1" のアドレスは、クライアント/サーバータグアドレスでは "Program:prog_1.tag_1" として指定されます。

構造体タグのアドレス指定

Logix 構造体タグ (グローバルまたはプログラム) は、1 つ以上のメンバータグを持つタグです。メンバータグのデータ型はアトミックまたは構造体型です。

<構造体名> . <サブ構造体名> . <アトミック型のタグ> <アトミック型のタグ>

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> . <アトミック型のタグ> <サブ構造体名> . <アトミック型のタグ>

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] . <アトミック型のタグ>

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> . <アトミック型のタグ> <サブ構造体の配列名> [次元 1, 次元 2, 次元 3] . <アトミック型のタグ>

● **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ほん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、Allen-Bradley または Rockwell のドキュメントを参照してください。

内部タグ

内部タグはサーバー構成には表示されませんが、OPC クライアントでブラウズすると、<チャンネル名>.<デバイス名> グループの下に表示されます。次のタグは、ControlLogix 5500 デバイスマジュールと CompactLogix 5300 デバイスマジュールでのみ有効です。

タグ名	サポート	データ型	アクセス
_CIPConnectionSizeRequested	最後に要求された CIP 接続サイズ。	Word	読み取り 書き込み *

タグ名	サポート	データ型	アクセス
_CIPConnectionSizeActual	使用している実際の CIP 接続サイズ。要求した値がデバイスでサポートされていない場合、その値は _CIPConnectionSizeRequested とは異なる値になります。	Word	読み取り専用
_LogicalAddressUploadCount	起動後またはリセット後に発生したコントローラプロジェクトのアップロード回数。「0」を設定すると、この回数がリセットされます。	Word	読み取り書き込み
_LogicalAddressUploadDuration	コントローラプロジェクトの最後のアップロードにかかった時間。次のアップロードが実行されるまで、この値がタグ内に保持されます。	Double	読み取り専用
_LastEditDetectedTimestamp	コントローラプロジェクトの最後の編集処理 (オンラインまたはオフライン) が検出されたときのタイムスタンプ。	String	読み取り専用
_LastLogicalAddressUploadTimestamp	コントローラプロジェクトの最後のアップロードが開始されたときのタイムスタンプ。	String	読み取り専用

*このタグは ENI Logix モデルでは読み取り専用です。

CIP 接続サイズの変更

_CIPConnectionSizeRequested タグを使用することで、CIP 接続サイズプロパティをリアルタイムで変更できます。クライアントが接続されている間は、接続サイズプロパティ (「**デバイスのプロパティ**」の「**Logix 通信パラメータ**」) とシステムタグの両方を設定できます。変更は次の読み取り書き込み要求が実行される前に適用されます。

定義済みの用語タグ

以下の表に示されているタグを使用して、ファームウェアバージョン 13 以上で動作している PLC から一般プロセッサ情報を取得できます。

タグ名	説明
#MODE	PLC の現在のキースイッチモードの説明。とりうる文字列値は Program、Run、Remote Program、Remote Run、Remote Debug です。サポートされるデータ型は String です。
#PLCTYPE	PLC の EDS ファイルで指定されている "ProdType" 属性に対応する整数値。サポートされるデータ型は String 以外のすべてです。
#REVISION	"<メジャー>.<マイナー>" として表示されるファームウェアバージョン。サポートされるデータ型は String です。
#PROCESSORNAME	PLC の EDS ファイルで指定されている "ProdName" 属性に対応するプロセッサ名。サポートされるデータ型は String です。
#STATUS	PLC のステータスを示します。とりうる値は OK (1) と Faulted (0) です。サポートされるデータ型は Date 以外のすべてです。
#PRODUCTCODE	PLC の EDS ファイルで指定されている "ProdCode" 属性に対応する整数値。サポートされるデータ型は String 以外のすべてです。
#VENDORID	PLC の EDS ファイルで指定されている "VendCode" 属性に対応する整数値。サポートされるデータ型は String 以外のすべてです。

アトミックデータ型のアドレス指定

アドレスフォーマットが使用可能な Logix データ型に推奨される使用法とアドレス指定の方法を以下に示します。補足するために例も示されています。「**詳細**」をクリックすると、そのアトミックデータ型の詳細なアドレス指定の方法が表示されます。

● **ヒント**: アトミックデータ型で複数のサーバーデータタイプがサポートされている場合、**自動タグデータベース生成** で使用されるデフォルトは**太字**で示されています。

● 注記: 空のセルは、必ずしもサポートされていないことを示すものではありません。

アトミックデータ型	標準	配列要素	オフセットがある/ない配列	ビット	文字列
BOOL					
クライアント/サーバーデータ型 詳細	Boolean	Boolean (BOOL 型 1 次元配列)	Boolean 配列 (BOOL 型 1 次元配列)		BOOL の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	BOOLTAG	BOOLARR[0]	BOOLARR[0]{32}		
SINT					
クライアント/サーバーデータ型 詳細	Byte、Char	Byte、Char	Byte 配列、Char 配列 (SINT 型 1/2/3 次元配列)	Boolean (SINT 内のビット)	文字列 (SINT 型 1/2/3 次元配列) SINT の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	SINTTAG	SINTARR[0]	SINTARR[0]{4}	SINTTAG.0	SINTARR/4
INT					
クライアント/サーバーデータ型 詳細	Word、Short	Word、Short	Word 配列、Short 配列 (INT 1/2/3 次元配列)	Boolean (INT 内のビット)	INT の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	INTTAG	INTARR[0]	INTARR[0]{4}	INTTAG.0	
DINT					
クライアント/サーバーデータ型 詳細	DWord、Long	DWord、Long	DWord 配列、Long 配列	Boolean (DINT 内のビット)	DINT の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	DINTTAG	DINTARR[0]	DINTARR[0]{4}	DINTTAG.0	
LINT					
クライアント/サーバーデータ型 詳細	Double、Date	Double、Date	Double 配列		LINT の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	LINTTAG	LINTARR[0]	LINTARR[0]{4}		
REAL					
クライアント/サーバーデータ型 詳細	Float	Float	Float 配列		REAL の詳細なアドレス指定 を参照してください。
クライアント/サーバータグの例	REALTAG	REALARR[0]	REALARR[0]{4}		
USINT					
クライアント/サーバーデータ型 詳細	Byte	Byte	Byte Array	Boolean (USINT 内のビット)	USINT の詳細なアドレス指定 を参照してください。
クライアント/サーバーデータ型	USINTTAG	USINTTARR[0]	USINTTARR[0]{4}	USINTTAG.0	

アトミックデータ型	標準	配列要素	オフセットがある/ない配列	ビット	文字列
UINT					
クライアント/サーバーデータ型 詳細	Word、BCD	Word、BCD	Word 配列、 BCD 配列	Boolean (UINT 内のビット)	UINT の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	UINTTAG	UINTARR[0]	UINTARR[0]{4}	UINTTAG.0	
UDINT					
クライアント/サーバーデータ型 詳細	DWord、 LBCD	DWord、 LBCD	DWord 配列、 LBCD 配列	Boolean	UDINT の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	UDINTTAG	UDINTARR[0]	UDINTARR[0] {4}	UDINTAG.0	
ULINT					
クライアント/サーバーデータ型 詳細	Double	Double	Double 配列		ULINT の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	ULINTTAG	ULINTARR[0]	ULINTARR[0] {4}		
LREAL					
クライアント/サーバーデータ型 詳細	Double	Double	Double 配列		LREAL の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	LREALTAG	LREALARR[0]	LREALARR[0] {4}		
TIME32					
クライアント/サーバーデータ型 詳細	String、Long	String、Long	Long 配列		TIME32 の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	TIME32TAG	TIME32ARR [0]	TIME32ARR[0] {4}		
TIME					
クライアント/サーバーデータ型 詳細	String、 LLong	String、LLong	LLong 配列		TIME の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	TIMETAG	TIMEARR[0]	TIMEARR[0]{4}		
LTIME					
クライアント/サーバーデータ型 詳細	String、 LLong	String、LLong	LLong 配列		LTIME の詳細なアドレス指定を参照してください。
クライアント/サーバータグの例	LTIMETAG	LTIMEARR[0]	LTIMEARR[0] {4}		

構造体データ型のアドレス指定

構造体レベルではアトミック構造体メンバーのみをアドレス指定できます。詳細については、以下の例を参照してください。

Logix タグ

MyTimer @ TIMER

クライアント/サーバータグ

1. 無効

TimerTag のアドレス = MyTimer

TimerTag のデータ型 = ??

2. 有効

TimerTag のアドレス = MyTimer.ACC

TimerTag のデータ型 = DWord

STRING データ型のアドレス指定

STRING は定義済みの Logix データ型であり、その構造体には DATA と LEN の 2 つのメンバーが含まれています。DATA は SINT の配列であり、STRING の文字が格納されます。LEN は DINT であり、クライアントに表示される DATA 内の文字数を表します。

LEN と DATA はアトムメンバーなので、これらはクライアント/サーバーから別個に参照される必要があります。構文は以下のとおりです。

説明	構文	例
STRING の値	DATA/<STRING の最大長>	MYSTRING.DATA/82
STRING の実際の長さ	LEN	MYSTRING.LEN

読み取り

DATA から読み取られた STRING は以下で打ち切られます。

- a. 出現した 1 つ目の Null 終端。
- b. a) が先に出現しない場合、LEN 内の値。
- c. a) または b) が先に出現しない場合、<STRING の最大長>。

例

PLC で MYSTRING.DATA には "Hello World" が格納されていますが、LEN が手動で 5 に設定されています。MYSTRING.DATA/82 を読み取ると "Hello" と表示されます。LEN を 20 に設定した場合、MYSTRING.DATA/82 では "Hello World" と表示されます。

書き込み

STRING の値が DATA に書き込まれると、書き込まれた DATA の長さも LEN に書き込まれます。なんらかの理由によって LEN への書き込みが失敗した場合、(コントローラへの DATA の書き込みが成功したにもかかわらず) DATA への書き込み操作も失敗したものと見なされます。

● **注記:** この動作は STRING 型の Logix タグおよび STRING から派生したカスタムタグ専用設計されました。独自の STRING を UDT に実装するユーザーは以下の点に注意してください。

- STRING として参照される DATA メンバーと DINT として参照される LEN メンバーを持つ UDT が存在する場合、その UDT で LEN が DATA の長さであるかどうかに関係なく LEN への書き込みは成功します。LEN が DATA の長さでない場合、この可能性を排除するため、UDT を設計する際には注意が必要です。
- STRING として参照される DATA メンバーを持つが LEN メンバーがない UDT が存在する場合、LEN への書き込みは警告なしで失敗し、DATA への影響はありません。

例

MYSTRING.DATA/82 に値 "Hello World" が格納されています。MYSTRING.LEN に 11 が格納されています。値 "Alarm Triggered" が MYSTRING.DATA/82 に書き込まれた場合、MYSTRING.LEN には 15 が書き込まれます。MYSTRING.LEN への書き込みに失敗した場合、MYSTRING.LEN には以前の値 11 が格納され、MYSTRING.DATA/82 では最初の 11 文字 ("Alarm Trigg") が表示されます。MYSTRING.DATA/82 への書き込みに失敗した場合、いずれのタグも影響を受けません。

LEN で文字列データを終了

論理アドレス指定モードでは、STRING.DATA を読み取ると STRING.LEN がシンボリックモードで自動的に読み取られます。これは、「LEN で文字列データを終了」オプションを無効にすることでバイパスできます。詳細については、[Logix オプション](#)を参照してください。

Logix 配列データの順序

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 [dim1, dim2]

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 [dim1, dim2, dim3]

