

kepware® kepserverex®

© 2019 PTC Inc. All Rights Reserved.

目次

目次	1
目次	2
目次	15
KEPServerEX V6	15
はじめに	16
システム要件	17
アプリケーションデータ	17
コンポーネント	19
プロセスモード	19
インタフェースと接続性	21
OPC DA	21
OPC AE	21
OPC UA インタフェース	23
OPC .NET	23
DDE	24
FastDDE/SuiteLink	24
iFIX ネイティブインタフェース	25
ThingWorx ネイティブインタフェース	25
管理メニューへのアクセス	26
設定	26
設定 - 管理	27
設定 - 構成	27
設定 - ランタイムプロセス	28
設定 - ランタイムオプション	29
設定 - イベントログ	30
設定 - ProglID リダイレクト	32
設定 - ユーザーマネージャ	33
設定 - ユーザーマネージャ ThingWorx Interface Users	37
設定 - Config API サービスの設定	38
設定 - 証明書ストア	40
設定 - サービスポート	42
ストアアンドフォワードサービス	43
ユーザーインタフェースのナビゲーション	44
プロジェクトのプロパティ	48
プロジェクトのプロパティ - 一般	48
プロジェクトのプロパティ - OPC UA	50
プロジェクトのプロパティ - DDE	52
プロジェクトのプロパティ - OPC .NET	53
プロジェクトのプロパティ - OPC AE	54

プロジェクトのプロパティ - FastDDE/SuiteLink	55
プロジェクトのプロパティ - iFIX PDB 設定	56
プロジェクトのプロパティ - OPC HDA	57
プロジェクトのプロパティ - ThingWorx	58
ストアアンドフォワード - 補充率の例	63
ストアアンドフォワード - システムタグ	64
サーバーのオプション	66
オプション - 一般	66
オプション - ランタイム接続	67
コンポーネントと概念	68
チャンネルとは	68
チャンネルのプロパティ - 一般	68
チャンネルのプロパティ - 詳細	69
チャンネルのプロパティ - イーサネット通信	70
チャンネルのプロパティ - シリアル通信	70
チャンネルのプロパティ - イーサネットカプセル化	72
チャンネルのプロパティ - 通信シリアル化	73
チャンネルのプロパティ - ネットワークインタフェース	74
チャンネルのプロパティ - 書き込み最適化	74
デバイスの検出手順	75
デバイスとは	76
デバイスのプロパティ - 一般	77
動作モード	77
デバイスのプロパティ - スキャンモード	78
デバイスのプロパティ - 自動格下げ	79
デバイスのプロパティ - 通信パラメータ	79
デバイスのプロパティ - イーサネットカプセル化	80
デバイスのプロパティ - タグ生成	80
デバイスのプロパティ - 時刻の同期化	82
デバイスのプロパティ - タイミング	82
デバイスのプロパティ - 冗長	83
タグとは	83
タグのプロパティ - 一般	84
複数タグの生成	86
タグのプロパティ - スケール変換	89
動的タグ	90
静的タグ (ユーザー定義)	91
タググループとは	91
タググループのプロパティ	91
エイリアスマップとは	92
エイリアスのプロパティ	93
イベントログとは	94
イベントログ	94

タグの管理	96
CSV のインポートとエクスポート	96
システムタグ	98
プロパティタグ	110
統計タグ	111
モデムタグ	112
通信シリアル化タグ	115
通信管理	118
サーバープロジェクトでのモデムの使用	118
電話帳	119
自動ダイヤル	120
プロジェクトの設計	122
サーバーの実行	122
新しいプロジェクトの開始	122
チャンネルの追加と構成	123
チャンネル作成ウィザード	123
デバイスの追加と構成	125
デバイス作成ウィザード	126
ユーザー定義のタグの追加 (例)	127
タグのブラウズ	129
複数のタグの生成	130
タグスケール変換の追加	132
プロジェクトの保存	133
暗号化されたプロジェクトを開く	135
プロジェクトのテスト	135
操作方法	142
デスクトップ対話の許可	142
エイリアスの作成および使用	142
サーバープロジェクトの最適化	144
チャンネル、デバイス、タグ、およびタググループの適切な名前の指定	145
サーバーに接続された DNS/DHCP デバイスで電源サイクルを行ったときの通信の問題の解決	145
エイリアスを使用したプロジェクトの最適化	146
サーバーでの DDE の使用	147
動的タグアドレス指定の使用	148
イーサネットカプセル化の使用	148
非正規化浮動小数点値の使用	150
デバイス要求ポール	151
iFIX アプリケーションからの構成	153
概要: iFIX アプリケーション内でのデータブロックの作成	153
iFIX 信号条件のオプション	155

iFIX アプリケーションのプロジェクトの起動	161
ストアアンドフォワードサービス	162
構成 API サービス	163
セキュリティ	163
ドキュメンテーション	163
Config API サービス - アーキテクチャ	163
Config API サービス - 同時クライアント	164
Config API サービス - コンテンツの取得	164
Config API サービス - データ	170
Config API サービス - チャンネルの作成	172
Config API サービス - チャンネルの更新	173
Config API サービス - チャンネルの除去	174
Config API サービス - デバイスの作成	174
Config API サービス - デバイスの更新	174
Config API サービス - デバイスの除去	175
Config API サービス - タグの作成	175
Config API サービス - タグの更新	176
Config API サービス - タグの除去	177
Config API サービス - タググループの作成	177
Config API サービス - タググループの更新	177
Config API サービス - タググループの除去	178
Config API サービス - サービス	178
Config API サービス - 応答コード	182
組み込み診断	183
OPC 診断ビューア	183
OPC DA のイベント	186
OPC UA サービス	192
通信診断	194
イベントログメッセージ	197
サーバーのサマリー情報	197
<名前> デバイスドライバは見つからなかったか、ロードできませんでした。	198
'<名前>'ドライバの複数のコピー ('<名前>'と'<名前>') が存在するので、これをロードできません。競合するドライバを除去してからアプリケーションを再起動してください。	199
プロジェクトファイルが無効です。	199
モデム回線 '<回線>'を開くことができませんでした [TAPI エラー = <コード>]。	199
ドライバレベルのエラーによってチャンネルを追加できませんでした。	199
ドライバレベルのエラーによってデバイスを追加できませんでした。	199
バージョンが一致しません。	199
無効な XML ドキュメント:	200
プロジェクト <名前> をロードできません:	200
プロジェクトファイルを '<パス>' にバックアップできませんでした [<理由>]。保存操作は中止されました。保存先ファイルがロックされておらず、読み取り/書き込みのアクセス権があることを確認してください。バックアップ	200

しないでこのプロジェクトの保存を続行するには、「ツール」 「オプション」 「一般」でバックアップオプションを選択解除してからこのプロジェクトを保存し直してください。	
<機能名>が見つからなかったか、ロードできませんでした。	200
プロジェクトファイル<名前>を保存できません:	201
デバイスの検出でデバイスの上限<数>を超えました。検出範囲を絞り込んでから再試行してください。 ...	201
このプロジェクトをロードするために<機能名>は必須です。	201
現在の言語ではXMLプロジェクトをロードできません。XMLプロジェクトをロードするには、サーバー管理で製品言語の選択を英語に変更してください。	201
オブジェクトが見つからないため、プロジェクトをロードできません。 オブジェクト = '<オブジェクト>'。	201
プロジェクトのロード中に無効なモデルが見つかりました。 デバイス = '<デバイス>'。	201
デバイスを追加できません。重複したデバイスがこのチャンネルにすでに存在している可能性があります。	201
自動生成されたタグ'<タグ>'はすでに存在し、上書きされません。	202
デバイス'<デバイス>'のタグデータベースを生成できませんでした。デバイスが応答していません。	202
デバイス'<デバイス>'のタグデータベースを生成できませんでした:	202
自動生成による上書きが多すぎるため、エラーメッセージの書き込みを停止しました。	202
アドレスが長すぎるのでタグ'<タグ>'を追加できません。アドレスの最大長は<数値>です。	202
回線'<回線>'はすでに使用されています。	203
回線'<回線>'でハードウェアエラーが発生しました。	203
回線'<回線>'への接続で通信ハンドルが提供されませんでした。	203
回線'<回線>'でダイヤルできません。	203
チャンネル'<名前>'でネットワークアダプタ'<アダプタ>'を使用できません。デフォルトのネットワークアダプタを使用します。	203
参照先デバイス'<チャンネルデバイス>'でのモデルタイプの変更を却下しています。	204
TAPI回線の初期化に失敗しました<コード>。	204
'<タグ>'での検証エラー:<エラー>。	204
ドライバーDLL'<名前>'をロードできません。	204
'<タグ>'での検証エラー:スケール変換パラメータが無効です。	204
回線'<回線>'にモデム構成を適用できません。	205
デバイス'<デバイス>'は自動的に格下げされました。	205
<ソース>:イーサネットカプセル化IP'<address>'が無効です。	205
'<製品>'ドライバーは現在のところXML永続をサポートしていません。デフォルトのファイルフォーマットを使用して保存してください。	205
プラグインDLL'<名前>'をロードできません。	206
'<デバイス>'に設定されているタイムゾーンは'<zone>'です。これはシステムで有効なタイムゾーンではありません。タイムゾーンをデフォルトの'<zone>'に戻します。	206
ドライバーDLL'<名前>'をロードできません。理由:	206
プラグインDLL'<名前>'をロードできません。理由:	206
自動ダイヤルを行うにはその電話帳に1つ以上の番号が含まれている必要があります。 チャンネル = '<チャンネル>'。	207
チャンネルが共有モデム接続を使用するためには、自動ダイヤルが有効になっていて、その電話帳に1つ以上の番号が含まれている必要があります。 チャンネル = '<チャンネル>'。	207
指定されたネットワークアダプタは、チャンネル'%1' アダプタ='%2'で無効です。	207
タグ生成要求によってタグは作成されませんでした。詳細はイベントログを参照してください。	207
TAPI構成が変更されました。再初期化しています...	207
<製品> デバイスドライバーが正常にロードされました。	208
<名前> デバイスドライバーを起動しています。	208

<名前> デバイスドライバを停止しています。	208
回線 '<modem>' で '<数値>' をダイヤルしています。	208
回線 '<モデム>' は切断されています。	208
回線 '<モデム>' でのダイヤルがユーザーによってキャンセルされました。	208
回線 '<モデム>' が <rate> ボーで接続されました。	208
'<モデム>' でリモート回線がビジー状態です。	208
'<モデム>' でリモート回線が応答していません。	208
'<モデム>' で発信音がありません。	208
電話番号が無効です (<数値>)。	208
'<モデム>' でダイヤルが中止されました。	208
'<モデム>' 上のリモートサイトで回線がドロップされました。	209
回線 '<モデム>' で着信呼び出しが検出されました。	209
モデム回線が開きました: '<モデム>'。	209
モデム回線が閉じました: '<モデム>'。	209
<製品> デバイスドライバがメモリからアンロードされました。	209
回線 '<モデム>' が接続されました。	209
デバイス '<デバイス>' でシミュレーションモードが有効になっています。	209
デバイス '<デバイス>' でシミュレーションモードが無効になっています。	209
デバイス '<デバイス>' にタグを自動生成しようとしています。	209
デバイス '<デバイス>' へのタグ自動生成が完了しました。	209
モデム回線 '<モデム>' の切断を開始しています。	209
クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが有効になりました。	209
デバイス '<デバイス>' でデータ収集が有効になっています。	210
デバイス '<デバイス>' でデータ収集が無効になっています。	210
オブジェクトタイプ '<名前>' はプロジェクトでは許可されません。	210
プロジェクト '<名前>' のバックアップが '<パス>' に作成されました。	210
通信を再確立可能かどうかを判定するためデバイス '<デバイス>' は自動昇格されました。	210
ライブラリ <名前> のロードに失敗しました。	210
マニフェスト作成リソース <名前> の読み取りに失敗しました。	210
プロジェクトファイルはこのソフトウェアのより新しいバージョンで作成されました。	210
クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが無効になりました。	210
電話番号の優先順位が変更されました。 電話番号名 = '<名前>'、更新後の優先順位 = '<優先順位>'。	210
デバイス '<デバイス>' のタグ生成結果。 作成されたタグ数 = <数>。	211
デバイス '<デバイス>' のタグ生成結果。 作成されたタグ数 = <数>、上書きされたタグ数 = <数>。	211
デバイス '<デバイス>' のタグ生成結果。 作成されたタグ数 = <数>、上書きされなかったタグ数 = <数>。	211
オブジェクトへのアクセスが拒否されました。 ユーザー = '<アカウント>'、オブジェクト = '<オブジェクトパス>'、アクセス許可 =	211
User moved from user group. User = '<name>', Old group = '<name>', New group = '<name>'。	211
ユーザーグループが作成されました。 グループ = '<名前>'。	211
ユーザーがユーザーグループに追加されました。 ユーザー = '<名前>'、グループ = '<名前>'。	211
ユーザーグループの名前が変更されました。 古い名前 = '<名前>'、新しい名前 = '<名前>'。	211
ユーザーグループでアクセス許可定義が変更されました。 グループ = '<名前>'。	211
ユーザーの名前が変更されました。 古い名前 = '<名前>'、新しい名前 = '<名前>'。	211
ユーザーが無効になりました。 ユーザー = '<名前>'。	212

ユーザーグループが無効になりました。 グループ = '<名前>'。	212
ユーザーが有効になりました。 ユーザー = '<名前>'。	212
ユーザーグループが有効になりました。 グループ = '<名前>'。	212
ユーザーのパスワードが変更されました。 ユーザー = '<名前>'。	212
アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者 にお問い合わせください。	212
ランタイム動作モードを変更しています。	212
ランタイム動作モードの変更が完了しました。	212
インストールを実行するためにシャットダウンしています。	212
OPC ProgID が ProgID リダイレクトリストに追加されました。 ProgID = '<ID>'。	212
OPC ProgID が ProgID リダイレクトリストから除去されました。 ProgID = '<ID>'。	212
無効な ProgID エントリが ProgID リダイレクトリストから削除されました。 ProgID = '<ID>'。	212
管理者のパスワードが現在のユーザーによってリセットされました。 管理者名 = '<名前>'、現在のユーザー = '<名前>'。	213
ユーザーがユーザーグループから移動しました。 ユーザー = '<名前>'、古いグループ = '<名前>'、新しいグ ループ '<名前>'。	213
ユーザーグループが作成されました。 グループ = '<名前>'。	213
ユーザーがユーザーグループに追加されました。 ユーザー = '<名前>'、グループ = '<名前>'。	213
インポートによってユーザー情報が置き換えられました。 インポートされたファイル = '<absolute file path>'。	213
ユーザーグループの名前が変更されました。 古い名前 = '<名前>'、新しい名前 = '<名前>'。	213
ユーザーグループでアクセス許可定義が変更されました。 グループ = '<名前>'。	213
ユーザーの名前が変更されました。 古い名前 = '<名前>'、新しい名前 = '<名前>'。	213
ユーザーが無効になりました。 ユーザー = '<名前>'。	213
ユーザーグループが無効になりました。 グループ = '<名前>'。	213
ユーザーが有効になりました。 ユーザー = '<名前>'。	213
ユーザーグループが有効になりました。 グループ = '<名前>'。	214
管理者のパスワードのリセットに失敗しました。 管理者名 = '<名前>'。	214
管理者のパスワードのリセットに失敗しました。現在のユーザーは Windows 管理者ではありません。 管理 者名 = '<名前>'、現在のユーザー = '<名前>'。	214
ユーザーのパスワードが変更されました。 ユーザー = '<名前>'。	214
CSV タグインポート時の一般エラー。	214
ランタイムへの接続に失敗しました。 ランタイムのホストアドレス = '<ホストアドレス>'、ユーザー = '<名前 >'、理由 = '<理由>'。	214
ユーザー情報が無効または見つかりません。	214
ランタイムプロジェクトを置き換えるにはユーザーアクセス許可が不十分です。	214
ランタイムプロジェクトの更新に失敗しました。	214
ランタイムプロジェクトの読み込みに失敗しました。	214
アクティブな参照カウントがあるので、チャンネル上のデバイスを置き換えられませんでした。 チャンネル = '<名前 >'。	214
チャンネル上の既存の自動生成されたデバイスの置き換えに失敗し、削除に失敗しました。 チャンネル = '<名 前>'。	215
チャンネルが有効でなくなりました。ユーザー入力を待機中に外部で除去された可能性があります。 チャンネル = '<名前>'。	215
デバイスドライバ DLL がロードされませんでした。	215
デバイスドライバは見つからなかったか、ロードできませんでした。 ドライバー = '<名前>'。	215
CSV データのインポート中にエラーが発生しました。\\n\\n識別レコードの読み取り中にフィールドバッファの オーバーフローが発生しました。	215

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が認識されません。 フィールド = '<名前>'。	215
CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が重複しています。 フィールド = '<名前>'。	215
CSV データのインポート中にエラーが発生しました。\\n\\nフィールド識別レコードが見つかりません。	215
CSV レコードのインポート中にエラーが発生しました。\\n\\nフィールドバッファのオーバーフロー。 レコードインデックス = '<数値>'。	215
CSV レコードのインポート中にエラーが発生しました。\\n\\n挿入に失敗しました。 レコードインデックス = '<数値>'、レコード名 = '<名前>'。	215
アプリケーションを起動できません。 アプリケーション = '<パス>'、OS エラー = '<コード>'。	216
CSV レコードのインポート中にエラーが発生しました。\\n\\nマッピング先のタグアドレスがこのプロジェクトには有効ではありません。 レコードインデックス = '<数値>'、タグアドレス = '<address>'。	216
CSV レコードのインポート中にエラーが発生しました。\\n\\nエイリアス名が無効です。名前に二重引用符を含めたり、名前をアンダースコアで始めることはできません。 レコードインデックス = '<数値>'。	216
無効な XML ドキュメント:	216
名前変更に失敗しました。その名前のオブジェクトがすでに存在します。 提案された名前 = '<名前>'。 ..	216
チャネル診断の開始に失敗しました	216
名前変更に失敗しました。名前にピリオドや二重引用符を含めたり、名前をアンダースコアで始めることはできません。 提案された名前 = '<名前>'。	216
リモートランタイムとの同期化に失敗しました。	216
アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。	216
CSV レコードのインポート中にエラーが発生しました。タグ名が無効です。 レコードインデックス = '<数値>'、タグ名 = '<名前>'。	216
CSV レコードのインポート中にエラーが発生しました。タグまたはグループの名前が最大長を超えています。 レコードインデックス = '<数値>'、名前の最大長 (文字数) = '<数値>'。	217
CSV レコードのインポート中にエラーが発生しました。アドレスが見つかりません。 レコードインデックス = '<数値>'。	217
CSV レコードのインポート中にエラーが発生しました。タググループ名が無効です。 レコードインデックス = '<数値>'、グループ名 = '<名前>'。	217
アクティブな接続があるので終了要求は無視されました。 アクティブな接続 = '<数>'。	217
埋め込み依存ファイルの保存に失敗しました。 ファイル = '<パス>'。	217
構成ユーティリティはサードパーティ製構成アプリケーションと同時に実行できません。両方のプログラムを閉じてから、使用するプログラムだけを開いてください。 製品 = '<名前>'。	217
プロジェクトを開いています。 プロジェクト = '<名前>'。	217
プロジェクトを閉じています。 プロジェクト = '<名前>'。	217
仮想ネットワークモードが変更されました。すべてのチャネルと仮想ネットワークがこの影響を受けます。仮想ネットワークモードの詳細についてはヘルプを参照してください。 新しいモード = '<モード>'。	217
チャネルでデバイス検出を開始しています。 チャネル = '<名前>'。	217
チャネルでデバイス検出が完了しました。 チャネル = '<名前>'、検出されたデバイス = '<数>'。	218
チャネルでデバイス検出がキャンセルされました。 チャネル = '<名前>'。	218
チャネルでデバイス検出がキャンセルされました。 チャネル = '<名前>'、検出されたデバイス = '<数>'。 ..	218
チャネルでデバイス検出を開始できません。 チャネル = '<名前>'。	218
インストールを実行するためにシャットダウンしています。	218
ランタイムプロジェクトがリセットされました。	218
ランタイムプロジェクトが置き換えられました。 新しいプロジェクト = '<パス>'。	218
イベントロガーサービスに接続していません。	218
アイテム '<名前>' の追加に失敗しました。	218