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]
```


Logix の詳細なアドレス指定

以下のアトミックデータ型では詳細なアドレス指定が可能です。以下のリストからリンクを選択すると、各データ型についての詳しい情報が表示されます。

[BOOL](#)
[SINT](#)
[INT](#)
[DINT](#)
[LINT](#)
[REAL](#)
[USINT](#)
[UINT](#)
[UDINT](#)
[ULINT](#)
[LREAL](#)
[TIME32](#)
[TIME](#)
[LTIME](#)

BOOL の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	Boolean Byte、Char Word、Short、BCD DWord、Long、LBCD Float*	なし
	Boolean	コントローラタグは 1 次元配列でなければなりません。
オフセットがない配列	Boolean 配列	<ol style="list-style-type: none"> 1. コントローラタグは 1 次元配列でなければなりません。 2. 要素の数は 8 の因数でなければなりません。
オフセットがない配列	Byte 配列、Char 配列 Word 配列、Short 配列、BCD 配列 DWord 配列、Long 配列、LBCD 配列 Float 配列*	サポートされていません。
オフセットがある配列	Boolean 配列	<ol style="list-style-type: none"> 1. コントローラタグは 1 次元配列でなければなりません。 2. オフセットは 32 ビット境界内に収める必要があります。 3. 要素の数は 8 の因数でなければなりません。
ビット	Boolean	<ol style="list-style-type: none"> 1. コントローラタグは 1 次元配列でなければなりません。 2. 範囲は 0 から 31 に制限されます。
文字列	文字列	サポートされていません。

*Float 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

例

ハイライトされている例は一般的な使用事例を示しています。

BOOL 型コントローラタグ - booltag = true

サーバータグアドレス	フォーマット	データ型	注記
booltag	標準	Boolean	値 = true
booltag	標準	Byte	値 = 1
booltag	標準	Word	値 = 1
booltag	標準	DWord	値 = 1
booltag	標準	Float	値 = 1.0
booltag [3]	配列要素	Boolean	無効: タグが配列ではありません
booltag [3]	配列要素	Word	無効: タグが配列ではありません
booltag {1}	オフセットがない配列	Word	無効: サポートされていません
booltag {1}	オフセットがない配列	Boolean	無効: サポートされていません
booltag [3] {32}	オフセットがある配列	Boolean	無効: タグが配列ではありません
booltag . 3	ビット	Boolean	無効: タグが配列ではありません
booltag / 1	String	文字列	無効: サポートされていません
booltag / 4	String	文字列	無効: サポートされていません

BOOL 配列コントローラタグ - bitarraytag = [0,1,0,1]

サーバータグアドレス	フォーマット	データ型	注記
bitarraytag	標準	Boolean	無効: タグが配列であってはなりません
bitarraytag	標準	Byte	無効: タグが配列であってはなりません
bitarraytag	標準	Word	無効: タグが配列であってはなりません
bitarraytag	標準	DWord	無効: タグが配列であってはなりません
bitarraytag	標準	Float	無効: タグが配列であってはなりません
bitarraytag [3]	配列要素	Boolean	値 = true
bitarraytag [3]	配列要素	Word	無効: データ型が不適切です
bitarraytag {3}	オフセットがない配列	Word	無効: タグが配列であってはなりません
bitarraytag {1}	オフセットがない配列	Word	無効: タグが配列であってはなりません
bitarraytag {1}	オフセットがない配列	Boolean	無効: 配列のサイズは 8 の因数でなければなりません
bitarraytag {32}	オフセットがない配列	Boolean	値 = [0,1,0,1,...]
bitarraytag [3] {32}	オフセットがある配列	Boolean	オフセットは 32 ビット境界で開始する必要があります
bitarraytag[0]{32}	オフセットがある配列	Boolean	値 = [0,1,0,1,...]
bitarraytag[32]{64}	オフセットがある配列	Boolean	値 = [...] 上記で示されていない値
bitarraytag . 3	ビット	Boolean	値 = true
bitarraytag / 1	String	文字列	無効: サポートされていません
bitarraytag / 4	String	文字列	無効: サポートされていません

SINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	Boolean*、Byte、Char、Word、Short、BCD、DWord、Long、LBCD、Float***	なし
配列要素	Byte、Char、Word、Short、BCD、DWord、Long、LBCD、Float***	コントローラタグは配列でなければなりません。

フォーマット	サポートされるデータ型	注記
<u>オフセットがない配列</u>	Boolean 配列	<ol style="list-style-type: none"> SINT 内のビットを配列形式にするにはこの事例に従います。 ● 注記: これは Boolean 表記の SINT の配列ではありません。 SINT 内のビットのみに適用されます。例: tag_1.0{8}。 ビットオフセットと配列サイズの和が 8 ビットを超えてはなりません。例: tag_1.1{8} は SINT を超えています、tag_1.0{8} は超えていません。 配列サイズは 8 の倍数でなければなりません。
<u>オフセットがない配列</u>	Byte 配列、Char 配列、Word 配列、Short 配列、BCD 配列**、DWord 配列、Long 配列、LBCD 配列**、Float 配列**、***	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	Byte 配列、Char 配列、Word 配列、Short 配列、BCD 配列**、DWord 配列、Long 配列、LBCD 配列**、Float 配列**、***	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<ol style="list-style-type: none"> 範囲は 0 から 7 に制限されます。 コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0。
<u>文字列</u>	String	<ol style="list-style-type: none"> 1 つの要素にアクセスする場合、コントローラタグが配列である必要はありません。 ● 注記: 文字列の値は SINT 値に相当する ASCII 文字です。例: SINT = 65 dec = "A"。 複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。文字列の値は文字列内のすべての SINT に相当する Null 終端 ASCII 文字です。文字列内の 1 文字 = 1 SINT。

*ゼロ以外の値は True にクランプされます。

**配列の各要素は SINT 配列内の要素に対応しています。配列はパックされません。

*** Float 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

例

ハイライトされている例は一般的な使用事例を示しています。

SINT コントローラタグ - sinttag = 122 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
sinttag	標準	Boolean	値 = true
sinttag	標準	Byte	値 = 122
sinttag	標準	Word	値 = 122
sinttag	標準	DWord	値 = 122
sinttag	標準	Float	値 = 122.0
sinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。さらに、

サーバータグアドレス	フォーマット	データ型	注記
			Boolean は無効です。
sinttag [3]	配列要素	Byte	無効: タグが配列ではありません。
sinttag {3}	オフセットがない配列	Byte	無効: タグが配列ではありません。
sinttag {1}	オフセットがない配列	Byte	値 = [122]
sinttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
sinttag [3] {1}	オフセットがある配列	Byte	無効: タグが配列ではありません。
sinttag . 3	ビット	Boolean	値 = true
sinttag . 0 {8}	オフセットがない配列	Boolean	値 = [0,1,0,1,1,1,1,0] 122 のビット値
sinttag / 1	文字列	String	値 = "z"
sinttag / 4	文字列	String	無効: タグが配列ではありません。

SINT 配列コントローラタグ - sintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	フォーマット	データ型	注記
sintarraytag	標準	Boolean	無効: タグが配列であってはなりません。
sintarraytag	標準	Byte	無効: タグが配列であってはなりません。
sintarraytag	標準	Word	無効: タグが配列であってはなりません。
sintarraytag	標準	DWord	無効: タグが配列であってはなりません。
sintarraytag	標準	Float	無効: タグが配列であってはなりません。
sintarraytag [3]	配列要素	Byte	無効: サーバータグで次元 2 のアドレスが欠落しています。
sintarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
sintarraytag [1,3]	配列要素	Byte	値 = 8
sintarraytag {10}	オフセットがない配列	Byte	値 = [83,73,78,84,5,6,7,8,9,10]
sintarraytag {2} {5}	オフセットがない配列	Word	値 = [83,73,78,84,5] [6,7,8,9,10]
sintarraytag {1}	オフセットがない配列	Byte	値 = 83
sintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
sintarraytag [1,3] {4}	オフセットがある配列	Byte	値 = [8,9,10,11]
sintarraytag . 3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
sintarraytag [1,3] . 3	ビット	Boolean	値 = 1
sintarraytag [1,3] . 0 {8}	オフセットがない配列	Boolean	値 = [0,0,0,1,0,0,0,0]
sintarraytag / 1	文字列	String	値 = "S"
sintarraytag / 4	文字列	String	値 = "SINT"

INT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
<u>標準</u>	Boolean*, Byte, Char**, Word, Short, BCD, DWord, Long, LBCD, Float****	なし
<u>配列要素</u>	Byte, Char**, Word, Short, BCD, DWord, Long, LBCD, Float****	コントローラタグは配列でなければなりません。
<u>オフセットがない配列</u>	Boolean 配列	1. INT 内のビットを配列形式にするにはこの事例に従います。

フォーマット	サポートされるデータ型	注記
		<ul style="list-style-type: none"> ● 注記: これは Boolean 表記の INT の配列ではありません。 2. INT 内のビットのみに適用されます。例: tag_1.0{16}。 3. ビットオフセットと配列サイズの和が 16 ビットを超えてはなりません。例: tag_1.1{16} は INT を超えていますが、tag_1.0{16} は超えていません。 4. 配列サイズは 8 の倍数でなければなりません。
<u>オフセットがない配列</u>	Byte 配列、Char 配列**、Word 配列、Short 配列、BCD 配列、DWord 配列、Long 配列、LBCD 配列***、Float 配列***、****	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	Byte 配列、Char 配列**、Word 配列、Short 配列、BCD 配列、DWord 配列、Long 配列、LBCD 配列***、Float 配列***、****	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<ol style="list-style-type: none"> 1. 範囲は 0 から 15 に制限されます。 2. コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0。
<u>文字列</u>	String	<ol style="list-style-type: none"> 1. 1 つの要素にアクセスする場合、コントローラタグが配列である必要はありません。 <ul style="list-style-type: none"> ● 注記: 文字列の値は INT 値 (255 にクランプ) に相当する ASCII 文字です。例: INT = 65 dec = "A"。 2. 複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。文字列の値は文字列内のすべての INT (255 にクランプ) に相当する Null 終端 ASCII 文字です。 <p>文字列内の 1 文字 = 1 INT、255 にクランプ</p> <p>INT 文字列はパッキングされません。効率を上げるには、代わりに SINT 型文字列または STRING 型構造体を使用します。</p>

*ゼロ以外の値は True にクランプされます。

**255 を超える値は 255 にクランプされます。

***配列の各要素は INT 配列内の要素に対応しています。配列はパッキングされません。

****Float 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

例

ハイライトされている例は一般的な使用事例を示しています。

INT コントローラタグ - inttag = 65534 (10 進)

サーバータグアドレス	クラス	データ型	注記
inttag	標準	Boolean	値 = true
inttag	標準	Byte	値 = 255
inttag	標準	Word	値 = 65534
inttag	標準	DWord	値 = 65534
inttag	標準	Float	値 = 65534.0
inttag [3]	配列要素	Boolean	無効: タグが配列ではありません。Boolean は無効です。
inttag [3]	配列要素	Word	無効: タグが配列ではありません。
inttag {3}	オフセットがない配列	Word	無効: タグが配列ではありません。
inttag {1}	オフセットがない配列	Word	値 = [65534]
inttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
inttag [3] {1}	オフセットがある配列	Word	無効: タグが配列ではありません。
inttag . 3	ビット	Boolean	値 = true
inttag . 0 {16}	オフセットがない配列	Boolean	値 = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] 65534 のビット値
inttag / 1	文字列	String	値 = 印刷不可文字 = 255 (10 進)。
inttag / 4	文字列	String	無効: タグが配列ではありません。

INT 配列コントローラタグ - intarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	クラス	データ型	注記
intarraytag	標準	Boolean	無効: タグが配列であってはなりません。
intarraytag	標準	Byte	無効: タグが配列であってはなりません。
intarraytag	標準	Word	無効: タグが配列であってはなりません。
intarraytag	標準	DWord	無効: タグが配列であってはなりません。
intarraytag	標準	Float	無効: タグが配列であってはなりません。
intarraytag [3]	配列要素	Word	無効: サーバータグで次元 2 のアドレスが欠落しています。
intarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
intarraytag [1,3]	配列要素	Word	値 = 259
intarraytag {10}	オフセットがない配列	Byte	値 = [73,78,84,255,255,255,255,255,9,10]
intarraytag {2} {5}	オフセットがない配列	Word	値 = [73,78,84,255,256] [257,258,259,9,10]
intarraytag {1}	オフセットがない配列	Word	値 = 73
intarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
intarraytag [1,3] {4}	オフセットがある配列	Word	値 = [259,9,10,11]
intarraytag . 3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
intarraytag [1,3] . 3	ビット	Boolean	値 = 0
intarraytag [1,3] . 0 {16}	オフセットがない配列	Boolean	値 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 259 のビット値
intarraytag / 1	文字列	String	値 = ""
intarraytag / 3	文字列	String	値 = "INT"

DINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	Boolean*, Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float ****	なし
配列要素	Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float ****	コントローラタグは配列でなければなりません。
オフセットがない配列	Boolean 配列	<ol style="list-style-type: none"> DINT 内のビットを配列形式にするにはこの事例に従います。 ● 注記: これは Boolean 表記の DINT の配列ではありません。 DINT 内のビットのみに適用されます。例: tag_1.0{32}。 ビットオフセットと配列サイズの和が 32 ビットを超えてはなりません。例: tag_1.1{32} は DINT を超えています。tag_1.0{32} は超えていません。 配列サイズは 8 の倍数でなければなりません。
オフセットがない配列	Byte 配列、Char 配列**, Word 配列、Short 配列、BCD 配列***、DWord 配列、Long 配列、LBCD 配列、Float 配列 ****	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	Byte 配列、Char 配列**, Word 配列、Short 配列、BCD 配列***、DWord 配列、Long 配列、LBCD 配列、Float 配列 ****	コントローラタグは配列でなければなりません。
ビット	Boolean	<ol style="list-style-type: none"> 範囲は 0 から 31 に制限されます。 コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0。
文字列	String	<ol style="list-style-type: none"> 1 つの要素にアクセスする場合、コントローラタグが配列である必要はありません。 ● 注記: 文字列の値は DINT 値 (255 にクランプ) に相当する ASCII 文字です。例: SINT = 65dec = "A"。 複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。文字列の値は文字列内のすべての DINT (255 にクランプ) に相当する Null 終端 ASCII 文字です。文字列内の 1 文字 = 1 DINT、255 にクランプ ● 注記: DINT 文字列はパックされません。効率を上げるには、代わりに SINT 型文字列または STRING 型構造体を使用します。

*ゼロ以外の値は True にクランプされます。

**255 を超える値は 255 にクランプされます。

***65535 を超える値は 65535 にクランプされます。

****Float 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

例

ハイライトされている例は一般的な使用事例を示しています。

DINT 型コントローラタグ - dinttag = 70000 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
dinttag	標準	Boolean	値 = true
dinttag	標準	Byte	値 = 255
dinttag	標準	Word	値 = 65535
dinttag	標準	DWord	値 = 70000
dinttag	標準	Float	値 = 70000.0
dinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。Boolean は無効です。
dinttag [3]	配列要素	DWord	無効: タグが配列ではありません。
dinttag {3}	オフセットがない配列	DWord	無効: タグが配列ではありません。
dinttag {1}	オフセットがない配列	DWord	値 = [70000]
dinttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です
dinttag [3]{1}	オフセットがある配列	DWord	無効: タグが配列ではありません。
dinttag . 3	ビット	Boolean	値 = false
dinttag . 0 {32}	オフセットがない配列	Boolean	値 = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,...0] 70000 のビット値
dinttag / 1	文字列	String	値 = 印刷不可文字 = 255 (10 進)
dinttag / 4	文字列	String	無効: タグが配列ではありません。

DINT 配列コントローラタグ - dintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	フォーマット	データ型	注記
dintarraytag	標準	Boolean	無効: タグが配列であってはなりません。
dintarraytag	標準	Byte	無効: タグが配列であってはなりません。
dintarraytag	標準	Word	無効: タグが配列であってはなりません。
dintarraytag	標準	DWord	無効: タグが配列であってはなりません。
dintarraytag	標準	Float	無効: タグが配列であってはなりません。
dintarraytag [3]	配列要素	DWord	無効: サーバータグで次元 2 のアドレスが欠落しています。
dintarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
dintarraytag [1,3]	配列要素	DWord	値 = 259
dintarraytag {10}	オフセットがない配列	Byte	値 = [68,73,78,84,255,255,255,255,9,10]
dintarraytag {2}{5}	オフセットがない配列	DWord	値 = [68,73,78,84,256] [257,258,259,9,10]
dintarraytag {1}	オフセットがない配列	DWord	値 = 68
dintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
dintarraytag [1,3]{4}	オフセットがある配列	DWord	値 = [259,9,10,11]
dintarraytag . 3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
dintarraytag [1,3] . 3	ビット	Boolean	値 = 0
dintarraytag [1,3] . 0 {32}	オフセットがない配列	Boolean	値 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 259 のビット値
dintarraytag / 1	文字列	String	値 = "D"
dintarraytag / 3	文字列	String	値 = "DINT"

LINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	Double*、Date**	なし
配列要素	Double*、Date**	コントローラタグは配列でなければなりません。
オフセットがない配列	Double、配列*	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	Double、配列*	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません。
文字列	該当なし	サポートされていません。

*Double 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

**Date 型の値は現地時刻ではなく協定世界時刻 (UTC) です。

例

ハイライトされている例は一般的な使用事例を示しています。

LINT コントローラタグ - linttag = 2007-01-01T16:46:40.000 (日付) == 1.16767E+15 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
linttag	標準	Boolean	無効: Boolean はサポートされていません。
linttag	標準	Byte	無効: Byte はサポートされていません。
linttag	標準	Word	無効: Word はサポートされていません。
linttag	標準	Double	値 = 1.16767E+15
linttag	標準	Date	値 = 2007-01-01T16:46:40.000*
linttag [3]	配列要素	Boolean	無効: タグが配列ではありません。 Boolean は無効です。
linttag [3]	配列要素	Double	無効: タグが配列ではありません。
linttag {3}	オフセットがない配列	Double	無効: タグが配列ではありません。
linttag {1}	オフセットがない配列	Double	値 = [1.16767E+15]
linttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
lintag [3]{1}	オフセットがある配列	Double	無効: タグが配列ではありません。
linttag .3	ビット	Boolean	無効: サポートされていない構文/データ型です。
linttag / 1	String	String	無効: サポートされていない構文/データ型です。

*Date 型の値は現地時刻ではなく協定世界時刻 (UTC) です。

LINT 配列コントローラタグ -

lintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 9.46746E+14]

1.16767E+15 == 2007-01-01T16:46:40.000 (日付)

9.4666E+14 == 1999-12-31T17:06:40.000

9.46746E+14 == 2000-01-1T17:00:00.000

0 == 1970-01-01T00:00:00.000

サーバータグアドレス	フォーマット	データ型	注記
lintarraytag	標準	Boolean	無効: Boolean はサポートされていません。
lintarraytag	標準	Byte	無効: Byte はサポートされていません。
lintarraytag	標準	Word	無効: Word はサポートされていません。
lintarraytag	標準	Double	無効: タグが配列であってはなりません。
lintarraytag	標準	Date	無効: タグが配列であってはなりません。

サーバータグアドレス	フォーマット	データ型	注記
lintarraytag [1]	配列要素	Double	無効: サーバータグで次元 2 のアドレスが欠落しています。
lintarraytag [1,1]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
lintarraytag [1,1]	配列要素	Double	値 = 9.46746E+14
lintarraytag [1,1]	配列要素	Date	値 = 2000-01-01T17:00:00.000*
lintarraytag {4}	オフセットがない配列	Double	値 = [0, 1.16767E+15, 9.4666E+14, 9.46746E+14]
lintarraytag {2} {2}	オフセットがない配列	Double	値 = [0, 1.16767E+15][9.4666E+14, 9.46746E+14]
lintarraytag {4}	オフセットがない配列	Date	無効: Date 配列はサポートされていません。
lintarraytag {1}	オフセットがない配列	Double	値 = 0
lintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
lintarraytag [0,1] {2}	オフセットがある配列	Double	値 = [1.16767E+15, 9.4666E+14]
lintarraytag .3	ビット	Boolean	無効: サポートされていない構文/データ型です。
lintarraytag / 1	String	String	無効: サポートされていない構文/データ型です。

*Date 型の値は現地時刻ではなく協定世界時刻 (UTC) です。

REAL の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
<u>標準</u>	Boolean*, Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float****	なし
<u>配列要素</u>	Byte, Char**, Word, Short, BCD***, DWord, Long, LBCD, Float****	コントローラタグは配列でなければなりません。
<u>オフセットがない配列</u>	Boolean 配列	<ol style="list-style-type: none"> REAL 内のビットを配列形式にするにはこの事例に従います。 ●注記: これは Boolean 表記の REAL の配列ではありません。 REAL 内のビットのみに適用されません。例: tag_1.0{32}。 ビットオフセットと配列サイズの和が 32 ビットを超えてはなりません。例: tag_1.1{32} は REAL を超えていますが、tag_1.0{32} は超えていません。 配列サイズは 8 の倍数でなければなりません。
<u>オフセットがない配列</u>	Byte 配列、Char 配列**, Word 配列、Short 配列、BCD 配列***、DWord 配列、Long 配列、LBCD 配列、Float 配列****	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	Byte 配列、Char 配列**, Word 配列、Short 配列、BCD 配列***、DWord 配列、Long 配列、LBCD 配列、Float 配列****	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<ol style="list-style-type: none"> 範囲は 0 から 31 に制限されます。 コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例:

フォーマット	サポートされるデータ型	注記
		tag_1 [2,2,3].0。 ●注記: Float はビットの参照を可能にするために DWord にキャストされます。
文字列	String	<ol style="list-style-type: none"> 1つの要素にアクセスする場合、コントローラタグが配列である必要はありません。 ●注記: 文字列の値は REAL 値 (255 にクランプ) に相当する ASCII 文字です。例: SINT = 65 dec = "A"。 複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。文字列の値は文字列内のすべての REAL (255 にクランプ) に相当する Null 終端 ASCII 文字です。文字列内の 1 文字 = 1 REAL、255 にクランプ。 ●注記: REAL 文字列はパックされません。効率を上げるには、代わりに SINT 型文字列または STRING 型構造体を使用します。

*ゼロ以外の値は True にクランプされます。

**255 を超える値は 255 にクランプされます。

***65535 を超える値は 65535 にクランプされます。

****Float 型の値は有効な IEEE 単精度浮動小数点数です。

例

ハイライトされている例は一般的な使用事例を示しています。

REAL コントローラタグ - realtag = 512.5 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
realtag	標準	Boolean	値 = true
realtag	標準	Byte	値 = 255
realtag	標準	Word	値 = 512
realtag	標準	DWord	値 = 512
realtag	標準	Float	値 = 512.5
realtag [3]	配列要素	Boolean	無効: タグが配列ではありません。さらに、Boolean は無効です。
realtag [3]	配列要素	DWord	無効: タグが配列ではありません。
realtag {3}	オフセットがない配列	DWord	無効: タグが配列ではありません。
realtag {1}	オフセットがない配列	Float	値 = [512.5]
realtag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
realtag [3]{1}	オフセットがある配列	Float	無効: タグが配列ではありません。
realtag .3	ビット	Boolean	値 = true
realtag .0 {32}	オフセットがない配列	Boolean	値 = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,...0] 512 のビット値
realtag / 1	文字列	String	値 = 印刷不可文字 = 255 (10 進)
realtag / 4	文字列	String	無効: タグが配列ではありません。

REAL 配列コントローラタグ - realarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

サーバータグアドレス	フォーマット	データ型	注記
realarraytag	標準	Boolean	無効: タグが配列であってはなりません。
realarraytag	標準	Byte	無効: タグが配列であってはなりません。
realarraytag	標準	Word	無効: タグが配列であってはなりません。
realarraytag	標準	DWord	無効: タグが配列であってはなりません。
realarraytag	標準	Float	無効: タグが配列であってはなりません。
realarraytag [3]	配列要素	Float	無効: サーバータグで次元 2 のアドレスが欠落しています。
realarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
realarraytag [1,3]	配列要素	Float	値 = 259.8
realarraytag {10}	オフセットがない配列	Byte	値 = [82,69,65,76,255,255,255,255,9,10]
realarraytag {2} {5}	オフセットがない配列	Float	値 = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
realarraytag {1}	オフセットがない配列	Float	値 = 82.1
realarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
realarraytag [1,3] {4}	オフセットがある配列	Float	値 = [259.8,9.0,10.0,11.0]
realarraytag . 3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
realarraytag [1,3] . 3	ビット	Boolean	値 = 0
realarraytag [1,3] . 0 {32}	オフセットがない配列	Boolean	値 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0] 259 のビット値
realarraytag / 1	文字列	String	値 = "R"
realarraytag / 3	文字列	String	値 = "REAL"

USINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
<u>標準</u>	Byte	なし
<u>配列要素</u>	Byte	コントローラタグは配列でなければなりません。
<u>オフセットがない配列</u>	Boolean 配列	<ol style="list-style-type: none"> USINT 内のビットを配列形式にするにはこの事例に従います。 ● 注記: これは Boolean 表記の USINT の配列ではありません。 USINT 内のビットのみに適用されません。例: tag_1.0{8}。 ビットオフセットと配列サイズの和が 8 ビットを超えてはなりません。例: tag_1.1{8} は USINT を超えていますが、tag_1.0{8} は超えていません。
<u>オフセットがない配列</u>	Byte Array	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	Byte Array	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<ol style="list-style-type: none"> 範囲は 0 から 7 に制限されます。 コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0。
<u>文字列</u>	該当なし	サポートしない

例

ハイライトされている例は一般的な使用事例を示しています。

USINT コントローラタグ - usinttag = 122 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
usinttag	標準	Boolean	値 = true
usinttag	標準	Byte	値 = 122
usinttag	標準	Word	無効: Word はサポートされていません。
usinttag	標準	DWord	無効: DWord はサポートされていません。
usinttag	標準	Float	無効: Float はサポートされていません
usinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。さらに、Boolean は無効です。
usinttag [3]	配列要素	Byte	無効: タグが配列ではありません。
usinttag {3}	オフセットがない配列	Byte	無効: タグが配列ではありません。
usinttag {1}	オフセットがない配列	Byte	値 = [122]
usinttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
usinttag [3] {1}	オフセットがある配列	Byte	無効: タグが配列ではありません。
usinttag .3	ビット	Boolean	値 = true
usinttag .0 {8}	オフセットがない配列	Boolean	値 = [0,1,0,1,1,1,1,0] 122 のビット値
usinttag / 1	String	String	値 = "z"
usinttag / 4	String	String	無効: タグが配列ではありません。

USINT 配列コントローラタグ - usintarraytag [4,4] = [[83,73,78,84],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	フォーマット	データ型	注記
usintarraytag	標準	Boolean	無効: タグが配列であってはなりません。
usintarraytag	標準	Byte	無効: タグが配列であってはなりません。
usintarraytag	標準	Word	無効: タグが配列であってはなりません。
usintarraytag	標準	DWord	無効: タグが配列であってはなりません。
usintarraytag	標準	Float	無効: タグが配列であってはなりません。
usintarraytag [3]	配列要素	Byte	無効: サーバータグで次元 2 のアドレスが欠落しています。
usintarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
usintarraytag [1,3]	配列要素	Byte	値 = 8
usintarraytag {10}	オフセットがない配列	Byte	値 = [83,73,78,84,5,6,7,8,9,10]
usintarraytag {2} {5}	オフセットがない配列	Word	値 = [83,73,78,84,5] [6,7,8,9,10]
usintarraytag {1}	オフセットがない配列	Byte	値 = 83
usintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
usintarraytag [1,3] {4}	オフセットがある配列	Byte	値 = [8,9,10,11]
usintarraytag .3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
usintarraytag [1,3] .3	ビット	Boolean	値 = 1
usintarraytag [1,3] .0 {8}	オフセットがない配列	Boolean	値 = [0,0,0,1,0,0,0,0]
usintarraytag / 1	String	String	無効: サポートされていない構文/データ型で

サーバータグアドレス	フォーマット	データ型	注記
			す。
usintarraytag / 4	String	String	無効: サポートされていない構文/データ型です。

UINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
<u>標準</u>	Word、BCD	なし
<u>配列要素</u>	Word、BCD	コントローラタグは配列でなければなりません。
<u>オフセットがない配列</u>	Boolean 配列	<ol style="list-style-type: none"> UINT 内のビットを配列形式にするにはこの事例に従います。 ● 注記: これは Boolean 表記の UINT の配列ではありません。 UINT 内のビットのみに適用されます。 例: tag_1.0{16}。 ビットオフセットと配列サイズの和が 8 ビットを超えてはなりません。例: tag_1.1{16} は UINT を超えています、tag_1.0{16} は超えていません。 配列サイズは 8 の倍数でなければなりません。