デバイスドライバ DLL がロードされませんでした。	218
無効なプロジェクトファイル: '<名前>'。	218
プロジェクトファイル: '<名前>'を開けませんでした。	218
使用中のプロジェクトと同じであるため、プロジェクト置換の要求を却下しています: '<名前>'。	219
ファイル名が既存のファイル: '<名前>'を上書きしないようにしてください。	219
ファイル名は空にはできません。	219
ファイル名は<サブディレクトリ>/<名前>.{json,opf,sopf}の形式でなければなりません	219
ファイル名には1つ以上の無効な文字が含まれています。	219
「プロジェクトファイルの暗号化」が有効になっているプロジェクトファイルの.OPFファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは.SOPFおよび.JSONです。	219
「プロジェクトファイルの暗号化」が無効になっているプロジェクトファイルの.SOPFファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは.OPFおよび.JSONです。	219
アカウント '<名前>'には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。	219
暗号化されたプロジェクトファイル(.SOPF)にはパスワードが必要です。	219
パスワードを使用した.OPFおよび.JSONプロジェクトファイルの保存はサポートされていません。暗号化されたプロジェクトファイルを保存するには、.SOPFを使用してください。	219
'<名前>'へのオブジェクトの追加に失敗しました: <理由>。	220
オブジェクト '<名前>'の移動に失敗しました: <理由>。	220
オブジェクト '<名前>'の更新に失敗しました: <理由>。	220
オブジェクト '<名前>'の削除に失敗しました: <理由>。	220
スタートアッププロジェクト '<名前>'をロードできません: <理由>。	220
スタートアッププロジェクト '<名前>'の更新に失敗しました: <理由>。	220
定義済みのスタートアッププロジェクトにランタイムプロジェクトが置き換えられました。次の再起動時にランタイムプロジェクトは '<名前>' から復元されます。	220
構成セッションがアクティブなのでユーザー定義のスタートアッププロジェクトは無視します。	220
読み取り専用アイテム参照 '<名前>'に対する書き込み要求が却下されました。	220
アイテム '<名前>'に書き込めません。	220
アイテム '<名前>'への書き込み要求が失敗しました。書き込みデータ型 '<タイプ>'をタグデータ型 '<タイプ>'に変換できません。	220
アイテム '<名前>'への書き込み要求が失敗しました。書き込みデータのスケール変換中にエラーが発生しました。	220
属しているデバイスが無効になっているのでアイテム参照 '<名前>'への書き込み要求は却下されました。	221
<名前>はシステムサービスとして実行するよう正常に設定されました。	221
<名前>はサービスコントロールマネージャデータベースから正常に除去されました。	221
ランタイムの再初期化が開始されました。	221
ランタイムの再初期化が完了しました。	221
スタートアッププロジェクト '<名前>'が更新されました。	221
ランタイムサービスが開始されました。	221
ランタイムプロセスが開始されました。	221
ランタイムが終了処理を実行しています。	221
ランタイムのシャットダウンが完了しました。	221
インストールを実行するためにシャットダウンしています。	221
'<名前>'から置き換えられたランタイムプロジェクトです。	221
アプリケーションデータディレクトリが見つかりません。	222
ランタイムプロジェクトは '<名前>'として保存されました。	222

ランタイムプロジェクトが置換されました。	222
構成セッションが<名前> (<名前>) によって開始されました。	222
<名前> に割り当てられている構成セッションが終了しました。	222
<名前> に割り当てられている構成セッションが書き込みアクセスに昇格されました。	222
<名前> に割り当てられている構成セッションが読み取り専用に格下げされました。	222
<名前> に割り当てられている構成セッションにアクセス許可の変更が適用されました。	222
Failed to start Script Engine server. Socket error occurred binding to local port. Error = <error>, Details = '<information>'.	222
スクリプトから未処理の例外が発生しました。 関数 = '<関数>', エラー = '<エラー>'.	222
Script Engine service stopping.	223
Script Engine service starting.	223
Config API SSL 証明書に不正な署名が含まれています。	223
Config API が SSL 証明書をロードできません。	223
Config API サービスを開始できません。ポートへのバインド中に問題が発生した可能性があります。	223
Config API SSL 証明書が期限切れになっています。	223
Config API SSL 証明書は自己署名されています。	223
ポート <ポート番号> で、構成 API が SSL なしで開始しました。	223
ポート <ポート番号> で、構成 API が SSL ありで開始しました。	223
OPC .NET サーバーの起動に失敗しました。詳細については、Windows アプリケーションのイベントログを参照してください。 .NET 3.5 Framework がインストールされていることも確認してください。 OS エラー = '<エラーの理由>'.	224
OPC .NET サーバーはインストールされていないため、起動できませんでした。インストールを再実行してください。	224
OPC .NET サーバーの起動がタイムアウトしました。OPC .NET Configuration Manager を使用することによってサーバーが動作していることを確認してください。	224
サーバーインスタンスの証明書 '<証明書の場所>' が見つかりません。OPC UA Configuration Manager を使用して証明書を再発行してください。	224
サーバーインスタンスの証明書 '<証明書の場所>' をインポートできませんでした。OPC UA Configuration Manager を使用して証明書を再発行してください。	224
UA Server の証明書が失効しました。OPC UA Configuration Manager を使用して証明書を再発行してください。	224
クライアント接続の待機中にソケットエラーが発生しました。 エンドポイント URL = '<エンドポイント URL>', エラー = <エラーコード>, 詳細 = '<description>'.	225
UA Server は UA Discovery Server を介した登録に失敗しました。 エンドポイント URL: '<エンドポイント URL>'.	225
証明書のロードに失敗したため、UA Server を開始できません。	225
UA Server は UA Discovery Server からの登録解除に失敗しました。 エンドポイント URL: '<エンドポイント URL>'.	225
UA Server は UA Discovery Server を介した登録に成功しました。 エンドポイント URL: '<エンドポイント URL>'.	226
UA Server は UA Discovery Server からの登録解除に成功しました。 エンドポイント URL: '<エンドポイント URL>'.	226
ReadProcessed 要求がタイムアウトになりました。 経過時間 = <秒> (秒)。	226
ReadAtTime 要求がタイムアウトになりました。 経過時間 = <秒> (秒)。	226
DDE アイテムの追加に失敗しました。 アイテム = '<アイテム名>'.	226
DDE クライアントがトピックの追加に失敗しました。 トピック = '<トピック>'.	226
アイテムに書き込めません。 アイテム = '<アイテム名>'.	226
指定された領域は有効ではありません。サブスクリプションフィルタを設定できませんでした。 領域 = '<領域>'	226

名>'。	
指定されたソースは有効ではありません。サブスクリプションフィルタを設定できませんでした。 ソース = '<ソース名>'。	227
Config API SSL 証明書に不正な署名が含まれています。	227
Config API が SSL 証明書をロードできません。	227
Config API サービスを開始できません。ポートへのバインド中に問題が発生した可能性があります。	227
Config API SSL 証明書が期限切れになっています。	227
Config API SSL 証明書は自己署名されています。	227
ポート <ポート番号> で、構成 API が SSL なしで開始しました。	227
ポート <ポート番号> で、構成 API が SSL ありで開始しました。	227
ThingWorx への接続に失敗しました。 プラットフォーム = <ホスト:ポートリソース>、エラー = <理由>。	227
アイテムの追加中にエラーが発生しました。 アイテム名 = '<アイテム名>'。	228
プラットフォーム上の自動バインド完了イベントのトリガーに失敗しました。	228
ThingWorx への接続が失敗しました (理由は不明)。 プラットフォーム <ホスト:ポートリソース>、エラー = <error>。	228
接続バッファ内の容量不足が原因で、1 つまたは複数の値変更の更新が失われました。 失われた更新の数 = <数>。	228
アイテムのパブリッシングに失敗しました。多次元配列はサポートされていません。 アイテム名 = '%s'。	229
ディスクが満杯であるため、「ストアアンドフォワード」データストアでデータを保存できません。	229
「ストアアンドフォワード」データストアのサイズの制限値に達しました。	229
ThingWorx への接続が切断されました。 プラットフォーム = <ホスト:ポートリソース>。	229
プロパティの自動バインドに失敗しました。 名前 = '<プロパティ名>'。	230
Thing を再起動できませんでした。 名前 = '<Thing 名>'。	230
プロパティへの書き込みに失敗しました。 プロパティ名 = '<名前>'、理由 = <理由>。	230
アイテムを追加する ThingWorx 要求が失敗しました。このアイテムはすでに追加されています。 アイテム名 = '<名前>'。	230
アイテムを除去する ThingWorx 要求が失敗しました。アイテムは存在しません。 アイテム名 = '<名前>'。	231
すべてのスキャンで更新を送信するようにサーバーが設定されていますが、1 つまたは複数のプロパティのプッシュタイプが、値が変更された場合にのみプッシュするように設定されています。 数 = <数>。	231
1 つまたは複数のプロパティのプッシュタイプが、更新をプラットフォームにプッシュしないように設定されています。 数 = <数>。	231
アイテムを除去するための ThingWorx 要求が失敗しました。このアイテムはバインドされており、force フラグが false に設定されています。 アイテム名 = '<名前>'。	231
プロパティへの書き込みに失敗しました。 Thing 名 = '<名前>'、プロパティ名 = '<名前>'、理由 = <理由>。	232
Thing へのプロパティ更新のプッシュでエラーが発生しました。 Thing 名 = '<名前>'。	232
「ストアアンドフォワード」データストアに接続または追加できません。インメモリ保存を使用します。 インメモリ保存サイズ (更新) = <数>。	232
ファイル IO エラーまたはデータストア破損のため、「ストアアンドフォワード」データストアがリセットされました。	232
プラットフォームによって開始された設定の変更を適用できません。アクセス許可が拒否されました。 ユーザー = '<ユーザー名>'。	233
ThingWorx Platform への構成の転送に失敗しました。	233
ThingWorx Platform への構成の転送に失敗しました。 理由 = '<理由>'	233
「ストアアンドフォワード」データストアで保存された更新の削除に失敗しました。	233
ThingWorx Platform からの構成の転送に失敗しました。	233
ThingWorx Platform からの構成の転送に失敗しました。 理由 = '<理由>'	233
使用しているアプリケーションキーが適切にフォーマットされ、有効であることを確認してください。	234

ThingWorx に接続しました。 プラットフォーム = <ホスト:ポートリソース>、Thing 名 = '<名前>'。	234
プロジェクト設定の変更による ThingWorx 接続の再初期化が、プラットフォームから開始されました。	234
インタフェースのシャットダウンまたは再初期化のために、保留中の自動バインドがドロップします。 数 = <数>。	234
1 つまたは複数の自動バインド要求が処理されました。 数 = <数>。	234
プロジェクト設定の変更が構成 API から開始されたため、ThingWorx 接続を再初期化しています。	235
Thing へのプロパティ更新のプッシュが再開しました: エラー状態は解決しました。 Thing 名 = '<名前>'。	235
ThingWorx からの構成転送が開始しました。	235
ThingWorx からの構成の転送が中断しました。	235
「ストアアンドフォワード」データストアを初期化しました。 データストアの場所: '<場所>'。	235
保存されたデータが「ストアアンドフォワード」データストアから削除されました。	235
ストアアンドフォワードモードが変更されました。 フォワードモード = '<モード>'。	235
「ストアアンドフォワード」データストアを初期化しました。 フォワードモード = '<モード>' データストアの場所 = '<場所>'。	236
無効なデータストアパスのため、データストアへの追加でエラーが発生しました。 パス = '<パス>'	236
「ストアアンドフォワード」サーバーの開始に失敗しました。ローカルポートへのバインド中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	236
「ストアアンドフォワード」サービスを中止しています。	236
「ストアアンドフォワード」サービスを開始しています。	236
データストアへの接続時にファイルが破損しました。データストアは再作成されました。 データストアのパス = '<パス>'。	236
構成が変更されたため、データストアは上書きされました。 データストアのパス = '<パス>'。	237
既存のデータストアは古いバージョンのサーバーで作成されたため、このデータストアに追加できません。データストアは再作成されました。 データストアのパス = '<パス>'。	237
COM ポートが別のアプリケーションによって使用されています。 ポート = '<ポート>'。	237
指定されたパラメータでは COM ポートを設定できません。 ポート = COM<数値>、OS エラー = <エラー>。	237
ドライバの初期化に失敗しました。	237
シリアル I/O スレッドを作成できません。	237
COM ポートが存在しません。 ポート = '<ポート>'。	238
COM ポートを開く際にエラーが発生しました。 ポート = '<ポート>'、OS エラー = <エラー>。	238
接続に失敗しました。アダプタにバインドできません。 アダプタ = '<名前>'。	238
Winsock のシャットダウンに失敗しました。 OS エラー = <エラー>。	238
Winsock の初期化に失敗しました。 OS エラー = <エラー>。	238
このドライバを使用するには Winsock V1.1 以上がインストールされている必要があります。	239
ローカルポートへのバインド中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	239
デバイスが応答していません。	239
デバイスが応答していません。 ID = '<デバイス>'。	239
チャネルでのシリアル通信エラー。 エラーマスク = <マスク>。	240
タグの書き込みで無効な配列サイズが検出されました。<デバイス名>.<アドレス>。	240
デバイスのアドレスに書き込めません。 アドレス = '<アドレス>'。	240
ドライバがタグを処理している間はこのページ上のアイテムを変更できません。	241
指定されたアドレスはデバイス上で有効ではありません。 無効なアドレス = '<アドレス>'。	241
アドレス '<アドレス>' はデバイス '<名前>' 上で有効ではありません。	241
ドライバがタグを処理している間にこのプロパティを変更することはできません。	241
デバイス '<名前>' 上のアドレス '<アドレス>' に書き込めません。	241
接続中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	242

データの受信中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	242
データの送信中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	242
読み取り可否のチェック中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	242
書き込み可否のチェック中にソケットエラーが発生しました。 エラー = <エラー>、詳細 = '<情報>'。	242
%s 	243
<名前> デバイスドライバ '名前'	243
リソース	244
索引	245



KEPServerEX V6

目次

[概要](#)

[インタフェースと接続性](#)

[管理メニューへのアクセス](#)

[構成のナビゲーション](#)

[基本的なサーバーコンポーネント](#)

[タグ管理](#)

[通信管理](#)

[組み込み診断](#)

[プロジェクトの設計](#)

[操作方法](#)

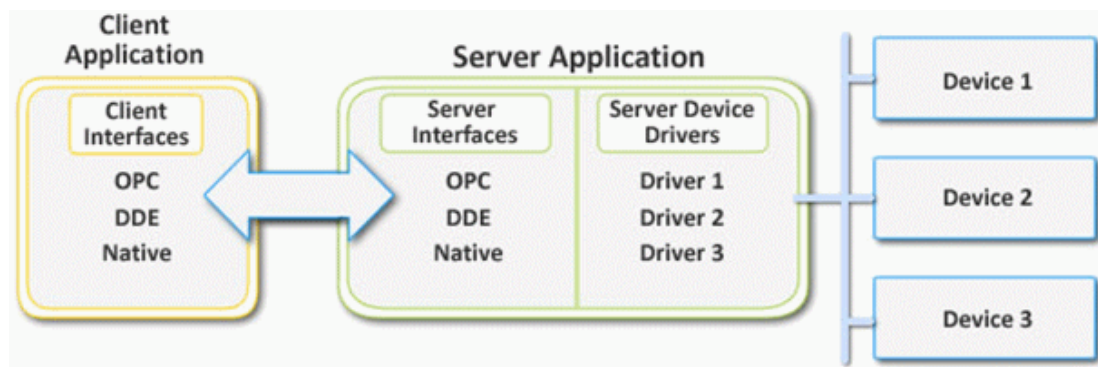
[イベントログメッセージ](#)

● 製品のライセンスについては、ライセンスユーティリティのヘルプファイルを参照してください。サーバーの構成メニューからヘルプファイルにアクセスするには、「ヘルプ」|「サーバーヘルプ」|「ライセンスユーティリティ」の順にクリックします。サーバーの管理メニューからヘルプファイルにアクセスするには、システムトレイにある KEPServerEX アイコンを右クリックし、「ヘルプ」|「ライセンスユーティリティ」の順に選択します。

はじめに

バージョン 1.590

このソフトウェアベースのサーバーは、正確な通信、迅速なセットアップ、および比類ないクライアントアプリケーション、産業用デバイス、システム間の相互運用性を実現することを目的として設計されています。このサーバーにはさまざまなプラグインとデバイスドライバおよびコンポーネントが用意されており、これらによってほとんどの通信要件に対応できます。プラグインの設計と単一のユーザーインターフェースによって、アプリケーションが規格ベースであるかどうかに関係なく、そのネイティブインターフェースからシームレスにアクセスできます。



システム要件

サーバーには、ソフトウェアとハードウェアの最小システム要件があります。アプリケーションを設計どおりに動作させるためには、これらの要件を満たす必要があります。

このアプリケーションは、以下の Microsoft Windows オペレーティングシステムをサポートします。

- Windows 10 x64 (Pro および Enterprise Edition)³
- Windows 10 x86 (Pro および Enterprise Edition)
- Windows 8.1 x64 (Windows 8、Pro、Enterprise Edition)³
- Windows 8.1 x86 (Windows 8、Pro、Enterprise Edition)
- Windows 8 x64 (Windows 8、Pro、Enterprise Edition)³
- Windows 8 x86 (Windows 8、Pro、Enterprise Edition)
- Windows 7 x64 (Professional、Ultimate、Enterprise Edition)³
- Windows 7 x86 (Professional、Ultimate、Enterprise Edition)
- Windows Server 2019 x64^{3,4}
- Windows Server 2016 x64^{3,4}
- Windows Server 2012 x64 R2³
- Windows Server 2012 x64³
- Windows Server 2008 x64 R2³

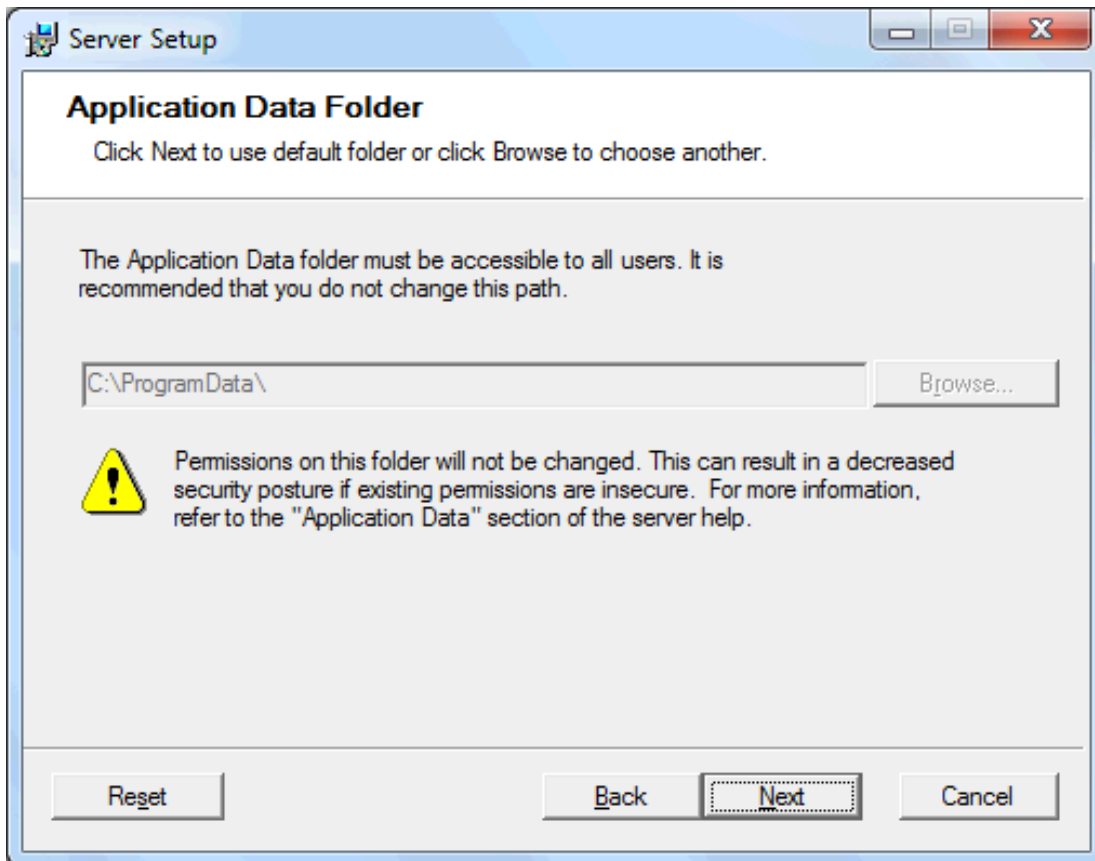
● 注記

1. 64 ビットオペレーティングシステムにインストールすると、アプリケーションは WOW64 (Windows-on-Windows 64 ビット) と呼ばれる Windows のサブシステムで実行されます。WOW64 は、Windows のすべての 64 ビットバージョンに含まれ、オペレーティングシステム間の差異をユーザーに対して透過的にするために設計されています。WOW64 では次の最小要件が必須です。
 - 1 GHz プロセッサ
 - 1 GB の RAM の搭載 (OS の提案に従う)
 - 180 MB の空きディスク容量
 - イーサネットカード
2. オペレーティングシステムに対する最新のセキュリティ更新がインストール済みであることを確認してください。
3. 32 ビット互換モードで実行されます。
4. Windows Server Core の展開はサポートされていません。

● Kepware の Web サイトと PTC の Web サイトでは、追加のリソースが提供されています。特に [KEPServerEX Install Guide](#) と [Secure KEPServerEX @ Deployment](#) は、各段階のプランニングに役立ちます。さらに複雑なシステム要件と推奨事項については、担当のシステムエンジニアにお問い合わせください。

アプリケーションデータ

Microsoft 標準ユーザーは、Application Data ディレクトリに適切なアクセス許可が必要です。このフォルダには、サーバーが正しく機能するための重要なファイルが含まれています。このフォルダのアクセス許可により、製品を設定できるユーザーが決定されます。デフォルトでは、サーバーは C:\ProgramData\<サーバー> にアプリケーションデータを格納します。この設定はインストール中に設定され、製品を再インストールすることによってのみ変更できます。アップグレードは前に構成された Windows セキュリティ設定を継承するため、アクセス許可は新しいインストール中にのみ構成する必要があります。次のダイアログに示すように、新しいインストールによってアプリケーションデータフォルダの場所を構成できます。



Microsoft 標準ユーザーには、フォルダとコンテンツに対し、読み取りアクセス許可と書き込みアクセス許可の両方が付与されている必要があります。サーバーを実行するために、実行アクセス許可は必要ありません。アプリケーションでは、このフォルダへのアクセス許可を追加するためのツールが提供されません。Windows エクスプローラを使用して付与する必要があります。適切なアクセス許可を持たないユーザーがアプリケーションを起動しようとした場合、次のエラーが発生します: 「このアカウントには、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。」

サーバーは、構成されているフォルダのアクセス許可を修正しません。その場所に構成されているデフォルトのアクセス許可を継承します。デフォルト (ProgramData) の場所では、ユーザーのデフォルトの Windows グループの読み取り専用アクセス許可が継承されます。読み取りアクセス許可だけでは製品を構成するために十分な権限は与えられていませんが、フォルダのコンテンツを読み取る権限がないはずのユーザーに対し許可する可能性があります。デフォルトでは、Windows 管理者には適切なアクセス許可が付与されています。

最低限の権限を実装するには、次の最良事例に従います。

- アプリケーションにアクセスする必要のあるユーザーまたはグループにのみアクセス許可を付与し、すべてのユーザーに対してアクセス許可を付与しません。一般的に、デフォルト Windows グループ「Users」のメンバーには、アプリケーションへの必要以上のアクセスが含まれています。
- アクセス権を持つべきではないユーザーに与えられたデフォルトのアクセス許可を除去します。たとえば、デフォルトディレクトリが使用されている場合、「Users」デフォルト Windows グループのメンバーに付与された、継承された読み取り専用アクセス許可を除去します。マシン上のすべてのユーザーが製品で作業できるようにしなければならぬ場合を除き、この操作を実行する必要があります。
- 個別のユーザーまたは「Users」デフォルト Windows グループでアクセス許可を管理しないでください。代わりに、カスタムユーザーグループを作成し、そのアクセス許可を構成します。アクセス許可を付与する必要があるユーザーを、そのグループに追加します。

ソフトウェア要件

次のいずれかのヒューマンマシンインタフェース (HMI) ソフトウェアパッケージを使用する必要があります。

iFIX バージョン 2.6 以降

iHistorian バージョン 1.0 以降

FIX32 バージョン 7.0
Cimplicity HMI バージョン 6.0 以降

コンポーネント

サーバーは、クライアント/サーバーアーキテクチャを実装します。コンポーネントには、構成、ランタイム、管理、およびイベントログがあります。

コンフィギュレーション

構成は、ランタイムプロジェクトを修正するために使用されるクライアント-ユーザーインターフェースです。複数のユーザーが構成を起動でき、リモートランタイム構成がサポートされています。

CSV のインポートとエクスポート

このサーバーでは、コンマ区切り変数 (CSV) ファイルのタグデータのインポートとエクスポートがサポートされています。CSV のインポートとエクスポートを使用すると、タグが目的のアプリケーションに素早く作成されます。

● 詳細については、[CSV のインポートとエクスポート](#)を参照してください。

ランタイム

ランタイムは、デフォルトでサービスとして起動するサーバーコンポーネントです。クライアントは、リモートまたはローカルでランタイムに接続できます。

管理

管理は、ユーザー管理とサーバーに関連する設定を表示または修正したり、アプリケーションを起動したりするために使用されます。デフォルトでは、管理はユーザーがオペレーティングシステムにログオンすると起動され、システムトレイに送信されます。

プロジェクト

プロジェクトファイルには、チャンネル、デバイス、タグの定義とともに、プリファレンスやその他の保存済み設定が含まれています。

● 詳細については、[プロジェクトの設計](#)を参照してください。

イベントログ

イベントログサービスは、情報、警告、エラー、およびセキュリティイベントを収集します。これらのイベントは、確認できるように構成の「イベントログ」ウィンドウに送信されます。

● 詳細については、[イベントログとは](#)を参照してください。

● 関連項目: [基本的なサーバーコンポーネント](#)

プロセスモード

ランタイムプロセスモードはサーバーの稼働中に変更できますが、クライアントが接続されている間に変更した場合、短い時間ながら接続が中断します。運用のモードは、システムサービスまたは対話型のいずれかです。

システムサービス

デフォルトでは、サーバーはサービスとしてインストールされ、実行されます。システムサービスを選択した場合、ランタイムはユーザーの介入を必要とせず、オペレーティングシステムが起動すると開始されます。ユーザーは、クライアントを通じて自動的にサーバーにアクセスできます。

対話型

対話型を選択した場合、ランタイムは、クライアントが接続を試みるまでは停止しています。開始されると、すべてのクライアントが切断されるまで稼働し、最後のクライアントが切断された時点でシャットダウンします。ランタイムは、ユーザーアカウントがオペレーティングシステムからログオフした場合もシャットダウンします。

● 注記: ランタイムプロセスモードは、クライアントアプリケーションでのニーズに応じて、「管理」設定ダイアログボックスで変更できます。

システムサービスが必要になる条件は、次のとおりです。

- UAC が有効になっているオペレーティングシステムで、iFIX の実行が要求される。

対話型が必要になる条件は、次のとおりです。

- 通信インタフェース (DDE など) がユーザーのデスクトップと情報を交換する必要があり、サーバーは、Windows Vista、Windows Server 2008、またはそれ以降のオペレーティングシステムにインストールされている。

● 関連項目:

[設定 - ランタイムプロセス](#)

[デスクトップとの対話を許可する方法](#)

インタフェースと接続性

この通信サーバーは、以下に示すクライアント/サーバーテクノロジーを同時にサポートしています。クライアントアプリケーションは、これらのテクノロジーを使用することにより、サーバーからのデータに同時にアクセスできます。特定のインタフェースの詳細については、以下のリストからリンクを選択してください。

[OPC DA](#)

[OPC AE](#)

[OPC UA](#)

[DDE](#)

[FastDDE/SuiteLink](#)

[iFIX ネイティブインタフェース](#)

[ThingWorx ネイティブインタフェース](#)

OPC DA

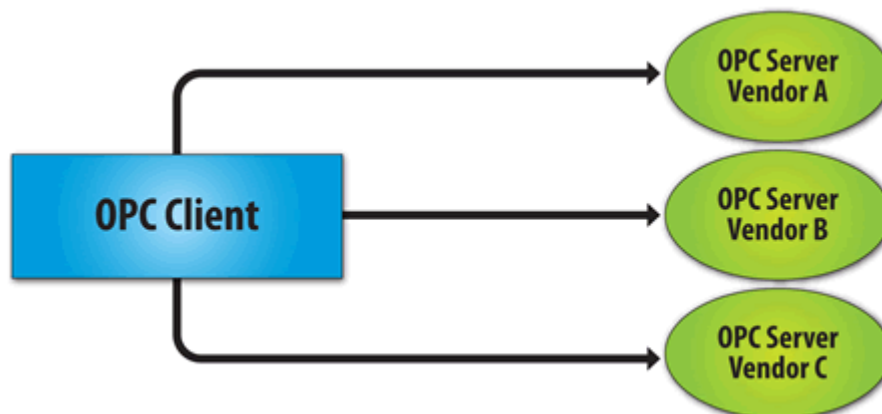
サポートされているバージョン

1.0a
2.05a
3.0

概要

"OPC" は、産業オートメーションや産業を支えるエンタープライズシステムにおける "オープン性、生産性、および接続性" (Open Productivity and Connectivity) の略称です。これは、1 つのアプリケーションがサーバー (データを提供) として、別のアプリケーションがクライアント (データを使用) として機能するクライアント/サーバーテクノロジーです。

OPC は一連の規格仕様で構成されており、最も成果の高い規格が OPC Data Access (DA) です。OPC DA は、広く採用されている産業通信規格であり、これを使用することによって、ベンダーが異なるデバイス間のデータ交換が可能になり、知的財産に関する制限を考慮することなくアプリケーションを制御できます。OPC サーバーは、店舗の PLC 間、作業現場の RTU 間、HMI ステーション間、およびデスクトップ PC のソフトウェアアプリケーション間で継続的にデータを通信できます。OPC に準拠することにより、継続的なリアルタイム通信が可能です (ハードウェアとソフトウェアのベンダーが異なる場合でも)。



1996 年に OPC 協議会によって開発された最初の仕様は OPC Data Access 1.0a です。これは、現在使用されている OPC クライアントアプリケーションの多くで引き続きサポートされていますが、OPC Data Access 2.0 Enhanced OPC では基礎となっている Microsoft COM テクノロジーがより効果的に活用されています。OPC DA インタフェースの最新バージョンは OPC Data Access 3.0 です。

● **関連項目:** [プロジェクトのプロパティ - OPC DA](#)

OPC AE

サポートされているバージョン

1.0
1.10

概要

OPC Alarms & Events (AE) は、アラームおよびイベント情報がシステム間で共有される方法を標準化するために OPC 協議会によって開発された仕様です。この規格を使用する AE クライアントは、装置の安全限界、システムエラー、およびその他の異常な状況に関するアラームおよびイベント通知を受け取ることができます。

単純イベント

単純イベントは、イベントログに表示されるサーバーイベント (情報、警告、エラー、セキュリティイベントなど) です。AE クライアントの単純イベントの場合、サーバーでは以下のフィルタオプションがサポートされています。

- **イベント種別** 単純。
- **イベントカテゴリ** サーバー定義のカテゴリによってフィルタします。各イベントに 1 つのカテゴリが割り当てられます。カテゴリの説明は次のとおりです。
 - **ランタイムエラーイベント** イベントログにエラーとして表示される単純イベント。
 - **ランタイム警告イベント** イベントログに警告として表示される単純イベント。
 - **ランタイム情報イベント** イベントログに情報として表示される単純イベント。

条件イベント

条件イベントは、サーバーの条件によって作成されます。これは、現時点では、Alarms & Events プラグインを使用することによってのみ構成可能です。AE クライアントの条件イベントの場合、サーバーでは以下のフィルタオプションがサポートされています。

1. **イベント 条件**。
2. **カテゴリ** サーバー定義のカテゴリによってフィルタします。各イベントに 1 つのカテゴリが割り当てられます。カテゴリの説明は次のとおりです。
 - **レベルアラーム** プロセスレベル条件によって生成されるイベント。(タンクレベル > 10 など)。
 - **偏差アラーム** 偏差条件によって生成されるイベント。(タンクレベル ± 10 など)。
 - **変更レートアラーム** 変更レート条件によって生成されるイベント。
3. **重要度** 重要度レベルによってフィルタします。レベルの範囲は 0 ~ 1000 です (1000 が最も重要)。各イベントに重要度が割り当てられます。
4. **エリア** プロセスエリアによってフィルタして、そのエリアのみからアラームとイベントを取得します。エリアは、アラームおよびイベント情報を整理するために使用されます。
5. **イベント発生元** イベント発生元によってフィルタして、そのイベント発生元のみからイベントを取得します。イベント発生元は、1 つのエリアに属する 1 つのイベント発生元 (サーバータグなど) によって作成された Alarms & Events エリアです。

● **注記:** Alarms & Events プラグインを使用することにより、サーバータグを使用して条件を構成できます。たとえば、Alarms & Events プラグインを使用して、最大値に到達するとイベントを生成する温度タグを構成できます。Alarms & Events プラグインの詳細については、OPC ベンダーにお問い合わせください。

● **関連項目:** [プロジェクトのプロパティ - OPC AE](#)

オプションのインタフェース

AE サーバーインタフェースでは、以下のオプションのインタフェースはサポートされていません。

- **IOPCEventServer::QueryEventAttributes** このインタフェースは、サーバーでサポートされていないイベント属性を管理します。属性を使用することにより、カスタム情報をイベントに追加できます (特殊なメッセージやサーバータグの値など)。これは IOPCEventSubscriptionMgt::SelectReturnedAttributes インタフェースと IOPCEventSubscriptionMgt::GetReturnedAttributes インタフェースにも当てはまります。
- **IOPCEventServer::TranslateToItemIds** このインタフェースを使用すると、AE クライアントはイベントに関連する OPC DA アイテムを取得できます。これは、イベントがサーバータグの値に関連付けられている場合があるからです。
- **IOPCEventServer2** このインタフェースを使用すると、クライアントはエリアとイベント発生元を有効/無効にできます。1 つのクライアントがすべてのクライアントのエリアまたはイベント発生元を有効/無効にできるため、このインタフェースはサーバーではサポートされていません。

● **注記:** AE サーバーインターフェースではイベントの追跡はサポートされていません。

OPC UA インタフェース

サポートされているバージョン

1.02 最適化されたバイナリ TCP

概要

OPC Unified Architecture (UA) は、OPC 協議会が数十のメンバー組織の協力のもと作成したオープン規格です。これは、作業現場データをビジネスシステムと共有する (店舗から経営責任者へ) もう 1 つの方法です。UA は、Microsoft DCOM に依存することなくクライアントからサーバーに安全に接続する方法でもあります。これを使用すると、ファイアウォールを介して VPN 接続経由で安全に接続できます。この UA サーバーの実装では、最適化されたバイナリ TCP と DA データモデルがサポートされています。

● **注記:** 現時点では、UA を HTTP/SOAP Web サービス経由で使用することも、複雑なデータに使用することもサポートされていません。詳細については、OPC UA Configuration Manager のヘルプファイルを参照してください。

OPC UA プロファイル

OPC UA はマルチパート仕様であり、フィーチャーと呼ばれる多数のサービスや情報モデルが定義されています。機能はプロファイルにグループ化され、これを使用して UA サーバーまたはクライアントによってサポートされる機能が表されます。すべての OPC UA プロファイルのリストとそれぞれの説明については、<http://www.opcfoundation.org/profilereporting/index.htm> を参照してください。

完全にサポートされる OPC UA プロファイル

- 標準 UA サーバープロファイル
- コアサーバーファセット
- データアクセスサーバーファセット
- セキュリティポリシー - Basic128Rsa15 (廃止予定)
- セキュリティポリシー - Basic256 (廃止予定)
- セキュリティポリシー - Basic256Sha256
- セキュリティポリシー - なし (セキュリティで保護されていない)
- UA-TCP UA-SC UA バイナリ

● **警告:** OPC UA 仕様バージョン 1.04 では、セキュリティポリシー Basic128Rsa15 および Basic256 は OPC 協議会によって廃止予定になっています。これらのポリシーによって提供される暗号化は安全性が低いと見なされるため、下位互換性を目的とする使用に制限してください。

部分的にサポートされる OPC UA プロファイル

- ベースサーバー動作ファセット

● **注記:** このプロファイルではセキュリティアドミニストレータ - XML スキーマがサポートされていません。

● **関連項目:** [プロジェクトのプロパティ - OPC UA](#)

OPC .NET

サポートされているバージョン

1.20.2

概要

OPC .NET は、Microsoft の .NET テクノロジーを活用して .NET クライアントがサーバーに接続できるようにする、OPC 協議会によって提供されている API ファミリーです。このサーバーでは OPC .NET 3.0 WCF (旧称 OPC Xi) がサポートされています。ほかの OPC .NET API とは異なり、OPC .NET 3.0 では接続性に Windows Communication Foundation (WCF) が使用されているため、DCOM の問題を回避でき、以下の利点があります。

- 複数の通信バインド (名前付きパイプ、TCP、Basic HTTP、Ws HTTP など) を介した安全な通信。
- OPC クラシックインターフェースの統合。
- Windows 環境の簡単な開発、構成、および配備。

サーバーは、OPC 協議会によって提供されているカスタマイズされたバージョンの OPC .NET 3.0 WCF Wrapper を使用して OPC .NET 3.0 のサポートを追加します。このラッパーは、"xi_server_runtime.exe" と呼ばれるシステムサービスとして実行されます。これは、既存のサーバーの OPC AE および DA インタフェースをラップして、WCF クライアントがサーバーのタグおよびアラームデータにアクセスできるようにします。これは、Historical Data Access (HDA) はサポートしていません。

● **注記:** OPC .NET サービスは、サーバーが起動し、インタフェースが有効になったときにのみ起動されます。OPC DA とは異なり、クライアントはサーバーを起動できません。構成の詳細については、[プロジェクトのプロパティ - OPC .NET](#) を参照してください。

要件

OPC .NET 3.0 をインストールして使用するには、サーバーをインストールする前にマシンに Microsoft .NET 3.5 がインストールされている必要があります。

DDE

サポートされているフォーマット

CF_Text
XL_Table
Advanced DDE

概要

このサーバーは OPC サーバーですが、データを共有するために動的データ交換 (DDE) を必要とするアプリケーションがまだ多数あります。このため、サーバーは、CF_Text、XL_Table、および Advanced DDE のいずれかの DDE フォーマットをサポートする DDE アプリケーションへのアクセスを提供します。CF_Text と XL_Table は、Microsoft によって開発された標準の DDE フォーマットであり、DDE を認識するすべてのアプリケーションで使用できます。Advanced DDE は、産業市場に特有のクライアントアプリケーションの多くでサポートされている高性能フォーマットです。

CF_Text と XL_Table

CF_Text は、Microsoft によって定義された標準の DDE フォーマットです。CF_Text は、DDE を認識するすべてのアプリケーションでサポートされています。XL_Table は、Microsoft によって定義された標準の DDE フォーマットであり、Excel で使用されています。DDE の詳細については、[サーバーで DDE を使用する方法](#) を参照してください。

Advanced DDE

Advanced DDE は、Rockwell Automation によって定義された DDE フォーマットです。現時点では、Advanced DDE は、すべての Rockwell クライアントアプリケーションで認識されます。Advanced DDE は、通常の CF_Text フォーマットのバリエーションであり、これを使用すると、より多くのデータをより速くアプリケーション間で転送できます (エラー処理もより効果的です)。

要件

DDE インタフェースがサーバーに接続するには、ランタイムがデスクトップと対話することが許可されている必要があります。詳細については、[デスクトップとの対話を許可する方法](#) を参照してください。

● **関連項目:** [プロジェクトのプロパティ - DDE](#)

FastDDE/SuiteLink

概要

FastDDE は、Wonderware Corporation によって定義された DDE フォーマットです。これを使用すると、通常の DDE より多くのデータをより速くアプリケーション間で転送できます (エラー処理もより効果的です)。SuiteLink は、FastDDE を継承するクライアント/サーバー通信方法です。これは TCP/IP ベースであり、帯域幅と速度が改善されています。FastDDE と SuiteLink はどちらも、すべての Wonderware クライアントアプリケーションでサポートされています。

● **注記:** Wonderware 接続性ツールキットを使用すると、OPC と FastDDE/SuiteLink の接続性が同時に提供されるとともに、中間でブリッジソフトウェアを使用することなくデバイスデータに素早くアクセスできます。

● セキュリティ上の理由から、最新の Wonderware DAServer Runtime Components を使用することをお勧めします。詳細および利用可能なダウンロードについては、Invensys グローバルテクニカルサポート WDN Web サイトを参照してください。

要件

FastDDE インタフェースがサーバーに接続するには、ランタイムがデスクトップと対話することが許可されている必要があります。詳細については、[デスクトップとの対話を許可する方法](#)を参照してください。

● **関連項目:** [プロジェクトのプロパティ - FastDDE/SuiteLink](#)

● FastDDE、SuiteLink、FactorySuite、InTouch、および Wonderware は、すべて Wonderware Corporation の商標です。

iFIX ネイティブインタフェース

概要

iFIX ネイティブインタフェースを使用すると、iFIX OPC Power Tool を使用することなくローカル iFIX アプリケーションに直接接続できるため、接続タスクが簡略化されます。このインタフェースがサポートされていると、サーバーと iFIX プロセスデータベース (PDB) 間の接続を微調整することもできます。

● **関連項目:** [プロジェクトのプロパティ - iFIX PDB 設定](#)

ThingWorx ネイティブインタフェース

概要

ThingWorx は、接続性プラットフォームの 1 つであり、これを使用すると、ユーザーのデバイスデータに基づいて、便利で実行可能なインテリジェンスを作成できます。ThingWorx ネイティブインタフェースでは、ThingWorx の Always On テクノロジーを使用して、追加の構成をほとんど行うことなく、ThingWorx プラットフォームにデータを提供できます。次世代 ThingWorx Composer の導入に伴い、ユーザーインタフェースをより効果的に Composer と統合できるように ThingWorx ネイティブインタフェースが更新されました。

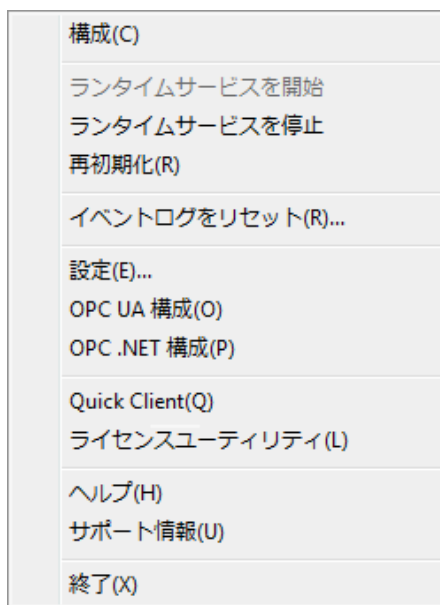
● ThingWorx ドキュメントに記載されているように、ThingWorx のアプリケーションキーの構成は、安全な環境を提供するために非常に重要です。サーバーインスタンスと ThingWorx プラットフォーム間で適切にデータを交換するには、使用されるアプリケーションキーによって必要な権限が付与される必要があります。

ThingWorx ネイティブインタフェースでは、産業用サーバーが ThingWorx プラットフォームから接続解除されたときにプロパティ更新をキャッシュするためのストアアンドフォワードがサポートされています。

● **関連項目:**
[プロジェクトのプロパティ - ThingWorx ネイティブインタフェース](#)
[補充率の例](#)
[ストアアンドフォワード - システムタグ](#)

管理メニューへのアクセス

管理メニューは、ユーザー管理設定を表示または修正したり、サーバーアプリケーションを起動したりするために使用されるツールです。管理メニューにアクセスするには、システムトレイにある「管理」アイコンを右クリックします。



「構成」: このオプションは、OPC サーバーの構成を起動します。

「ランタイムサービスを開始」: このオプションは、サーバーのランタイムプロセスを開始し、デフォルトのランタイムプロジェクトをロードします。

「ランタイムサービスを停止」: このオプションは、すべてのクライアントを切断し、サーバーのランタイムプロセスを停止する前にデフォルトのランタイムプロジェクトを保存します。