<u>オフセットがない配列</u>	Word 配列、BCD 配列	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	Word 配列、BCD 配列	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<ol style="list-style-type: none"> 範囲は 0 から 15 に制限されます。 コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0。
<u>文字列</u>	該当なし	サポートしない

例

ハイライトされている例は一般的な使用事例を示しています。

UINT コントローラタグ - uinttag = 65534 (10 進)

サーバータグアドレス	クラス	データ型	注記
uinttag	標準	Boolean	無効: Boolean はサポートされていません。
uinttag	標準	Byte	無効: Byte はサポートされていません。
uinttag	標準	Word	値 = 65534
uinttag	標準	DWord	無効: DWord はサポートされていません。
uinttag	標準	Float	無効: Float はサポートされていません。
uinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。Boolean は無効です。
uinttag [3]	配列要素	Byte	無効: タグが配列ではありません。
uinttag {3}	オフセットがない配列	Byte	無効: タグが配列ではありません。
uinttag {1}	オフセットがない配列	Byte	値 = [65534]

サーバータグアドレス	クラス	データ型	注記
uinttag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
uinttag [3] {1}	オフセットがある配列	Word	無効: タグが配列ではありません。
uinttag .3	ビット	Boolean	値 = true
uinttag .0 {16}	オフセットがない配列	Boolean	値 = [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] 65534 のビット値
uinttag / 1	String	String	無効: サポートされていない構文/データ型です。
uinttag / 4	String	String	無効: サポートされていない構文/データ型です。

UINT 配列コントローラタグ - uintarraytag [4,4] = [[73,78,84,255],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	フォーマット	データ型	注記
uintarraytag	標準	Boolean	無効: タグが配列であってはなりません。
uintarraytag	標準	Byte	無効: タグが配列であってはなりません。
uintarraytag	標準	Word	無効: タグが配列であってはなりません。
uintarraytag	標準	DWord	無効: タグが配列であってはなりません。
uintarraytag	標準	Float	無効: タグが配列であってはなりません。
uintarraytag [3]	配列要素	Word	無効: サーバータグで次元 2 のアドレスが欠落しています。
uintarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
uintarraytag [1,3]	配列要素	Word	値 = 259
uintarraytag {10}	オフセットがない配列	BCD	値 = [49,54,54,165,100,101,102,103,9,10]
uintarraytag {2} {5}	オフセットがない配列	Word	値 = [73,78,84,255,256] [257,258,259,9,10]
uintarraytag {1}	オフセットがない配列	Word	値 = 73
uintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
uintarraytag [1,3] {4}	オフセットがある配列	Byte	値 = [259,9,10,11]
uintarraytag .3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
uintarraytag [1,3] .3	ビット	Boolean	値 = 0
uintarraytag [1,3] .0 {16}	オフセットがない配列	Boolean	値 = [0,0,0,1,0,0,0,0] 値 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0] 259 のビット値
uintarraytag / 1	String	String	無効: サポートされていない構文/データ型です。
uintarraytag / 3	String	String	無効: サポートされていない構文/データ型です。

UDINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	DWord、LBCD	なし
配列要素	DWord、LBCD	コントローラタグは配列でなければなりません。
オフセットがない配列	Boolean 配列	1. UDINT 内のビットを配列形式にするにはこの事例に従います。 ● 注記: これは Boolean 表記の

フォーマット	サポートされるデータ型	注記
		<p>UDINT の配列ではありません。</p> <p>2. UDINT 内のビットのみに適用されます。例: tag_1.0{32}。</p> <p>3. ビットオフセットと配列サイズの和が 32 ビットを超えてはなりません。例: tag_1.1{32} は UINT を超えていますが、tag_1.0{32} は超えていません。</p> <p>4. 配列サイズは 8 の倍数でなければなりません。</p>
<u>オフセットがない配列</u>	DWord 配列、LBCD 配列	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
<u>オフセットがある配列</u>	DWord 配列、LBCD 配列	コントローラタグは配列でなければなりません。
<u>ビット</u>	Boolean	<p>1. 範囲は 0 から 31 に制限されます。</p> <p>2. コントローラタグが配列である場合、ビットクラス参照の先頭に配列要素クラス参照を付ける必要があります。例: tag_1 [2,2,3].0</p>
<u>文字列</u>	該当なし	サポートされていません

例

ハイライトされている例は一般的な使用事例を示しています。

UDINT コントローラタグ - udinttag = 70000 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
udinttag	標準	Boolean	無効: Boolean はサポートされていません
udinttag	標準	Byte	無効: Byte はサポートされていません
udinttag	標準	Word	無効: Word はサポートされていません
udinttag	標準	DWord	値 = 70000
udinttag	標準	LBCD	値 = 11170
udinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。Boolean は無効です。
udinttag [3]	配列要素	DWord	無効: タグが配列ではありません
udinttag {3}	オフセットがない配列	DWord	無効: タグが配列ではありません
udinttag {1}	オフセットがない配列	DWord	値 = [70000]
udinttag {1}	オフセットがない配列	Boolean	無効: Boolean 配列はサポートされていません
udinttag [3]{1}	オフセットがある配列	DWord	無効: タグが配列ではありません
udinttag .3	ビット	Boolean	値 = False
udinttag .0 {32}	オフセットがない配列	Boolean	値 = [0,0,0,0,1,1,1,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,...0] 70000 のビット値
udinttag / 1	String	String	無効: サポートされていない構文/データ型です
udinttag / 4	String	String	無効: サポートされていない構文/データ型です

UDINT 配列コントローラタグ - udintarraytag [4,4] = [[68,73,78,84],[256,257,258,259],[9,10,11,12],[13,14,15,16]]

サーバータグアドレス	フォーマット	データ型	注記
udintarraytag	標準	Boolean	無効: Boolean はサポートされていません
udintarraytag	標準	Byte	無効: Byte はサポートされていません
udintarraytag	標準	Word	無効: Word はサポートされていません
udintarraytag	標準	DWord	無効: タグが配列であってはなりません
udintarraytag	標準	Float	無効: Float はサポートされていません
udintarraytag [3]	配列要素	DWord	無効: サーバータグで次元 2 のアドレスが欠落しています。
udintarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません
udintarraytag [1,3]	配列要素	DWord	値 = 259
udintarraytag {10}	オフセットがない配列	LCBD	値 = [44,49,54,54,100,101,102,103,9,10]
udintarraytag {2}{5}	オフセットがない配列	DWord	値 = [68,73,78,84,256] [257,258,259,9,10]
udintarraytag {1}	オフセットがない配列	DWord	値 = 68
udintarraytag {1}	オフセットがない配列	Boolean	無効: データ型が不適切です。
udintarraytag [1,3]{4}	オフセットがある配列	DWord	値 = [259,9,10,11]
udintarraytag .3	ビット	Boolean	無効: タグはアトミックの場所を参照する必要があります。
udintarraytag [1,3].3	ビット	Boolean	値 = False
udintarraytag [1,3].0 {32}	オフセットがない配列	Boolean	値 = [1,1,0,0,0,0,0,0,1,0,0,0,0,0,0]
udintarraytag / 1	String	String	無効: サポートされていない構文/データ型です
udintarraytag / 3	String	String	無効: サポートされていない構文/データ型です

ULINT の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
標準	Double*	なし
配列要素	Double*	コントローラタグは配列でなければなりません。
オフセットがない配列	Double、配列*	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	Double、配列*	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません
文字列	該当なし	サポートされていません

*Double 型の値は Float 形式のコントローラタグの額面と等しくなります (非 IEEE 浮動小数点数)。

例

ハイライトされている例は一般的な使用事例を示しています。

ULINT コントローラタグ - ulinttag = 1.8446744073709560e+19 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
ulinttag	標準	Boolean	無効: Boolean はサポートされていません
ulinttag	標準	Byte	無効: Byte はサポートされていません
ulinttag	標準	Word	無効: Word はサポートされていません
ulinttag	標準	Double	値 = 1.8446744073709560e+19
ulinttag [3]	配列要素	Boolean	無効: タグが配列ではありません。 Boolean は

サーバータグアドレス	フォーマット	データ型	注記
			無効です。
ulinttag [3]	配列要素	Double	無効: タグが配列ではありません
ulinttag {3}	オフセットがない配列	Double	無効: タグが配列ではありません
ulinttag {1}	オフセットがない配列	Double	値 = [1.8446744073709560e+19]
ulinttag {1}	オフセットがない配列	Boolean	無効: Boolean 配列はサポートされていません
ulinttag [3]{1}	オフセットがある配列	Double	無効: タグが配列ではありません
ulinttag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
ulinttag / 1	String	String	無効: サポートされていない構文/データ型です

ULINT 配列コントローラタグ -

ulintarraytag [2,2] = [0, 1.16767E+15],[9.4666E+14, 1.8446744073709560e+19]

サーバータグアドレス	フォーマット	データ型	注記
ulintarraytag	標準	Boolean	無効: Boolean はサポートされていません
ulintarraytag	標準	Byte	無効: Byte はサポートされていません
ulintarraytag	標準	Word	無効: Word はサポートされていません
ulintarraytag	標準	Double	無効: タグが配列であってはなりません
ulintarraytag	標準	Date	無効: Date はサポートされていません
ulintarraytag [1]	配列要素	Double	無効: サーバータグで次元 2 のアドレスが欠落しています。
ulintarraytag [1,1]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
ulintarraytag [1,1]	配列要素	Double	値 = 1.8446744073709560e+19
ulintarraytag {4}	オフセットがない配列	Double	値 = [0, 1.16767E+15, 9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {2}{2}	オフセットがない配列	Double	値 = [0, 1.16767E+15][9.4666E+14, 1.8446744073709560e+19]
ulintarraytag {4}	オフセットがない配列	Date	無効: Date 配列はサポートされていません
ulintarraytag {1}	オフセットがない配列	Double	値 = 0
ulintarraytag {1}	オフセットがない配列	Boolean	無効: Boolean 配列はサポートされていません
ulintarraytag [0,1]{2}	オフセットがある配列	Double	値 = [1.16767E+15, 9.4666E+14]
ulintarraytag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
ulintarraytag / 1	String	String	無効: サポートされていない構文/データ型です

LREAL の詳細なアドレス指定

フォーマット	サポートされるデータ型	注記
<u>標準</u>	Double	なし
<u>配列要素</u>	Double	コントローラタグは配列でなければなりません。
<u>オフセットがない配列</u>	Double 配列	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。

フォーマット	サポートされるデータ型	注記
オフセットがある配列	Double 配列	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません
文字列	該当なし	サポートされていません

例

ハイライトされている例は一般的な使用事例を示しています。

LREAL コントローラタグ - lrealtag = 1.7976931348623157E+308 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
lrealtag	標準	Boolean	無効: Boolean はサポートされていません
lrealtag	標準	Byte	無効: Byte はサポートされていません
lrealtag	標準	Word	無効: Word はサポートされていません
lrealtag	標準	Double	値 = 1.7976931348623157E+308
lrealtag [3]	配列要素	Boolean	無効: タグが配列ではありません。さらに、Boolean は無効です。
lrealtag {1}	オフセットがない配列	Double	値 = [1.7976931348623157E+308]
lrealtag {1}	オフセットがない配列	Boolean	無効: Boolean 配列はサポートされていません
lrealtag [3]{1}	オフセットがある配列	Double	無効: タグが配列ではありません
lrealtag . 3	ビット	Boolean	無効: サポートされていない構文/データ型です
lrealtag . 0 {32}	オフセットがない配列	Boolean	無効: サポートされていない構文/データ型です
lrealtag / 1	文字列	String	無効: サポートされていない構文/データ型です
lrealtag / 4	文字列	String	無効: サポートされていない構文/データ型です

LREAL 配列コントローラタグ - lrealarraytag [4,4] = [[82.1,69.2,65.3,76.4],[256.5,257.6,258.7,259.8],[9.0,10.0,11.0,12.0],[13.0,14.0,15.0,16.0]]

サーバータグアドレス	フォーマット	データ型	注記
lrealarraytag	標準	Boolean	無効: タグが配列であってはなりません。
lrealarraytag	標準	Byte	無効: タグが配列であってはなりません。
lrealarraytag	標準	Word	無効: タグが配列であってはなりません。
lrealarraytag	標準	Double	無効: タグが配列であってはなりません。
lrealarraytag [3]	配列要素	Double	無効: サーバータグで次元 2 のアドレスが欠落しています。
lrealarraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
lrealarraytag [1,1]	配列要素	Double	値 = 257.6
lrealarraytag {2} {5}	オフセットがない配列	Double	値 = [82.1,69.2,65.3,76.4,256.5] [257.6,258.7,259.8,9,10]
lrealarraytag {1}	オフセットがない配列	Double	値 = 82.1
lrealarraytag {1}	オフセットがない配列	Boolean	無効: Boolean 配列はサポートされていません
lrealarraytag [1,3]{4}	オフセットがある配列	Double	値 = [259.8,9.0,10.0,11.0]
lrealarraytag . 3	ビット	Boolean	無効: サポートされていない構文/データ型で

サーバータグアドレス	フォーマット	データ型	注記
			す。
lrealarraytag / 1	文字列	String	無効: サポートされていない構文/データ型です。

詳細なアドレス指定: TIME32

フォーマット	サポートされるデータ型	注記
標準	String、Long	なし
配列要素	String、Long	コントローラタグは配列でなければなりません。
オフセットがない配列	Long Array	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	Long Array	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません
文字列	該当なし	長さ指定子を持つ文字列はサポートされていません。

● **ヒント:** 太字のデータ型は、デフォルトのデータ型を表します。

例

ハイライトされている例は一般的な使用事例を示しています。

TIME32 コントローラタグ - time32tag = -2147483647 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
time32tag	標準	String	値 = T32#-35m_47s_483ms_647us
time32tag	標準	Long	-2147483647
time32tag	標準	DWord	無効: DWord はサポートされていません
time32tag	標準	Boolean	無効: Boolean はサポートされていません
time32tag [3]	配列要素	String	無効: タグが配列ではありません
time32tag {1}	オフセットがない配列	Long Array	値 = [-2147483647]
time32tag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
time32tag [3] {1}	オフセットがある配列	Long Array	無効: タグが配列ではありません
time32tag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
time32tag .0 {32}	オフセットがない配列	Boolean 配列	無効: サポートされていない構文/データ型です
time32tag / 1	String	String	無効: String 長さはサポートされていません
time32tag / 4	String	String	無効: String 長さはサポートされていません

TIME32 配列コントローラタグ - time32arraytag [4,4] = [[1,2,3,4],[500,600,700,800],[90000,100000,110000,120000],[13000000,14000000,15000000,16000000]] (10 進)

サーバータグアドレス	フォーマット	データ型	注記
time32arraytag	標準	String	T32#1us (最初の要素のみ読み取り)
time32arraytag	標準	Long	1 (最初の要素のみ読み取り)
time32arraytag	標準	Boolean	無効: タグが配列であってはなりません。Boolean はサポートされていません。

サーバータグアドレス	フォーマット	データ型	注記
time32arraytag [1,1]	配列要素	String	T32#600us
time32arraytag [2,2]	配列要素	Long	値 = 110000
time32arraytag [3]	配列要素	Long	無効: サーバータグで次元 2 のアドレスが欠落しています。
time32arraytag [1.3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
time32arraytag {2} {5}	オフセットがない配列	Long Array	値 = [1,2,3,4,500] [600,700,800,90000,100000]
time32arraytag {1}	オフセットがない配列	Long Array	値 = [1]
time32arraytag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
time32arraytag [1,2] {3}	オフセットがある配列	Long Array	値 = [700,800,90000]
time32arraytag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
time32arraytag / 1	String	String	無効: String 長さはサポートされていません

詳細なアドレス指定: TIME

フォーマット	サポートされるデータ型	注記
標準	String、LLong	なし
配列要素	String、LLong	コントローラタグは配列でなければなりません。
オフセットがない配列	LLong Array	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	LLong Array	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません
文字列	該当なし	長さ指定子を持つ文字列はサポートされていません。

● ヒント: 太字のデータ型は、デフォルトのデータ型を表します。

例

ハイライトされている例は一般的な使用事例を示しています。

TIME コントローラタグ - timetag = -2725199999999 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
timetag	標準	String	値 = T#-31d_12h_59m_59s_999ms_999us
timetag	標準	LLong	-2725199999999
timetag	標準	QWord	無効: QWord 型はサポートされていません
timetag	標準	Boolean	無効: Boolean はサポートされていません
timetag [3]	配列要素	String	無効: タグが配列ではありません
timetag {1}	オフセットがない配列	LLong Array	値 = [-2725199999999]
timetag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
timetag [3] {1}	オフセットがある配列	LLong Array	無効: タグが配列ではありません
timetag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
timetag .0 {32}	オフセットがない配列	Boolean 配列	無効: サポートされていない構文/データ型で

サーバータグアドレス	フォーマット	データ型	注記
			す
timetag / 1	String	String	無効: String 長さはサポートされていません
timetag / 4	String	String	無効: String 長さはサポートされていません

TIME 配列コントローラタグ - timearraytag [4,4] = [[1,2,3,4],[500,600,700,800],[90000,100000,110000,120000],[13000000,14000000,15000000,16000000]] (10 進)

サーバータグアドレス	フォーマット	データ型	注記
timearraytag	標準	String	T#1us (最初の要素のみ読み取り)
timearraytag	標準	LLong	1 (最初の要素のみ読み取り)
timearraytag	標準	Boolean	無効: タグが配列であってはなりません。 Boolean はサポートされていません。
timearraytag [1,1]	配列要素	String	T#600us
timearraytag [2,2]	配列要素	LLong	値 = 110000
timearraytag [3]	配列要素	LLong	無効: サーバータグで次元 2 のアドレスが欠落しています。
timearraytag [1.3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
timearraytag {2} {5}	オフセットがない配列	LLong Array	値 = [1,2,3,4,500] [600,700,800,90000,100000]
timearraytag {1}	オフセットがない配列	LLong Array	値 = [1]
timearraytag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
timearraytag [1,3] {4}	オフセットがある配列	LLong Array	値 = [800,90000,100000,110000]
timearraytag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
timearraytag / 1	String	String	無効: String 長さはサポートされていません

詳細なアドレス指定: LTIME

フォーマット	サポートされるデータ型	注記
標準	String、LLong	なし
配列要素	String、LLong	コントローラタグは配列でなければなりません。
オフセットがない配列	LLong 配列	複数の要素にアクセスする場合、コントローラタグは配列でなければなりません。
オフセットがある配列	LLong 配列	コントローラタグは配列でなければなりません。
ビット	該当なし	サポートされていません
文字列	該当なし	長さ指定子を持つ文字列はサポートされていません。

● ヒント: 太字のデータ型は、デフォルトのデータ型を表します。

例

ハイライトされている例は一般的な使用事例を示しています。

LTIME コントローラタグ - ltimetag = -272519999999999 (10 進)

サーバータグアドレス	フォーマット	データ型	注記
ltime tag	標準	String	値 = LT#-31d_12h_59m_59s_999ms_999us_999ns
ltime tag	標準	LLong	-2725199999999999
ltime tag	標準	QWord	無効: QWord 型はサポートされていません
ltime tag	標準	Boolean	無効: Boolean はサポートされていません
ltime tag [3]	配列要素	String	無効: タグが配列ではありません
ltime tag {1}	オフセットがない配列	LLong Array	値 = [-2725199999999999]
ltime tag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
ltime tag [3] {1}	オフセットがある配列	LLong Array	無効: タグが配列ではありません
ltime tag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
ltime tag .0 {32}	オフセットがない配列	Boolean 配列	無効: サポートされていない構文/データ型です
ltime tag / 1	String	String	無効: String 長さはサポートされていません
ltime tag / 4	String	String	無効: String 長さはサポートされていません

LTIME 配列コントローラタグ - ltimearraytag [4,4] = [[1,2,3,4],[500,600,700,800],[90000,100000,110000,120000],[13000000,14000000,15000000,16000000]] (10 進)

サーバータグアドレス	フォーマット	データ型	注記
ltimearraytag	標準	String	T#1us (最初の要素のみ読み取り)
ltimearraytag	標準	LLong	1 (最初の要素のみ読み取り)
ltimearraytag	標準	Boolean	無効: タグが配列であってはなりません。Boolean はサポートされていません。
ltimearraytag [1,1]	配列要素	String	T#600ns
ltimearraytag [2,2]	配列要素	LLong	値 = 110000
ltimearraytag [3]	配列要素	LLong	無効: サーバータグで次元 2 のアドレスが欠落しています。
ltimearraytag [1,3]	配列要素	Boolean	無効: 配列要素には Boolean を使用できません。
ltimearraytag {2} {5}	オフセットがない配列	LLong Array	値 = [1,2,3,4,500] [600,700,800,90000,100000]
ltimearraytag {1}	オフセットがない配列	LLong Array	値 = [1]
ltimearraytag {1}	オフセットがない配列	String Array	無効: String Array はサポートされていません
ltimearraytag [2,2] {4}	オフセットがある配列	LLong Array	値 = [110000,120000,13000000,14000000]
ltimearraytag .3	ビット	Boolean	無効: サポートされていない構文/データ型です
ltimearraytag / 1	String	String	無効: String 長さはサポートされていません

ファイル一覧

以下のリストからリンクを選択すると、各種デバイスモデルによってサポートされている各ファイルの情報が表示されます。

[出力ファイル](#)

[入力ファイル](#)

[ステータスファイル](#)

[バイナリファイル](#)

[タイマーファイル](#)

[カウンタファイル](#)

[制御ファイル](#)

[整数ファイル](#)

[Float ファイル](#)

[ASCII ファイル](#)

[文字列ファイル](#)

[BCD ファイル](#)

[Long ファイル](#)

[MicroLogix PID ファイル](#)

[PID ファイル](#)

[MicroLogix メッセージファイル](#)

[メッセージファイル](#)

[ブロック転送ファイル](#)

ファンクションファイル一覧

[高速カウンタファイル \(HSC\)](#)

[リアルタイムクロックファイル \(RTC\)](#)

[チャンネル0 通信ステータスファイル \(CS0\)](#)

[チャンネル1 通信ステータスファイル \(CS1\)](#)

[I/O モジュールステータスファイル \(IOS\)](#)

● デバイスモデルとそのサポートされているファイルの詳細については、[アドレスの説明](#)を参照してください。

出力ファイル

出力ファイル内のデータにアクセスするための構文は PLC モデルによって異なります。出力ファイルでは配列はサポートされていません。デフォルトのデータ型を太字で示しています。

PLC-5 での構文

構文	データ型	アクセス
O:<Word>	Short、 Word 、BCD	読み取り/書き込み
O:<Word>/<ビット>	Boolean	読み取り/書き込み
O/ビット	Boolean	読み取り/書き込み

● **注記:** PLC-5 モデルでの Word とビットのアドレス情報は 8 進で表記します。これはプログラミングソフトウェアの規則に従います。

MicroLogix での構文

構文	データ型	アクセス
O:<Word>	Short、 Word 、BCD	読み取り/書き込み
O:<Word>/<ビット>	Boolean	読み取り/書き込み
O/ビット	Boolean	読み取り/書き込み

MicroLogix モデルには、埋め込み I/O と拡張 I/O の 2 つのタイプの I/O があります (Micrologix 1000 を除く)。埋め込み I/O は CPU 基本ユニットに存在し、拡張 I/O は CPU 基本ユニットにプラグイン接続します。各 MicroLogix モデルの I/O 機能を以下の表に示します。

MicroLogix モデル	埋め込み I/O	拡張 I/O
1000	スロット 0	該当なし
1100	スロット 0	スロット 1-4
1200	スロット 0	スロット 1-6
1400	スロット 0	スロット 1-7
1500	スロット 0	スロット 1-16

MicroLogix I/O のアドレス構文は、スロットではなく、ゼロベースの Word オフセットを参照します。特定のスロットに対する Word オフセットをユーザーが計算する必要があります。これには各モジュールとそのサイズ (Word 数) についての知識が必要です。以下の表にはいくつかの使用可能なモジュールのサイズが示されていますが、MicroLogix のドキュメントとコントローラプロジェクトを参照してモジュールの正しい Word サイズを調べることをお勧めします。

MicroLogix 埋め込み I/O の Word サイズ

MicroLogix モデル	入力 Word 数	出力 Word 数
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix 拡張 I/O の Word サイズ

モジュール:	入力 Word 数	出力 Word 数
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1

モジュール:	入力 Word 数	出力 Word 数
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

計算

スロット x の出力 Word オフセット = スロット 0 からスロット (x-1) での出力 Word 数。

● 注記:

1. 拡張 I/O にオフセットする場合には埋め込み I/O を考慮する必要があります。
2. 出力 Word のオフセットの計算では入力 Word 数は考慮されません。

I/O の例

以下の場合

スロット 0 = MicroLogix 1500 LRP シリーズ C = 出力 Word 数は 4

スロット 1 = 1769-OF2 = 出力 Word 数は 2

スロット 2 = 1769-OW8 = 出力 Word 数は 1

スロット 3 = 1769-IA16 = 出力 Word 数は 0

スロット 4 = 1769-OF8V = 出力 Word 数は 9

スロット 4 (= 4 + 2 + 1 = 7 Word) のビット 5 = O:7/5

SLC 500 での構文

デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
O:<スロット>	Short、 Word 、BCD	読み取り専用
O:<スロット>.<Word>	Short、 Word 、BCD	読み取り専用
O:<スロット>/<ビット>	Boolean	読み取り専用
O:<スロット>.<Word>/<ビット>	Boolean	読み取り専用

範囲

PLC モデル	最小スロット	最大スロット	最大 Word
MicroLogix	該当なし	なし	2047
SLC 500 固定 I/O	該当なし	なし	1
SLC 500 モジュール I/O	1	30	*
PLC-5 シリーズ	該当なし	なし	277 (8 進)

*各 I/O モジュールで使用可能な入力/出力 Word の数については、[SLC 500 モジュール I/O 選択ガイド](#)を参照してください。

例

MicroLogix	説明
O:0	Word 0
O/2	ビット 2
O:0/5	ビット 5

SLC 500 固定 I/O	説明
O:0	Word 0
O:1	Word 1
O/16	ビット 16
O:1/0	ビット 0 Word 1 (O/16 と同じ)

PLC5*	説明
O:0	Word 0
O:37	Word 31 (8 進の 37 = 10 進の 31)
O/42	ビット 34 (8 進の 42 = 10 進の 34)
O:2/2	ビット 2 Word 2 (O/42 と同じ)

*アドレスは 8 進で示されています。

SLC 500 モジュール I/O	説明
O:1	Word 0 スロット 1
O:1.0	Word 0 スロット 1 (O:1 と同じ)
O:12	Word 0 スロット 12
O:12.2	Word 2 スロット 12
O:4.0/0	ビット 0 Word 0 スロット 4
O:4/0	ビット 0 スロット 4 (O:4.0/0 と同じ)
O:4.2/0	ビット 0 Word 2 スロット 4
O:4/32	ビット 32 スロット 4 (O:4.2/0 と同じ)

入力ファイル

入力ファイル内のデータにアクセスするための構文は PLC モデルによって異なります。入力ファイルでは配列はサポートされていません。デフォルトのデータ型を**太字**で示しています。

PLC-5 での構文

構文	データ型	アクセス
I:<Word>	Short、 Word 、BCD	読み取り/書き込み
I:<Word>/<ビット>	Boolean	読み取り/書き込み
I/ビット	Boolean	読み取り/書き込み

● **注意:** PLC-5 モデルでの Word とビットのアドレス情報は 8 進で表記します。これはプログラミングソフトウェアの規則に従います。

MicroLogix での構文

構文	データ型	アクセス
I:<Word>	Short、Word、BCD	読み取り/書き込み
I:<Word>/<ビット>	Boolean	読み取り/書き込み
I/ビット	Boolean	読み取り/書き込み

MicroLogix モデルには、埋め込み I/O と拡張 I/O の 2 つのタイプの I/O があります (Micrologix 1000 を除く)。埋め込み I/O は CPU 基本ユニットに存在し、拡張 I/O は CPU 基本ユニットにプラグイン接続します。各 MicroLogix モデルの I/O 機能を以下の表に示します。

MicroLogix モデル	埋め込み I/O	拡張 I/O
1000	スロット 0	該当なし
1100	スロット 0	スロット 1-4
1200	スロット 0	スロット 1-6
1400	スロット 0	スロット 1-7
1500	スロット 0	スロット 1-16

MicroLogix I/O のアドレス構文は、スロットではなく、ゼロベースの Word オフセットを参照します。特定のスロットに対する Word オフセットをユーザーが計算する必要があります。これには各モジュールとそのサイズ (Word 数) についての知識が必要です。以下の表にはいくつかの使用可能なモジュールのサイズが示されていますが、MicroLogix のドキュメントとコントローラプロジェクトを参照してモジュールの正しい Word サイズを調べることをお勧めします。

MicroLogix 埋め込み I/O の Word サイズ

MicroLogix モデル	入力 Word 数	出力 Word 数
1000	2	1
1100	6	4
1200	4	4
1400	8	6
1500	4	4

MicroLogix 拡張 I/O の Word サイズ

モジュール:	入力 Word 数	出力 Word 数
1769-HSC	35	34
1769-IA8I	1	0
1769-IA16	1	0
1769-IF4	6	0
1769-IF4XOF2	8	2
1769-IF8	12	1
1769-IM12	1	0
1769-IQ16	1	0
1769-IQ6XOW4	1	1
1769-IQ16F	1	0
1769-IQ32	2	0
1769-IR6	8	0
1769-IT6	8	0
1769-OA8	0	1
1769-OA16	0	1

モジュール:	入力 Word 数	出力 Word 数
1769-OB8	0	1
1769-OB16	0	1
1769-OB16P	0	1
1769-OB32	0	2
1769-OF2	2	2
1769-OF8C	11	9
1769-OF8V	11	9
1769-OV16	0	1
1769-OW8	0	1
1769-OW16	0	1
1769-OW8I	0	1
1769-SDN	66	2
1769-SM1	12	12
1769-SM2	7	7
1769-ASCII	108	108
1762-IA8	1	0
1762-IF2OF2	6	2
1762-IF4	7	0
1762-IQ8	1	0
1762-IQ8OW6	1	1
1762-IQ16	1	0
1762-OA8	0	1
1762-OB8	0	1
1762-OB16	0	1
1762-OW8	0	1
1762-OW16	0	1
1762-IT4	6	0
1762-IR4	6	0
1762-OF4	2	4
1762-OX6I	0	1

計算

スロット x の入力 Word オフセット = スロット 0 からスロット (x-1) での入力 Word 数。

● 注記:

1. 拡張 I/O にオフセットする場合には埋め込み I/O を考慮する必要があります。
2. 入力 Word のオフセットの計算では出力 Word 数は考慮されません。

I/O の例

以下の場合

スロット 0 = MicroLogix 1500 LRP シリーズ C = 入力 Word 数は 4

スロット 1 = 1769-OF2 = 入力 Word 数は 2

スロット 2 = 1769-OW8 = 入力 Word 数は 0

スロット 3 = 1769-IA16 = 入力 Word 数は 1

スロット 4 = 1769-OF8V = 入力 Word 数は 11

スロット 3 (= 4 + 2 = 6 Word) のビット 5 = I:6/5

SLC 500 での構文

構文	データ型	アクセス
I:<スロット>	Short、Word、BCD	読み取り専用
I:<スロット>.<Word>	Short、Word、BCD	読み取り専用
I:<スロット>/<ビット>	Boolean	読み取り専用
I:<スロット>.<Word>/<ビット>	Boolean	読み取り専用

範囲

PLC モデル	最小スロット	最大スロット	最大 Word
MicroLogix	該当なし	なし	2047
SLC 500 固定 I/O	該当なし	なし	1
SLC 500 モジュール I/O	1	30	*
PLC-5 シリーズ	該当なし	なし	277 (8 進)

*各 I/O モジュールで使用可能な入力/出力 Word の数については、[SLC 500 モジュール I/O 選択ガイド](#)を参照してください。

例

MicroLogix	説明
I:0	Word 0
I/2	ビット 2
I:1/5	ビット 5 Word 1