「再初期化」: このオプションは、すべてのクライアントを切断し Runtime サーバーをリセットします。これは、サーバーのランタイムプロセスを停止することなく、デフォルトのランタイムプロジェクトを自動的に保存し、再ロードします。

「イベントログをリセット」: このオプションは、イベントログをリセットします。リセットの日付、時刻、およびソースが構成ウィンドウ内のイベントログに追加されます。

「設定...」: このオプションは、設定グループを起動します。詳細については、[設定](#)を参照してください。

「OPC UA 構成」: このオプションは、OPC UA Configuration Manager を起動します (使用可能な場合)。

「OPC.NET 構成」: このオプションは、OPC .NET Configuration Manager を起動します。

「Quick Client」: このオプションは、Quick Client を起動します。

「ライセンスユーティリティ」: このオプションは、サーバーのライセンスユーティリティを起動します。

「ヘルプ」: このオプションは、サーバーのヘルプドキュメントを起動します。

「サポート情報」: このオプションは、サーバーと、それを使用するために現在インストールされているドライバーの両方に関する基本的なサマリー情報を含んでいるダイアログを起動します。詳細については、[サーバーのサマリー情報](#)を参照してください。

「終了」: このオプションは、管理を閉じ、システムトレイから除去します。これを再び表示するには、Windows の「スタート」メニューから選択します。

設定

「設定」グループにアクセスするには、システムトレイにある「管理」アイコンを右クリックします。「設定」を選択します。詳細については、以下のリストからリンクを選択してください。

[設定 - 管理](#)

[設定 - 構成](#)

[設定 - ランタイムプロセス](#)

[設定 - ランタイムオプション](#)

[設定 - イベントログ](#)

[設定 - ProgID リダイレクト](#)

[設定 - ユーザーマネージャ](#)

[設定 - 構成 API サービス](#)

[設定 - 証明書ストア](#)

[設定 - サービスポート](#)

Security Policies - ユーザーのアクセス許可およびアクセス制御用に、プラグインが用意されています。製品のヘルプシステムを参照してください。

Local Historian - データの保存およびアクセス用に、プラグインが用意されています。製品のヘルプシステムを参照してください。

IoT Gateway - モノのインターネットの統合用に、プラグインが用意されています。製品のヘルプシステムを参照してください。

設定 - 管理

「管理」グループは、ランタイム管理の操作を設定するために使用します。

管理を自動的に開始: 有効にすると、「管理」の自動開始が有効になります。「管理」は、「設定」コンソール、構成、ライセンスユーティリティ、ユーザーマネージャコンソール、ランタイムサービスの停止/開始コントロールを含め、各種のサーバーツールに素早くアクセスするためのリンクが用意されたシステムトレイアプリケーションです。

「製品言語の選択」: ドロップダウンメニューから適切なユーザーインターフェースの言語を選択します。

ヒント: 言語設定のデフォルトはインストール時の言語で、インストール時の言語は、デフォルト設定ではオペレーティングシステムの言語設定と同じです。

設定 - 構成

「構成」グループは、構成がどのようにランタイムと接続し、対話するかを設定するために使用します。

接続

構成クライアントがランタイムと通信するために開く必要がある TCP/IP ポートの番号を入力します。場合によってはこのポートでの通信を許可するためにネットワークファイアウォール設定を行う必要があります。

ポートを使用して通信(P) デフォルト(D)

☐ ランタイムでリモート接続の受け入れを許可(L)

セッション管理

同時構成接続の最大数(C)

アイドルセッションのタイムアウト秒数(S)

接続

「ポートを使用して通信」: このプロパティは、構成とランタイムの間で発生する通信に使用される TCP/IP ポートです。デフォルト設定を取得するには、「デフォルト」をクリックします。

「ランタイムでリモート接続の受け入れを許可」: 有効にすると、ランタイムがリモート接続を受け入れるようになります。デフォルトでは無効に設定されています。

セッション管理

「同時構成接続の最大数」: ランタイムに対して同時に作成できる構成接続の数を指定します。範囲は 1 から 64 です。デフォルトは 10 です。

「アイドルセッションのタイムアウト」: 対話できる状態のコンソール接続がシャットダウンされるまでの時間の長さを設定します。有効な範囲は 10 から 3600 秒です。デフォルトは 60 秒です。

設定 - ランタイムプロセス

「ランタイムプロセス」グループは、サーバーランタイムのプロセスモードとともに、ランタイムによる PC リソースの利用方法を指定するために使用します。

プロセスモード

サーバーランタイムはシステムサービスとして動作するか特定のユーザーセッションで対話的に実行できます。この設定を変更した場合、サーバーが再起動し、ユーザーが設定した DCOM 設定がデフォルトに戻ります。

選択したモード(S): システムサービス ▼

プロセス優先度

高優先度の分類でサーバープロセスを実行するには次のボックスをオンにします。

☐ 高優先度(H)

プロセッサアフィニティ

この PC に複数の CPU が搭載されている場合、以下のリストから 1 つ以上の CPU に実行を制限できます。

☒ CPU 0
☒ CPU 1
☒ CPU 2
☒ CPU 3

「**選択したモード**」: このプロパティは、サーバーを「システムサービス」または「対話型」のどちらとして実行するかを指定するために使用します。デフォルトでは、サーバーは「システムサービス」としてインストールされ、実行されます。この設定を変更すると、構成とプロセスを含むすべてのクライアントが切断され、サーバーは、停止して再起動されます。また、ユーザーが設定した DCOM 設定もデフォルトに戻されます。

「**高優先度**」: このプロパティは、サーバープロセスの優先度を「高」に設定するために使用します。デフォルトの設定では「通常」になっています。有効にした場合、サーバーがリソースに優先的にアクセスできます。

● **注記**: アプリケーションの優先度を「高」に設定すると、同一システム上のほかのアプリケーションに悪影響が及ぶ可能性もあることから、Microsoft 社はそのように設定することを推奨していません。

「**プロセッサアフィニティ**」: このプロパティは、複数の CPU を搭載する PC 上でサーバーを実行するときに、どの CPU で実行できるかを指定するために使用します。

設定 - ランタイムオプション

「ランタイムオプション」グループは、ランタイムで実行されるプロジェクトの設定を変更するために使用します。

OPC 接続のセキュリティ

OPC 接続のセキュリティ

DCOM 構成ユーティリティから提供されるセキュリティ設定を使用するには次のボックスをオンにします。ランタイムを再起動する必要があります。

☒ DCOM 構成の設定を使用(D) 設定(N)...

プロジェクトのバックアップ

☒ 置換前にランタイムプロジェクトをバックアップ(B)

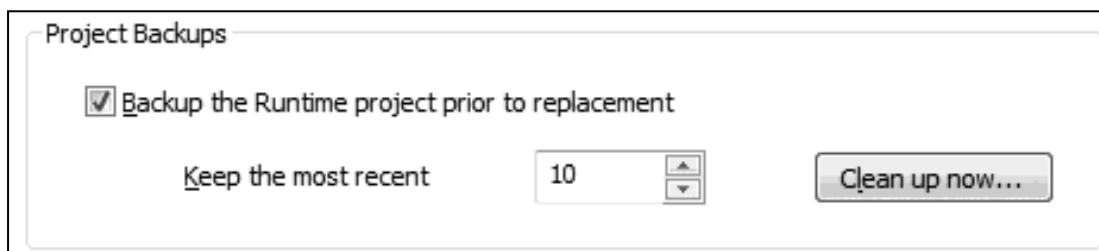
最新を保持(K) 10 すぐクリーンアップ(L)

「**DCOM 構成の設定を使用**」: 有効にすると、DCOM 構成ユーティリティから取得される認証情報とセキュリティ情報が使用されます。

「**設定...**」をクリックすると、ある一定のユーザーやアプリケーションについてセキュリティレベルを指定し、アクセスを制限するための DCOM 構成ユーティリティが起動します。

●この設定が無効になっている場合、サーバーは、アプリケーションに対して設定されている DCOM 設定を無効にし、クライアントアプリケーションから受け取る呼び出しについて、認証を実行しません。サーバーがクライアントアプリケーションに代わって何らかの操作を実行するときは、クライアントアプリケーションのセキュリティ設定を擬装します。この設定を無効にした場合、最小レベルのセキュリティしか提供されなくなるため、無効にすることはお勧めしません。この設定を選択する場合は、クライアントアプリケーションとサーバーアプリケーションのセキュリティが侵害されることのないよう、アプリケーションが安全な環境で実行されていることを確認してください。

プロジェクトのバックアップ



「**置換前にランタイムプロジェクトをバックアップ**」: 有効にすると、ランタイムプロジェクトの上書き前に、ランタイムプロジェクトがバックアップされます。バックアップ先はイベントログに表示されます。このオプションはデフォルトで有効です。

●**注記**: ランタイムプロジェクトが上書きされるのは、ランタイムへの接続時に、「新規」または「開く」が選択されている場合です。また、プロジェクトをオフラインで操作している間にランタイムに接続すると、ランタイムプロジェクトが置き換えられる可能性があります。

「**最新を保持**」: このプロパティは、ディスクに保存されるバックアップファイルの数を制限します。範囲は 1 から 1000 です。デフォルトは 10 です。

「**今すぐクリーンアップ...**」: このプロパティを使用すると、ランタイムプロジェクトのバックアップをすべて削除できる確認ダイアログボックスが表示されます。バックアップを削除しても、実行中のプロジェクトには影響しません。

●**ヒント**: 災害復旧用にプロジェクトファイルのコピーを定期的に保存することをお勧めします。これらのバックアップのデフォルトディレクトリは次のとおりです。

64 ビット OS バージョンの場合、バックアッププロジェクトファイルは以下の場所に保存されます。
C:\ProgramData\Kepware\KEPServerEX\V6\Project Backups

32 ビット OS バージョンの場合、バックアッププロジェクトファイルは以下の場所に保存されます。
C:\ProgramData(x86)\Kepware\KEPServerEX\V6\Project Backups

●**ヒント**: 別の場所にファイルが保存されている場合は、*.opf、*.sopf、または*.json をサーチして、有効なプロジェクトファイルを見つけください。

設定 - イベントログ

「イベントログ」グループは、次のログについて、通信設定と永続設定を定義するために使用します: イベントログ、OPC 診断ログ、通信診断ログ。

●各タイプのログの設定は、ほかのタイプのログの設定からは独立しています。

接続

「**ポート**」: ログとランタイムの間で発生する通信に使用される TCP/IP ポートを指定します。有効な範囲は 49152 から 65535 です。デフォルトのポート設定を復元するには、空白値を入力します。

イベントログの設定

「永続モード」: ログの永続モードを指定します。指定できるオプションは、「メモリ」、「単一ファイル」、および「拡張データストア」です。イベントログ設定のデフォルト設定は、単一ファイルです。OPC 診断ログの設定、および、通信診断ログの設定のデフォルト設定は、いずれも「メモリ (永続性なし)」です。オプションの説明は次のとおりです。

- ・「メモリ (永続性なし)」: このモードを選択すると、イベントはすべてメモリに記録され、ディスクログは生成されません。特定の数のレコードが保持され、その数を超えると最も古いレコードから削除されます。コンテンツは、サーバーが起動するたびに除去されます。
- ・「単一ファイル」: このモードを選択すると、単一のログファイルがディスクに生成されます。特定の数のレコードが保持され、その数を超えると最も古いレコードから削除されます。コンテンツは、サーバーの起動時に、ディスク上のこのファイルから復元されます。
- ・「拡張データストア」: このモードを選択すると、ディスク上の多数のファイルにわたって分散したデータストアに、大量のレコードが保存される可能性があります。レコードは特定の日数にわたって保持され、この日数が経過するとディスクから除去されます。コンテンツは、サーバーの起動時に、ディスク上の分散ファイルストアから復元されます。

「最大レコード数」: ログシステムが保持するレコードの数を指定します。この数を超えると、最も古いレコードから順に削除されはじめます。「永続モード」が「メモリ」または「単一ファイル」に設定されている場合のみ指定できます。有効な範囲は 100 から 100,000 レコードです。デフォルト設定は 25,000 レコードです。

● **注記**: このプロパティをログの現在のサイズよりも小さい値に設定すると、ログが切り捨てられます。

「ログファイルのパス」: ディスクログの保存場所を指定します。「永続モード」が「単一ファイル」または「拡張データストア」に設定されている場合のみ指定できます。

● **注記**: イベントログサービスは SYSTEM アカウントのコンテキストで実行されており、ローカルホスト上のマッピングされたドライブへのアクセス権を持っていないため、マッピングされたパスを使用して診断データを保存しようとすると、失敗する場合があります。マッピングされたパスは、自己判断で使用してください。かわりに、汎用名前付け規則 (UNC) パスを使用することをお勧めします。

「単一ファイル最大サイズ」: 1 つのデータストアファイルのサイズについて、上限を指定します。このサイズに達すると、新しいデータストアファイルが作成されます。「永続モード」が「拡張データストア」に設定されている場合のみ指定できます。有効な範囲は 100 から 10000 KB です。デフォルトの設定は 1000 KB です。

「最小保持日数」: データストアファイルに保存される直近のレコードについて、最低限の保持日数を指定します。この日数を超えると、データストアファイルをディスクから削除できるようになります。「永続モード」が「拡張データストア」に設定されている場合のみ指定できます。有効な範囲は 1 から 90 日です。デフォルトの設定は 30 日です。

● **関連項目**: [組み込み診断](#)

● ファイルに保存するときは、ディスクへのデータの保存に関連するエラーがないかどうか、Windows イベントビューアを観察してください。

ディスクからの永続データストアの復元

イベントログは、起動時または次の場合に、ディスクからレコードを復元します。

1. 「永続モード」が「単一ファイル」または「拡張データストア」に設定されている。

● **注記**: 「単一ファイル」による永続化を選択している場合、サーバーは、すべての永続レコードをディスクからロードした後に、レコードをクライアントが利用できるようにします。

2. ログファイルのパスが、有効な永続ログデータが含まれたディレクトリに設定されている。

拡張データストアによる永続化

「永続モード」を「拡張データストア」にしている場合は、ディスクから膨大な数のレコードがロードされる可能性があります。ログサービスクライアントは、応答性を維持するため、ディスクからレコードがロードされている間にレコードをリクエストします。レコードストアのロードが進行している間は、フィルタリングとは無関係に、ログに含まれているすべてのレコードがクライアントに提供されます。すべてのレコードがロードされると、サーバーがレコードにフィルタを適用し、時系列に沿ってレコードを並べ替えます。クライアントのビューは、自動的に更新されます。

● **注記**: 大規模なレコードストアをロードする際は、ログサーバーの応答性が通常時よりも低下する場合があります。ロードおよび処理が完了すると、サーバーは完全な応答性を取り戻します。リソース使用率は、ロードの進行中は通常時よりも高くなり、完了すると通常時の状態に戻ります。

Disk Full Behavior

「永続モード」を「拡張データストア」にしている場合、特に OPC 診断情報を保存しているときは、ストレージメディアが短時間でいっぱいになる可能性があります。レコードの保存中にディスクエラーが発生した場合は、Windows イベントビューアにエラーが記録されます。

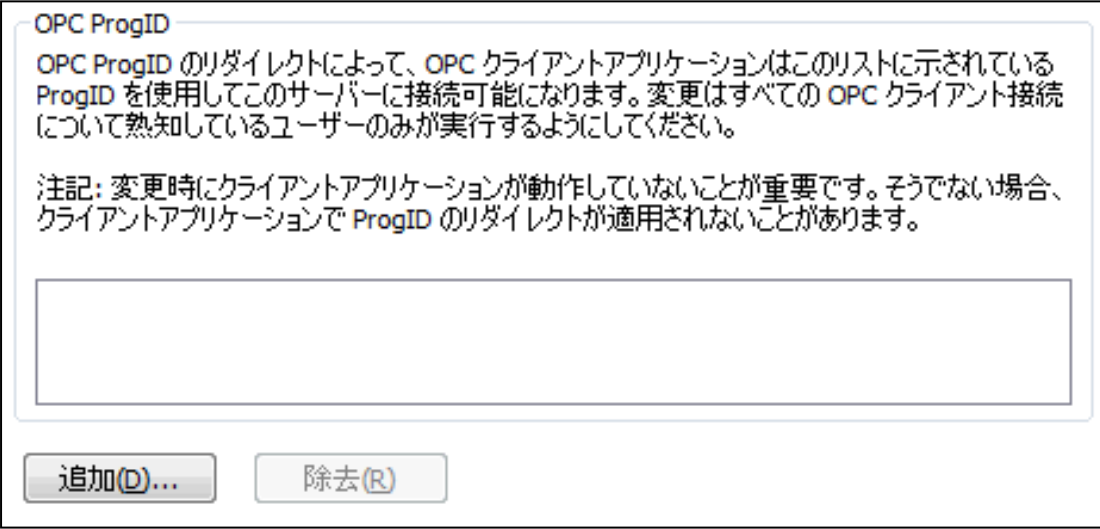
● 関連項目: [OPC 診断ビューア](#)

● イベントログシステムは、イベントログの内容を保護するメカニズムがない場合、役に立たなくなります。オペレータがこれらのプロパティを変更したり、ログをリセットしたりできると、目的が失われます。オペレータのアクセスできる機能を限定するには、ユーザーマネージャを使用します。

設定 - ProgID リダイレクト

多くの OPC クライアントアプリケーションは、OPC サーバーの ProgID を通じて OPC サーバーに接続します。新しい OPC サーバーに移行またはアップグレードする必要があるユーザーは、(OPC サーバーの ProgID にリンクした数千個のタグが保持されている) タグデータベースを変更しないまま、それらを実施することが少なくありません。このサーバーでは、これらの転換においてユーザーを支援する、ProgID リダイレクト機能が提供されています。

ProgID リダイレクト機能を利用すると、レガシーサーバーの ProgID をユーザーが入力できます。必要な Windows レジストリエントリがサーバーによって作成され、クライアントアプリケーションは、レガシーサーバーの ProgID を使用してサーバーに接続できるようになります。



追加: このボタンは、ProgID をリダイレクトリストに追加するために使用します。クリックすると、「新しい ProgID を追加」ダイアログボックスが表示されます。詳細については、以下の「新しい ProgID の追加」を参照してください。

除去: このボタンは、選択した ProgID をリダイレクトリストから除去するために使用します。

● **注記:** リダイレクトされる ProgID は、OpcEnum サービスを使用して OPC サーバーの位置を特定している OPC クライアントアプリケーションではブラウズできません。ほとんどの場合は、ユーザーがクライアントアプリケーションに手動で入力できます。

新しい ProgID の追加

詳細については、次の手順を参照してください。

1. 「ProgID リダイレクト」グループで、「追加」をクリックします。

2. 「ProgID」で、レガシーサーバーの ProgID を入力します。

3. 完了後、「OK」をクリックします。

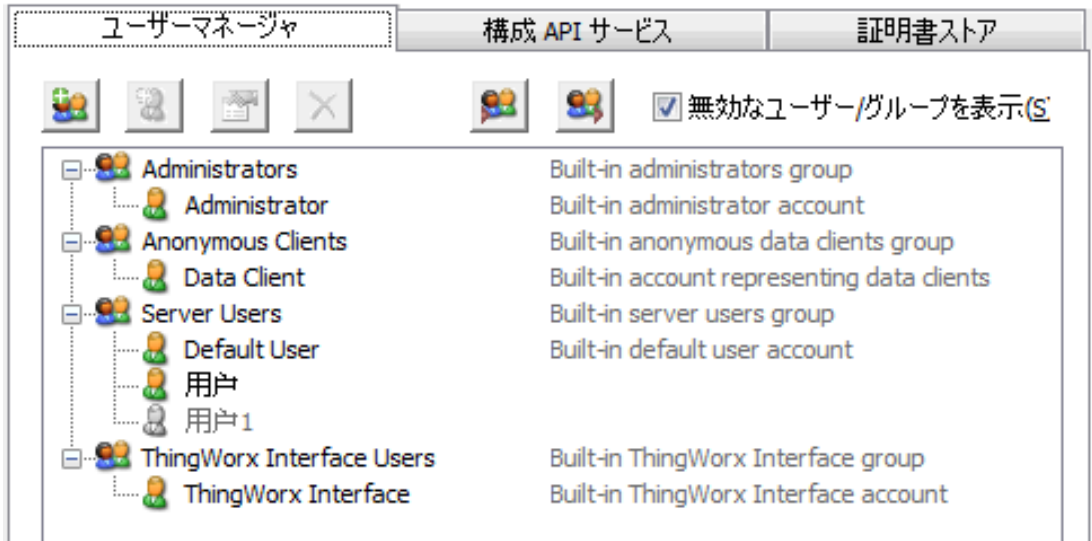
●レガシーサーバーの ProgID をリダイレクトリストに追加している間は、クライアントアプリケーションを実行しないでください。この警告事項を遵守しない場合、新しくリダイレクトされる ProgID がクライアントアプリケーションで適用されない可能性があります。

設定 - ユーザーマネージャ

ユーザーマネージャは、プロジェクトのオブジェクト (チャネル、デバイス、タグなど) と、それに対応する機能へのアクセスを制御します。ユーザーマネージャでは、ユーザーグループごとにアクセス許可を指定できます。たとえば、ユーザーマネージャでは、プロジェクトタグデータに対するデータクライアントユーザーアクセスを、匿名クライアントユーザーグループからそのユーザーに付与されるアクセス許可に基づいて制限できます。ユーザーマネージャは、そのインポート/エクスポート機能によって、サーバーインストール間でユーザー情報を転送することもできます。