SLC 500 固定 I/O	説明
I:0	Word 0
I:1	Word 1
I/16	ビット 16
I:1/0	ビット 0 Word 1 (I/16 と同じ)

PLC5*	説明
I:0	Word 0
I:10	Word 8 (8 進の 10 = 10 進の 8)
I/20	ビット 16 (8 進の 20 = 10 進の 16)
I:1/0	ビット 0 Word 1 (I/20 と同じ)

*アドレスは 8 進で示されています。

SLC 500 モジュール I/O	説明
I:1	Word 0 スロット 1
I:1.0	Word 0 スロット 1 (I:1 と同じ)
I:12	Word 0 スロット 12
I:12.2	Word 2 スロット 12
I:4.0/0	ビット 0 Word 0 スロット 4
I:4/0	ビット 0 スロット 4 (I:4.0/0 と同じ)
I:4.2/0	ビット 0 Word 2 スロット 4
I:4/32	ビット 32 スロット 4 (I:4.2/0 と同じ)

ステータスファイル

ステータスファイルにアクセスするには、Word およびオプションで Word 内のビットを指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
S:<Word>	Short、Word、BCD、DWord、Long、LBCD	読み取り/書き込み
S:<Word> [行数][列数]	Short、Word、BCD、DWord、Long、LBCD (配列タイプ)	読み取り/書き込み
S:<Word> [列数]	Short、Word、BCD、DWord、Long、LBCD (配列タイプ)	読み取り/書き込み
S:<Word>/<ビット>	Boolean	読み取り/書き込み
S/ビット	Boolean	読み取り/書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 16 Word を超えてはなりません。

範囲

PLC モデル	最大 Word
MicroLogix	999
SLC 500 固定 I/O	96
SLC 500 モジュール I/O	999
PLC-5 シリーズ	999

32 ビットデータ型 (Long、DWord、Long BCD など) としてアクセスする場合、最大 Word 位置は 1 小さくなります。

例

例	説明
S:0	Word 0
S/26	ビット 26
S:4/15	ビット 15 Word 4
S:10 [16]	Word 10 で始まる 16 要素の配列
S:0 [4] [8]	Word 0 で始まる 4 x 8 要素の配列

バイナリファイル

バイナリファイルにアクセスするには、ファイル番号、Word、および Word 内のビット (オプション) を指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
B<ファイル>:<Word>	Short、 Word 、BCD、DWord、Long、LBCD	読み取り/書き込み
B<ファイル>:<Word> [行数][列数]	Short、 Word 、BCD、DWord、Long、LBCD (配列タイプ)	読み取り/書き込み
B<ファイル>:<Word> [列数]	Short、 Word 、BCD、DWord、Long、LBCD (配列タイプ)	読み取り/書き込み
B<ファイル>:<Word>/<ビット>	Boolean	読み取り/書き込み
B<ファイル>/bit	Boolean	読み取り/書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 16 Word を超えてはなりません。

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	3, 9-999	999
SLC 500 固定 I/O	3, 9-255	255
SLC 500 モジュール I/O	3, 9-999	999
PLC-5 シリーズ	3-999	1999

32 ビットデータ型 (Long、DWord、Long BCD など) としてアクセスする場合、最大 Word 位置は 1 小さくなります。

例

例	説明
B3:0	Word 0
B3/26	ビット 26
B12:4/15	ビット 15 Word 4
B3:10 [20]	Word 10 で始まる 20 要素の配列
B15:0 [6][6]	Word 0 で始まる 6 x 6 要素の配列

タイマーファイル

タイマーファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
T<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
ACC	Short、Word	読み取り/書き込み
PRE	Short、Word	読み取り/書き込み
DN	Boolean	読み取り専用
TT	Boolean	読み取り専用
EN	Boolean	読み取り専用

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	4, 9-999	999
SLC 500 固定 I/O	4, 9-255	255
SLC 500 モジュール I/O	4, 9-999	999
PLC-5 シリーズ	3-999	1999

例

例	説明
T4:0.ACC	タイマー 0 ファイル 4 のアキュムレータ
T4:10.DN	タイマー 10 ファイル 4 の完了ビット
T15:0.PRE	タイマー 0 ファイル 15 のプリセット

カウンタファイル

カウンタファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
C<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
ACC	Word 、Short	読み取り/書き込み
PRE	Word 、Short	読み取り/書き込み
UA	Boolean	読み取り専用
UN	Boolean	読み取り専用
OV	Boolean	読み取り専用
DN	Boolean	読み取り専用
CD	Boolean	読み取り専用
CU	Boolean	読み取り専用

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	5, 9-999	999
SLC 500 固定 I/O	5, 9-255	255
SLC 500 モジュール I/O	5, 9-999	999
PLC-5 シリーズ	3-999	1999

例

例	説明
C5:0.ACC	カウンタ 0 ファイル 5 のアキュムレータ
C5:10.DN	カウンタ 10 ファイル 5 の完了ビット
C15:0.PRE	カウンタ 0 ファイル 15 のプリセット

制御ファイル

制御ファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
R<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
LEN	Word、Short	読み取り書き込み
POS	Word、Short	読み取り書き込み
FD	Boolean	読み取り専用
IN	Boolean	読み取り専用
UL	Boolean	読み取り専用
ER	Boolean	読み取り専用
EM	Boolean	読み取り専用
DN	Boolean	読み取り専用
EU	Boolean	読み取り専用
EN	Boolean	読み取り専用

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	6, 9-999	999
SLC 500 固定 I/O	6, 9-255	255
SLC 500 モジュール I/O	6, 9-999	999
PLC-5 シリーズ	3-999	1999

例

例	説明
R6:0.LEN	制御 0 ファイル 6 の長さフィールド
R6:10.DN	制御 10 ファイル 6 の完了ビット
R15:18.POS	制御 18 ファイル 15 の位置フィールド

整数ファイル

整数ファイルにアクセスするには、ファイル番号、Word、および Word 内のビット (オプション) を指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
N<ファイル>:<Word>	Short、 Word 、BCD、DWord、Long、LBCD	読み取り書き込み
N<ファイル>:<Word> [行数][列数]	Short、 Word 、BCD、DWord、Long、LBCD (配列タイプ)	読み取り書き込み
N<ファイル>:<Word> [列数]	Short、 Word 、BCD、DWord、Long、LBCD (配列タイプ)	読み取り書き込み
N<ファイル>:<Word>/<ビット>	Boolean	読み取り書き込み
N<ファイル>/ビット	Boolean	読み取り書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 16 Word を超えてはなりません。

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	7, 9-999	999
SLC 500 固定 I/O	7, 9-255	255
SLC 500 モジュール I/O	7, 9-999	999
PLC-5 シリーズ	3-999	1999

32 ビットデータ型 (Long、DWord、Long BCD など) としてアクセスする場合、最大 Word 位置は 1 小さくなります。

例

例	説明
N7:0	Word 0
N7/26	ビット 26
N12:4/15	ビット 15 Word 4
N7:10 [8]	Word 10 で始まる 8 要素の配列
N15:0 [4] [5]	Word 0 で始まる 4 x 5 要素の配列

Float ファイル

Float ファイルにアクセスするには、ファイル番号と要素を指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
F<ファイル>:<要素>	Float	読み取り/書き込み
F<ファイル>:<要素> [行数][列数]	Float (配列タイプ)	読み取り/書き込み
F<ファイル>:<要素> [列数]	Float (配列タイプ)	読み取り/書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 8 Float を超えてはなりません。

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	8-999	999
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	8-999	999
PLC-5 シリーズ	3-999	1999

例

例	説明
F8:0	Float 0
F8:10 [16]	Word 10 で始まる 16 要素の配列
F15:0 [4] [4]	Word 0 で始まる 16 要素の配列

ASCII ファイル

ASCII ファイルのデータにアクセスするには、ファイル番号と文字位置を指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
A<ファイル>:<文字>	Char, Byte*	読み取り/書き込み
A<ファイル>:<文字> [行数][列数]	Char, Byte*	読み取り/書き込み
A<ファイル>:<文字> [列数]	Char, Byte*	読み取り/書き込み
A<ファイル>:<Word オフセット>/長さ	String**	読み取り/書き込み

*指定されているブロック要求サイズを配列要素の数を超えてはなりません。PLC は内部で 1 Word あたり 2 文字をファイルにパックし、上位バイトには 1 つ目の文字が含まれ、下位バイトには 2 つ目の文字が含まれます。PLC プログラミングソフトウェアでは Word レベル (2 文字レベル) でのアクセスが可能です。Allen-Bradley ControlLogix Ethernet ドライバーでは文字レベルでのアクセスが可能です。

プログラミングソフトウェア "A10:0 = AB" を使用した場合、A10:0 の上位バイトに 'A' が格納され、下位バイトに 'B' が格納されます。Allen-Bradley ControlLogix Ethernet ドライバーを使用して "A10:0 = A" と "A10:1 = B" の 2 つを指定すると、同じデータが PLC メモリに格納されます。

**このファイルを文字列データとして参照することで、プログラミングソフトウェアのように Word 境界上のデータにアクセスできます。長さは最大 232 文字です。デバイスに送信された文字列がアドレスによって指定されている長さより短い場合、ドライバーはその文字列を Null 終端してからコントローラに送信します。

範囲

PLC モデル	ファイル番号	最大文字
MicroLogix	3-255	511
SLC 500 固定 I/O	該当なし	なし

PLC モデル	ファイル番号	最大文字
SLC 500 モジュール I/O	9-999	1999
PLC-5 シリーズ	3-999	1999

● **注記:** 一部の MicroLogix デバイスと SLC 500 PLC デバイスでは ASCII ファイルタイプがサポートされていません。詳細については、PLC のドキュメントを参照してください。

例

例	説明
A9:0	文字 0 (Word 0 の上位バイト)
A27:10 [80]	文字 10 で始まる 80 文字の配列
A15:0 [4] [16]	文字 0 で始まる 4 x 16 文字の配列
A62:0/32	Word オフセット 0 で始まる 32 文字の文字列

文字列ファイル

文字列ファイルにアクセスするには、ファイル番号と要素を指定します。文字列は 82 文字から成る Null 終端配列です。このドライバーは PLC から返された文字列長に基づいて Null 終端を配置します。デフォルトのデータ型を太字で示しています。

● **注記:** 文字列ファイルでは配列はサポートされていません。

構文	データ型	アクセス
ST<ファイル>:<要素>.<フィールド>	String	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	9-999	999
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	9-999	999
PLC-5 シリーズ	3-999	999

例

例	説明
ST9:0	文字列 0
ST18:10	文字列 10

BCD ファイル

BCD ファイルにアクセスするには、ファイル番号と Word を指定します。デフォルトのデータ型を太字で示しています。

PLC-5 での構文

構文	データ型	アクセス
D<ファイル>:<Word>	BCD 、LBCD	読み取り/書き込み
D<ファイル>:<Word> [行数][列数]	BCD 、LBCD (配列タイプ)	読み取り/書き込み
D<ファイル>:<Word> [列数]	BCD 、LBCD (配列タイプ)	読み取り/書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 16 BCD を超えてはなりません。

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	該当なし	なし
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	該当なし	なし
PLC-5 シリーズ	3-999	999

例

例	説明
D9:0	Word 0
D27:10 [16]	Word 10 で始まる 16 要素の配列
D15:0 [4][8]	Word 0 で始まる 32 要素の配列

Long ファイル

長整数ファイルにアクセスするには、ファイル番号と要素を指定します。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
L<ファイル>:<DWord>	Long、 DWord 、LBCD	読み取り書き込み
L<ファイル>:<DWord> [行数][列数]	Long、 DWord 、LBCD (配列タイプ)	読み取り書き込み
L<ファイル>:<DWord> [列数]	Long、 DWord 、LBCD (配列タイプ)	読み取り書き込み

指定されているブロック要求サイズを配列要素の数 (バイト数) が超えてはなりません。つまり、ブロック要求サイズが 32 バイトの場合、配列サイズが 8 Long を超えてはなりません。

範囲

PLC モデル	ファイル番号	最大 Word
MicroLogix	9-999	999
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	該当なし	なし
PLC-5 シリーズ	該当なし	なし

例

例	説明
L9:0	Word 0
L9:10 [8]	Word 10 で始まる 8 要素の配列
L15:0 [4] [5]	Word 0 で始まる 4 x 5 要素の配列

MicroLogix PID ファイル

PID ファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
PD<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
SPS	Word 、Short	読み取り書き込み
KC	Word 、Short	読み取り書き込み
TI	Word 、Short	読み取り書き込み

要素フィールド	データ型	アクセス
TD	Word、Short	読み取り/書き込み
MAXS	Word、Short	読み取り/書き込み
MINS	Word、Short	読み取り/書き込み
ZCD	Word、Short	読み取り/書き込み
CVH	Word、Short	読み取り/書き込み
CVL	Word、Short	読み取り/書き込み
LUT	Word、Short	読み取り/書き込み
SPV	Word、Short	読み取り/書き込み
CVP	Word、Short	読み取り/書き込み
TM	Boolean	読み取り/書き込み
AM	Boolean	読み取り/書き込み
CM	Boolean	読み取り/書き込み
OL	Boolean	読み取り/書き込み
RG	Boolean	読み取り/書き込み
SC	Boolean	読み取り/書き込み
TF	Boolean	読み取り/書き込み
DA	Boolean	読み取り/書き込み
DB	Boolean	読み取り/書き込み
UL	Boolean	読み取り/書き込み
LL	Boolean	読み取り/書き込み
SP	Boolean	読み取り/書き込み
PV	Boolean	読み取り/書き込み
DN	Boolean	読み取り/書き込み
EN	Boolean	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	3-255	255
すべての SLC	該当なし	なし
PLC-5	PID ファイル	PID ファイル

例

例	説明
PD14:0.KC	PD 0 ファイル 14 の比例ゲイン
PD18:6.EN	PD 6 ファイル 18 の PID 有効化ビット

PID ファイル

PID ファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

PLC-5 での構文

構文	データ型	アクセス
PD<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
SP	Real	読み取り書き込み
KP	Real	読み取り書き込み
KI	Real	読み取り書き込み
KD	Real	読み取り書き込み
BIAS	Real	読み取り書き込み
MAXS	Real	読み取り書き込み
MINS	Real	読み取り書き込み
DB	Real	読み取り書き込み
SO	Real	読み取り書き込み
MAXO	Real	読み取り書き込み
MINO	Real	読み取り書き込み
UPD	Real	読み取り書き込み
PV	Real	読み取り書き込み
ERR	Real	読み取り書き込み
OUT	Real	読み取り書き込み
PVH	Real	読み取り書き込み
PVL	Real	読み取り書き込み
DVP	Real	読み取り書き込み
DVN	Real	読み取り書き込み
PVDB	Real	読み取り書き込み
DVDB	Real	読み取り書き込み
MAXI	Real	読み取り書き込み
MINI	Real	読み取り書き込み
TIE	Real	読み取り書き込み
FILE	Word、Short	読み取り書き込み
ELEM	Word、Short	読み取り書き込み
EN	Boolean	読み取り書き込み
CT	Boolean	読み取り書き込み
CL	Boolean	読み取り書き込み
PVT	Boolean	読み取り書き込み
DO	Boolean	読み取り書き込み
SWM	Boolean	読み取り書き込み
CA	Boolean	読み取り書き込み
MO	Boolean	読み取り書き込み
PE	Boolean	読み取り書き込み
INI	Boolean	読み取り書き込み
SPOR	Boolean	読み取り書き込み
OLL	Boolean	読み取り書き込み
OLH	Boolean	読み取り書き込み
EWD	Boolean	読み取り書き込み
DVNA	Boolean	読み取り書き込み
DVHA	Boolean	読み取り書き込み
PVLA	Boolean	読み取り書き込み
PVHA	Boolean	読み取り書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	なし
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	該当なし	なし
PLC-5 シリーズ	3-999	999

例

例	説明
PD14:0.SP	PD 0 ファイル 14 のセットポイントフィールド
PD18:6.EN	PD 6 ファイル 18 のステータス有効化ビット

MicroLogix メッセージファイル

メッセージファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

構文	データ型	アクセス
MG<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
IA	Word , Short	読み取り/書き込み
RBL	Word , Short	読み取り/書き込み
LBN	Word , Short	読み取り/書き込み
RBN	Word , Short	読み取り/書き込み
CHN	Word , Short	読み取り/書き込み
NOD	Word , Short	読み取り/書き込み
MTO	Word , Short	読み取り/書き込み
NB	Word , Short	読み取り/書き込み
TFT	Word , Short	読み取り/書き込み
TFN	Word , Short	読み取り/書き込み
ELE	Word , Short	読み取り/書き込み
SEL	Word , Short	読み取り/書き込み
TO	Boolean	読み取り/書き込み
CO	Boolean	読み取り/書き込み
EN	Boolean	読み取り/書き込み
RN	Boolean	読み取り/書き込み
EW	Boolean	読み取り/書き込み
ER	Boolean	読み取り/書き込み
DN	Boolean	読み取り/書き込み
ST	Boolean	読み取り/書き込み
BK	Boolean	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	3-255	255
すべての SLC	該当なし	なし
PLC5	<u>メッセージファイル</u>	<u>メッセージファイル</u>

例

例	説明
MG14:0.TO	データファイル 14 の MSG 要素 0 のタイムアウトビット
MG18:6.CO	データファイル 18 の MSG 要素 6 の継続ビット

メッセージファイル

メッセージファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

PLC-5 での構文

構文	データ型	アクセス
MG<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
ERR	Short 、Word	読み取り書き込み
RLEN	Short 、Word	読み取り書き込み
DLEN	Short 、Word	読み取り書き込み
EN	Boolean	読み取り書き込み
ST	Boolean	読み取り書き込み
DN	Boolean	読み取り書き込み
ER	Boolean	読み取り書き込み
CO	Boolean	読み取り書き込み
EW	Boolean	読み取り書き込み
NR	Boolean	読み取り書き込み
TO	Boolean	読み取り書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	なし
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	該当なし	なし
PLC-5 シリーズ	3-999	999

例

例	説明
MG14:0.RLEN	MG 0 ファイル 14 の要求された長さのフィールド
MG18:6.CO	MG 6 ファイル 18 の継続ビット

ブロック転送ファイル

ブロック転送ファイルは、ファイル番号、要素、フィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

PLC-5 での構文

構文	データ型	アクセス
BT<ファイル>:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
RLEN	Word、Short	読み取り/書き込み
DLEN	Word、Short	読み取り/書き込み
FILE	Word、Short	読み取り/書き込み
ELEM	Word、Short	読み取り/書き込み
RW	Boolean	読み取り/書き込み
ST	Boolean	読み取り/書き込み
DN	Boolean	読み取り/書き込み
ER	Boolean	読み取り/書き込み
CO	Boolean	読み取り/書き込み
EW	Boolean	読み取り/書き込み
NR	Boolean	読み取り/書き込み
TO	Boolean	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	なし
SLC 500 固定 I/O	該当なし	なし
SLC 500 モジュール I/O	該当なし	なし
PLC-5 シリーズ	3-999	1999

例

例	説明
BT14:0.RLEN	BT 0 ファイル 14 の要求された長さのフィールド
BT18:6.CO	BT 6 ファイル 18 の継続ビット

ファンクションファイル

ENI MicroLogix および MicroLogix 1100 デバイスモデルによってサポートされているファイルについては、以下のリストからリンクを選択してください。

[高速カウンタファイル\(HSC\)](#)

[リアルタイムクロックファイル\(RTC\)](#)

[チャンネル0 通信ステータスファイル\(CS0\)](#)

[チャンネル1 通信ステータスファイル\(CS1\)](#)

[I/O モジュールステータスファイル\(IOS\)](#)

● デバイスモデルとそのサポートされているファイルの詳細については、[アドレスの説明](#)を参照してください。

高速カウンタファイル(HSC)

HSC ファイルは、要素とフィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

● **関連項目:** [ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)

構文	データ型	アクセス
HSC:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	デフォルトの型	アクセス
ACC	DWord 、Long	読み取り専用
HIP	DWord 、Long	読み取り書き込み
LOP	DWord 、Long	読み取り書き込み
OVF	DWord 、Long	読み取り書き込み
UNF	DWord 、Long	読み取り書き込み
PFN	Word 、Short	読み取り専用
ER	Word 、Short	読み取り専用
MOD	Word 、Short	読み取り専用
OMB	Word 、Short	読み取り専用
HPO	Word 、Short	読み取り書き込み
LPO	Word 、Short	読み取り書き込み
UIX	Boolean	読み取り専用
UIP	Boolean	読み取り専用
AS	Boolean	読み取り専用
ED	Boolean	読み取り専用
SP	Boolean	読み取り専用
LPR	Boolean	読み取り専用
HPR	Boolean	読み取り専用
DIR	Boolean	読み取り専用
CD	Boolean	読み取り専用
CU	Boolean	読み取り専用
UIE	Boolean	読み取り書き込み
UIL	Boolean	読み取り書き込み
FE	Boolean	読み取り書き込み
CE	Boolean	読み取り書き込み

要素フィールド	デフォルトの型	アクセス
LPM	Boolean	読み取り/書き込み
HPM	Boolean	読み取り/書き込み
UFM	Boolean	読み取り/書き込み
OFM	Boolean	読み取り/書き込み
LPI	Boolean	読み取り/書き込み
HPI	Boolean	読み取り/書き込み
UFI	Boolean	読み取り/書き込み
OFI	Boolean	読み取り/書き込み
UF	Boolean	読み取り/書き込み
OF	Boolean	読み取り/書き込み
MD	Boolean	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	254
すべての SLC	該当なし	なし
PLC5	該当なし	なし

例

例	説明
HSC:0.OMB	高速カウンタ 0 の出力マスク設定。
HSC:1.ED	高速カウンタ 1 のエラー検出インジケータ。

リアルタイムクロックファイル (RTC)

RTC ファイルは、要素とフィールドを指定することによってデータにアクセスする構造体タイプのファイルです。デフォルトのデータ型を太字で示しています。

● **関連項目:** [ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)

構文	データ型	アクセス
RTC:<要素>.<フィールド>	フィールドによって異なる	フィールドによって異なる

各要素には次のフィールドを使用できます。各フィールドの意味については、PLC のドキュメントを参照してください。

要素フィールド	データ型	アクセス
YR	Word 、Short	読み取り/書き込み
MON	Word 、Short	読み取り/書き込み
DAY	Word 、Short	読み取り/書き込み
HR	Word 、Short	読み取り/書き込み
MIN	Word 、Short	読み取り/書き込み
SEC	Word 、Short	読み取り/書き込み
DOW	Word 、Short	読み取り/書き込み
DS	Boolean	読み取り専用
BL	Boolean	読み取り専用
_SET (ブロック書き込み)	Boolean	読み取り/書き込み

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	254
すべての SLC	該当なし	なし
PLC5	該当なし	なし

例

例	説明
RTC:0.YR	リアルタイムクロック 0 の年設定。
RTC:0.BL	リアルタイムクロック 0 のバッテリー低下インジケータ。

チャンネル 0 通信ステータスファイル (CS0)

チャンネル 0 の通信ステータスファイルにアクセスするには、Word (およびオプションで Word 内のビット) を指定します。デフォルトのデータ型を太字で示しています。

● 関連項目: [ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)

構文	データ型	アクセス
CS0:<Word>	Short、 Word 、BCD、DWord、Long、LBCD	<Word> と <ビット> による
CS0:<Word>/<ビット>	Boolean	<Word> と <ビット> による
CS0/ビット	Boolean	<Word> と <ビット> による

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	254
すべての SLC	該当なし	なし
PLC5	該当なし	なし

例

例	説明
CS0:0	Word 0。
CS0:4/2	ビット 2 Word 4 = MCP。

● CS0 の Word/ビットの意味については、Rockwell のドキュメントを参照してください。

チャンネル 1 通信ステータスファイル (CS1)

チャンネル 1 の通信ステータスファイルにアクセスするには、Word (およびオプションで Word 内のビット) を指定します。デフォルトのデータ型を太字で示しています。

● 関連項目: [ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)

構文	データ型	アクセス
CS1:<Word>	Short、 Word 、BCD、DWord、Long、LBCD	<Word> と <ビット> による
CS1:<Word>/<ビット>	Boolean	<Word> と <ビット> による
CS1/ビット	Boolean	<Word> と <ビット> による

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	254
すべての SLC	該当なし	なし
PLC5	該当なし	なし

例

例	説明
CS1:0	Word 0。
CS1:4/2	ビット 2 Word 4 = MCP。

- CS1 の Word/ビットの意味については、Rockwell のドキュメントを参照してください。

I/O モジュールステータスファイル (IOS)

I/O モジュールステータスファイルにアクセスするには、Word、およびオプションでビットを指定します。デフォルトのデータ型を太字で示しています。

- 関連項目: [ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)

構文	データ型	アクセス
IOS:<Word>	Short、 Word 、BCD、DWord、Long、LBCD	<Word> と <ビット> による
IOS:<Word>/<ビット>	Boolean	<Word> と <ビット> による
IOS/ビット	Boolean	<Word> と <ビット> による

範囲

PLC モデル	ファイル番号	最大要素
MicroLogix	該当なし	254
すべての SLC	該当なし	なし
PLC5	該当なし	なし

例

例	説明
IOS:0	Word 0
IOS:4/2	ビット 2 Word 4

- 1769 拡張 I/O ステータスコードの一覧については、製造元のマニュアルを参照してください。

エラーコード

以降のセクションでは、サーバーのイベントログに記録されるエラーコードを定義しています。特定のエラーコードタイプの詳細については、以下のリストからリンクを選択してください。

[カプセル化エラーコード](#)

[CIP エラーコード](#)

カプセル化 エラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	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。
0070	ターゲット ID が無効です。

CIP エラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	ログコード	説明
0001	0x01	接続エラー*
0002	0x02	リソースが不足しています
0003	0x03	値が無効です
0004	0x04	IOI を解釈できなかったタグが存在しません ● 注記: このエラーは、アドレスが有効で、セキュリティアクセス許可によってクライアントアクセスが許可されていない場合にも発生することがあります。
0005	0x05	宛先が不明です
0006	0x06	要求されたデータは応答パケットに収まりません
0007	0x07	接続が失われました
0008	0x08	サポートされていないサービスです
0009	0x09	データセグメントにエラーがあるか属性値が無効です
000A	0x0A	属性リストエラー
000B	0x0B	状態がすでに存在します
000C	0x0C	オブジェクトモデルが競合しています
000D	0x0D	オブジェクトがすでに存在します
000E	0x0E	属性を設定できません
000F	0x0F	Permission denied
0010	0x10	デバイスの状態が競合しています
0011	0x11	応答が大きすぎます
0012	0x12	プリミティブがフラグメント化されています
0013	0x13	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
0014	0x14	属性がサポートされていません
0015	0x15	指定されたデータでは多すぎます
001A	0x1A	ブリッジ要求が大きすぎます
001B	0x1B	ブリッジ応答が大きすぎます

エラーコード	ログコード	説明
001C	0x1C	属性リストが不足しています
001D	0x1D	属性リストが無効です
001E	0x1E	組み込みサービスエラー
001F	0x1F	接続中にエラーが発生しました**
0022	0x22	無効な応答を受信しました
0025	0x25	キーセグメントエラー
0026	0x26	指定された IOI Word の数は IOI Word 数と一致しません
0027	0x27	リストに予期しない属性があります

● *関連項目: [0x0001 拡張エラーコード](#)

● **関連項目: [0x001F 拡張エラーコード](#)

Logix5000 固有 (1756-L1) のエラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	説明
00FF	一般的なエラー*

● *関連項目: [0x00FF 拡張エラーコード](#)

● 関連項目: 一覧にないエラーコードについては、Rockwell のドキュメントを参照してください。

0x0001 拡張エラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	説明
0100	接続が使用中です
0103	転送はサポートされていません
0106	オーナーシップが競合しています
0107	接続が見つかりません
0108	接続タイプが無効です
0109	接続サイズが無効です
0110	モジュールが設定されていません
0111	EPR はサポートされていません
0114	モジュールが間違っています
0115	デバイスタイプが間違っています
0116	リビジョンが間違っています
0118	構成フォーマットが無効です
011A	アプリケーションが接続されていません
0203	接続がタイムアウトになりました
0204	未接続メッセージがタイムアウトになりました
0205	未接続送信パラメータエラー
0206	メッセージが大きすぎます
0301	バッファメモリがありません
0302	帯域幅を使用できません
0303	スクリーナを使用できません
0305	署名が一致しています
0311	ポートを使用できません
0312	リンクアドレスを使用できません

エラーコード	説明
0315	セグメントタイプが無効です
0317	接続がスケジュールされていません
0318	自己へのリンクアドレスは無効です

● 一覧にないエラーコードについては、Rockwell のドキュメントを参照してください。

0x0C 拡張エラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	説明
0203	接続がタイムアウトになりました

● 一覧にないエラーコードについては、Rockwell のドキュメントを参照してください。

0x00FF 拡張エラーコード

次のエラーコードは 16 進数で表示されます。

エラーコード	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
2106	データは使用中です
2107	データ型が無効であるかサポートされていません。

● 一覧にないエラーコードについては、Rockwell のドキュメントを参照してください。

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リングとソートについては、OPC 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。

デバイスからコントローラプロジェクトをアップロード中に次のエラーが発生しました。シンボリックプロトコルを使用します。

エラータイプ:
エラー

同期化中に無効または破損したコントローラプロジェクトが検出されました。まもなく同期化を再試行します。

エラータイプ:
エラー

考えられる原因:
同期化中に無効または破損したコントローラプロジェクトが検出されました。

解決策:
操作は必要ありません。ドライバーは 30 秒経過すると再び同期化を試みます。

● **注記:**
論理アドレス指定モードではプロジェクトの同期化が必要です。

同期化中にプロジェクトのダウンロードが検出されました。まもなく同期化を再試行します。

エラータイプ:
エラー

考えられる原因:
デバイスがコントローラプロジェクトと同期化しているときにプロジェクトのダウンロードが試みられました。

解決策:
操作は必要ありません。ドライバーは 30 秒経過すると再び同期化を試みます。

● **注記:**
論理アドレス指定モードではプロジェクトの同期化が必要です。

データベースエラー。参照タグのデータ型が不明です。エイリアスタグのデータ型をデフォルトに設定します。| 参照タグ = '<タグ>'、エイリアスタグ = '<タグ>'、デフォルトデータ型 = '<タイプ>'。

エラータイプ:
エラー

考えられる原因:
エイリアスタグの宣言で参照される "Alias For" タグのデータ型がタグインポートファイルで見つかりませんでした。エイリアスタグを正しく生成するためにはこのデータ型が必要です。

解決策:
エイリアスタグはデフォルトの型として指定されているデータ型を使用します。

● 注記:

RSLogix5000 で、「Edit Tags」タブのタグビューに「Alias For」列があり、このタグへの参照、構造タグメンバー、またはエイリアスタグが表すビットが入力されています。

● 関連項目:

Logix オプション

データベースエラー。タグインポートファイルでメンバーのデータ型が見つかりません。データ型をデフォルトに設定します。| メンバーのデータ型 = '<タイプ>'、UDT = '<タイプ>'、デフォルトデータ型 = '<タイプ>'。

エラータイプ:

エラー

考えられる原因:

ユーザー定義型のメンバーのデータ型の定義がタグインポートファイルで見つかりませんでした。メンバーはデバイスのプロパティで指定されているデフォルトの型をとります。

解決策:

指定されているタグのユーザー定義データ型の定義を確認して修正し、インポートを再試行してください。

● 関連項目:

Logix オプション

データベースエラー。タグインポートファイルでデータ型が見つかりません。タグは追加されません。| データ型 = '<タイプ>'、タグ名 = '<タグ>'。

エラータイプ:

エラー

考えられる原因:

指定されているタグのデータ型の定義がタグインポートファイルで見つかりませんでした。タグはデータベースに追加されません。

解決策:

指定されているタグのデータ型の定義を確認して修正し、インポートを再試行してください。

データベースエラー。エイリアスタグの処理中にエラーが発生しました。タグは追加されませんでした。| エイリアスタグ = '<タグ>'。

エラータイプ:

エラー

考えられる原因:

エイリアスタグの処理中に内部エラーが発生しました。エイリアスタグを生成できませんでした。

解決策:

指定されているタグのデータ型の定義を確認して修正し、インポートを再試行してください。

データベースエラー。レジスタセッションの要求時にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

要求時に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。要求内のすべての読み取りと書き込みが失敗しました。

解決策:

このようなエラーからの回復はドライバーが自動的に試みます。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。エラー 0x02 はドライバー関連ではなくデバイス関連なので除外されます。

● 関連項目:

カプセル化エラーコード

データベースエラー。レジスタセッションの要求時にフレーミングエラーが発生しました。

エラータイプ:

エラー

データベースエラー。フォワードオープンの要求時にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。

エラータイプ:

エラー

データベースエラー。フォワードオープンの要求時にフレーミングエラーが発生しました。

エラータイプ:

エラー

データベースエラー。フォワードオープンの要求時にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

エラー

データベースエラー。プロジェクト情報のアップロード中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● 関連項目:

カプセル化エラーコード

データベースエラー。プロジェクト情報のアップロード中にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● 関連項目:

CIP エラーコード

データベースエラー。プロジェクト情報のアップロード中にフレーミング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:

1. パケットに不整列が発生しています (原因はPC とデバイス間の接続/切断)。
2. デバイスのケーブル接続の不良によりノイズが発生しています。

解決策:

1. ノイズが少ないネットワーク上にデバイスを配置してください。
2. 要求タイムアウト、再試行回数、またはその両方の値を増やしてください。
3. サーバーを再起動してから、もう一度試してください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

データベースエラー。内部エラーが発生しました。**エラータイプ:**

エラー

データベースエラー。プログラム情報のアップロード中にカプセル化エラーが発生しました。| プログラム名 = '<名前>'、カプセル化エラー = <コード>。**エラータイプ:**

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● 関連項目:

カプセル化エラーコード

データベースエラー。プログラム情報のアップロード中にエラーが発生しました。| プログラム名 = '<名前>'、CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● **注記:**

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● **関連項目:**

CIP エラーコード

データベースエラー。プログラム情報のアップロード中にフレーミングエラーが発生しました。| プログラム名 = '<名前>'。

エラータイプ:

エラー

考えられる原因:

1. パケットに不整列が発生しています (原因は PC とデバイス間の接続/切断)。
2. デバイスのケーブル接続の不良によりノイズが発生しています。

解決策:

1. ノイズが少ないネットワーク上にデバイスを配置してください。
2. 要求タイムアウト、再試行回数、またはその両方の値を増やしてください。
3. サーバーを再起動してから、もう一度試してください。

● **注記:**

論理アドレス指定モードではプロジェクトのアップロードが必要です。

データベースエラー。タグの CIP データ型を解決できません。デフォルトの型に設定します。| CIP データ型 = <タイプ>、タグ名 = '<タグ>'、デフォルト データ型 = '<タイプ>'。

エラータイプ:

エラー

考えられる原因:

1. インポートファイル内の CIP データ型が不明です。
2. インポートファイルにエラーが含まれている可能性があります。

解決策:

RSLogix 内のエラーを解決し、タグエクスポートプロセスを再試行して新しいタグインポートファイルを生成してください。

● **関連項目:**

自動タグデータベース生成の準備

プロジェクト情報のアップロード中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● 関連項目:

カプセル化エラーコード

プロジェクト情報のアップロード中にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

エラー

考えられる原因:

コントローラプロジェクトをアップロード中に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

● 関連項目:

CIP エラーコード

プロジェクト情報のアップロード中にフレーミング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:

1. パケットに不整列が発生しています (原因は PC とデバイス間の接続/切断)。
2. デバイスのケーブル接続の不良によりノイズが発生しています。

解決策:

1. ノイズが少ないネットワーク上にデバイスを配置してください。
2. 要求タイムアウト、再試行回数、またはその両方の値を増やしてください。
3. サーバーを再起動してから、もう一度試してください。

● 注記:

論理アドレス指定モードではプロジェクトのアップロードが必要です。

プログラム情報のアップロード中にカプセル化エラーが発生しました。| プログラム名 = '<名前>', カプセル化エラー = <コード>。

エラータイプ:
エラー

プログラム情報のアップロード中にエラーが発生しました。| プログラム名 = '<名前>', CIP エラー = <コード>, 拡張エラー = <コード>。

エラータイプ:
エラー

プログラム情報のアップロード中にフレーミングエラーが発生しました。| プログラム名 = '<名前>'。

エラータイプ:
エラー

コントローラプログラム情報のアップロード中にカプセル化エラーが発生しました。カプセル化エラー = <コード>。

エラータイプ:
エラー

コントローラプログラム情報のアップロード中にエラーが発生しました。CIP エラー = <コード>, 拡張エラー = <コード>。

エラータイプ:
エラー

コントローラプログラム情報のアップロード中にフレーミングエラーが発生しました。

エラータイプ:
エラー

プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。

エラータイプ:
エラー

考えられる原因:

不活動ウォッチドッグに設定されている値が小さすぎるため、プロジェクトをロードできません。

解決策:

不活動ウォッチドッグの値を増やしてから、もう一度試してください。

データベースエラー。プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。

エラータイプ:
エラー

考えられる原因:

不活動ウォッチドッグに設定されている値が小さすぎるため、プロジェクトをロードできません。

解決策:

不活動ウォッチドッグの値を増やしてから、もう一度試してください。

データベースエラー。フォワードオープンに要求に利用可能な接続はもうありません。

エラータイプ:

エラー

タグデータベースのインポート用のファイルを開くときにエラーが発生しました。| OS エラー = '<コード>'。

エラータイプ:

エラー

サポートされていないコントローラです。| ベンダー ID = <ID>、製品タイプ = <タイプ>、製品コード = <コード>、製品名 = '<名前>'。

エラータイプ:

警告

デバイスから受信したフレームにエラーが含まれています。

エラータイプ:

警告

考えられる原因:

1. PC とデバイス間の接続/切断によってパケットに不整列が発生しています。
2. デバイスのケーブル接続の不良によりノイズが発生しています。

解決策:

1. ノイズが少ないネットワーク上にデバイスを配置してください。
2. 要求タイムアウト、再試行回数、またはその両方の値を増やしてください。

フレーミングエラーにより書き込み要求が失敗しました。| 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:

1. 不正な要求サービスコードがあります。
2. ドライバーは予想されるバイト数よりも多いかまたは少ないバイト数を受信しました。
3. このエラーが頻繁に発生する場合、ケーブル接続またはデバイスに問題がある可能性があります。

解決策:

1. ドライバーがこのエラーから回復できるようにするには、再試行回数を増やしてください。
2. ケーブル接続とデバイスが適切に機能していることを確認してください。

フレーミングエラーによりタグの読み取り要求が失敗しました。| 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:

1. 不正な要求サービスコードがあります。
2. ドライバーは予想されるバイト数よりも多いかまたは少ないバイト数を受信しました。
3. このエラーが頻繁に発生する場合、ケーブル接続またはデバイスに問題がある可能性があります。

解決策:

1. ドライバーがこのエラーから回復できるようにするには、再試行回数を増やしてください。
2. ケーブル接続とデバイスが適切に機能していることを確認してください。

フレーミングエラーによりブロック読み取り要求が失敗しました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。**エラータイプ:**

警告

考えられる原因:

1. デバイスとホスト PC 間のイーサネット接続が切断しています。
2. イーサネット接続の通信パラメータが不正です。
3. この名前のデバイスに不正な IP アドレスが割り当てられている可能性があります。

解決策:

1. PC とデバイス間のケーブル接続を確認してください。
2. この名前のデバイスに正しいポートが指定されていることを確認してください。
3. この名前のデバイスに指定した IP アドレスが実際のデバイスのアドレスと一致することを確認してください。

フレーミングエラーによりブロック読み取り要求が失敗しました。| ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。**エラータイプ:**

警告

考えられる原因:

1. 不正な要求サービスコードがあります。
2. ドライバーは予想されるバイト数よりも多いかまたは少ないバイト数を受信しました。
3. このエラーが頻繁に発生する場合、ケーブル接続またはデバイスに問題がある可能性があります。

解決策:

1. ドライバーがこのエラーから回復できるようにするには、再試行回数を増やしてください。
2. ケーブル接続とデバイスが適切に機能していることを確認してください。

タグに書き込めません。| タグアドレス = '<アドレス>', CIP エラー = <コード>、拡張エラー = <コード>。**エラータイプ:**

警告

考えられる原因:

指定されたタグの書き込み要求時に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。

● 関連項目:

CIP エラーコード

タグを読み取れません。| タグアドレス = '<アドレス>', CIP エラー = <コード>, 拡張エラー = <コード>。

エラータイプ:

警告

考えられる原因:

指定されたタグの読み取り要求時に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。

● 関連項目:

CIP エラーコード

ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>', CIP エラー = <コード>, 拡張エラー = <コード>。

エラータイプ:

警告

ブロックを読み取れません。| ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>', CIP エラー = <コード>, 拡張エラー = <コード>。

エラータイプ:

警告

タグに書き込めません。コントローラタグのデータ型が不明です。| タグアドレス = '<アドレス>', データ型 = <タイプ>。

エラータイプ:

警告

考えられる原因:

コントローラタグのデータ型がサポートされていないため、指定されたタグの書き込み要求は失敗しました。

解決策:

テクニカルサポートまでご連絡の上、この型に関するサポートの追加をご要望ください。

● 関連項目:

アトミックデータ型のアドレス指定

タグを読み取れません。コントローラタグのデータ型が不明です。タグは非アクティブ化されました。| タグアドレス = '<アドレス>', データ型 = <タイプ>。

エラータイプ:

警告

考えられる原因:

コントローラタグのデータ型がサポートされていないため、指定されたタグの読み取り要求は失敗しました。

解決策:

テクニカルサポートまでご連絡の上、この型に関するサポートの追加をご要望ください。

● 関連項目:

アトミックデータ型のアドレス指定

ブロックを読み取れません。コントローラタグのデータ型が不明です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

ブロック内のコントローラタグのデータ型がサポートされていないため、指定されたブロックの読み取り要求は失敗しました。

解決策:

テクニカルサポートまでご連絡の上、この型に関するサポートの追加をご要望ください。

● 関連項目:

アトミックデータ型のアドレス指定

タグに書き込めません。データ型がサポートされていません。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

クライアントタグのデータ型がサポートされていないため、指定されたタグの書き込み要求は失敗しました。

解決策:

タグのデータ型をサポート対象の型に変更してください。

● 関連項目:

アトミックデータ型のアドレス指定

タグを読み取れません。データ型がサポートされていません。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

コントローラタグのデータ型がサポートされていないため、指定されたタグの読み取り要求は失敗しました。

解決策:

テクニカルサポートまでご連絡の上、この型に関するサポートの追加をご要望ください。

● 関連項目:

アトミックデータ型のアドレス指定

ブロックを読み取れません。データ型がサポートされていません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

ブロック内のコントローラタグのデータ型がサポートされていないため、指定されたブロックの読み取り要求は失敗しました。

解決策:

テクニカルサポートまでご連絡の上、この型に関するサポートの追加をご要望ください。

● **関連項目:**

アトミックデータ型のアドレス指定

タグに書き込めません。このタグには不正なデータ型です。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

クライアントタグのデータ型が示されたコントローラタグに対して不正であるため、指定されたタグの書き込み要求は失敗しました。

解決策:

タグのデータ型をサポート対象の型に変更してください。たとえば、BOOL 配列のコントローラタグにデータ型 Short は不正です。その場合、データ型を Boolean に変更すると問題は解決します。

● **関連項目:**

アトミックデータ型のアドレス指定

タグを読み取れません。このタグには不正なデータ型です。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

クライアントタグのデータ型が示されたコントローラタグに対して不正であるため、指定されたタグの読み取り要求は失敗しました。

解決策:

タグのデータ型をサポート対象の型に変更してください。たとえば、BOOL 配列のコントローラタグにデータ型 Short は不正です。その場合、データ型を Boolean に変更すると問題を解決できます。

● **関連項目:**

アトミックデータ型のアドレス指定

ブロックを読み取れません。このブロックには不正なデータ型です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

クライアントタグのデータ型が示されたコントローラタグに対して不正であるため、ブロックの読み取り要求は失敗しました。

解決策:

このブロック内のタグのデータ型をサポート対象の型に変更してください。たとえば、BOOL 配列のコントローラタグにデータ型 Short は不正です。その場合、データ型を Boolean に変更すると問題は解決します。

● 関連項目:

アトミックデータ型のアドレス指定

タグに書き込めません。タグは複数要素の配列をサポートしません。| 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:

複数要素の配列から示されたコントローラタグへのアクセスをドライバーがサポートしないため、指定されたタグの読み取り要求は失敗しました。

解決策:

タグのデータ型またはアドレスをサポート対象のものに変更してください。

● 関連項目:

アトミックデータ型のアドレス指定

タグを読み取れません。タグは複数要素の配列をサポートしません。タグは非アクティブ化されました。| 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:

複数要素の配列から示されたコントローラタグへのアクセスをドライバーがサポートしないため、指定されたタグの読み取り要求は失敗しました。

解決策:

タグのデータ型またはアドレスをサポート対象のものに変更してください。このエラーによりタグは非アクティブ化され、再度処理されることはありません。

● 関連項目:

アトミックデータ型のアドレス指定

ブロックを読み取れません。ブロックは複数要素の配列をサポートしません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

複数要素の配列から示されたコントローラタグへのアクセスをドライバーがサポートしないため、このブロック内のタグの読み取り要求は失敗しました。

解決策:

このブロック内のタグのデータ型またはアドレスをサポートされているものに変更してください。このエラーによりブロックの <count> 個の要素は非アクティブ化され、再度処理されることはありません。

● 関連項目:

アトミックデータ型のアドレス指定

タグに書き込めません。ネイティブタグのサイズが不一致です。| 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:

ネイティブタグのサイズ (フットプリント) が、プロジェクトのアップロードによって決まった予想されるサイズと一致しません。

解決策:

1. プロトコルモードをシンボリックモードに変更してから、もう一度試してください。
2. テクニカルサポートまでご連絡の上、この問題についてご報告ください。

タグを読み取れません。ネイティブタグのサイズが不一致です。| 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:

ネイティブタグのサイズ (フットプリント) が、プロジェクトのアップロードによって決まった予想されるサイズと一致しません。

解決策:

1. プロトコルモードをシンボリックモードに変更してから、もう一度試してください。
2. テクニカルサポートまでご連絡の上、この問題についてご報告ください。

ブロックを読み取れません。ネイティブタグのサイズが一致しません。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

ネイティブタグのブロックのサイズ (フットプリント) が、プロジェクトのアップロードによって決まった予想されるサイズと一致しません。

解決策:

1. プロトコルモードをシンボリックモードに変更してから、もう一度試してください。
2. テクニカルサポートまでご連絡の上、この問題についてご報告ください。

ブロックを読み取れません。ネイティブタグのサイズが一致しません。| ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。

エラータイプ:

警告

考えられる原因:

ネイティブタグのブロックのサイズ (フットプリント) が、プロジェクトのアップロードによって決まった予想されるサイズと一致しません。

解決策:

1. プロトコルモードをシンボリックモードに変更してから、もう一度試してください。
2. テクニカルサポートまでご連絡の上、この問題についてご報告ください。

タグに書き込めません。| 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:

1. デバイスとホスト PC 間のイーサネット接続が切断しています。
2. イーサネット接続の通信パラメータが不正です。
3. この名前のデバイスに不正な IP アドレスが割り当てられている可能性があります。

解決策:

1. PC とデバイス間のケーブル接続を確認してください。
2. この名前のデバイスに正しいポートが指定されていることを確認してください。
3. この名前のデバイスに指定した IP アドレスが実際のデバイスのアドレスと一致することを確認してください。

タグを読み取れません。タグは非アクティブ化されました。| 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:

1. デバイスとホスト PC 間のイーサネット接続が切断しています。
2. イーサネット接続の通信パラメータが不正です。
3. この名前のデバイスに不正な IP アドレスが割り当てられている可能性があります。

解決策:

1. PC とデバイス間のケーブル接続を確認してください。
2. この名前のデバイスに正しいポートが指定されていることを確認してください。
3. この名前のデバイスに指定した IP アドレスが実際のデバイスのアドレスと一致することを確認してください。

● 注記:

このエラーによりタグは非アクティブ化され、再度処理されることはありません。

ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

1. デバイスとホスト PC 間のイーサネット接続が切断しています。
2. イーサネット接続の通信パラメータが不正です。

3. この名前のデバイスに不正な IP アドレスが割り当てられている可能性があります。

解決策:

1. PC とデバイス間のケーブル接続を確認してください。
2. この名前のデバイスに正しいポートが指定されていることを確認してください。
3. この名前のデバイスに指定した IP アドレスが実際のデバイスのアドレスと一致することを確認してください。

● 注記:

このエラーによりブロックの要素は非アクティブ化され、再度処理されることはありません。

ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>'。

エラータイプ:

警告

考えられる原因:

1. デバイスとホスト PC 間のイーサネット接続が切断しています。
2. イーサネット接続の通信パラメータが不正です。
3. この名前のデバイスに不正な IP アドレスが割り当てられている可能性があります。

解決策:

1. PC とデバイス間のケーブル接続を確認してください。
2. この名前のデバイスに正しいポートが指定されていることを確認してください。
3. この名前のデバイスに指定した IP アドレスが実際のデバイスのアドレスと一致することを確認してください。

● 注記:

このエラーによりブロックの要素は非アクティブ化され、再度処理されることはありません。

デバイスへの要求中にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

警告

考えられる原因:

要求時に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。要求内のすべての読み取りと書き込みが失敗しました。

解決策:

返されたエラーコードによって解決策が異なります。

● 関連項目:

CIP エラーコード

デバイスへの要求中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。

エラータイプ:

警告

考えられる原因:

要求時に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。要求内のすべての読み取りと書き込みが失敗しました。

解決策:

このようなエラーからの回復はドライバーが自動的に試みます。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。エラー 0x02 はドライバー関連ではなくデバイス関連なので除外されます。

● 関連項目:

カプセル化エラーコード

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。

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。

ブロックを読み取れません。受信したフレームにエラーが含まれています。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'。**エラータイプ:**

警告

考えられる原因:

1. 不正な要求サービスコードがあります。
2. ドライバーは予想されるバイト数よりも多いかまたは少ないバイト数を受信しました。

解決策:

1. ドライバーがこのエラーから回復できるようにするには、再試行回数を増やしてください。
2. このエラーが頻繁に発生する場合、ケーブル接続またはデバイス自体に問題がある可能性があります。特定のタグでこのエラーが頻繁に発生する場合、テクニカルサポートまでご連絡ください。

デバイスからファンクションファイルを読み取れません。受信したフレームにエラーが含まれています。| ファンクションファイル = '<名前>'。**エラータイプ:**

警告

ブロックを読み取れません。タグは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。**エラータイプ:**

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

デバイスからファンクションファイルを読み取れません。タグは非アクティブ化されました。| ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

アドレスに書き込めません。受信したフレームにエラーが含まれています。| アドレス = '<アドレス>'。

エラータイプ:

警告

ファンクションファイルに書き込めません。受信したフレームにエラーが含まれています。| ファンクションファイル = '<名前>'。

エラータイプ:

警告

ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

アドレスが PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルを読み取れません。| ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

ブロックを読み取れません。タグは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルを読み取れません。タグは非アクティブ化されました。| ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● 注記:

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

アドレスに書き込みません。 | アドレス = '<アドレス>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● 注記:

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルに書き込みません。 | ファンクションファイル = '<名前>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● 注記:

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>', DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルを読み取れません。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

アドレスに書き込めません。| アドレス = '<アドレス>', DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● **注記:**

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルに書き込めません。 | ファンクションファイル = '<名前>'、DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このアドレスは PLC に存在しません。

解決策:

PLC から返されたステータスコードと拡張ステータスコードを確認してください。拡張ステータスコードは必ず返るわけではなく、エラー情報はステータスコードに含まれています。これらのコードは 16 進数で表示されます。

● 注記:

ステータスコードの下位ニブルのステータスコードエラーは、ローカルノードによって検出されたエラーを示します。ローカルノードによって検出されたエラーは、KF モジュールが何らかの理由によってネットワーク上で宛先 PLC を見つけられない場合に発生します。ステータスコードの上位ニブルのステータスコードエラーは、PLC によって検出されたエラーを示します。これらのエラーは、データ位置が PLC で使用できないか書き込み不能の場合に生成されます。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

タグを読み取れません。内部メモリが無効です。 | タグアドレス = '<アドレス>'。

エラータイプ:

警告

タグを読み取れません。このタグには不正なデータ型です。 | タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。

エラータイプ:

警告

考えられる原因:

クライアントタグのデータ型が示されたコントローラタグに対して不正であるため、指定されたタグの読み取り要求は失敗しました。

解決策:

タグのデータ型をサポート対象の型に変更してください。たとえば、BOOL 配列のコントローラタグにデータ型 Short は不正です。その場合、データ型を Boolean に変更すると問題を解決できます。

● 関連項目:

アトミックデータ型のアドレス指定

ブロックを読み取れません。内部メモリが無効です。タグは非アクティブ化されました。 | タグアドレス = '<アドレス>'。

エラータイプ:

警告

ブロックを読み取れません。内部メモリが無効です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。

エラータイプ:

警告

アドレスに書き込めません。内部メモリが無効です。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>', CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

警告

考えられる原因:

指定されたブロックの読み取り要求時に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。

解決策:

返されたエラーコードによって解決策が異なります。

● 関連項目:

CIP エラーコード

デバイスが応答していません。ローカルノードがエラーを返しました。| DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

PLC はローカルノードからの要求に応答しませんでした。ローカルノードが 1756-DHRIO、1756-CNB、1761-NET-ENI などの中間ノードである可能性があります。

解決策:

エラーコード定義に関する Allen-Bradley ドキュメントを参照してください。たとえば、STS コード '0x02'(hex) が返された場合、リモートノード (PLC) とローカルノード間のケーブル接続を確認してください。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

ファンクションファイルに書き込めません。ローカルノードがエラーを返しました。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このエラーは、PLC がローカルノードからの書き込み要求に応答しなかったことを意味します。ローカルノードが 1756-DHRIO、1756-CNB、1761-NET-ENI などの中間ノードである可能性があります。

解決策:

STS エラーコード定義に関する Allen-Bradley ドキュメントを参照してください。たとえば、STS コード '0x02'(hex) が返された場合、リモートノード (PLC) とローカルノード間のケーブル接続を確認してください。

● 関連項目:

エラーコード定義に関する Allen-Bradley ドキュメント

アドレスに書き込めません。ローカルノードがエラーを返しました。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。

エラータイプ:

警告

考えられる原因:

このエラーは、PLC がローカルノードからの書き込み要求に応答しなかったことを意味します。ローカルノードが 1756-DHRIO、1756-CNB、1761-NET-ENI などの中間ノードである可能性があります。

解決策:

STS エラーコード定義に関する Allen-Bradley ドキュメントを参照してください。たとえば、STS コード '0x02'(hex) が返された場合、リモートノード (PLC) とローカルノード間のケーブル接続を確認してください。

● **関連項目:**

エラーコード定義に関する Allen-Bradley ドキュメント

タグで予期しないオフセットが見つかりました。タグはシンボリックプロトコルを使用します。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

タグで予期しないオフセットが見つかりました。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

タグで予期しないオフセット/スパンが見つかりました。| タグアドレス = '<アドレス>'。

エラータイプ:

警告

プロジェクトのダウンロードが進行中であるかプロジェクトが存在しません。

エラータイプ:

警告

プロジェクトのダウンロードが完了しました。

エラータイプ:

警告

プロジェクトのオンライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。

エラータイプ:

警告

プロジェクトのオフライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。

エラータイプ:

警告

デバイスからコントローラプロジェクトをアップロード中に次のエラーが発生しました。シンボリックプロトコルを使用します。

エラータイプ:

警告

デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。| カプセル化エラー = <コード>。

エラータイプ:

警告

考えられる原因:

要求時に Ethernet/IP パケットのカプセル化部分の範囲でデバイスがエラーを返しました。問題が解決するまで、論理モードに設定されているデバイスはシンボリックモードに戻ります。

解決策:

このようなエラーからの回復はドライバーが自動的に試みます。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。エラー 0x02 はドライバー関連ではなくデバイス関連なので除外されます。

● 関連項目:

カプセル化エラーコード

デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。| CIP エラー = <コード>、拡張エラー = <コード>。

エラータイプ:

警告

考えられる原因:

要求時に Ethernet/IP パケットの CIP 部分の範囲でデバイスがエラーを返しました。問題が解決するまで、論理モードに設定されているデバイスはシンボリックモードに戻ります。

解決策:

返されたエラーコードによって解決策が異なります。問題が引き続き発生する場合、テクニカルサポートまでご連絡ください。

● 関連項目:

CIP エラーコード

デバイスの識別情報を取得できません。受信したフレームにエラーが含まれています。すべてのタグがシンボリックプロトコルを使用します。

エラータイプ:

警告

考えられる原因:

1. PC とデバイス間の接続/切断によってパケットに不整列が発生しています。
2. デバイス間のケーブル接続の不良によりノイズが発生しています。
3. 不正なフレームサイズを受信しました。
4. TNS の不一致があります。
5. デバイスから無効な応答コマンドが返されました。
6. このデバイスでは Ethernet/IP が有効になっていません。

解決策:

1. 介入しなくてもドライバーはこのエラーから回復します。このエラーが頻繁に発生する場合、ケーブル接続、ネットワーク、またはデバイス自体に問題がある可能性があります。
2. 通信先のデバイスがイーサネット対応デバイスであることを確認してください。

要求された CIP 接続サイズはこのデバイスによってサポートされていません。自動的に最大サイズにフォールバックします。| 要求されたサイズ = <数値> (バイト)、最大サイズ = <数値> (バイト)。

エラータイプ:

警告

考えられる原因:

要求された CIP 接続サイズはこのデバイスによってサポートされていません。

解決策:

このデバイスによってサポートされているサイズに CIP 接続サイズを変更してください。

● 関連項目:

Logix 通信パラメータ

タグのインポートファイル名が無効です。ファイルパスは使用できません。

エラータイプ:

警告

考えられる原因:

タグのインポートファイル名にはパスが含まれます。

解決策:

ファイル名からパスを除去します。

デバイスへの読み取り/書き込み要求が中止しました。デバイスプロジェクトからの論理アドレスを更新しています。

エラータイプ:

警告

デバイスへの読み取り/書き込み要求が再開しました。デバイスからの論理アドレスの更新が完了しました。現在、論理アドレス指定を使用しています。

エラータイプ:

警告

タグに書き込めません。書き込まれた値には構文エラーが含まれています。| タグアドレス = '<アドレス>'、必要なフォーマット = '<フォーマット>'。

エラータイプ:

警告

考えられる原因:

書き込まれた文字列の値は、必要な '<format>' に準拠していません。

解決策:

1. 必要なフォーマットに準拠している文字列の値を、余分な文字や埋め込みの空白なしで書き込んでください。
2. 時間タイプでは、ドライバーは Studio 5000 によってサポートされているものと同じリテラル文字列をサポートしているため、参照として使用できます。たとえば、すべての時間の部分はありませんが、少なくとも 1 つの部分が必要です: T32#0us は使用できますが、T32# は使用できません。

タグに書き込めません。書き込まれた値は範囲外です。| 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:

1. 書き込まれた文字列の値は、コントローラタグのデータ型で許可されている最小値または最大値を超えます。
2. 書き込まれた整数値は、コントローラタグのデータ型で許可されている最小値または最大値を超えています。

解決策:

コントローラタグのデータ型の範囲を超えていない文字列の値または整数値を書き込みます。たとえば、TIME 整数最大値は 2725199999999 です。これは、文字列リテラル 'T#31d_12h_59m_59s_999ms_999us' と関連付けられています; 'T#31d_13h' は範囲外になります。

データベースステータス。非エイリアスタグをインポートしています。

エラータイプ:

情報

データベースステータス。エイリアスタグをインポートしています。

エラータイプ:

情報

データベースステータス。タグプロジェクトを構築しています。お待ちください。| タグプロジェクト数 = <数値>。

エラータイプ:

情報

データベースエラー。最大文字長さを超えているため、タグ名が変更されました。| タグ名 = '<タグ>'、最大長さ = <数値>、新しいタグ名 = '<タグ>'。

エラータイプ:

情報

データベースエラー。最大文字長さを超えているため、配列タグの名前が変更されました。| 配列タグ = '<タグ>'、最大長さ = <数値>、新しい配列タグ = '<tags>'。

エラータイプ:

情報

データベースエラー。プログラムグループの名前が最大文字長さを超えています。プログラムグループの名前が変更されました。| グループ名 = '<名前>'、最大長さ = <数値>、新しいグループ名 = '<名前>'。

エラータイプ:

情報

データベースステータス。コントローラプロジェクトを読み込んでいます。

エラータイプ:
情報

データベースステータス。| プログラムの数 = <数値>、データ型の数 = <数値>、インポートされたタグの数 = <数値>。

エラータイプ:
情報

データベースステータス。OPC タグを生成しています。

エラータイプ:
情報

メモリリソース量が低下しています。

エラータイプ:
情報

不明なエラーが発生しました。

エラータイプ:
情報

データベースステータス。.L5X ファイルからタグをインポートしています。| スキーマリビジョン = '<値>'、ソフトウェアリビジョン = '<値>'。

エラータイプ:
情報

詳細。| IP = '<アドレス>'、ベンダー ID = <ベンダー>、製品タイプ = <タイプ>、製品コード = <コード>、リビジョン = '<値>'、製品名 = '<名前>'、製品シリアル番号 = <数値>。

エラータイプ:
情報

経過時間 = <数値> (秒)。

エラータイプ:
情報

シンボリックデバイスの読み取り回数 = <数値>。

エラータイプ:
情報

シンボリック配列ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:
情報

シンボリック配列ブロックキャッシュの読み取り回数 = <数値>。

エラータイプ:
情報

シンボリックインスタンス非ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

シンボリックインスタンス非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

シンボリックインスタンス非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。

エラータイプ:

情報

シンボリックインスタンスブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

シンボリックインスタンスブロックキャッシュの読み取り回数 = <数値>。

エラータイプ:

情報

物理非ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

物理非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

物理非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。

エラータイプ:

情報

物理ブロックデバイスの読み取り回数 = <数値>。

エラータイプ:

情報

物理ブロックキャッシュの読み取り回数 = <数値>。

エラータイプ:

情報

読み取りタグ数 = <数値>。

エラータイプ:

情報

送信 パケット数 = <数値>。

エラータイプ:

情報

受信 パケット数 = <数値>。

エラータイプ:

情報

初期化トランザクション数 = <数値>。

エラータイプ:
情報

読み取り書き込みトランザクション数 = <数値>。

エラータイプ:
情報

1 秒あたり平均送信パケット数 = <数値>。

エラータイプ:
情報

1 秒あたり平均受信パケット数 = <数値>。

エラータイプ:
情報

1 秒あたり平均タグ読み取り回数 = <数値>。

エラータイプ:
情報

1 トランザクションあたり平均タグ数 = <数値>。

エラータイプ:
情報

-

エラータイプ:
情報

%s | デバイス統計

エラータイプ:
情報

デバイス平均ターンアラウンドタイム = <数値> (ミリ秒)

エラータイプ:
情報

%s | チャネル統計

エラータイプ:
情報

ドライバー統計

エラータイプ:
情報

デバイスタグのインポートが中断しました。

エラータイプ:
情報

インポートファイル '%s' はパス '%s' に見つかりません。

エラータイプ:
情報

コントローラプロジェクトの読み込み中にエラーが発生しました。

エラータイプ:
情報

内部ドライバーエラーが発生しました。

エラータイプ:
情報

同期化中に無効または破損したコントローラプロジェクトが検出されました。後でもう一度試してください。

エラータイプ:
情報

同期化中にプロジェクトのダウンロードが検出されました。後でもう一度試してください。

エラータイプ:
情報

メモリリソース量が低下しています。

エラータイプ:
情報

L5K ファイルが無効であるか破損しています。

エラータイプ:
情報

不明なエラーが発生しました。

エラータイプ:
情報

データベースエラー。PLC5/SLC/MicroLogix デバイスはこの機能をサポートしていません。

エラータイプ:
情報

L5X ファイルが無効であるか破損しています。

エラータイプ:
情報

インポートファイル '<空>' はパス '<空>' に見つかりません。

エラータイプ:
情報

インポートファイル '%s' はパス '<空>' に見つかりません。

エラータイプ:
情報

インポートファイル '<空>' はパス '%s' に見つかりません。

エラータイプ:

情報

XML 要素がポストスキーマの検証に失敗しました。デバイスからのタグのインポートはこのモデルではサポートされていません。代替要素を使用してください。| XML 要素 = '{<名前空間><要素>', サポートしていないモデル = '<モデル>', 代替 XML 要素 = '{<名前空間><要素>'。

エラータイプ:

セキュリティ

この値はこのモデルの XML 要素ではサポートされていません。新しい値に自動的に設定します。| 値 = '<値>', XML 要素 = '{<名前空間><要素>', モデル = '<モデル>', 新しい値 = '<値>'。

エラータイプ:

セキュリティ

付録

特定のトピックの詳細については、以下のリストからリンクを選択してください。

[チャンネルプロパティ](#)
[デバイスのプロパティ](#)
[タグのプロパティ](#)

[Logix 設定](#)
[1761-NET-ENI の設定](#)
[Data Highway Plus ゲートウェイの設定](#)
[通信のルーティング](#)
[シリアルゲートウェイの設定](#)
[Data Highway Plus ゲートウェイ](#)
[ControlNet ゲートウェイ](#)
[EtherNet/IP ゲートウェイの設定](#)
[MicroLogix 1100 の設定](#)
[プロトコルモードの選択](#)
[コントローラプロジェクト内の変更の検出](#)
[SoftLogix 5800 の接続の注意事項](#)
[用語集](#)

Allen-Bradley ControlLogix Ethernet チャンネルのプロパティ

以下は Allen-Bradley ControlLogix Ethernet チャンネルレベルのプロパティのリストです。