ユーザーマネージャには3つの組み込みのグループがあり、それぞれに組み込みのユーザーが含まれます。デフォルトのグループは、管理者、サーバーユーザーおよび匿名クライアントです。デフォルトのユーザーは、管理者、デフォルトのユーザー、およびデータクライアントです。ユーザーは、名前の変更も説明フィールドの変更もできません。デフォルトグループとデフォルトユーザーは、いずれも無効にできません。

● **注記:** 管理者の設定は変更できませんが、管理者ユーザーを追加することはできます。



「新しいグループ」: このボタンをクリックすると、新規ユーザーグループが追加されます。

● 詳細については、[ユーザーグループのプロパティ](#)を参照してください。

「新しいユーザー」: このボタンをクリックすると、選択したユーザーグループに新しいユーザーが追加されます。この機能は匿名クライアントでは無効です。

● 詳細については、[ユーザープロパティ](#)を参照してください。

「プロパティを編集」: このボタンをクリックすると、選択したユーザーまたはユーザーグループのプロパティを編集できます。

● **ヒント:** 複数のアクセス許可を同時に更新するには、プロパティグループを右クリックし、必要なアクセス許可を選択します。

Project Modification		
Server Permissions		
Manage Licenses	Allow All	Deny
Reset OPC D	Deny All	Deny
Reset Communications Diagnostics Log		Deny
Modify Server Settings		Deny
Disconnect Clients		Deny
Reset Event Log		Deny
Manage OPC UA/.NET Configuration		Deny
Config API Log Access		Deny
Replace Runtime Project		Allow

「選択したユーザー/グループを無効にする」: このボタンをクリックすると、選択したユーザーまたはユーザーグループが無効になります。この機能は、カスタムユーザーとユーザーグループだけが使用できます。ユーザーグループを無効にすると、そのグループ内のすべてのユーザーが無効になります。

● **注記:** ユーザーまたはユーザーグループを無効にすると、「無効なユーザー/グループを表示」オプションが呼び出されます。このオプションを有効にすると、無効になっていたすべてのユーザーおよびユーザーグループが、ユーザーグループとユーザーのリストに表示されます。

「選択したユーザー/グループを回復」: このボタンをクリックすると、選択したユーザーまたはユーザーグループが回復します。ユーザーグループを復元すると、そのグループ内のユーザーの状態が、無効にする前の状態に戻ります。このアイコンは、ユーザーまたはユーザーグループが無効になっている場合にのみ使用できます。

● **注記:** 無効なユーザーとユーザーグループをすべて復元すると、「無効なユーザー/グループを表示」オプションが表示されなくなります。

「ユーザー情報をインポート」: このボタンをクリックすると、XML ファイルからユーザー情報がインポートされます。正常にインポートするには、選択したファイルが、サーバーの管理ユーティリティからエクスポートされている必要があります。この機能は、組み込みの管理者がログインしている場合にのみ有効です。

「ユーザー情報をエクスポート」: このボタンをクリックすると、ユーザー情報が XML ファイルにエクスポートされます。これは、1 台のマシンから別のマシンにプロジェクトを移動する必要があるユーザーにとって便利です。管理者にも、XML ファイルをパスワードで保護するオプションがあります。このオプションを使用すると、新しいマシンに正常にインポートするために正しいパスワードを入力する必要があります。XML ファイルは編集や再インポートができません。この機能は常に有効になっています。

● **注記:** ユーザー情報のインポート/エクスポート機能は、サーバーバージョン 5.12 でリリースされました。以前のサーバーバージョンの使用中に設定したすべてのユーザーパスワードは、エクスポートを試みる前に 5.12 で変更する必要があります。変更しないと、エクスポートは失敗します。

● サーバーをアップグレードした場合やユーザー情報をインポートした場合は、アクセス許可が正しく設定されているかどうかをユーザーマネージャで確認することをお勧めします。

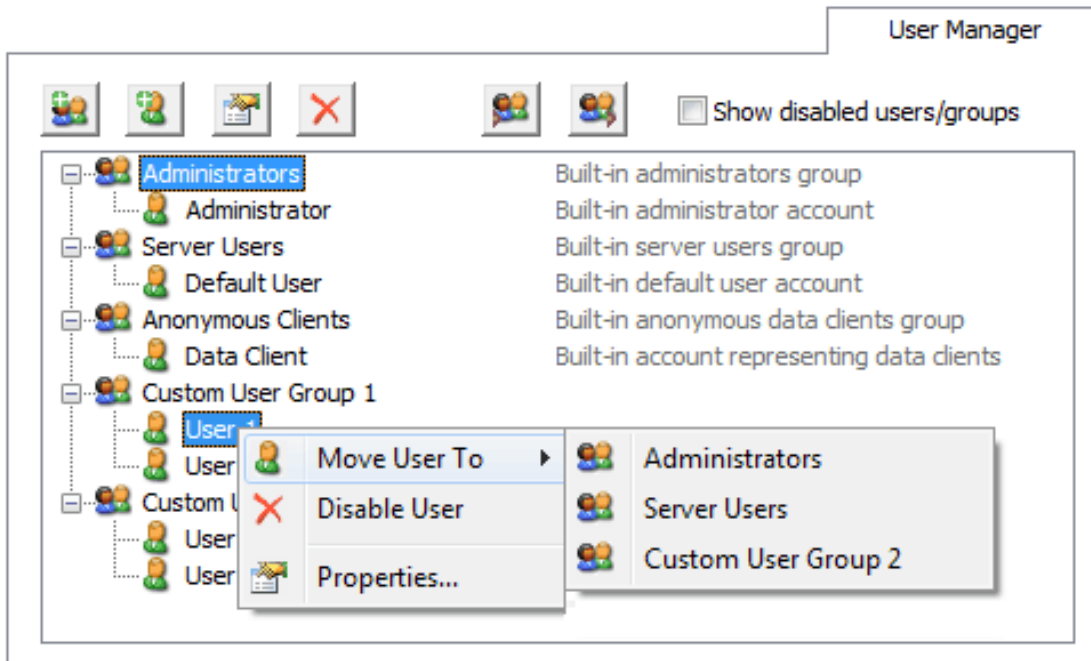
● **注記:** カスタムユーザーとユーザーグループは、一度作成すると削除できませんが、「ユーザー情報をインポート」オプションを使用すると、既存のユーザーとユーザーグループを、インポートしたユーザーとユーザーグループで置き換えることができます (管理者組み込みユーザーを除く)。

● プロジェクト保護のために、完了したユーザー情報のコピーをエクスポートすることをお勧めします。正しいユーザー情報がないと、プロジェクトをロードできません。

● **関連項目:** ThingWorx プラットフォームに接続する場合は、[ThingWorx インタフェースユーザー](#)を参照してください。

その他の設定へのアクセス

ショートカットとその他の設定には、ユーザーグループおよびユーザーのコンテキストメニューからアクセスできます。



「ユーザーの移動先」このオプションによってユーザーが別のユーザーグループに移動します。グループのステータスにかかわらず、有効なグループと無効なグループの両方がリストに表示されます。アクティブなユーザーを無効なグループに移動すると、そのユーザーも無効になります。有効なグループに無効なユーザーを移動すると、そのステータスは、変更するまで維持されます。

ユーザーグループのプロパティ

ユーザーグループを右クリックし、「プロパティ」を選択することでも、ユーザーグループのプロパティにアクセスできます。

ヒント: カテゴリ内のすべてのオプションをすばやく許可または拒否するには、そのカテゴリを右クリックし、「すべて許可」または「すべて拒否」を選択します。太字のテキストで表示される設定は、その値が変更されたことを示します。変更を保存すると、テキストが通常どおり表示されます。

「名前」: アイコンをクリックして、新しいユーザーグループの名前を指定します。使用可能な最大文字数は31です。名前の重複は許可されません。

「説明」: このオプションのプロパティは、ユーザーグループの簡単な説明を提供します。これは、オペレータが新しいユーザーアカウントを作成するときに特に役立ちます。使用可能な最大文字数は128です。

「このユーザーグループに割り当てられているアクセス許可」: このフィールドでは、選択したユーザーグループのアクセス許可を指定します。アクセス許可は、プロジェクトの修正、サーバーのアクセス許可、I/O タグアクセス、システムタグアクセス、内部タグアクセス、およびプロジェクト名前空間のブラウズのカテゴリに分かれています。カテゴリの詳細は次のとおりです。

- ・「プロジェクトの修正」: このカテゴリでは、デフォルトのプロジェクトの修正を制御するアクセス許可を指定します。
- ・「サーバーアクセス許可」: このカテゴリでは、サーバー機能へのアクセスを制御するアクセス許可を指定します。これらのアクセス許可は、匿名クライアントではサポートされていません。
- ・「I/O タグアクセス」: このカテゴリでは、デバイスレベルの I/O タグデータへのアクセスを制御するアクセス許可を指定します。これらのタグにはデバイス通信が必要で、サーバーでは静的タグとして記述されます。
- ・「システムタグアクセス」: このカテゴリでは、システムタグへのアクセスを制御するアクセス許可を指定します。これらのタグは先頭がアンダースコアで、サーバー定義の場所にあります。詳細については、[システムタグ](#)を参照してください。
- ・「内部タグアクセス」: このカテゴリでは、内部タグへのアクセスを制御するアクセス許可を指定します。これらのタグは、ドライバー管理（ドライバーの操作の一部を制御）またはユーザー指定（プラグインレベル）のいずれかです。
- ・「プロジェクト名前空間のブラウズ」: このカテゴリでは、ブラウズをサポートするクライアントのプロジェクトの名前空間に対するブラウズアクセスを制御するアクセス許可を指定します。この時点では、これは数個のクライアントタイプでのみサポートされています。

● **ヒント**: カテゴリの特定のオブジェクトの詳細を表示するには、そのオブジェクトを選択します。

ユーザープロパティ

ユーザープロパティには、ユーザーをダブルクリックするか、ユーザーを右クリックして「プロパティ...」を選択することでもアクセスできます。



「古いパスワード」: このフィールドには、このユーザーについてアクティブであったパスワードが表示されます。

「パスワード」: このユーザーがシステムにログインするために入力する必要がある新しい (更新後の) パスワードを入力します。大文字と小文字が区別され、最大 512 文字を使用できます。

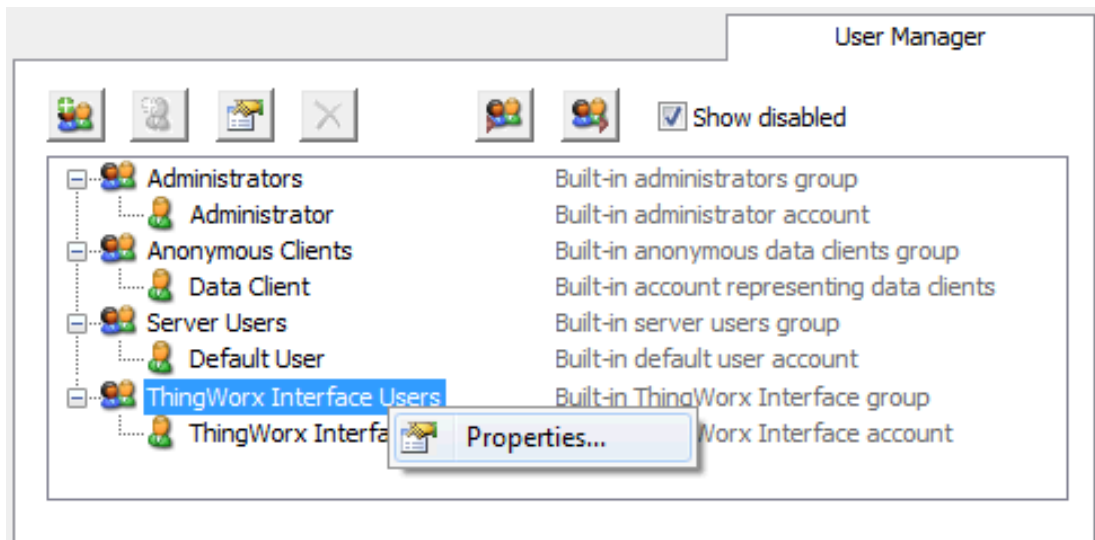
「パスワードを確認」: 同じパスワードを再入力します。「新しいパスワード」フィールドと「パスワードを確認」フィールドの両方にまったく同じパスワードを入力する必要があります。

● **注記**: パスワードの長さは少なくとも 14 文字で、大文字と小文字の両方、数字、および特殊文字を含めることをお勧めします。広く知られたパスワード、簡単に推測できるパスワード、一般的なパスワードは避けてください。

設定 - ユーザーマネージャ ThingWorx Interface Users

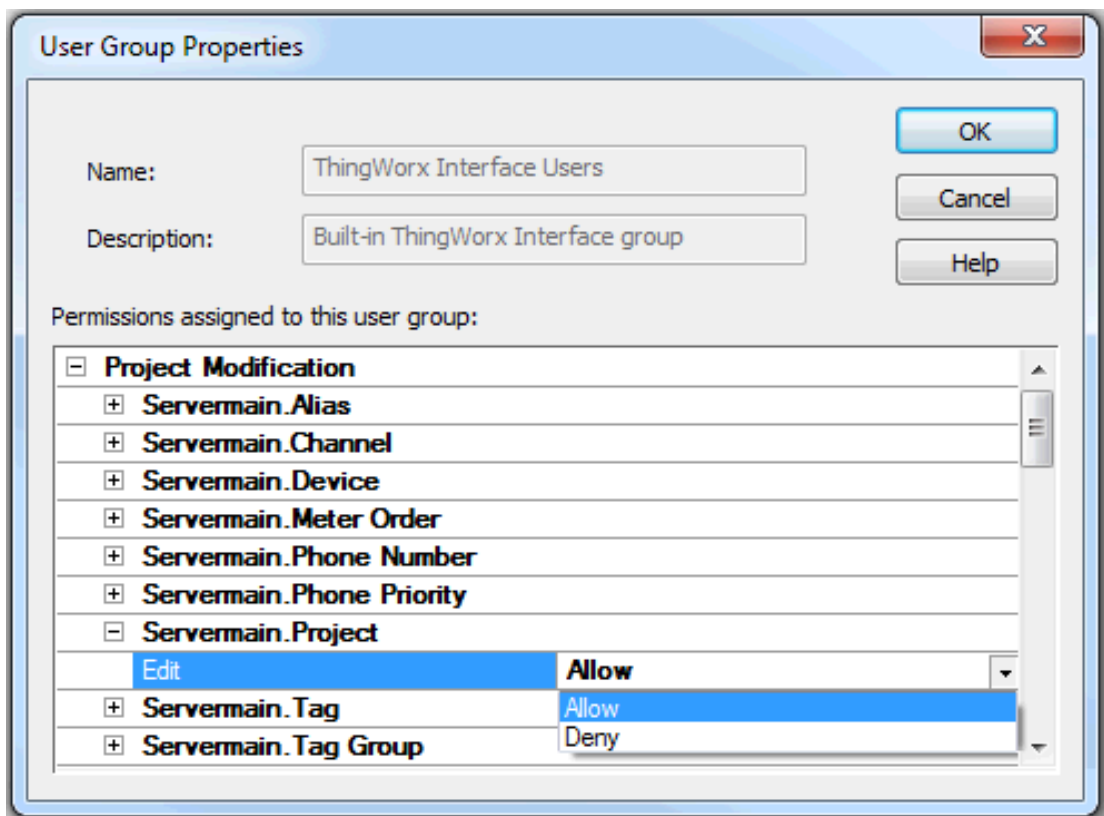
ユーザーマネージャは、プロジェクトオブジェクトとそれに対応する関数へのクライアントアクセスを制御します。すべてのボタンとコントロールは、ユーザーマネージャの「一般」セクションに記載されているとおりに機能します。「ThingWorx Interface Users」グループは、接続されている ThingWorx プラットフォームインスタンスに対するアクセス、データ交換、解析を制御します。

● 関連項目: [ユーザーマネージャ](#)



サーバーと ThingWorx プラットフォームとの間でデータを転送するための適切なアクセス権を付与するには、「プロジェクトの修正」オプションと「ストアアンドフォワード」オプションを有効にする必要があります。この機能に対して正しいアクセス権を付与するには、以下の手順を実行します。

1. 「ThingWorx Interface Users」グループを選択します。
2. 右クリックして「プロパティ...」を選択します。
3. 「プロジェクトの修正」グループを展開します。
4. 「Servermain.Project」権限を探して展開します。
5. ドロップダウンメニューを使用して「許可」を選択し、プロジェクトファイルの変更許可を付与します。
6. 「OK」をクリックして閉じます。



設定 - Config API サービスの設定

構成 API サービスは、インストール時に構成されます。設定を調整する必要がある場合は、システムトレイにある「管理」アイコンを右クリックし、「設定」|「構成 API サービス」の順に選択します。

● システムトレイに「管理」アイコンがない場合は、次のように選択してトレイを再起動します:「スタート」|「すべてのプログラム」|「Kepware」|「KEPServerEX 6」|「KEPServerEX 6 Administration」|「設定」|「スタート」|「すべてのプログラム」|「PTC」|「OPC Aggregator」|「OPC Aggregator Administration」|「設定」。

管理	構成	ランタイムプロセス	ランタイムオプション	イベントログ	ProgIDリダイレクト
ユーザーマネージャ	構成 API サービス	Security Policies	Local Historian	IoT Gateway	
有効化		いいえ			
HTTP を有効にする		いいえ			
HTTP ポート		57412			
HTTPS ポート		57512			
CORS で許可されるオリジン					
デフォルトに戻す		デフォルトに戻す			
ブラウザで表示		http://127.0.0.1:57412/config			
ブラウザで表示 (SSL)		https://127.0.0.1:57512/config			

「Enable」: 構成 API サーバーを有効にする場合は、「はい」を選択します。無効（「いいえ」）になっていると、サービスは実行されますが、HTTP および HTTPS ポートにバインドせず、クライアントがサーバーにアクセスできません。

「HTTP を有効にする」: セキュリティで保護された暗号化されたプロトコルとエンドポイントのみにデータ転送を制限するには、「いいえ」を選択します。暗号化されていないデータ転送を許可するには、「はい」を選択します。

● ヒント:

1. ユーザー認証、アプリケーションキー、およびその他の機密データはプレーンテキストとして転送されるので、HTTP は内部ネットワークに対してのみお勧めします。
2. 保護されていない HTTP 経由での外部アクセスを防ぐため、このポートを Windows ファイアウォールでブロックすることをお勧めします。

「HTTP ポート」: 暗号化されていない HTTP 経由で REST クライアントが通信するための TCP/IP ポートを指定します。有効な範囲は 1 から 65535 です。HTTP ポートと HTTPS ポートが同じであってはなりません。デフォルトのポート番号: 57412。

「HTTPS ポート」: 保護されている HTTPS 経由で REST クライアントが通信するための TCP/IP ポートを指定します。有効な範囲は 1 から 65535 です。HTTP ポートと HTTPS ポートが同じであってはなりません。デフォルトのポート番号: 57512。

「CORS で許可されるオリジン」: 構成 API サーバーにアクセスして Cross Origin Resource Sharing (CORS) 要求を処理できるコンマ区切りのドメイン仕様の承認済み "ホワイトリスト" を指定します。

「デフォルトを回復」: 右側にある青色のリンクをクリックすると、デフォルトの HTTP および HTTPS ポート値が復元されます。

「ブラウザで表示」: 右側にある青色のアドレスリンクをクリックすると、ブラウザで構成 API のドキュメントのトップページが開きます。

「ブラウザで表示 (SSL)」: 右側にある青色のアドレスリンクをクリックすると、ブラウザで保護されている URL を使用して構成 API のドキュメントのトップページが開きます。

管理

構成

ランタイムプロセス

ランタイムオプション

イベントログ

ProgID リダイレクト

ユーザーマネージャ

構成 API サービス

Security Policies

Local Historian

IoT Gateway

☐

トランザクションログ

永続モード	メモリ (永続なし)
最大レコード数	1000
ログファイルのパス	C:\ProgramData\Kepware\KEP ServerE...
単一ファイル最大サイズ (KB)	1000
最小保持日数	30
詳細	いいえ

「トランザクションログ」

「永続モード」: システムログのレコード保持方法を選択します。デフォルトの設定は「メモリ (永続なし)」です。以下のオプションがあります。

- 「メモリ (永続なし)」: メモリ内のすべてのイベントを記録しますが、ディスクに保存されるログの生成は行いません。特定の数のレコードが保持され、その数を超えると最も古いレコードから削除されます。コンテンツは、サーバーの実行中にのみ使用できます。
- 「単一ファイル」: ディスクに保存される記録済みログファイルを生成します。特定の数のレコードが保持され、その数を超えると最も古いレコードから削除されます。コンテンツは、サーバーの起動時に、このファイルから復元されます。
- 「拡張データストア」: レコードが膨大な数にのぼる場合に、複数のファイルに分散してディスクに保存します。レコードは特定の日数にわたって保持され、この日数が経過するとディスクから除去されます。コンテンツは、サーバーの起動時に、ディスク上の分散されているファイルから復元されます。

「最大レコード数」: ログに保持されるトランザクションの数を指定します。この数を超えると最も古いレコードが削除されます。「永続モード」が「メモリ」または「単一ファイル」に設定されている場合に使用できます。有効な範囲は 100 から 30000 レコードです。デフォルトの設定は 1000 レコードです。

● 注記: このパラメータを現在のログのサイズよりも小さい値に設定すると、ログは切り詰められます。

「**ログファイルのパス**」: ディスク上のログが保存されている場所を示します。「永続モード」が「単一ファイル」または「拡張データストア」に設定されている場合に使用できます。