```
{
"common.ALLTYPES_NAME": "MyChannel",
"common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
"servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false,
"servermain.CHANNEL_UNIQUE_ID": 4126021724,
"servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING": "",
"servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2,
"servermain.CHANNEL_WRITE_OPTIMIZATIONS_DUTY_CYCLE": 10,
"servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_HANDLING": 0,
}
```

Allen-Bradley ControlLogix Ethernet デバイスのプロパティ

以下は Allen-Bradley ControlLogix Ethernet デバイスレベルのプロパティのリストです。

```
{
"common.ALLTYPES_NAME": "MyDevice",
"common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Allen-Bradley ControlLogix Ethernet",
"servermain.DEVICE_MODEL": 0,
"servermain.DEVICE_UNIQUE_ID": 1286734974,
"servermain.DEVICE_CHANNEL_ASSIGNMENT": "Channel1",
"servermain.DEVICE_ID_FORMAT": 0,
"servermain.DEVICE_ID_STRING": "<10.10.110.15>,1,0",
"servermain.DEVICE_ID_HEXADECIMAL": 0,
"servermain.DEVICE_ID_DECIMAL": 0,
"servermain.DEVICE_ID_OCTAL": 0,
"servermain.DEVICE_DATA_COLLECTION": true,
"servermain.DEVICE_SIMULATED": false,
"servermain.DEVICE_SCAN_MODE": 0,
"servermain.DEVICE_SCAN_MODE_RATE_MS": 1000,
"servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE": false,
}
```



```
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0、  
0  
]  
}  
}
```

Allen-Bradley ControlLogix Ethernet タグのプロパティ

以下は Allen-Bradley ControlLogix Ethernet タグのプロパティのリストです。

```
{  
  "common.ALLTYPES_NAME": "MyTag1",  
  "common.ALLTYPES_DESCRIPTION": "",  
  "servermain.TAG_ADDRESS": "MyTag1",  
  "servermain.TAG_DATA_TYPE": 6,  
  "servermain.TAG_READ_WRITE_ACCESS": 1,  
  "servermain.TAG_SCAN_RATE_MILLISECONDS": 100,  
  "servermain.TAG_AUTOGENERATED": true,  
  "servermain.TAG_SCALING_TYPE": 0,  
  "servermain.TAG_SCALING_RAW_LOW": 0,  
  "servermain.TAG_SCALING_RAW_HIGH": 1000,  
  "servermain.TAG_SCALING_SCALED_DATA_TYPE": 9,  
  "servermain.TAG_SCALING_SCALED_LOW": 0,  
  "servermain.TAG_SCALING_SCALED_HIGH": 1000,  
  "servermain.TAG_SCALING_CLAMP_LOW": false,  
  "servermain.TAG_SCALING_CLAMP_HIGH": false,  
  "servermain.TAG_SCALING_NEGATE_VALUE": false,  
  "servermain.TAG_SCALING_UNITS": ""  
}
```

Logix デバイス ID

ENI デバイスの ID の設定については、[1761-NET-ENI の設定](#)を参照してください。

ControlLogix 5500 Ethernet

デバイス ID には、デバイスの IP アドレス、およびコントローラの CPU があるスロットの番号を指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,[<オプションのルーティングパス>],<CPU スロット>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	IP アドレスまたはホスト名。	10 進	0-255
1	ポート ID	バックプレーンへのポート。	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	EtherNet/IP インタフェースモジュールからの経路を指定し、これは 1 (バックプレーンへのポート) でなければなりません。	10 進	*
CPU スロット	リンクアドレス	ControlLogix プロセッサのスロット番号。	10 進	0-255

*詳細については、[接続パスの指定](#)を参照してください。

例

123.123.123.123,1,0

これは Ethernet/IP 123.123.123.123 に相当します。ポート ID は 1 で CPU はスロット 0 にあります。

CompactLogix 5300 イーサネットのデバイス ID

デバイス ID には、デバイスの IP アドレス、およびコントローラの CPU があるスロットの番号を指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,[<オプションのルーティングパス>],<CPU スロット>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	CompactLogix Ethernet の IP アドレスまたはホスト名。	10 進	0-255
1	ポート ID	バックプレーンへのポート。	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	イーサネットポートからの経路を指定し、これは 1 (バックプレーンへのポート) でなければなりません。	10 進	*
CPU スロット	リンクアドレス	CompactLogix プロセッサのスロット番号。	10 進	0-255

*詳細については、[接続パスの指定](#)を参照してください。

例

123.123.123.123,1,0

これは CompactLogix IP 123.123.123.123 に相当します。ポート ID は 1 で CPU はスロット 0 にあります。

FlexLogix 5400 Ethernet のデバイス ID

デバイス ID には、デバイスの IP アドレス、およびコントローラの CPU があるスロットの番号を指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,[<オプションのルーティングパス>],<CPU スロット>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	1788-ENBT の IP アドレスまたはホスト名。	10 進	0-255
1	ポート ID	バックプレーンへのポート。	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	1788-ENBT インタフェースモジュールからの経路を指定し、これは 1 (バックプレーンへのポート) でなければなりません。	10 進	*
CPU スロット	リンクアドレス	FlexLogix プロセッサのスロット番号。	10 進	0-255

*詳細については、[接続パスの指定](#)を参照してください。

例

123.123.123.123,1,0

これは 1788-ENBT IP 123.123.123.123 に相当します。ポート ID は 1 で CPU はスロット 0 にあります。

SoftLogix 5800 のデバイス ID

デバイス ID には、SoftLogix PC の IP アドレス、およびコントローラの CPU がある仮想スロットの番号を指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,<オプションのルーティングパス>,<CPU スロット>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	SoftLogix PC NIC の IP アドレスまたはホスト名。	10 進	0-255
1	ポート ID	バックプレーンへのポート。	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	EtherNet/IP メッセージングモジュールからの経路を指定し、これは 1 (仮想バックプレーンへのポート) でなければなりません。	10 進	*
CPU スロット	リンクアドレス	仮想バックプレーン内の SoftLogix プロセッサのスロット番号。	10 進	0-255

*詳細については、[接続パスの指定](#)を参照してください。

例

123.123.123.123,1,1

これは SoftLogix PC の IP アドレス 123.123.123.123 に相当します。ポート ID は 1 で CPU はスロット 1 にあります。

* デバイス ID にリモートバックプレーンへのルーティングパスを追加する方法については、[通信のルーティング](#)を参照してください。

* **関連項目:** [SoftLogix 5800 の接続の注意事項](#)

1761-NET-ENI の設定

1761-NET-ENI は Allen-Bradley ControlLogix Ethernet ドライバー を使用してイーサネット上の ControlLogix、CompactLogix、FlexLogix、MicroLogix、SLC 500、および PLC-5 シリーズ PLC と通信する手段を提供します。

要件

CH0 RS232 チャンネルを利用する全二重 DF1 をサポートしている MicroLogix、SLC 500、または PLC-5 シリーズ PLC。

1761-NET-ENI デバイスシリーズ A、B、C、または D。

CH0 RS232 チャンネルを利用する ControlLogix、CompactLogix、または FlexLogix PLC。
1761-NET-ENI デバイスシリーズ B 以降。

● 注記:

1. ENI ControlLogix、CompactLogix、および FlexLogix のユーザーは、通信パラメータ、データベース設定、プロジェクト/プロトコルオプションについて、目次から「Logix Setup」ブックを参照してください。
2. (このユーティリティの「ENI IP Addr」タブにある)「CompactLogix Routing」オプションを表示するには、Allen-Bradley から提供されている ENI / ENIW ユーティリティを使用します。これはファームウェアリビジョン 2.31 を使用する ENI モジュールでテストされました。

● ENI モジュールでは TCP 接続の数が制限されています。このため、ドライバーが接続を使用できるようにするため、ユーザーはモジュールと通信するアプリケーション (RSLinx/RWho など) の使用を避ける必要があります。

ENI デバイス ID

デバイス ID では 1761-NET-ENI の IP アドレスを指定します。デバイス ID は次のように指定します。

<IP アドレス>

指定子	指定子のタイプ	説明	フォーマット	範囲
IP アドレス	該当なし	1761-NET-ENI の IP アドレス	10 進	0-255

例

123.123.123.123

これは ENI IP 123.123.123.123 に相当します。このデバイスは全二重 DF1 のみをサポートするため、ノード ID は必要ありません。

● 通信パラメータの詳細については、[Logix 通信パラメータ](#)を参照してください。

Data Highway Plus ゲートウェイの設定

DH+ ゲートウェイは Allen-Bradley ControlLogix Ethernet ドライバーを使用して DH+ 上の SLC 500 および PLC-5 シリーズ PLC と通信する手段を提供します。

要件

EtherNet/IP インタフェースモジュール。

DH+ 用に適切なチャネルが設定されている 1756-DHRIO インタフェースモジュール。

DH+ ネットワーク上の SLC500 または PLC-5 シリーズ PLC。

● 注記: DH+ ゲートウェイモデルは自動タグデータベース生成をサポートしていません。

DH+ ゲートウェイのデバイス ID

デバイス ID では、接続を行うために必要な DH+ パラメータに加え、デバイスの IP アドレスを指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,[<オプションのルーティングパス>],<DHRIO スロット>.<DHRIO チャネル>.<DH+ ノード ID (10 進)>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	IP アドレスまたはホスト名	10 進	0-255
1	ポート ID	バックプレーンへのポート	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	EtherNet/IP インタフェースモジュールからの経路として 1 (バックプレーンへのポート) を指定します	10 進	*
DHRIO スロット	リンクアドレス	1756-DHRIO インタフェースモジュールのスロット番号	10 進	0-255
DHRIO チャネル		使用する DH+ チャネル	英字	A と B
DH+ ノード ID		ターゲット PLC の DH+ ノード ID (10 進フォーマット)**	10 進	0-

指定子	指定子のタイプ*	説明	フォーマット	範囲
				99

● *詳細については、[接続パスの指定](#)を参照してください。

● **詳細については、以下の「8進フォーマットでのノード ID のアドレス指定」を参照してください。

例

123.123.123.123,1,2,A,3

これは Ethernet/IP 123.123.123.123 に相当します。DH+ カードはスロット 2 にあります。DH+ チャネル A およびアドレス指定ターゲットとして DH+ ノード ID 3 (10 進) を使用します。

8 進フォーマットでのノード ID のアドレス指定

PLC では DH+ ノード ID が 8 進フォーマットで指定されているため、DH+ ゲートウェイデバイス ID で使用するためには 10 進フォーマットに変換する必要があります。ノード ID は RSLinx 内の RSWho にあります。これは 8 進フォーマットで表示されます。

例

RSWho での DH+ ノード 10 (8 進) = DH+ ゲートウェイデバイス ID での DH+ ノード 8 (10 進)。

適切なコントローラとの通信を確認することが重要です。上記の例では、DH+ ゲートウェイデバイス ID での DH+ ノード ID として 10 を入力した場合、ノード 10 (8 進) ではなくノード 12 (10 進の 10 に相当する 8 進数) との通信が確立されず、ノード 12 (8 進) が存在しない場合、DHRIO モジュールは DF1 STS 0x02 を返します。これはリンクレイヤーがパケットの送信を保証できないことを意味します。つまり、DH+ ネットワーク上で DH+ ノードが見つかりません。

● デバイス ID にリモート DH+ ノードへのルーティングパスを追加する方法については、[通信のルーティング](#)を参照してください。

● 通信パラメータの詳細については、[ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)を参照してください。

シリアルゲートウェイの設定

シリアルゲートウェイは Allen-Bradley ControlLogix Ethernet ドライバーを使用してシリアルネットワーク上の ControlLogix、CompactLogix、FlexLogix および SoftLogix PLC と通信する手段を提供します。

要件

EtherNet/IP インタフェースモジュール。

シリアルポートを備えたローカル CPU。

シリアルポートを備えたリモート ControlLogix、CompactLogix、FlexLogix、または SoftLogix CPU。

● 注記:

1. ローカル CPU とリモート CPU が同じシリアルネットワーク上に存在する必要があります。
2. シリアルゲートウェイモデルは自動タグデータベース生成をサポートしていません。

シリアルゲートウェイのデバイス ID

デバイス ID では、接続を行うために必要なリモートデバイスステーション ID に加え、ローカルデバイスの IP アドレスを指定します。デバイス ID は次のように指定します。

<IP またはホスト名>,1,<オプションのルーティングパス>,<CPU スロット>,<シリアルポートチャネル>,<ステーション ID (10 進)>

指定子	指定子のタイプ*	説明	フォーマット	範囲
IP/ホスト名	該当なし	IP アドレスまたはホスト名	10 進	0-255
1	ポート ID	バックプレーンへのポート	10 進	1
オプションのルーティングパス	リンクとポートの複数のペア	EtherNet/IP インタフェースモジュールからの経路として 1 (バックプレーンへのポート) を指定します	10 進	*

指定子	指定子のタイプ*	説明	フォーマット	範囲
CPU スロット	リンクアドレス	通通信に使用されるシリアルポートを備えた CPU モジュールのスロット番号	10 進	0-255
シリアルポートチャンネル		使用するシリアルポートチャンネル	英字	A と B
ステーション ID		ターゲット PLC の DF1 ステーション ID (10 進フォーマット)**	10 進	0-255

*詳細については、[接続パスの指定](#)を参照してください。

例

123.123.123.123,1,0.A.3

これは EtherNet/IP 123.123.123.123 に相当します。CPU カードはスロット 0 にあります。チャンネル A (シリアルポート) およびアドレス指定ターゲットとしてステーション ID 3 (10 進) を使用します。

注記:

1. デバイス ID にリモートシリアルノードへのルーティングパスを追加する方法については、[通信のルーティング](#)を参照してください。
2. 通信パラメータの詳細については、[Logix 通信パラメータ](#)を参照してください。
3. デバイス ID を設定する際には、RSLinx 経由で同じルートを使用してそのデバイスを検出可能であることを確認してください。

MicroLogix 1100 の設定

MicroLogix 1100 のデバイス ID

デバイス ID では MicroLogix 1100 の IP アドレスを指定します。デバイス ID は次のように指定します。

<IP またはホスト名>

指定子	指定子のタイプ	説明	フォーマット	範囲
IP/ホスト名	該当なし	IP アドレスまたはホスト名	10 進	0-255

例

123.123.123.123

これは IP 123.123.123.123 に相当します。

*通信パラメータの詳細については、[ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ](#)を参照してください。

通信のルーティング

ルーティングは各種ネットワークを介してリモートデバイスと通信するための手段を提供します。これはローカルデバイスとリモートデバイスとの間のブリッジと考えることができ、これらのデバイスは 2 つの異なるフィールドバスネットワーク上に存在していても構いません。リモート (宛先) バックプレーンにアクセスすることで、このバックプレーン上のサポートされているモジュールと直接通信することが可能になります。サポートされているモジュールは次のとおりです。

- ControlLogix アプリケーション用 ControlLogix 5500 プロセッサ。
- SoftLogix アプリケーション用 SoftLogix 5800 プロセッサ。
- DH+ ゲートウェイアプリケーション用 1756-DHRIO インタフェースモジュール。
- ControlNet ゲートウェイアプリケーション用 1756-CNB および 1756-CNBR インタフェースモジュール。

ルーティングパスは一連のバックプレーンホップであり、その最後のホップは宛先バックプレーンを指します。ホップごとに (Logix プロセッサではなく) Logix バックプレーンが必要です。個々のホップは次のいずれかのネットワークをそのメディアとして利用できます。

- ControlNet
- DH+
- TCP/IP (Ethernet/IP)

● **重要:** ENI モデルと MicroLogix 1100 モデルではルーティングはサポートされません。

接続パスの指定

デバイス ID ではルーティングパスが指定されています。非ルーティングアプリケーションと同様に、通信は PC 上の Allen-Bradley ControlLogix Ethernet ドライバー からローカルイーサネット モジュールに向けて発信されます。このローカルイーサネット モジュールに到達すると、非ルーティングアプリケーションと同様に、デバイス ID によってモジュールからバックプレーンへの発信経路が指定されます。ルーティングパスに従って宛先の Logix バックプレーンにメッセージが送信されます。デバイス ID では通信に使用するデバイスも指定されます (ControlLogix プロセッサ、SoftLogix プロセッサ、DH+ ノード、ControlNet ノードなど)。

ルーティングパスの指定は左括弧 ([) で始まり右括弧 (]) で終了します。パス自体は一連のポート/リンクアドレスのペアであり、RSLogix 5000 「Message Configuration」ダイアログでの通信パスの構文と同じです。

指定子のタイプ	説明	フォーマット	範囲
ポート ID	対象のインタフェースモジュールからの発信経路を指定します。*	10 進	0-65535
リンクアドレス	対応するポートがバックプレーンである場合、リンクアドレスは発信元のインタフェースモジュールの-slot 番号です。 対応するポートがインタフェースモジュールポートである場合、リンクアドレスには次のように宛先ノードを指定します。 - DH+/ControlNet: ノード ID - イーサネット/IP 通信モジュール: IP アドレス - SoftLogix イーサネット/IP モジュール: IP アドレス	10 進	0-255

*詳細については、以下の「ポート参照」を参照してください。

単一ホップ

IP アドレス, ポート ID 0, [リンクアドレス 0, ポート ID 1, リンクアドレス 1, ポート ID 2], リンクアドレス 2。

マルチホップ (N ホップ)

IP アドレス, ポート ID 0, [リンクアドレス 0, ポート ID 1, リンクアドレス 1, ポート ID 2, リンクアドレス 2, ... ポート ID (N+1), リンクアドレス (N+1), ポート ID (N+2)], リンクアドレス (N+2)。

注記:

1. パスの最後のポート ID (単一ホップではポート ID 2、マルチホップではポート ID (N+2)) は 1 (バックプレーンのポート) でなければなりません。
2. ポート ID 0 は 1 (バックプレーンのポート) でなければなりません。リンクアドレス 2 とリンクアドレス (N+2) はリモート Logix プロセッサ/1756-DHRIO モジュール/1756-CNB モジュールの-slot 番号です。

ポート参照

インタフェースモジュール	ポート 1	ポート 2	ポート 3
イーサネット/IP 通信モジュール	バックプレーン	イーサネットネットワーク	該当なし
SoftLogix イーサネット/IP メッセージングモジュール	仮想バックプレーン	イーサネットネットワーク	該当なし
1756-DHRIO	バックプレーン	チャネル A 上の DH+ ネットワーク	チャネル B 上の DH+ ネットワーク
1756-CNB	バックプレーン	ControlNet ネットワーク	該当なし

アプリケーションに関する注意事項

1. パス内で同じインタフェースモジュールチャンネルにメッセージを複数回ルーティングすることはできません。これを行った場合 CIP エラー 0x01 拡張エラー 0x100B が発生します。
2. 複数チャンネルインタフェースモジュールの場合、メッセージを先にバックプレーンに転送するかバックプレーンをまとめて回避するかにかかわらず、メッセージをあるモジュールにルーティングした後ただちに(異なるチャンネルを使用して)そのモジュールからルーティングすることはできません。前述のように、各ホップが ControlLogix バックプレーンを必要とするので、後者はサポートされていません。この例として、ある DH+ リンク (1756-DHRIO のチャンネル A など) から別の DH+ リンク (同じ 1756-DHRIO のチャンネル B) に 1 つの 1756-DHRIO インタフェースモジュールを経由して DH+ メッセージをルーティングすることが挙げられます。これは一般的にリモート DH+ メッセージングと呼ばれ、サポートされていません。

ルーティングの例

以下のルーティングの例には、ローカル 1756-ENBT の IP 以外のすべてのデバイス ID が記載されています。デバイス ID/ルーティングパスの図はローカル 1756-ENBT モジュールから見たものです。ホップの説明は以下の形式になっています。

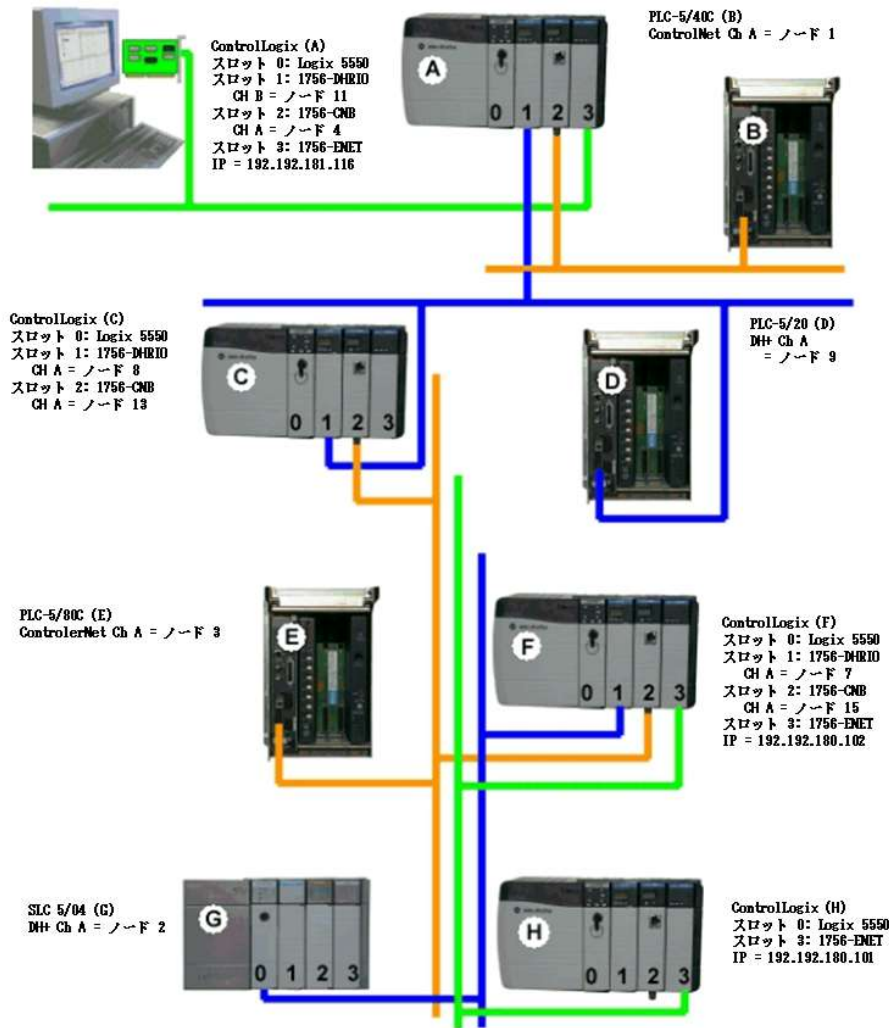
リンクアドレス (N), ポート ID (N+1), リンクアドレス (N+1), ポート ID (N+2)

● 詳細については、[接続パスの指定](#)を参照してください。接続/ルーティングパスの構築の詳細については、Allen-Bradley ドキュメント 1756-6.5.14, pp. 4-5 から 4-8 を参照してください。

以下の図では、DH+/ControlNet のノード ID はすべて 10 進フォーマットで示されています。PLC で指定するノード ID および RSWho で表示されるノード ID は 8 進フォーマットです。色の説明は次のとおりです。

- 緑色 = イーサネット
- 青色 = DH+
- オレンジ色 = ControlNet

● 詳細については、[Data Highway Plus ゲートウェイの設定](#)および [ControlNet ゲートウェイの設定](#)を参照してください。

**例 1**

DH+ ゲートウェイ経由での Logix5550 から PLC-5 へのルーティング。

宛先ノード	モデル	ルーティング	デバイスID (IPなし)
PLC-5/20 (D)	DH+ ゲートウェイ	いいえ	1,1.B.9

例 2

CN ゲートウェイ経由での Logix5550 から PLC-5C へのルーティング。

宛先ノード	モデル	ルーティング	デバイスID (IPなし)
PLC-5/40C (B)	CN ゲートウェイ	いいえ	1,2.A.1

例 3

DH+ を使用した Logix5550 から Logix5550 へのルーティング。

宛先ノード	モデル	ルーティング	デバイスID (IPなし)
Logix5550 (C)	ControlLogix 5550	はい	1,[1,2,8,1],0

例 3 でのルーティングパスの内訳。

ホップ	セグメント	説明
1	1,2,8,1	スロット 1 (DHRIO) -> ポート 2 (DH+ チャンネル A) -> DH+ ノード 8 -> Logix C バックプレーン

例 4

DH+ を使用した CN ゲートウェイ経由の Logix5550 から PLC-5C へのルーティング。

宛先ノード	モデル	ルーティング	デバイス ID (IP なし)
PLC-5/80C (E)	CN ゲートウェイ	はい	1,[1,2,8,1],2.A.3

例 4 でのルーティングパスの内訳。

ホップ	セグメント	説明
1	1,2,8,1	スロット 1 (DHRIO) -> ポート 2 (DH+ チャンネル A) -> DH+ ノード 8 -> Logix C バックプレーン

例 5

DH+ と ControlNet を使用した Logix5550 から Logix5550 へのルーティング

宛先ノード	モデル	ルーティング	デバイス ID (IP なし)
Logix5550 (F)	ControlLogix 5550	はい	1,[1,2,8,1,2,2,15,1],0

例 5 でのルーティングパスの内訳。

ホップ	セグメント	説明
1	1,2,8,1	スロット 1 (DHRIO) -> ポート 2 (DH+ チャンネル A) -> DH+ ノード 8 -> Logix C バックプレーン
2	2,2,15,1	スロット 2 (CNB) -> ポート 2 (CN チャンネル A) -> CN ノード 15 -> Logix F バックプレーン

例 6

DH+ と ControlNet を使用した Logix5550 から SLC 5/04 へのルーティング。

宛先ノード	モデル	ルーティング	デバイス ID (IP なし)
SLC 5/04 (G)	DH+ ゲートウェイ	はい	1,[1,2,8,1,2,2,15,1],1.A.2

例 6 でのルーティングパスの内訳。

ホップ	セグメント	説明
1	1,2,8,1	スロット 1 (DHRIO) -> ポート 2 (DH+ チャンネル A) -> DH+ ノード 8 -> Logix C バックプレーン
2	2,2,15,1	スロット 2 (CNB) -> ポート 2 (CN チャンネル A) -> CN ノード 15 -> Logix F バックプレーン

例 7

DH+, ControlNet, イーサネットを使用した Logix5550 から Logix5550 へのルーティング。

宛先ノード	モデル	ルーティング	デバイス ID (IP なし)
Logix5550 (H)	ControlLogix 5550	はい	1,[1,2,8,1,2,2,15,1,3,2,192.192.180.101,1],0

例 7 でのルーティングパスの内訳。

ホップ	セグメント	説明
1	1,2,8,1	スロット 1 (DHRIO) -> ポート 2 (DH+ チャネル A) -> DH+ ノード 8 -> Logix C バックプレーン
2	2,2,15,1	スロット 2 (CNB) -> ポート 2 (CN チャネル A) -> CN ノード 15 -> Logix F バックプレーン
3	3,2,192.192.180.101,1	スロット 3 (ENBT) -> ポート 2 -> リモート 1756-ENBT IP -> Logix H バックプレーン

プロトコルモードの選択

シンボリックモード

シンボリックモードはパケット内の各クライアント/サーバータグアドレスであり、その ASCII 文字名によって表されます。

メリット	デメリット
<ol style="list-style-type: none"> データ要求を行うために必要なすべての情報は、クライアント/サーバータグのアドレスにあります。 PLC から要求されるデータはクライアント/サーバータグでアクセスされているデータのみです。 下位互換性があります。 	<ol style="list-style-type: none"> シンボリックアドレスの処理時に、デバイスの応答時間が長くなります。 各要求のサイズが変化するので、複数要求パケットにつき要求の数が少なくなります。

注記:

- 複数要求パケット最適化を利用するには、できるだけ多くのタグを1つのパケットで表す必要があります。タグアドレスはパケット内の ASCII 文字の名前によって表されるので、タグのアドレスはできるだけ短くしてください。たとえば、"mytag" は "MyVeryLongTagNameThatContains36Chars" よりも優先されます。
- デフォルトのデータ型プロパティが "default" に設定されている場合、自動タグ生成によって、コントローラ内のタイプと一致するデータ型を持つタグが作成されます。

論理モード

論理非ブロックと論理ブロックは2つの読み取りプロトコルをカプセル化します。使用されるプロトコルはドライバーによって自動的に決定され、コントローラのリビジョンに基づいています。次の表は、これらのモードとそれらがマッピングするプロトコルの概要を示しています。

プロトコルモード	使用されている読み取りプロトコル		使用される書き込みプロトコル
	FRN V21 以上	FRN v20 以下と下側	すべての FRN
シンボリック	シンボリック (非ブロック)	シンボリック (非ブロック)	シンボリック
論理非ブロック	シンボルインスタンスの非ブロック	物理非ブロック*	シンボルインスタンス
論理ブロック	シンボルインスタンスのブロック	物理ブロック*	シンボルインスタンス

* V21 で廃止されました。

論理読み取りを実行するために必要な情報は、ドライバーによって自動的に実行されるコントローラプロジェクトのアップロードシーケンスで取得されます。簡潔にするため、「論理アドレス」という用語は、使用されているプロトコルに応じて、シンボルインスタンス ID または物理アドレスを表します。論理モードでは、すべてのシンボリックリクエストに必要な時間がかかるアドレスの解析とルックアップが回避されます。

● **注記:** これらの論理モードは、シリアルゲートウェイモデルでは使用できません。

論理非ブロックモード

論理非ブロックモードでは、すべてのクライアント/サーバータグが個別に、固定サイズで要求されます。

メリット	デメリット
<ol style="list-style-type: none"> 各要求のサイズが固定されるので、複数要求パケットあたりの要求の数を最大にできます。 パケットのクライアント/サーバータグが論理アドレスで指定されるので、デバイスの応答時間が短くな 	論理アドレスを特定するためにプロジェクトをアップロードする際の初期化でオーバーヘッドが生じます。

メリット	デメリット
<p>ります。</p> <p>3. PLC から要求されるデータはクライアント/サーバータグでアクセスされているデータのみです。</p>	

● **注記:** このモードは、構造体タグメンバーの少数部分がクライアント/サーバーによって参照されている場合に推奨されません。

論理ブロックモード

論理ブロックでは、1つのクライアント/サーバータグのみによって開始される1つの要求で、Logix タグのすべてのデータが取得されます。データブロックが受信されると、ドライバー内のキャッシュに格納され、タイムスタンプが付きまます。指定された Logix タグに属する連続するクライアント/サーバータグは、このキャッシュからデータを取得します。すべてのタグが更新されると、キャッシュが古くなったことを示す新しいリクエストが開始されます。現在時刻 > キャッシュのタイムスタンプ + タグスキャン速度になっている場合、キャッシュは古くなっています。このケースが含まれている場合、デバイスに対して別のブロックリクエストが作成され、キャッシュが更新され、サイクルが繰り返されます。

メリット	デメリット
<p>1. コンテンツは読み取りのたびに読み込まれます。</p> <p>2. パケットのクライアント/サーバータグが論理アドレスで指定されるので、デバイスの応答時間が短くなります。</p> <p>3. 各要求のサイズが固定されるので、複数要求パケットあたりの要求の数を最大にできます。</p>	<p>1. 論理アドレスを特定するためにプロジェクトをアップロードする際の初期化でオーバーヘッドが生じます。</p> <p>2. 一部の Logix タグが参照されている場合、論理非ブロックモードよりも低速になります (クライアント/サーバータグで参照されているより多くのデータが PLC からアクセスされているため)。</p>

● **注記:** このモードは、構造体タグメンバーの大部分がクライアント/サーバーによって参照されている場合に推奨されます。

● **関連項目:** [パフォーマンス統計とチューニング](#)

シンボルインスタンス vs 物理プロトコル

シンボルインスタンスの読み取りは、CIP インスタンス ID を使用して読み取り要求でネイティブタグを指定する CIP 要求です。非ブロックモードでは、構造体メンバーと配列要素へのパスを完全に修飾するために、CIP メンバー ID が必要になる場合があります。たとえば、CIP インスタンス id は構造体を表し、CIP メンバー id は構造内のメンバーを表します。クライアント/サーバータグを完全に限定するには、CIP メンバー id を追加する必要があるため、要求のサイズが異なる場合があります。構造のネスタリングが深いほど、タグを指定するために必要な CIP メンバー ID の数が少なくなり、1つのパケットに適合するリクエストの数が少なくなります。シンボルインスタンスの読み取りは FRN V21 で導入されました。

物理読み取りは、DMA アドレスが読み取り要求でネイティブタグを指定するために使用される CIP 要求です。非ブロックモードでは、構造メンバーと配列要素へのパスを完全に修飾するためにバイトオフセットが必要になる場合があります。たとえば、開始 DMA アドレスは構造を表し、バイトオフセットは構造内のメンバーを表します。最終的に開始 + オフセットは、リクエストで指定されている DMA アドレスです。すべてのリクエストはサイズに固定されています (シンボルインスタンスの読み取りとは異なります)。深さの構造がどのようにネストされている場合でも、その要求はパケット内で同じサイズになります。FRN V21 の場合、物理読み取りは廃止されました。

コントローラプロジェクト内の変更の検出

プロジェクトの Allen-Bradley ControlLogix Ethernet ドライバー モニターが変更され、進行中のダウンロード、オンラインでの編集、オフライン編集が検出されます。プロトコルが Logical に設定されている場合、ドライバーのプロジェクトイメージをコントローラプロジェクトのイメージと同期化するオプションがユーザーに与えられます。同期化によって、読み取りと書き込みの実行時に、ドライバーは各ネイティブタグの現在の論理アドレスを使用するようになります。

- **ダウンロード中:** ドライバーは、すべてのリクエストでオンライン編集とオフライン編集の両方を監視します。ネイティブタグの読み取りまたは書き込み中にダウンロードが発生したかどうかを検出し、そのモードに応じてプロジェクトの変更手順に従います。この同期化を有効にするには、デバイスを右クリックし、「プロパティ...」を選択します。「Logix オプション」グループで、「オンライン編集後に同期化」または「オフライン編集後に同期化」を見つけ、「はい」を選択します。

- ・「**オンライン編集後に同期化**」: ドライバーは、すべてのリクエストでオンライン編集を監視します。この関数は、次の読み取りまたは書き込み操作で、コントローラでオンライン編集が発生したかどうかを検出し、そのモードに応じてプロジェクトの変更手順に従います。この同期化を有効にするには、デバイスを右クリックし、「プロパティ...」を選択します。「Logix オプション」グループで、「**オンライン編集後に同期化**」を見つけ、「はい」を選択します。