● マッピングされているパスを使用して診断データを永続化する試みは失敗する可能性があります。トランザクションログサービスはシステムアカウントのコンテキストで実行されており、ローカルホスト上のマッピングされているドライブにアクセスできないからです。マッピングされているドライブパスを使用する際には注意が必要です。汎用名前付け規則 (UNC) パスを使用することをお勧めします。

「**単一ファイル最大サイズ**」: 新しいデータストアファイルが開始される単一のデータストアファイルのサイズ制限を KB 単位で示します。「永続モード」が「拡張データストア」に設定されている場合に使用できます。有効な範囲は 100 から 10000 KB です。デフォルトの設定は 1000 KB です。

「**最小保持日数**」: 個々のデータストアファイルが保持される日数を指定します。この日数が経過するとディスクから削除されます。「永続モード」が「拡張データストア」に設定されている場合に使用できます。有効な範囲は 1 から 90 日です。デフォルトの設定は 30 日です。

「**詳細**」: 詳細なデータをログに記録する場合は、「はい」を選択します。詳細なログには、それ以外のログに含まれているパラメータに加えて、HTTP 要求および応答ボディが含まれます。詳細については、[詳細ログ](#)を参照してください。より少ないデータを記録して、ログファイルが大きくなるようにする場合は、「いいえ」を選択します。

管理	構成	ランタイムプロセス	ランタイムオプション	イベントログ	ProgID リダイレクト
ユーザーマネージャ	構成 API サービス	Security Policies	Local Historian	IoT Gateway	

証明書管理

証明書を表示	証明書を表示
証明書をエクスポート	証明書をエクスポート
証明書を再発行	証明書を再発行
証明書をインポート	証明書をインポート

「証明書管理」

● **注記**: クライアントと REST サーバー間の SSL 通信は、X.509 証明書を使用して確立されます。REST サーバーがインストールされているにもかかわらず、保護されているネットワークの外部からサーバーにアクセスするために信頼されている証明書が必要な場合は、デフォルトの自己署名証明書が生成されます。

「**証明書を表示**」: 右側にある青色のリンクをクリックすると、現在の証明書が開き、その内容をレビューできます。

「**証明書をエクスポート**」: 右側にある青色のリンクをクリックすると、現在の証明書が .PEM フォーマットで保存されます (サードパーティの REST クライアントにインポートするためなど)。

「**証明書を再発行**」: 右側にある青色のリンクをクリックすると、新しい証明書が作成され、現在の証明書に置き換わります。

「**証明書をインポート**」: 右側にある青色のリンクをクリックすると、証明書が .PEM フォーマットでインポートされます。

● **注記**: 証明書はインストール時に追加の構成を行うことなく作成されます。証明書を再発行またはインポートした場合、構成 API が停止し、Windows のサービスコントロールマネージャまたはシステムの再起動によって再起動されるまで、新しい証明書は適用されません。

設定 - 証明書ストア

証明書ストアを使用して、トランスポート層セキュリティ (TLS) またはその古いバリエーション、Secure Socket Layer (SSL) を使用して安全に通信する機能の証明書を構成することができます。このタブは、利用できる機能がインストールされている場合 ([ThingWorx ネイティブインターフェイス](#)など) にのみ表示されます。

証明書ストア	
機能	ThingWorx Native Interface
<input checked="" type="checkbox"/> インスタンス証明書	
表示	表示
エクスポート	エクスポート
再発行	再発行
インポート	インポート
<input checked="" type="checkbox"/> 信頼ストアを管理	
証明書	<空>
表示	表示
エクスポート	エクスポート
削除	削除
<input checked="" type="checkbox"/> 信頼ストアを拡張	
インポート	インポート

「インスタンス証明書」

「表示」: 現在選択されている機能のインスタンス証明書を表示するには、「表示」リンクをクリックします。

「エクスポート」: 現在選択されている機能のインスタンス証明書を、ユーザーが選択したディレクトリに保存します。推奨されるファイル名は証明書の拇印ですが、ユーザーはこれを自由に変更することができます。出力は PEM エンコードされ、単一の証明書を含みます。

「再発行」: 現在選択されている機能のインスタンス証明書を再発行します。証明書ストアによって生成された証明書は自己署名であり、10 年で失効します。

「インポート」: 現在選択されている機能のインスタンス証明書をインポートします。このオプションを使用して、TLS/SSL ピアによって信頼されている証明機関によって署名された証明書をインポートします。

「信頼ストアを管理」

「証明書」: 信頼ストアには、証明書が含まれていない場合も、1 つ以上の証明書が含まれている場合もあります。ユーザーは、表示、エクスポート、または削除する証明書を選択する必要があります。

「表示」: 現在選択されている機能の現在選択されている信頼証明書を表示します。

「エクスポート」: 現在選択されている機能の現在選択されている信頼証明書をエクスポートします。インスタンス証明書と同様に、出力ファイルは PEM エンコードされ、これには 1 つの証明書が含まれます。

「削除」: 現在選択されている機能の現在選択されている信頼証明書を削除します。この機能は、信頼チェーンにこの証明書を含む証明書を提示するピアを信頼しなくなりました。

「信頼ストアを拡張」

「インポート」: 1 つまたは複数の証明機関または自己署名証明書を信頼ストアにインポートします。この機能は、この証明書を提示する TLS/SSL ピア、またはインポートされた証明書によって署名された証明書を信頼します。

インスタンス証明書のインポート動作

- ・ インポートファイルには、証明書および暗号化されていない秘密キーが含まれている必要があります。
- ・ 無効な署名が含まれている場合、証明書をインポートできません。
- ・ 証明書の有効期限が切れると、ユーザーにメッセージが表示されます。TLS/SSL ピアは、期限切れの証明書を拒否することがあります。

信頼証明書のインポート動作

- ・インポートファイルには、1 つ以上の証明書が含まれている必要があります。
- ・秘密キーは必要ありませんが、ファイルに存在していても問題はありません。
- ・1 つ以上の証明書に無効な署名がある場合、インポートは成功しません。
- ・1 つ以上の証明書がすでに信頼ストアに存在する証明書を複製した場合、インポートは成功しません。
- ・インポートファイル内のいずれかの証明書の有効期限が切れている場合、ユーザーに確認メッセージが表示されます。この機能は、信頼チェーン内の期限切れの証明書に依存する証明書を拒否する場合があります。

設定 - サービスポート

「管理」グループは、ランタイム管理の操作を設定するために使用します。「サービスポート」管理設定は、インストール時に自動的に設定されます。設定を更新する必要がある場合は、システムトレイにある「管理」アイコンを右クリックし、「設定」|「サービスポート」の順に選択して、サービスポートのシステム設定にアクセスします。

The screenshot shows a window titled 'Service Ports'. It has two main sections. The first section, 'Store and Forward', contains the text 'Enter the TCP/IP port number that allows communication with the ThingWorx Native Interface Store and Forward Service.' Below this is a text input field labeled 'Port:' and a button labeled 'Default'. The second section, 'Security', contains the text 'Enter the TCP/IP port number that allows communication with the Key Service.' Below this is a text input field labeled 'Preferred Port:' and a button labeled 'Default'.

蓄積転送

「ポート」: ストアアンドフォワードクライアントがストアアンドフォワードサービスとの通信に使用する TCP/IP ポートを指定します。有効な範囲は 1024 から 65535 です。デフォルトはサーバーによって設定されます。

「デフォルト」: クリックすると、このフィールドにデフォルトのポート番号が指定されます。

ヒント:

- ・別のサーバーアプリケーションで使用されていない場合は、デフォルトのポートを使用することをお勧めします。
- ・ストアアンドフォワードサービスはリモート接続を受け付けないため、このポート割り当てをファイアウォールに関連付けられないようにしてください。
- ・ユーザーによる SAF の有効化を許可するためのアクセス許可には、プロジェクトの変更が含まれます。[ユーザーマネージャ](#)を使用して、servermain プロジェクトを変更するための許可を、ユーザーまたはグループ (通常は匿名クライアント) に付与します。ThingWorx ユーザーは、[ユーザーマネージャ ThingWorx Interface Users](#)に記載されている説明に従い、ThingWorx Interface Users グループ経由で、同じアクセス許可を設定する必要があります。

●関連項目: [プロジェクトのプロパティ - ThingWorx](#)

セキュリティ

「推奨ポート」: キーサービスがサーバー内での通信に使用できる TCP/IP ポートを指定します。有効な範囲は 1024 から 65535 です。デフォルトはサーバーによって設定されます。何らかの理由で「推奨ポート」が使用できないか不適切な場合、サービスは別のポートのセキュリティを確保しようとします。

「デフォルト」: クリックすると、このフィールドにデフォルトのポート番号が指定されます。

ストアアンドフォワードサービス

ストアアンドフォワードサービスを使用すると、各種サーバーコンポーネントは一定期間中、データをローカルディスクに保存できます。このサービスは、ストアアンドフォワード機能を必要とするコンポーネントとともにインストールされます。ストアアンドフォワードサービスは、ストアアンドフォワードをサポートする機能に基づいて自動的に起動および停止します。

●関連項目:

[ThingWorx のプロジェクトのプロパティ](#)

[ストアアンドフォワードの構成設定](#)

[ストアアンドフォワード - システムタグ](#)

[ThingWorx のアクセス権](#)

ユーザーインターフェースのナビゲーション

構成はサーバーランタイムを操作する一般的な方法を提供します。さまざまなプラグインとドライバによってボタン、メニュー、およびアイコンが追加されますが、標準のインターフェース要素とその説明は次のとおりです。

タイトルバー

















アプリケーション名を表示します。コンフィギュレーションがランタイムに接続されている場合、および該当する場合は現在のランタイムプロジェクトが表示されます。

メニューバー

- 「**ファイル**」 「保存」、「開く」、「インポート」、「エクスポート」などのプロジェクトレベルのコマンドがあります。
- 「**編集**」 「コピー」、「貼り付け」、「新しいチャネル」などの操作コマンドがあります。
- 「**表示**」 表示コマンドがあります。たとえば、どのユーザーインターフェース要素を表示するか非表示にするか、どのタイプのツリー組織を表示するかなどを制御します。
- 「**ツール**」 一般オプション、接続設定、イベントログフィルタなどの構成コマンドがあります。
- 「**ランタイム**」 「接続...」、「切断」、「最初期化」などのサーバー接続性コマンドがあります。
- 「**ヘルプ**」 サーバー別、ドライバ別、またはプラグイン別の製品ドキュメントにアクセスするためのコマンドがあります。

ボタンバー

標準のボタンの説明は次のとおりです。プラグインとドライバにより、アクティブなアイテムとビューに使用できる機能に基づいて、ボタンが追加、除去、有効化、および無効化されます。

-  「**新しいプロジェクト**」: アクティブなプロジェクトを置き換える新しいプロジェクトファイルの作成を開始します。[プロジェクトファイル](#)により、接続されているデバイス、その接続されているデバイスの設定、それらをグループ化する方法が定義されます。
-  「**プロジェクトを開く**」: アクティブなプロジェクトを置き換えるためにロードする既存のプロジェクトファイルをブラウズできます。
-  「**プロジェクトを保存**」: 最近の変更を実装し、アクティブなプロジェクトファイルをディスクに書き込みます。
-  「**名前を付けて保存**」: アクティブなプロジェクトを変更とともに新しい場所またはファイル名などに書き込みます。
-  「**新しいチャネル**」: 新規グループまたはデータ収集のメディアを作成します。
-  「**新しいデバイス**」: データ収集の新しいハードウェアコンポーネントまたはPLCを定義します。
-  「**新しいタググループ**」: データポイントの新しいコレクション、または単一の単位として構成できるタグを定義します。
-  「**新しいタグ**」: コレクションの新しいデータポイントを定義します。
-  「**タググループ作成**」: ターゲットデバイスまたは環境で検出されたタグを定義します。
-  「**タグの複製**」: 選択したタグのコピーを作成します。
-  「**プロパティ**」: 選択したアイテムのパラメータを表示および編集できます。
-  「**元に戻す**」: 値またはアイテムを、最も最近の変更の前の構成にリセットします。
-  「**切り取り**」: 選択したアイテムを除去し、クリップボードに保存します。
-  「**コピー**」: 選択したアイテムの複製を作成し、クリップボードに保存します。
-  「**貼り付け**」: 現在クリップボードにあるアイテムを、選択した領域に挿入します。
-  「**削除**」: 選択したアイテムまたはその定義、あるいはその両方を除去します。



「Quick Client」: 統合されたクライアントインターフェースを実行します。

Figure 1-10 shows two screenshots of the Quick Client interface. The top screenshot displays the 'Project' tree on the left, listing various channels and devices. The main pane shows a table of tags, with 'Tag1' selected. The bottom screenshot shows the 'Event' log, listing recent events such as 'BACnet/IP Device Driver' and 'ABB Totalflow Device Driver'.

Top Screenshot: Project Tree and Tag Table

タグ名	アドレス	データ型	スキャン...	スケール...	説明
Tag1	AD00005	DWord	100	Square Root	

Bottom Screenshot: Event Log

日付	時刻	ソース	イベント
9/28/2016	10:24:29 AM	BACnet/IP	BACnet/IP デバイスドライバ 'V6.0.1956.0'
9/28/2016	10:24:29 AM	KEPServerEX\Runtime	BACnet/IP デバイスドライバを起動しています。
9/28/2016	10:24:29 AM	Allen-Bradley Micro800 Et...	Allen-Bradley Micro800 Ethernet デバイスドライバ 'V6.0.1956.0'
9/28/2016	10:24:29 AM	KEPServerEX\Runtime	Allen-Bradley Micro800 Ethernet デバイスドライバを起動しています。
9/28/2016	10:24:29 AM	ABB Totalflow	ABB Totalflow Device Driver 'V6.0.1956.0'
9/28/2016	10:24:29 AM	ABB Totalflow	Loaded TCI Toolkit extradata.ini file, revision 77.
9/28/2016	10:24:29 AM	KEPServerEX\Runtime	ABB Totalflow デバイスドライバを起動しています。
9/28/2016	10:24:29 AM	KEPServerEX\Runtime	プロジェクト 'C:\ProgramData\Kepware\KEPServerEX\V6\default.opf' のバックアップ

準備完了 Default User クライアント: 0 アクティブなタグ: 0 / 0

Bottom Screenshot: Project Tree and Tag Table

Tag Name	Address	Data Type	Scan Rate	Scaling	Description
Tag1	AD00005	DWord	100	Square Root	

Bottom Screenshot: Event Log

Date	Time	Source	Event
9/28/2016	10:24:29 AM	BACnet/IP	BACnet/IP Device Driver 'V6.0.1956.0'
9/28/2016	10:24:29 AM	Runtime	Starting BACnet/IP device driver.
9/28/2016	10:24:29 AM	Allen-Bradley Micro800 Et...	Allen-Bradley Micro800 Ethernet Device Driver 'V6.0.1956.0'
9/28/2016	10:24:29 AM	Runtime	Starting Allen-Bradley Micro800 Ethernet device driver.
9/28/2016	10:24:29 AM	ABB Totalflow	ABB Totalflow Device Driver 'V6.0.1956.0'
9/28/2016	10:24:29 AM	ABB Totalflow	Loaded TCI Toolkit extradata.ini file, revision 77.
9/28/2016	10:24:29 AM	Runtime	Starting ABB Totalflow device driver.
9/28/2016	10:24:29 AM	Runtime	Created backup of project C:\ProgramData\default.opf

Ready www.ptc.com Default User Clients: 0 Active tags: 0 of 0

プロジェクトツリービュー

このビューには、現在のプロジェクトのコンテンツ、組織、および設定が階層表示されます。プロジェクトツリービューは、プロジェクトのすべての側面が一元化された場所として設計されています。ノードを展開すると、デバイス、タググループ、またはタグレベルまでの詳細がドリルダウンで表示されます。機能とプラグインはツリービューではノードとして表示されるため、構成を1つの場所で簡単に行うことができます。ツリーの主なノードは次のとおりです。

プロジェクト - アクティブなプロジェクトのグローバル設定が保存および更新されます。

接続性 - チャネルとデバイスが表示され、右クリック操作を実行でき、詳細枠にプロパティが表示されます。

エイリアス - システムリソース、レガシーパス、複雑なルーティングへのマッピングを短く表すことができ、よりユーザーフレンドリであり、名前とショートカットはSCADAとの互換性があります。

Advanced Tags - 操作や解析をタグ処理内に構築して保存できます。これは独立した製品プラグインです。

Alarms & Events - システムモニターを定義して管理できます。これは独立した製品プラグインです。

DataLogger - ODBC 準拠のデータベースにデータを整理して保存できます。これは独立した製品プラグインです。

EFM Exporter - フローデータとトレンドデータを取り込んで調整できます。これは独立した製品プラグインです。

IDF for Splunk - データ管理とデータマイニングへのデータフィードを設定できます。これは独立した製品プラグインです。

IoT Gateway - エンタープライズシステム、モニター、解析との接続を管理します。これは独立した製品プラグインです。

Local Historian - データ収集、ログ、ストレージ、保存を定義します。これは独立した製品プラグインです。

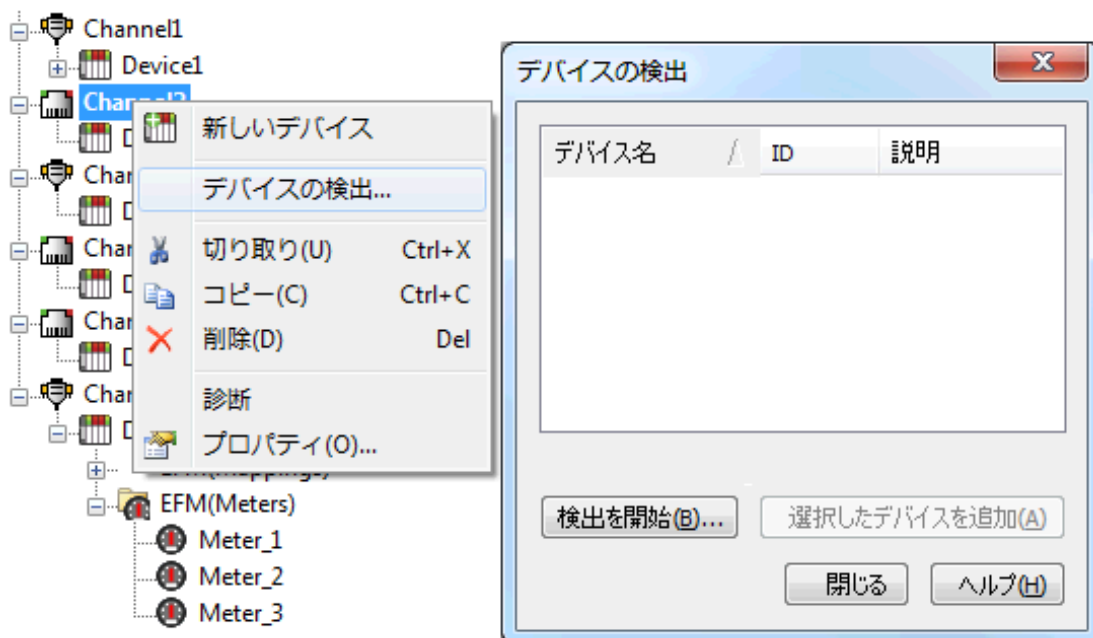
Scheduler - データ収集、パブリッシング、帯域幅管理を調整できます。これは独立した製品プラグインです。

SNMP Agent - 情報テクノロジーとSNMPプロトコルへの通信ブリッジを作成できます。これは独立した製品プラグインです。

非常に大規模なプロジェクトの場合、または一部の機能がほかの機能よりも頻繁に使用される場合は、ツリーをフィルタによってカスタマイズできます。「表示」メニューでツリーのノードの表示と非表示を切り替えます。

プロジェクトツリーの右クリックメニューには、状況に応じた各種オプションが表示されます。たとえば、デバイスやチャネルをコピーして貼り付けることで、既存の選択や設定に基づいて新しい構成を開始できます。名前は複製された後、一意になるように番号が追加されます(多数のアイテムを貼り付けた場合には増分します)。追加機能をサポートするドライバーでは、これらを右クリックメニューから実行することもできます。

たとえば、「**デバイスの検出**」では、アクセス可能なネットワークで互換性のあるデバイスがサーチされ、これらが自動的に追加されます。



詳細ビュー

このビューには、アクティブなプロジェクトに対していくつか用意されている構成選択オプションの1つが表示されます。その情報は、現在のプロジェクトツリービューに関連するものです。

● **注記:** 別のプロジェクトツリービューを選択しても、詳細ビューの列は、チャンネルまたはデバイスを選択するまで保持されます。チャンネルまたはデバイスを選択すると、これらの列が更新され、選択したチャンネルまたはデバイスに関連するデバイスまたはタグ情報が表示されます。

イベントログ

枠の最下部にあるこのビューには、ログに記録されたメッセージが表示されます。これには、サーバー、ドライバ、またはプラグインからの一般情報、セキュリティ警告、警告、およびエラーの4つのタイプがあります。デフォルトでは、ログエントリは日付、時刻、イベント発生元、およびイベントの説明で構成されています。詳細については、[イベントログオプション](#)を参照してください。

プロパティエディタ

プロパティエディタで一部のプロパティを編集できます。プロパティエディタの標準ボタンには次のような機能があります。

プロパティグループ 一般 スキャンモード	識別	
	名前	Device1
	説明	
	ドライバ	Simulator
	モデル	16 Bit Device
	チャンネル割り当て	Channel1
	ID フォーマット	10 進数
	ID	44
	動作モード	
	データコレクション	有効化