- ・「**オフライン編集後に同期化**」: ドライバーは、すべてのリクエストでオフライン編集を監視します。次の読み取りまたは書き込み操作でコントローラを使用してオフライン編集が発生したかどうかを検出され、そのモードに応じてプロジェクト変更手順に従います。この同期化を有効にするには、デバイスを右クリックし、「プロパティ...」を選択します。「Logix オプション」グループで、「**オフライン編集後に同期化**」を見つけ、「はい」を選択します。

プロジェクトの変更手順 (シンボリックモード)

1. プロジェクトの変更が検出されます。
2. 変更が検出されたことを示すメッセージがイベントログに記録されます。
3. プロジェクトの変更中にダウンロードが発生した場合は以下のように動作します。
 - ・ 進行中のすべての読み取りと書き込みが停止して失敗します。
 - ・ プロジェクトの変更の完了をモニターするため、コントローラは 2 秒ごとにポーリングされます。
 - ・ プロジェクトの変更は検出されなくなります。
 - ・ 変更が検出されなくなったことを示すメッセージがイベントログに記録されます。
4. プロジェクトの変更中にオンライン編集およびオフライン編集が発生した場合は以下のように動作します。
 - ・ 応答データは無視されます。
 - ・ 進行中のすべての読み取りと書き込みが再試行されます。
5. 読み取りと書き込みがシンボリックモードで再開されます。

プロジェクトの変更手順 (論理モード)

1. プロジェクトの変更が検出されます。
2. 変更が検出されたことを示すメッセージがイベントログに記録されます。
3. プロジェクトの変更中にダウンロードが発生した場合は以下のように動作します。
 - ・ 進行中のすべての読み取りと書き込みが停止して失敗します。
 - ・ プロジェクトの変更の完了をモニターするため、コントローラは 2 秒ごとにポーリングされます。
 - ・ プロジェクトの変更は検出されなくなります。
 - ・ 変更が検出されなくなったことを示すメッセージがイベントログに記録されます。
4. プロジェクトの変更中にオンライン編集およびオフライン編集が発生した場合は以下のように動作します。
 - ・ 応答データは無視されます。
 - ・ 進行中のすべての読み取りと書き込みが再試行されます。
5. 読み取りと書き込みがシンボリックモードで再開されます。
6. コントローラとの同期化オプションが有効になっている場合:
 - ・ シンボリックモードが 30 秒経過すると、ドライバーはコントローラからプロジェクトをアップロード (同期化) します。
 - ・ 読み取りと書き込みは、論理モードを使用して新しい論理アドレスでレジュームされます。
7. コントローラとの同期化オプションが無効になっている場合、読み取りと書き込みは、古い論理アドレスとの論理モードを使用して再開されます。

SoftLogix 5800 の接続の注意事項

適切な操作を行うには、イーサネットデバイス、ゲートウェイを介したリモートデバイスなど) が SoftLogix PC の RSLinx にインストールされている必要があります。1 つまたは複数のイーサネットベースのドライバーがインストールされている場合、要求は CIP エラー 0x5、拡張エラー 0x1、および CIP エラー 0x8 を返します。

OPC サーバーと同じ PC 上の SoftLogix Soft PLC への接続

Allen-Bradley ControlLogix Ethernet ドライバーをサーバーと同じ PC 上で実行されている SoftLogix Soft PLC に接続するには、以下の手順に従います。

1. PC 上に RSLinx で現在実行されているイーサネットベースのドライバーがないことを確認してください。
2. **EtherNet/IP メッセージモジュール** が SoftLogix 仮想シャーシにインストールされていることを確認します。
3. 「**デバイスプロパティ**」| 「**一般**」グループで、デバイス ID の値を見つけます。"127.0.0.1, 1, <PLC CPU スロット>" は指定できません。デバイス ID は "<PC の特定の IP アドレス>, 1, <PLC CPU スロット>" に設定する必要があります。

たとえば、PC の IP アドレスが 192.168.3.4 で、SoftLogix CPU が仮想シャーシのスロット 2 にある場合、正しいデバイス ID は "192.168.3.4, 1, 2" になります。

索引

-

----- 152

%

%s | チャネル統計 152

%s | デバイス統計 152

0

0000 ジェネリックモジュール 28

0x0001 拡張エラーコード 121

0x00FF 拡張エラーコード 122

0x01 120

0x02 120

0x03 120

0x04 120

0x05 120

0x06 120

0x07 120

0x08 120

0x09 120

0x0A 120

0x0B 120

0x0C 120

0x0C 拡張エラーコード 122

0x0D 120

0x0E 120

0x0F 120

0x10 120

0x11 120

0x12 120

0x13 120

0x14 120

0x15 120

0x1A 120

0x1B 120

0x1C 121

0x1D 121

0x1E 121

0x1F 121
0x22 121
0x25 121
0x26 121
0x27 121

1

1 トランザクションあたり平均タグ数 = <数値>。 152
1 秒あたり平均タグ読み取り回数 = <数値>。 152
1 秒あたり平均受信パケット数 = <数値>。 152
1 秒あたり平均送信パケット数 = <数値>。 152
1761-NET-ENI 160

A

ASCII ファイル 108

B

BCD 56
BCD ファイル 109
BOOL の詳細なアドレス指定 73
Boolean 56
Byte 56

C

Char 56
CIP エラーコード、エラーコード 120
ControlLogix 通信パラメータ 22
ControlNet での PLC-5 シリーズのアドレス指定 63

D

DataHighwayPlus (TM) ゲートウェイの設定 161
Date 56
DH+ ゲートウェイのデバイス ID 161
DH+ での SLC 500 モジュール I/O のアドレス指定 62
DINT の詳細なアドレス指定 79
Double 56
DWord 56

E

ENI DF1/DH+/ControlNet ゲートウェイ通信パラメータ 26
ENI での CompactLogix 5300 のアドレス指定 59
ENI での ControlLogix 5500 のアドレス指定 58
ENI での FlexLogix 5400 のアドレス指定 59
ENI での MicroLogix のアドレス指定 60
ENI での SLC 500 モジュール I/O のアドレス指定 63
ENI での SLC 500 固定 I/O のアドレス指定 62
ENI デバイス ID 161

F

Float 56
Float、Float ファイル 108

I

I/O モジュールステータスファイル (IOS) 119
ID 18
INT の詳細なアドレス指定 76

L

L5K ファイルが無効であるか破損しています。 153
L5X ファイルが無効であるか破損しています。 153
LBCD 56
LEN で文字列データを終了 24
LINT の詳細なアドレス指定 81
Logix アドレス指定、ControlLogix 5000 アドレス指定、Logix タグベースのアドレス指定 65
Logix デバイス ID、ControlLogix 5500 Ethernet 159
Logix のアドレス指定 58
Logix の詳細なアドレス指定 73
Logix 通信パラメータ 22
Logix 配列データの順序 72
Long 56
Long ファイル 110
LREAL の詳細なアドレス指定 90

M

MicroLogix 1100 のデバイス ID 163

MicroLogix 1100 の設定 163
MicroLogix PID ファイル 110
MicroLogix のアドレス指定 59
MicroLogix メッセージファイル 113
Module 28

P

PID ファイル 111
PLC-5 シリーズのアドレス指定 63

R

REAL の詳細なアドレス指定 82

S

Short 56
SINT の詳細なアドレス指定 74
SLC 500 スロット構成 27
SLC 500 モジュール I/O のアドレス指定 62
SLC 500 モジュール I/O 選択ガイド 30
SLC 500 固定 I/O のアドレス指定 62
SoftLogix 5800 のアドレス指定 59
SoftLogix Soft PLC 接続の注意事項 171
SoftLogix オプション、ControlLogix オプション、Logix オプション 24
SoftLogix データベースの設定、ControlLogix データベースの設定、Logix データベースの設定 25
SoftLogix 通信パラメータ 22
String 56
STRING データ型のアドレス指定 71

T

TCP/IP ポート 22, 26

U

UDINT の詳細なアドレス指定 87
UINT の詳細なアドレス指定 86
ULINT の詳細なアドレス指定 89
USINT の詳細なアドレス指定 84

W

Word 56

X

XML 要素がポストスキーマの検証に失敗しました。デバイスからのタグのインポートはこのモデルではサポートされていません。代替要素を使用してください。| XML 要素 = '{<名前空間><要素>', サポートしていないモデル = '<モデル>', 代替 XML 要素 = '{<名前空間><要素>'。 154

あ

アトミックデータ型のアドレス指定 68

アドレスに書き込めません。| アドレス = '<アドレス>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。 142

アドレスに書き込めません。| アドレス = '<アドレス>', DF1 ステータス = <コード>。 143

アドレスに書き込めません。ローカルノードがエラーを返しました。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。 146

アドレスに書き込めません。受信したフレームにエラーが含まれています。| アドレス = '<アドレス>'。 140

アドレスに書き込めません。内部メモリが無効です。| タグアドレス = '<アドレス>'。 145

アドレスの説明 58

アプリケーションの最適化 43

い

イーサネット/IP ゲートウェイでの MicroLogix のアドレス指定 59

イーサネット/IP ゲートウェイでの PLC-5 シリーズのアドレス指定 64

イーサネット/IP ゲートウェイでの SLC 500 モジュラー I/O のアドレス指定 62

イーサネット/IP ゲートウェイでの SLC 500 固定 I/O のアドレス指定 62

イーサネットでの CompactLogix 5300 のアドレス指定 59

イーサネットでの ControlLogix 5500 のアドレス指定 58

イーサネットでの FlexLogix 5400 のアドレス指定 59

イーサネット設定 16

イベントログメッセージ 123

インポートファイル '%s' はパス '%s' に見つかりません。 153

インポートファイル '%s' はパス '<空>' に見つかりません。 153

インポートファイル '<空>' はパス '%s' に見つかりません。 154

インポートファイル '<空>' はパス '<空>' に見つかりません。 153

インポートファイルから作成 25

え

エラーコード 120

エラー時に格下げ 20

お

オフライン編集後に同期化 24

オンライン編集後に同期化 24

か

カウンタファイル 105

カプセル化エラーコード 120

き

キャッシュからの初期更新 19

く

グローバルタグ 67

こ

この値はこのモデルの XML 要素ではサポートされていません。新しい値に自動的に設定します。| 値 = '<値>', XML 要素 = '{<名前空間><要素>', モデル = '<モデル>', 新しい値 = '<値>'。 154

コントローラからサーバーへの変換 40

コントローラプログラム情報のアップロード中にエラーが発生しました。CIP エラー = <コード>, 拡張エラー = <コード>。 129

コントローラプログラム情報のアップロード中にカプセル化エラーが発生しました。カプセル化エラー = <コード>。 129

コントローラプログラム情報のアップロード中にフレーミングエラーが発生しました。 129

コントローラプロジェクトの読み込み中にエラーが発生しました。 153

コントローラプロジェクト内の変更の検出 169

さ

サブグループを許可 22

サポートされていないコントローラです。| ベンダー ID = <ID>, 製品タイプ = <タイプ>, 製品コード = <コード>, 製品名 = '<名前>'。 130

サポートされるデバイス 12

し

シミュレーション 18

シリアルゲートウェイでの CompactLogix 5300 のアドレス指定 59

シリアルゲートウェイでの ControlLogix 5500 のアドレス指定 59

シリアルゲートウェイでの FlexLogix 5400 のアドレス指定 59
シリアルゲートウェイでの SoftLogix 5800 のアドレス指定 59
シリアルゲートウェイのデバイス ID 162
シリアルゲートウェイの設定 162
シンボリックインスタンスブロックキャッシュの読み取り回数 = <数値>。 151
シンボリックインスタンスブロックデバイスの読み取り回数 = <数値>。 151
シンボリックインスタンス非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。 151
シンボリックインスタンス非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。 151
シンボリックインスタンス非ブロックデバイスの読み取り回数 = <数値>。 150
シンボリックデバイスの読み取り回数 = <数値>。 150
シンボリック配列ブロックキャッシュの読み取り回数 = <数値>。 150
シンボリック配列ブロックデバイスの読み取り回数 = <数値>。 150

す

スキャンしない、要求ポールのみ 19
スキャンモード 18
ステータスファイル 102
すべてのタグのすべての値を書き込み 16
すべてのタグの最新の値のみを書き込み 16
スロット 28

せ

ゼロで置換 17

た

タイマーファイル 104
タイミング 19
タイムアウト前の試行回数 20
タグインポートファイル 25
タグデータベースのインポート用のファイルを開くときにエラーが発生しました。| OS エラー = '<コード>'。 130
タグで予期しないオフセット/スパンが見つかりました。| タグアドレス = '<アドレス>'。 146
タグで予期しないオフセットが見つかりました。| タグアドレス = '<アドレス>'。 146
タグで予期しないオフセットが見つかりました。タグはシンボリックプロトコルを使用します。| タグアドレス = '<アドレス>'。
146
タグに指定のスキャン速度を適用 19
タグに書き込めません。| タグアドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。 131
タグに書き込めません。| タグアドレス = '<アドレス>'。 137
タグに書き込めません。このタグには不正なデータ型です。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。 134
タグに書き込めません。コントローラタグのデータ型が不明です。| タグアドレス = '<アドレス>'、データ型 = <タイプ>。
132
タグに書き込めません。タグは複数要素の配列をサポートしません。| タグアドレス = '<アドレス>'。 135

- タグに書き込めません。データ型がサポートされていません。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。 133
- タグに書き込めません。ネイティブタグのサイズが不一致です。| タグアドレス = '<アドレス>'。 136
- タグに書き込めません。書き込まれた値には構文エラーが含まれています。| タグアドレス = '<アドレス>'、必要なフォーマット = '<フォーマット>'。 148
- タグに書き込めません。書き込まれた値は範囲外です。| タグアドレス = '<アドレス>'。 149
- タグのインポートファイル名が無効です。ファイルパスは使用できません。 148
- タグを読み取れません。| タグアドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。 132
- タグを読み取れません。このタグには不正なデータ型です。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。 144
- タグを読み取れません。このタグには不正なデータ型です。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。 134
- タグを読み取れません。コントローラタグのデータ型が不明です。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'、データ型 = <タイプ>。 132
- タグを読み取れません。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'。 137
- タグを読み取れません。タグは複数要素の配列をサポートしません。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'。 135
- タグを読み取れません。データ型がサポートされていません。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'、データ型 = '<タイプ>'。 133
- タグを読み取れません。ネイティブタグのサイズが不一致です。| タグアドレス = '<アドレス>'。 136
- タグを読み取れません。内部メモリが無効です。| タグアドレス = '<アドレス>'。 144
- タグ階層 38
- タグ数 15
- タグ生成 20

ち

- チャンネル 0 通信ステータスファイル 118
- チャンネル 1 通信ステータスファイル 118
- チャンネルのプロパティ -イーサネット通信 15
- チャンネルのプロパティ - 一般 15
- チャンネルのプロパティ - 書き込み最適化 16
- チャンネルのプロパティ - 詳細 16
- チャンネル割り当て 17

て

- データコレクション 18
- データベースエラー。PLC5/SLC/MicroLogix デバイスはこの機能をサポートしていません。 153
- データベースエラー。エイリアスタグの処理中にエラーが発生しました。タグは追加されませんでした。| エイリアスタグ = '<タグ>'。 124
- データベースエラー。タグインポートファイルでデータ型が見つかりません。タグは追加されません。| データ型 = '<タイプ>'、タグ名 = '<タグ>'。 124
- データベースエラー。タグインポートファイルでメンバーのデータ型が見つかりません。データ型をデフォルトに設定します。| メンバーのデータ型 = '<タイプ>'、UDT = '<タイプ>'、デフォルトデータ型 = '<タイプ>'。 124
- データベースエラー。タグの CIP データ型を解決できません。デフォルトの型に設定します。| CIP データ型 = <タイプ>、タグ名 = '<タグ>'、デフォルトデータ型 = '<タイプ>'。 127
- データベースエラー。フォワードオープンの要求に利用可能な接続はもうありません。 130

- データベースエラー。フォワードオープンの要求時にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。 125
- データベースエラー。フォワードオープンの要求時にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。 125
- データベースエラー。フォワードオープンの要求時にフレーミングエラーが発生しました。 125
- データベースエラー。プログラムグループの名前が最大文字長さを超えています。プログラムグループの名前が変更されました。| グループ名 = '<名前>'、最大長さ = <数値>、新しいグループ名 = '<名前>'。 149
- データベースエラー。プログラム情報のアップロード中にエラーが発生しました。| プログラム名 = '<名前>'、CIP エラー = <コード>、拡張エラー = <コード>。 127
- データベースエラー。プログラム情報のアップロード中にカプセル化エラーが発生しました。| プログラム名 = '<名前>'、カプセル化エラー = <コード>。 126
- データベースエラー。プログラム情報のアップロード中にフレーミングエラーが発生しました。| プログラム名 = '<名前>'。 127
- データベースエラー。プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。 129
- データベースエラー。プロジェクト情報のアップロード中にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。 125
- データベースエラー。プロジェクト情報のアップロード中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。 125
- データベースエラー。プロジェクト情報のアップロード中にフレーミングエラーが発生しました。 126
- データベースエラー。レジスタセッションの要求時にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。 124
- データベースエラー。レジスタセッションの要求時にフレーミングエラーが発生しました。 125
- データベースエラー。最大文字長さを超えているため、タグ名が変更されました。| タグ名 = '<タグ>'、最大長さ = <数値>、新しいタグ名 = '<タグ>'。 149
- データベースエラー。最大文字長さを超えているため、配列タグの名前が変更されました。| 配列タグ = '<タグ>'、最大長さ = <数値>、新しい配列タグ = '<tags>'。 149
- データベースエラー。参照タグのデータ型が不明です。エイリアスタグのデータ型をデフォルトに設定します。| 参照タグ = '<タグ>'、エイリアスタグ = '<タグ>'、デフォルトデータ型 = '<タイプ>'。 123
- データベースエラー。内部エラーが発生しました。 126
- データベースステータス。L5X ファイルからタグをインポートしています。| スキーマリビジョン = '<値>'、ソフトウェアリビジョン = '<値>'。 150
- データベースステータス。| プログラムの数 = <数値>、データ型の数 = <数値>、インポートされたタグの数 = <数値>。 150
- データベースステータス。OPC タグを生成しています。 150
- データベースステータス。エイリアスタグをインポートしています。 149
- データベースステータス。コントローラプロジェクトを読み込んでいます。 150
- データベースステータス。タグプロジェクトを構築しています。お待ちください。| タグプロジェクト数 = <数値>。 149
- データベースステータス。非エイリアスタグをインポートしています。 149
- データベースのインポート方法 25
- データ型の説明 56
- デバイスからコントローラプロジェクトをアップロード中に次のエラーが発生しました。シンボリックプロトコルを使用しません。 123, 147
- デバイスからファンクションファイルを読み取れません。タグは非アクティブ化されました。| ファンクションファイル = '<名前>'、DF1 ステータス = <コード>、拡張ステータス = <コード>。 140
- デバイスからファンクションファイルを読み取れません。受信したフレームにエラーが含まれています。| ファンクションファイル = '<名前>'。 139
- デバイスから作成 25
- デバイスから受信したフレームにエラーが含まれています。 130
- デバイスが応答していません。ローカルノードがエラーを返しました。| DF1 ステータス = <コード>。 145

- デバイスタグのインポートが中断しました。 152
- デバイスのプロパティ- タイミング 19
- デバイスのプロパティ- タグ生成 20
- デバイスのプロパティ- 自動格下げ 20
- デバイスのプロパティ- 冗長 29
- デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。| CIP エラー = <コード>、拡張エラー = <コード>。 147
- デバイスの識別情報を取得できません。すべてのタグがシンボリックプロトコルを使用します。| カプセル化エラー = <コード>。 147
- デバイスの識別情報を取得できません。受信したフレームにエラーが含まれています。すべてのタグがシンボリックプロトコルを使用します。 147
- デバイスへの読み取り書き込み要求が再開しました。デバイスからの論理アドレスの更新が完了しました。現在、論理アドレス指定を使用しています。 148
- デバイスへの読み取り書き込み要求が中止しました。デバイスプロジェクトからの論理アドレスを更新しています。 148
- デバイスへの要求中にエラーが発生しました。| CIP エラー = <コード>、拡張エラー = <コード>。 138
- デバイスへの要求中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。 138
- デバイス間遅延 17
- デバイス起動時 21
- デバイス平均ターンアラウンドタイム = <数値> (ミリ秒) 152
- デフォルトデータ型の条件 57
- デューティサイクル 16

と

- ドライバー 17
- ドライバー統計 152

ね

- ネットワークアダプタ 16

は

- バイナリファイル 104
- パフォーマンスチューニングの例 45
- パフォーマンスの最適化 42
- パフォーマンス統計 24
- パフォーマンス統計とチューニング 44

ふ

- ファイル一覧 96
- ファンクションファイル 116

- ファンクションファイルに書き込めません。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。 142
- ファンクションファイルに書き込めません。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。 144
- ファンクションファイルに書き込めません。ローカルノードがエラーを返しました。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。 145
- ファンクションファイルに書き込めません。受信したフレームにエラーが含まれています。| ファンクションファイル = '<名前>'。 140
- ファンクションファイルのブロック書き込みを許可 26
- ファンクションファイルを読み取れません。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。 141
- ファンクションファイルを読み取れません。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。 143
- ファンクションファイルを読み取れません。タグは非アクティブ化されました。| ファンクションファイル = '<名前>', DF1 ステータス = <コード>。 141
- フィルタ 26
- フレーミングエラーによりタグの読み取り要求が失敗しました。| タグアドレス = '<アドレス>'。 130
- フレーミングエラーによりブロック読み取り要求が失敗しました。| ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。 131
- フレーミングエラーによりブロック読み取り要求が失敗しました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。 131
- フレーミングエラーにより書き込み要求が失敗しました。| タグアドレス = '<アドレス>'。 130
- プログラムタグ 67
- プログラム情報のアップロード中にエラーが発生しました。| プログラム名 = '<名前>', CIP エラー = <コード>, 拡張エラー = <コード>。 129
- プログラム情報のアップロード中にカプセル化エラーが発生しました。| プログラム名 = '<名前>', カプセル化エラー = <コード>。 129
- プログラム情報のアップロード中にフレーミングエラーが発生しました。| プログラム名 = '<名前>'。 129
- プロジェクトオプション 24
- プロジェクトのオフライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。 146
- プロジェクトのオンライン編集が検出されました。現在、シンボリックのアドレス指定を使用しています。 146
- プロジェクトのダウンロードが完了しました。 146
- プロジェクトのダウンロードが進行中であるかプロジェクトが存在しません。 146
- プロジェクト情報のアップロード中に CIP 接続がタイムアウトしました。 129
- プロジェクト情報のアップロード中にエラーが発生しました。| CIP エラー = <コード>, 拡張エラー = <コード>。 128
- プロジェクト情報のアップロード中にカプセル化エラーが発生しました。| カプセル化エラー = <コード>。 128
- プロジェクト情報のアップロード中にフレーミングエラーが発生しました。 128
- ブロックを読み取れません。| ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>', CIP エラー = <コード>, 拡張エラー = <コード>。 132
- ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>', CIP エラー = <コード>, 拡張エラー = <コード>。 132
- ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。 140
- ブロックを読み取れません。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>', DF1 ステータス = <コード>。 143
- ブロックを読み取れません。このブロックには不正なデータ型です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>', データ型 = '<タイプ>'。 134
- ブロックを読み取れません。コントローラタグのデータ型が不明です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>', データ型 = '<タイプ>'。 133
- ブロックを読み取れません。タグは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>', DF1 ステータス = <コード>, 拡張ステータス = <コード>。 139, 141

- ブロックを読み取れません。データ型がサポートされていません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、データ型 = '<タイプ>'。 134
- ブロックを読み取れません。ネイティブタグのサイズが一致しません。| ブロックサイズ = <数値> (バイト)、ブロック名 = '<名前>'。 136
- ブロックを読み取れません。ネイティブタグのサイズが一致しません。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。 136
- ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (バイト)、タグ名 = '<タグ>'。 138
- ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'、CIP エラー = <コード>、拡張エラー = <コード>。 145
- ブロックを読み取れません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。 137
- ブロックを読み取れません。ブロックは複数要素の配列をサポートしません。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。 135
- ブロックを読み取れません。受信したフレームにエラーが含まれています。| ブロックサイズ = <数値> (要素)、開始アドレス = '<アドレス>'。 139
- ブロックを読み取れません。内部メモリが無効です。タグは非アクティブ化されました。| タグアドレス = '<アドレス>'。 144
- ブロックを読み取れません。内部メモリが無効です。ブロックは非アクティブ化されました。| ブロックサイズ = <数値> (要素)、ブロック開始アドレス = '<アドレス>'。 145
- ブロック転送ファイル 114
- プロトコル 24
- プロトコルモード 24
- プロトコルモードの選択 168
- プロパティ変更時 21

へ

- ヘルプの目次 10

ほ

- ポート ID 164

め

- メッセージファイル 114
- メモリリソース量が低下しています。 150, 153
- 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>'。 139

も

- モデル 17

リ

リアルタイムクロックファイル (RTC) 117

リンクアドレス 164

る

ルーティングの例 165

漢字

概要 11

格下げまでのタイムアウト回数 20

格下げ期間 20

格下げ時に要求を破棄 20

経過時間 = <数値> (秒)。 150

構造体タグのアドレス指定 67

構造体タグのアドレス指定、タグの有効範囲 67

構造体データ型のアドレス指定 70

高速カウンタファイル (HSC) 116

最適化方法 16

作成 22

削除 21

自動タグデータベース生成、長いコントローラプログラム、タグ名 37

自動タグデータベース生成の準備 41

自動格下げ 20

識別 15, 17

受信パケット数 = <数値>。 151

重複タグ 21

出力 Word 数 28

出力ファイル 96

初期化トランザクション数 = <数値>。 152

詳細。| IP = '<アドレス>'、ベンダー ID = <ベンダー>、製品タイプ = <タイプ>、製品コード = <コード>、リビジョン = '<値>'、製品名 = '<名前>'、製品シリアル番号 = <数値>。 150

詳細なアドレス指定

 LTIME 94

 TIME 93

 TIME32 92

上書き 21

冗長 29

親グループ 22

診断 15

制御ファイル 105

整数ファイル 106

生成 21

接続サイズ 22
接続のタイムアウト 19
接続パスの指定 164
設定 11
説明の表示 25
先頭のアンダースコア 40
送信パケット数 = <数値>。 151
通信タイムアウト 19
通信のルーティング 163
通信の最適化 42
通信プロトコル 13
定義済みの用語タグ 68
統計 44
統計タイプ 44
動作モード 18
同期化中にプロジェクトのダウンロードが検出されました。まもなく同期化を再試行します。 123
同期化中にプロジェクトのダウンロードが検出されました。後でもう一度試してください。 153
同期化中に無効または破損したコントローラプロジェクトが検出されました。まもなく同期化を再試行します。 123
同期化中に無効または破損したコントローラプロジェクトが検出されました。後でもう一度試してください。 153
読み取り書き込みトランザクション数 = <数値>。 152
読み取りタグ数 = <数値>。 151
内部タグ 67
内部ドライバーエラーが発生しました。 153
入力 Word 28
入力ファイル 99
配列タグ 38
配列のタグ。アドレスの形式 66
配列の制限を適用 26
配列ブロックサイズ 22
配列要素の上限 26
非 Boolean タグの最新の値のみを書き込み 16
非正規化浮動小数点処理 17
不活動ウォッチドッグ 22
不明なエラーが発生しました。 150, 153
付録 155
物理ブロックキャッシュの読み取り回数 = <数値>。 151
物理ブロックデバイスの読み取り回数 = <数値>。 151
物理非ブロック、配列ブロックキャッシュの読み取り回数 = <数値>。 151
物理非ブロック、配列ブロックデバイスの読み取り回数 = <数値>。 151
物理非ブロックデバイスの読み取り回数 = <数値>。 151
文字列ファイル 109
未修正 17
名前の長さを制限 26
要求サイズ 26
要求された CIP 接続サイズはこのデバイスによってサポートされていません。自動的に最大サイズにフォールバックしま

す。| 要求されたサイズ = <数値> (バイト)、最大サイズ = <数値> (バイト)。 148
要求のタイムアウト 19