「**デフォルト**」: 選択したプロパティグループの設定をデフォルト値に戻します (適用された変更と保留中の変更の両方)。

「**OK**」: プロパティエディタを終了し、すべての変更を実装します。

「**キャンセル**」: 保留中の変更を実装しないでプロパティエディタを終了します。プロパティエディタを閉じて同じ効果があります。

「**適用**」: すべてのプロパティグループ内の保留中の変更を実装します。

「**ヘルプ**」: 選択したプロパティのヘルプを開きます。

● 保留中の変更は適用されるまで太字で表示されます。

ステータスバー

設定の現在のステータス (接続中、準備中など)、およびメニューバーとボタンバーアイテムのマウスカーソルのヒントが表示されます。

プロジェクトのプロパティ

構成から「プロジェクトのプロパティ」グループにアクセスするには、「編集」「プロジェクトのプロパティ」の順にクリックします。詳細については、以下のリストからリンクを選択してください。

- [プロジェクトのプロパティ - 一般](#)
- [プロジェクトのプロパティ - OPC DA](#)
- [プロジェクトのプロパティ - DDE](#)
- [プロジェクトのプロパティ - FastDDE/SuiteLink](#)
- [プロジェクトのプロパティ - iFIX PDB 設定](#)
- [プロジェクトのプロパティ - OPC UA](#)
- [プロジェクトのプロパティ - OPC AE](#)
- [プロジェクトのプロパティ - OPC HDA](#)
- [プロジェクトのプロパティ - ThingWorx](#)

プロジェクトのプロパティ - 一般

一般プロパティを使用して、タイトルとコメントを参照用としてプロジェクトに添付するとともに、プロジェクトのセキュリティ設定を管理します。「タイトル」フィールドに記述できる文字列は最大 64 文字ですが、「説明」フィールドは実質上無制限となっています。ただし、フィールドに入力できる程度に説明の長さを抑えることで、プロジェクトのロード時間が短くなります。

プロパティグループ	<input type="checkbox"/> 識別	
一般	説明	
OPC DA	タイトル	Simulation Driver Demo
OPC UA	定義されているタグ	1027
DDE	<input type="checkbox"/> プロジェクトファイルの暗号化	
OPC .NET	有効化	はい
OPC AE	プロジェクトパスワード	*****
OPC HDA		
ThingWorx		

識別

「説明」: レポート内やモニターシステムでこのプロジェクトを識別する際に役立つ語句を入力します (オプション)。

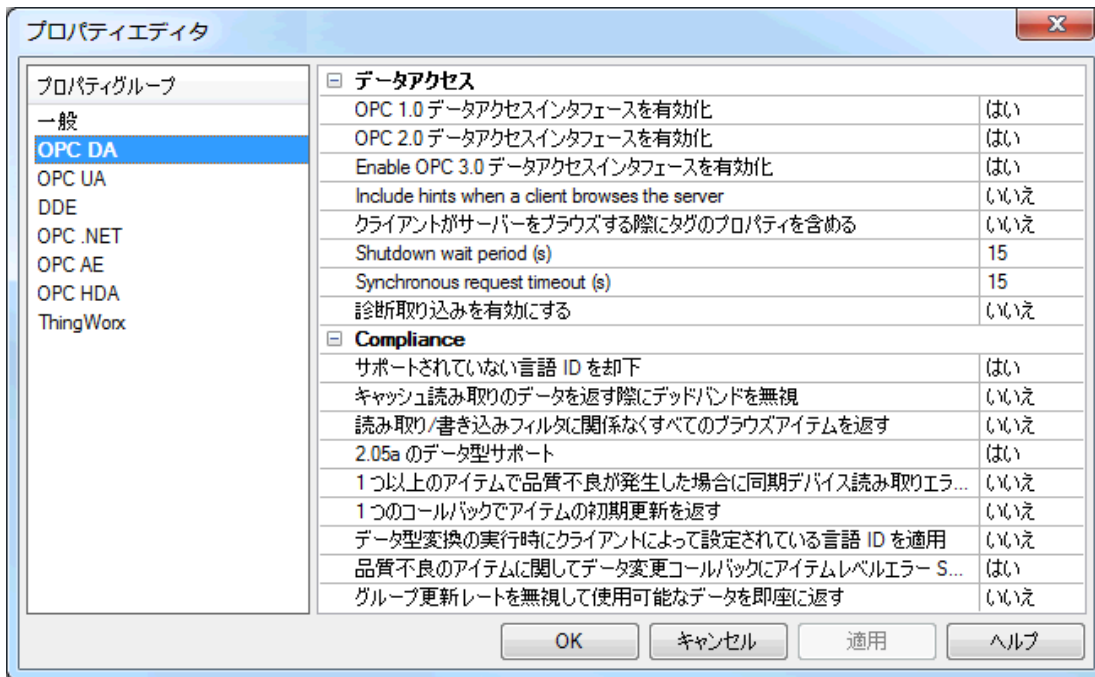
「タイトル」: ファイル名やレポートでこのプロジェクトを識別する単語または語句を入力します (オプション)。

「定義されているタグ」: このプロジェクト (および該当する場合にはライセンス) で予想されるデータコレクションにタグ数が対応していることを確認します。

●設定をデフォルト値/プリセット値に復元するには、「デフォルト」ボタンを使用します。

プロジェクトのプロパティ - OPC DA

このサーバーは、OPC Foundation の仕様との互換性が最大限に保たれるよう設計されています。その一方で、仕様に完全に準拠することと、すべての OPC クライアントアプリケーションと連携できることは、別の問題であることがテストで判明しています。「OPC DA コンプライアンス」ダイアログボックスでは、ごく一部の OPC クライアントのニーズに合わせて、サーバーの挙動をカスタマイズできます。これらのオプションは、大多数の OPC クライアントアプリケーションについては調整不要です。



「データアクセス」

「OPC 1.0 データアクセスインタフェースを有効化」: 「はい」を選択した場合、1.0 仕様をサポートしている OPC クライアントからの OPC クライアント接続をサーバーが受け入れることが許可されます。デフォルト設定では有効になっています。

「OPC 2.0 データアクセスインタフェースを有効化」: 「はい」を選択した場合、2.0 仕様をサポートしている OPC クライアントからの OPC クライアント接続をサーバーが受け入れることが許可されます。デフォルト設定では有効になっています。

「OPC 3.0 データアクセスインタフェースを有効化」: 「はい」を選択した場合、3.0 仕様をサポートしている OPC クライアントからの OPC クライアント接続をサーバーが受け入れることが許可されます。デフォルト設定では有効になっています。

「ブラウズする際にヒントを含める」: 「はい」を選択した場合、通信ドライバーごとに用意されているアドレスフォーマットのヒントを OPC クライアントアプリケーションでブラウズすることが許可されます。このヒントは、特定デバイスのデータの取り扱いに関するクイックリファレンスです。OPC クライアントから動的タグを入力する場合に活用できます。ヒントアイテムは、有効な OPC タグではありません。一部の OPC クライアントアプリケーションは、自身のタグデータベースに Hint タグを登録しようとする場合があります。この場合、クライアントはサーバーからエラーを受け取ります。このエラーは、ほとんどのクライアントでは問題になることはありませんが、クライアントによっては、タグの自動登録が停止するか、エラーが報告される場合があります。これらを防止するには、ヒントを無効にします。デフォルトでは無効（「いいえ」）に設定されています。

「ブラウズする際にタグのプロパティを含める」: 「はい」を選択した場合、アドレス空間内のタグごとに用意されているタグプロパティを OPC クライアントアプリケーションでブラウズすることが許可されます。デフォルトでは無効に設定されています。

「シャットダウン待機時間」: OPC クライアントがサーバーシャットダウンイベントから復帰するまで、サーバーが待機する時間の長さを指定します。クライアントアプリケーションがこのタイムアウト期限以内に復帰しない場合、サーバーは、シャットダウンと終了の処理を完了します。有効な範囲は 10 から 60 秒です。デフォルトの設定は 15 秒です。

「同期要求タイムアウト」: サーバーが同期読み取り操作の完了を待機する時間の長さを指定します。同期操作が進行している間にタイムアウト値を超えた場合、サーバーは、操作を強制的に完了してクライアントにエラーを通知します。これにより、同期操作の使用時にクライアントが異常停止することを防止できます。有効な範囲は 5 から 60 秒です。デフォルトの設定は 15 秒です。

● **注記:** 同期書き込みでは、このプロパティの設定値は使用されません。このプロパティが使用されるのは、読み取りまたはリクエストに限られます。

「診断取り込みを有効にする」: 「はい」を選択した場合、OPC 診断データのログを保存のためにイベントログに作成することが許可されます（通常はトラブルシューティングに利用します）。デフォルトでは無効（「いいえ」）に設定されています。

● OPC のデータアクセス 1.0、2.0、3.0 カスタム仕様の詳細については、OPC Foundation の Web サイト (www.opcfoundation.org) を参照してください。

コンプライアンス

「サポートされていない言語 ID を却下」:「はい」を選択した場合、サーバーでネイティブサポートされている言語 ID のみが許可されます。OPC クライアントアプリケーションがサーバーに OPC グループを追加しようとしたとき、一般エラーを受信した場合は、クライアントが、ネイティブでサポートされていない言語 ID をサーバーに提示した可能性があります。この場合、サーバーはグループの追加を拒否します。この問題を解決するには、コンプライアンス機能が無効にして、あらゆる言語 ID がサーバーで受け入れられるようにします。

「キャッシュ読み取りのデッドバンドを無視する」:「はい」を選択した場合、サーバーに追加されている OPC グループのデッドバンド設定をサーバーが無視します。一部の OPC クライアントでは、デッドバンドに関して、適切な値を渡すと(たとえば、更新頻度が低いように見える、あるいはまったく更新されていないように見える場合であっても、適切なデータを保持していると)問題が生じます。この状態に陥ることはまれです。したがって、通常、このオプションはデフォルトの無効状態のままにしておいてください。

「ブラウズフィルタを無視する」:「はい」を選択した場合、ブラウザ要求が発行されたとき、サーバーは、OPC クライアントのタグブラウザに適用されているアクセスフィルタにかかわらず、すべてのタグを OPC クライアントアプリケーションに返します。

「2.05a のデータ型サポート」:「はい」を選択した場合、2.05a 仕様に追加された、データ型の要件およびデータ型強制に関して期待される挙動をサーバーが遵守します。

「品質不良でエラーを返す」:「はい」を選択した場合、同期的なデバイス読み取りで、1 つ以上のアイテムに関して読み取り結果の品質が不良であった場合、サーバーはエラーを返します。仕様に準拠する場合は、1 つ以上のアイテムのデータに品質不良もしくは品質不明のものが含まれている場合も、要求の完遂を示す成功通知をサーバーが返すことが要件です。

「初期更新をグループ化する」:「はい」を選択した場合、サーバーは、未解決の初期アイテム更新を 1 つのコールバックですべて返すようになります。無効になっている場合、サーバーは、利用可能になった初期更新のみを返します(コールバックが複数になる可能性があります)。

●このオプションを有効にすると、一方的に送信されるデバイスプロトコルに関して、データのバッファリング(イベントのプレイバック)をサポートするドライバーを使用している場合、バッファ済みのデータが失われる恐れがあります。バッファ済みデータの喪失が懸念事項となる場合は、このコンプライアンスオプションを無効にしてください。

「クライアントロケールを適用する」:「はい」を選択した場合、データ型変換の実行時に、サーバーで稼働中の Windows オペレーティングシステムのロケール ID を使用するのか、OPC クライアントによって設定されるロケール ID を使用するのかが、このオプションによって決定されます。たとえば、浮動小数点数を表す 1,200 などの文字列は、英語メトリックスを使用している場合、変換が発生すると 1,200 に変換されますが、ドイツ語メトリックスを使用している場合は、変換が発生すると 1.2 になります。英語版の OS でドイツ語版のソフトウェアを実行している場合は、コンマがどのように取り扱われるのかをユーザーが決定する必要があります。この設定値によって、そうした柔軟性を確保できます。デフォルトでは、これまでの実装を背景として、サーバーのオペレーティングシステムのロケール ID が優先されます。

「品質不良のアイテムに S_FALSE を返す」:「はい」を選択した場合、品質不良のアイテムに関して、アイテムエラー配列でサーバーが S_FALSE を返すようになります。この値は、仕様に完全に準拠するよう設定されている既存プロジェクトの場合、デフォルトでは「はい」に設定され、それ以外のプロジェクトでは「いいえ」に設定されます。「いいえ」に設定すると、以前と同様に E_FAIL (0x80004005) を返すという挙動になります。

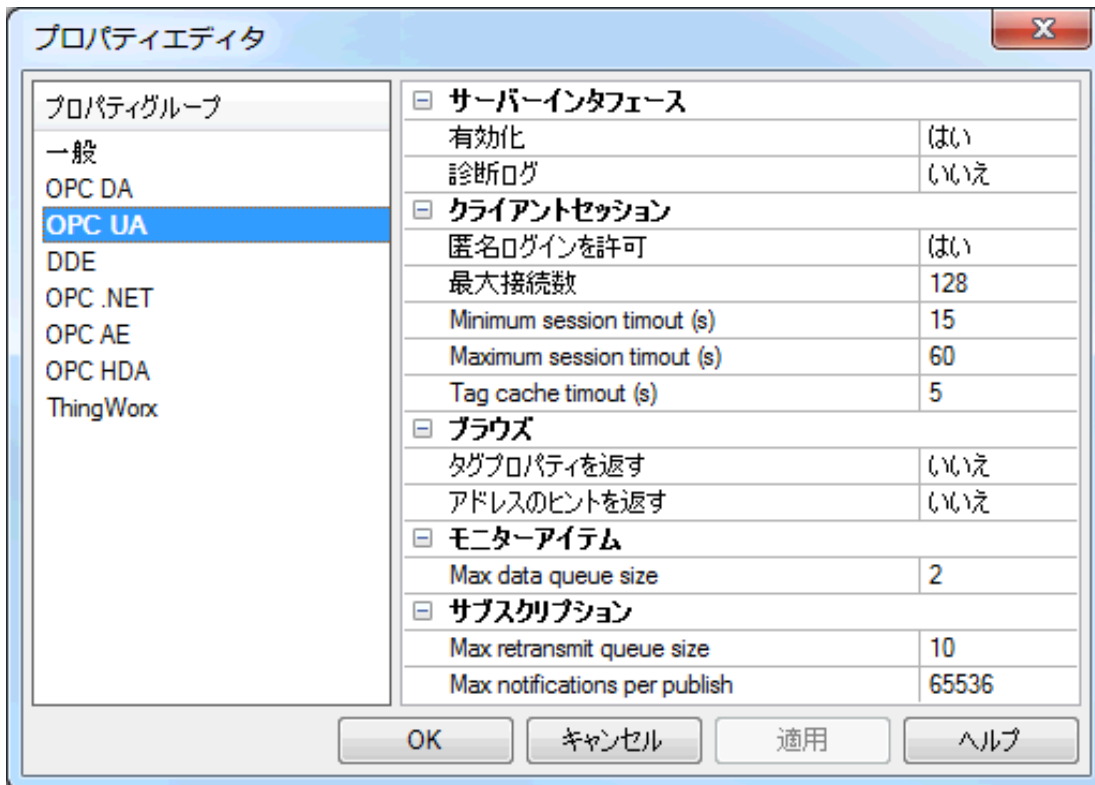
「データを即座に返す」:「はい」を選択した場合、クライアントを更新するすべてのグループが有効になります。このオプションが有効になっている場合は、アクティブなアイテムの値または品質に変化が生じた時点で、クライアント更新が開始されます。当該のグループに追加されているアイテムに関する、クライアントの要求によるスキャンの頻度は、クライアントによって指定されたグループ更新頻度を使用して設定されます。デフォルトでは無効に設定されています。

●設定をデフォルト値/プリセット値に復元するには、「デフォルト」ボタンを使用します。

プロジェクトのプロパティ - OPC UA

OPC Unified Architecture (UA) は、プラットフォームに依存しない、相互運用の基準を提供するものです。OPC のデータアクセス (DA) テクノロジーを置き換えるものではなく、大多数の業種別アプリケーションにとっては、既存の DA アーキテクチャを補完または拡張する存在です。「OPC UA」グループには、サーバーの現在の OPC UA 設定が表示されます。

●**注記:** 設定を変更するには、個々のプロパティの 2 番目の列をクリックします。ドロップダウンメニューが表示され、選択できるオプションが表示されます。



サーバーインタフェース

「**Enable**」: 有効にすると、UA サーバーインタフェースが初期化され、クライアント接続が受け入れられるようになります。無効にしている場合、このページの残りのプロパティは無効になります。

「**診断ログ**」: 有効にすると、OPC UA スタックの診断情報が OPC 診断ビューアに記録されます。このプロパティは、トラブルシューティングを目的とする場合のみ有効にしてください。

クライアントセッション

「**匿名ログインを許可**」: このプロパティでは、接続を確立する際にユーザー名とパスワードが必要であるかどうかを指定します。セキュリティのため、デフォルト設定は「いいえ」であり、匿名アクセスは許可されず、資格証明が必要です。

● **注記**: この設定が無効の場合、ユーザーは、ユーザーマネージャでデフォルトユーザーとしてログインできません。ユーザーマネージャでパスワードが設定されており、ログインに使用されている場合、ユーザーは管理者としてログインできます。

● クライアントが接続時にパスワードを入力すると、サーバーはエンドポイントのセキュリティポリシーによって定義されている暗号化アルゴリズムを使用してパスワードを解読します。

「**最大接続数**」: サポートされる接続の最大数を指定します。有効な範囲は 1 から 128 です。デフォルトの設定は 128 です。

「**セッションのタイムアウトの最小時間**」: セッションの確立に関して、UA クライアントのタイムアウト下限を指定します。値は、アプリケーションでのニーズに応じて変更できます。デフォルト値は 15 秒です。

「**セッションのタイムアウトの最大時間**」: セッションの確立に関して、UA クライアントのタイムアウト上限を指定します。値は、アプリケーションでのニーズに応じて変更できます。デフォルト値は 60 秒です。

「**タグキャッシュのタイムアウト**」: タグキャッシュのタイムアウトを指定します。有効な範囲は 0 から 60 秒です。デフォルトの設定は 5 秒です。

● **注記**: このタイムアウトは、UA クライアントがタグの使用を完了した後、タグがキャッシュされる期間の長さを制御するものです。UA クライアントが、設定済みの間隔で未登録タグへの読み取りまたは書き込みを実行する場合は、タイムアウトを大きくすることにより、パフォーマンスを改善できます。たとえば、クライアントが 5 秒間隔で未登録タグを読み取る場合は、タグキャッシュのタイムアウトを 6 秒に設定してください。クライアントからのリクエストごとにタグが必ず再作成されることがなくなり、パフォーマンスが向上します。

ブラウズ

「**タグプロパティを返す**」: 有効にすると、アドレス空間のタグごとに用意されているタグプロパティを、UA クライアントアプリケーションでブラウズできるようになります。この設定は、デフォルトでは無効になっています。

「**アドレスのヒントを返す**」: 有効にすると、アイテムごとに用意されているアドレスフォーマットのヒントを、UA クライアントアプリケーションでブラウズできるようになります。ヒントは有効な UA タグではありませんが、ある一定の UA クライアントアプリケーションは、タグデータベースにタグを追加しようとする場合があります。この場合、クライアントはサーバーからエラーを受け取ります。このエラーが原因となって、クライアントがエラーを報告するか、タグの自動追加を停止する場合があります。この動作を防止するには、このプロパティを必ず無効にしてください。この設定は、デフォルトでは無効になっています。

モニターアイテム

「**データキューの最大サイズ**」: アイテムに関して、キューに格納するデータ通知数の最大値を指定します。有効な範囲は 1 から 100 です。デフォルトの設定は 2 です。

● **注記**: データキューは、監視対象アイテムの更新レートがサブスクリプションのパブリッシングレートよりも高い場合に使用されます。たとえば、監視対象アイテムの更新レートが 1 秒であり、サブスクリプションのパブリッシング頻度が 10 秒である場合、10 秒ごとに、アイテムに関して 10 個のデータ通知がパブリッシングされます。データのキューイングではメモリが消費されるため、メモリが懸念事項である場合は、この値を大きくしすぎないようにしてください。

サブスクリプション

「**再転送キューの最大サイズ**」: キューに格納されるパブリッシングの数について、サブスクリプションあたりの最大値を指定します。有効な範囲は 1 から 100 です。値を 0 にすると、再送信は無効になります。デフォルトの設定は 0 です。

● **注記**: サブスクリプションのパブリッシングイベントは、クライアントからのリクエストに応じてキューに保存され、再送信されます。キューイングではメモリが消費されるため、メモリが懸念事項である場合は、この値を大きくしすぎないようにしてください。

「**パブリッシングあたりの最大通知数**」: パブリッシングあたりの通知の数について、最大値を指定します。有効な範囲は 1 から 65536 です。デフォルトの設定は 65536 です。

● **注記**: この値によって、サーバーからクライアントに送信されるパケットのサイズが制限されることにより、接続のパフォーマンスに影響を受ける可能性があります。一般に、高帯域幅の接続には大きな値を、低帯域幅の接続には小さな値を使用してください。

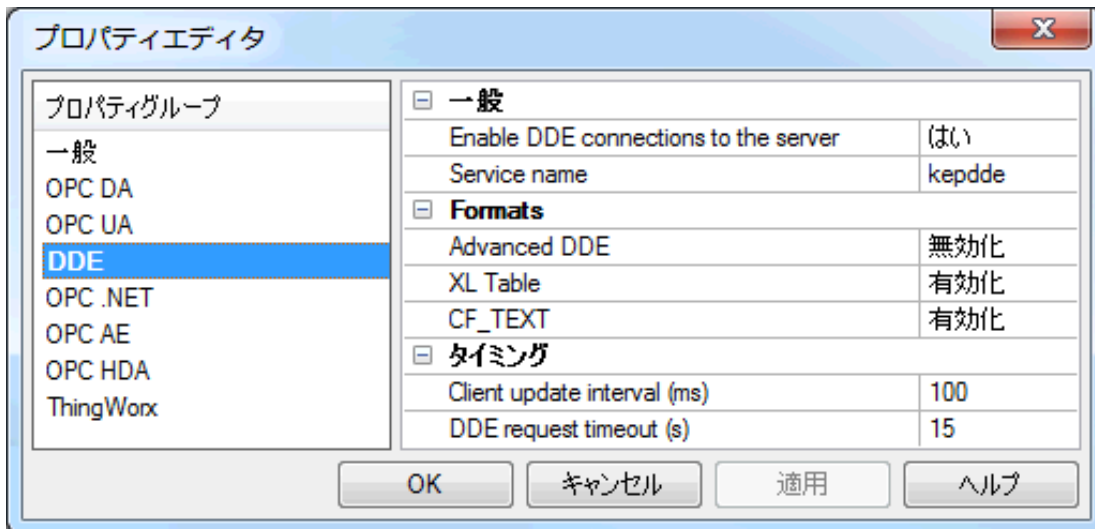
● 設定をデフォルト値/プリセット値に復元するには、「**デフォルト**」ボタンを使用します。

プロジェクトのプロパティ - DDE

サーバーは OPC サーバーですが、一部のアプリケーションはデータを共有するために**動的データ交換 (DDE)**を必要とします。サーバーは、**CF_Text**、**XL_Table**、および **Advanced DDE** のいずれかの DDE フォーマットをサポートする DDE アプリケーションへのアクセスを提供します。CF_Text と XL_Table は、Microsoft によって開発された標準の DDE フォーマットであり、DDE を認識するすべてのアプリケーションで使用できます。Advanced DDE は、産業市場に特有のクライアントアプリケーションの多くでサポートされている高性能フォーマットです。

● DDE インタフェースがサーバーに接続するには、ランタイムがデスクトップと対話することが許可されている必要があります。詳細については、[デスクトップとの対話を許可する方法](#)を参照してください。

構成を介して DDE サーバーの設定にアクセスするには、「**編集**」|「**プロジェクトのプロパティ**」の順にクリックし、DDE プロパティを見つけます。これらのプロパティを使用して、アプリケーションの要件に合うように DDE 操作を調整できます。



一般

「サーバーへの DDE 接続を有効化」: このプロパティでは、サーバーの DDE サーバー部分を有効にするか無効にするかを決定します。DDE 操作が無効になっていると、サーバーは DDE データの要求に応答しません。サーバーを OPC サーバーとしてのみ使用する場合は、DDE 操作を無効にすることをお勧めします。これにより、データセキュリティとサーバーの全体的なパフォーマンスを向上させることができます。DDE はデフォルトでは無効になっています。

● 関連項目: [サーバーで DDE を使用する方法](#)

「サービス名」: このプロパティにより、サーバーが DDE クライアントに対してどのようにアプリケーション名として表示されるかを変更できます。この名前は、最初は、前のバージョンのサーバーとの互換性が確保されるように設定されます。既存のサーバーを置き換える必要がある場合は、サーバーのサービス名を、置き換える DDE サーバーと一致するように変更できます。1 ～ 32 文字の文字列をサービス名として入力できます。

フォーマット

このプロパティにより、クライアントアプリケーションで使用する DDE フォーマットを構成できます。「Advanced DDE」、「XL Table」、および「CF_Text」を有効または無効にします。デフォルトでは、すべてのフォーマットが有効になっています。これは、DDE クライアントアプリケーションからサーバーに接続する際に問題が発生した場合に特に役立ちます。これらの DDE フォーマットを個別に無効にすることにより、特定のフォーマットをテストする目的で分離できます。

● 注記: DDE を認識するすべてのアプリケーションで、少なくとも CF_Text がサポートされています。

タイミング

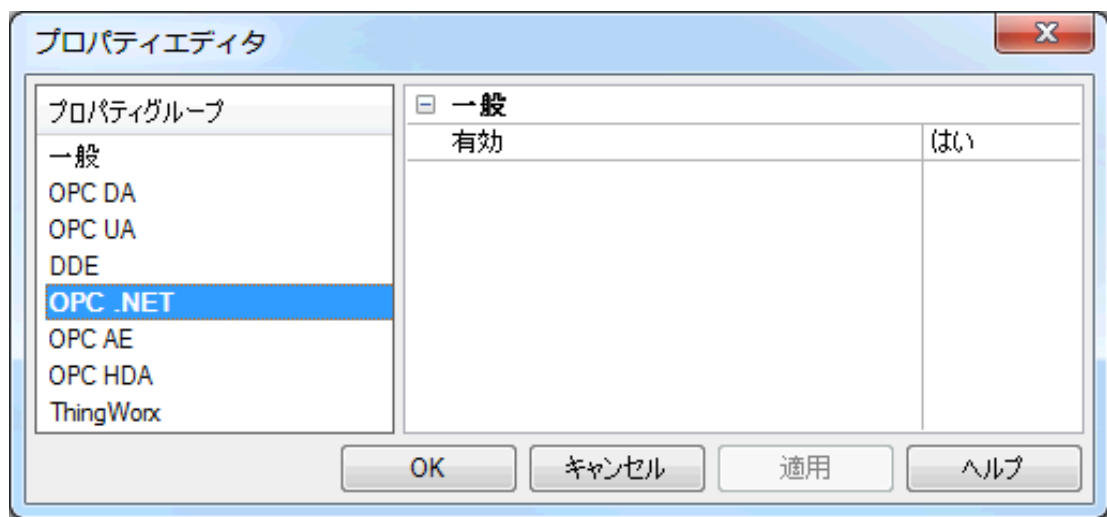
「クライアント更新間隔」: この間隔設定により、DDE データを 1 つにまとめてクライアントアプリケーションに転送できます。DDE フォーマットを使用することによるパフォーマンスの向上は、大規模なサーバーデータブロックを単一の DDE 応答で送信できる場合にのみ達成されます。サーバーが大規模なデータブロックを収集する能力を向上させるために、更新タイマーを設定して、新しいデータがクライアントアプリケーションに送信されるまでプールに蓄積されるようにできます。更新タイマーの有効な範囲は 20 から 60000 ミリ秒です。デフォルトの設定は 100 ミリ秒です。

「DDE 要求タイムアウト」: このプロパティは、DDE 要求の完了のタイムアウトを構成するために使用されます。サーバー上の DDE クライアント要求 (読み取りまたは書き込み操作) が指定したタイムアウト内に完了しなかった場合は、DDE クライアントにエラーが返されます。有効な範囲は 1 から 30 秒です。デフォルトの設定は 15 秒です。

● 注記: 変更を有効にするためにサーバーランタイムを再初期化する必要がある場合があります。

プロジェクトのプロパティ - OPC .NET

構成を介して OPC .NET サーバーの設定にアクセスするには、「編集」|「プロジェクトのプロパティ」の順にクリックし、「OPC .NET」タブを選択します。



「有効」: 有効にすると、OPC .NET ラッパーが初期化され、クライアント接続が受け入れられるようになります。

● ヒント:

1. OPC .NET ラッパーは、"xi_server_runtime.exe" と呼ばれるシステムサービスとして実行されます。このラッパーが起動されるのは、サーバーが起動し、上で説明したオプションが有効になっている場合のみです。OPC DA とは異なり、クライアントはサーバーを起動できません。
2. OPC .NET をインストールして使用するには、サーバーをインストールする前に、マシンに Microsoft .NET 3.5 が存在している必要があります。

● 設定をデフォルト値/プリセット値に復元するには、「デフォルト」ボタンを使用します。

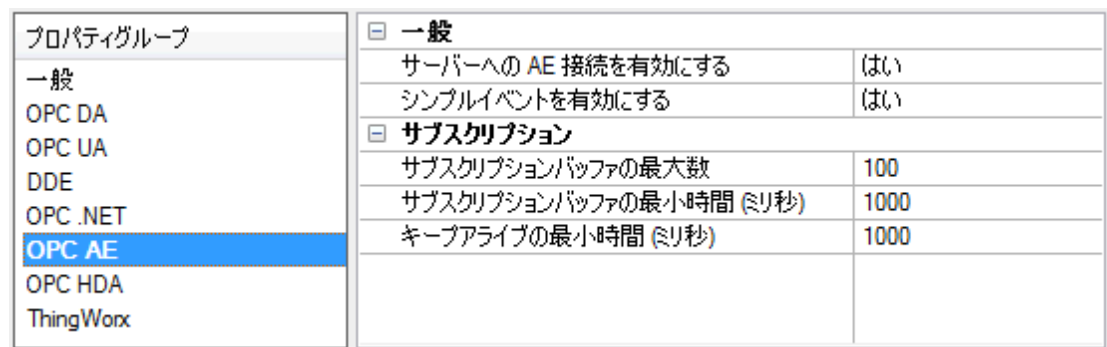
プロジェクトのプロパティ - OPC AE

イベントは、サーバーで発生した現象を通知するために使用されるもので、OPC データアクセスでのデータ更新と似ています。OPC AE 機能では、システムの起動と停止のメッセージ、警告、エラーなどを含め、単純なイベントをサーバーから受信できます。これらのイベントは、イベントログに表示されます。

プロジェクトレベルで AE 設定の数を指定するには、OPC AE のグループを使用します。これらの設定に加えた変更は、すべてのアラーム & イベント (A&E) クライアントがサーバーから切断された後に適用されます。

Alarms & Events プラグインを使用すると、A&E クライアントで A&E のデータを OPC サーバーから受信できます。このプラグインは、OPC サーバーのイベントを A&E フォーマットに変換するとともに、OPC サーバータグを使用してカスタムアラームを作成するために使用します。

● 詳細については、OPC ベンダーに問い合わせてください。



一般

「サーバーへの AE 接続を有効にする」: このプロパティによって、OPC AE サーバーのオンとオフを切り替えます。

「**シンプルイベントを有効にする**」: 有効にすると、クライアントが単純なイベントを利用できるようになります。無効にした場合は、通常のイベントが送信されます。デフォルト設定では有効になっています。

サブスクリプション

「**サブスクリプションバッファの最大数**」: 1 回の送信呼び出しでクライアントに送信されるイベント数について、最大値を指定します。範囲は 0 から 1000 です。デフォルトの設定は 100 です。0 は、制限しないことを意味します。

「**サブスクリプションバッファの最小時間**」: クライアントへの送信呼び出しの間隔について、最小値を指定します。サポートされる範囲は 100 から 60000 ミリ秒です。デフォルトの設定は 1000 ミリ秒です。

「**キープアライブの最小時間**」: サーバーからクライアントに送信されるキープアライブメッセージの間隔について、最小値を指定します。サポートされる範囲は 100 から 60000 ミリ秒です。デフォルトの設定は 1000 ミリ秒です。

● 設定をデフォルト値/プリセット値に復元するには、「**デフォルト**」ボタンを使用します。

プロジェクトのプロパティ - FastDDE/SuiteLink

サーバーは Wonderware Corporation の FastDDE と SuiteLink をサポートしているため、簡単にサーバーを FactorySuite アプリケーションに接続できます。Wonderware 接続性ツールキットを使用すると、OPC と FastDDE/SuiteLink の接続性が同時に提供されるとともに、中間でブリッジソフトウェアを使用することなくデバイスデータに素早くアクセスできます。

● **FastDDE** インタフェースがサーバーに接続するには、ランタイムがデスクトップと対話することが許可されている必要があります。詳細については、[デスクトップとの対話を許可する方法](#)を参照してください。

● **注記**: FastDDE/SuiteLink を適切に動作させるには (およびこのタブを「プロジェクトのプロパティ」で使用するには)、Wonderware FS2000 Common Components または InTouch Runtime Component バージョン 8.0 以降が PC にインストールされている必要があります。

Property Groups	<div> <div>General</div> <div>OPC DA</div> <div>OPC UA</div> <div>DDE</div> <div>FastDDE/SuiteLink</div> <div>OPC AE</div> <div>OPC HDA</div> <div>ThingWorx</div> </div>	
	<div> <div>General</div> <div>Enable FastDDE/SuiteLink connections to the server</div> <div>Yes</div> </div>	
	<div> <div>Application name</div> <div>server_runtime</div> </div>	
	<div> <div>Timing</div> <div>Client update interval (ms)</div> <div>100</div> </div>	

「**FastDDE/SuiteLink のサーバーへの接続を有効にする**」: このプロパティでは、クライアント/サーバープロトコルのサポートを有効または無効にします。Wonderware 製品が PC にインストールされている場合、この設定を有効にできます。FastDDE/SuiteLink 操作が無効になっていると、サーバーは FastDDE または SuiteLink データに対するすべての要求に応答しません。

● **ヒント**: より高いパフォーマンスとセキュリティを確保するため、サーバーが OPC の接続性だけにのみ使用されている場合は、この設定を無効にすることをお勧めします。

「**アプリケーション名**」: アプリケーションの名前を指定します。デフォルト設定は "server_runtime" です。

● **注記**: この名前は、特定のエンドユーザーのニーズに合わせてカスタマイズできます。たとえば、インストール中に「**除去およびリダイレクト**」を選択した場合に、特定の FactorySuite アプリケーションを修正することなく機能させるには、この設定を "servermain" に変更する必要があります。

「**クライアント更新間隔 (ミリ秒)**」: 新しいデータが FastDDE/SuiteLink クライアントアプリケーションに送信される頻度を指定します。範囲は 20 から 32000 ミリ秒です。デフォルトの設定は 100 ミリ秒です。タイマーを使用することで、FastDDE/SuiteLink データを 1 つにまとめてクライアントアプリケーションに転送できます。FastDDE や SuiteLink などのクライアント/サーバープロトコルを使用することによるパフォーマンスの向上は、大規模なサーバーデータブロックを単一の応答で送信できる場合にのみ達成されます。サーバーが大規模なデータブロックを収集する能力を向上させるために、更新タイマーを設定して、新しいデータがクライアントアプリケーションに送信されるまでブールに蓄積されるようにできます。

● **注記**:

1. 更新レートは、データがデバイスから読み取られる頻度ではなく、データがクライアントアプリケーションに送信される頻度を制御します。スキャン速度を使用して、接続されているデバイスからサーバーがデータを取得する速度の増減を調整できます。詳細については、[タグのプロパティ - 一般](#)を参照してください。
2. 変更を有効にするためにサーバーランタイムの再初期化が必要になる場合があります。

● 設定をデフォルト値/プリセット値に復元するには、「**デフォルト**」ボタンを使用します。

プロジェクトのプロパティ - iFIX PDB 設定

「iFIX PDB 設定」ダイアログには、iFIX プロセスデータベース (PDB) タグとサーバータグの処理の間の動作を調整するプロパティが含まれています。アクセスするには、「**編集**」|「**プロジェクトのプロパティ**」の順にクリックします。

● **注記:** 「iFIX PDB 設定」は、iFIX がコンピュータにインストールされている場合にのみ「プロジェクトのプロパティ」に表示されます。

● iFIX PDB インタフェースをランタイムと連動させるために、「プロセスモード」を「システムサービス」に設定する必要がある場合があります。詳細については、[プロセスモード](#)を参照してください。

Property Groups General iFIX PDB Settings	[-] General	
	Enable connectivity to iFIX PDB	Yes
	Enable latched data	No
	Enable update per poll	No
	Use iFIX startup configuration file	Yes
	Use unconfirmed updates	No
	[-] Timing	
	PDB-to-server request timeout (s)	5
	Deactive tags on PDB read inactivity	Yes
	Inactivity timeout (s)	15

● **注記:** 各フィールドをデフォルト値のままに維持することをお勧めします。また、設定がアプリケーションの要件を満たしていることを確認することもお勧めします。

一般

「**iFIX PDB への接続性を有効にする**」: クライアント/サーバープロトコルのサポートを有効または無効にします。iFIX PDB 操作が無効になっていると、サーバーは iFIX PDB データに対するすべての要求に応答しません。より高いパフォーマンスとセキュリティを確保するため、サーバーが OPC の接続性にのみ使用されている場合は、このプロパティを無効にします。

「**ラッチされたデータを有効にする**」: 通常、通信障害が発生すると、iFIX アプリケーションのデータリンクに一連の疑問符 ("????") などが表示されます。ただし、常に値が表示されているほうが適切な場合があります。ラッチされたデータを有効にすると、最後に正常に読み取られた値が画面に表示されたままになります。デフォルト設定では有効になっています。

● **注記:** AR および DR ブロックのデータラッチングはサポートされていません。

「**ポーリングごとの更新を有効にする**」: これが有効になっていると、ドライバーがデバイスのポーリングを行うたびに、サーバーは現在の値、品質、およびタイムスタンプを iFIX に配信します。これが無効になっていると、サーバーは値または品質が変更されたことを確認したときに更新のみを iFIX に配信します。デフォルトでは無効に設定されています。

● **注記:** これは動的な設定であるため、このオプションを適用すると、サーバーはただちにデバイスのスキャン速度で更新を iFIX クライアントに配信します。

「**iFIX 起動構成ファイルを使用する**」: iFIX クライアントがアクセスするすべてのアイテムを含んでいるこのファイルを iFIX で作成する場合は、これを有効にします。これは、iFIX がアイテムデータを要求する前に自動的にアイテムのスキャンを開始します。デフォルト設定では有効になっています。

● **関連項目:** [iFIX アプリケーションのプロジェクトの起動](#)

「**未確定の更新を使用する**」: サーバーが NIO インタフェースを介した書き込みに続いて iFIX のローカルキャッシュを更新する方法を制御します。デフォルト設定 (無効) では、サーバーは読み取りによって値が確定されるまでローカルキャッシュを更新しません。ほとんどのアプリケーションの場合、データの整合性の観点からは、デフォルト設定で操作することが最適です。iFIX Easy Database Access (EDA) を活用しているアプリケーションの場合は、未確定の更新を有効にして、試された書き込み値ですぐに iFIX のローカルキャッシュを更新することをお勧めします。

● **注記:** データの整合性の観点からは、未確定の更新を使用することによって、書き込みが正常に行われたという誤った情報が示され、iFIX に不正確なデータが表示されることがあります。また、未確定の更新を使用することにより、iFIX に表示されるデータが、一時的に未確定の更新 (試された書き込み値) の後に確定済みの更新 (実際に読み取られたアイテムの値) が続くことが原因で "明滅" する場合があります。

タイミング

「PDB に対するサーバー要求のタイムアウト」: iFIX PDB が追加、除去、読み取り、または書き込み要求からの応答を待ち、それがタイムアウトするまでの時間を指定します。タイムアウトすると、要求はサーバーに代わって自動的に破棄されます。タイムアウトは、サーバーがほかの要求を処理しているためビジー状態である場合や、サーバーが iFIX PDB との通信を失った場合に発生します。通信を失った場合は、タイムアウトがさらに発生しないように、iFIX PDB が自動的にサーバーとの通信を再確立します。有効な範囲は 5 から 60 秒です。デフォルトの設定は 5 秒です。

「PDB の読み取りが非アクティブの場合にタグを非アクティブにする」: サーバーが、指定した時間内に iFIX によって読み取られなかったタグを自動的に非アクティブにします。これにより、プロセスハードウェアの不要なポーリングが低減されます。有効にした場合、サーバーは指定した間隔で (この図では 15 秒) タグのリストを読み取り、アイドル状態のものがあれば非アクティブにします。指定した時間内に iFIX がタグの読み取り要求を実行しなかった場合、タグはアイドル状態と見なされます。サーバーは 15 秒間隔でアイドル状態のタグがあるかどうかをチェックするため、タグが最後に読み取られてからちょうど 15 秒後に非アクティブにならない場合があります。チェック間隔内のどの時点で最後の読み取りが行われたかによっては、非アクティブ化が最大で 15 秒遅くなる可能性があります。iFIX が、非アクティブになったタグからのデータを要求すると、サーバーはそのタグを再びアクティブにしてハードウェアのポーリングを再開します。デフォルトでは無効に設定されています。この機能は、有効にすると、すべてのプロジェクトに適用されます。15 から 607999 (15 秒から 1 週間) の範囲でアイドル時間を指定できます。

● この機能は、レジスタタグのみを使用対象としているため、その他のタグはスキャン停止になる可能性があります。この機能を使用する場合に、この状況を回避するには、非アクティブ化タイマーを、iFIX データベースで構成されている最も長いスキャン時間よりも長い時間に設定します。

「不活動タイムアウト」: iFIX PDB がアクティビティを待ち、それがタイムアウトするまでの時間を指定します。通信を失った場合は、タイムアウトがさらに発生しないように、iFIX PDB が自動的にサーバーとの通信を再確立します。有効な範囲は 5 から 60 秒です。デフォルトの設定は 5 秒です。

● 設定をデフォルト値/プリセット値に復元するには、「デフォルト」ボタンを使用します。

プロジェクトのプロパティ - OPC HDA

構成を介して OPC HDA サーバーの設定にアクセスするには、「編集」「プロジェクトのプロパティ」の順にクリックし、「OPC HDA」グループを展開します。

プロパティグループ	一般	
一般		
OPC DA		
OPC UA		
DDE		
OPC .NET		
OPC AE		
OPC HDA		
ThingWorx		
	サーバーへの HDA 接続を有効にする	はい
	診断を有効にする	いいえ

「サーバーへの HDA 接続を有効にする」: 有効にすると、このサーバーによって公開されている HDA サーバーに HDA クライアントが接続できるようになります。無効にすると、クライアントの HDA 接続が無効になります。これらの設定はランタイムを再起動することなく適用できますが、サーバーは、接続済みのクライアントを接続解除することはないものの、新しいクライアント接続も受け入れなくなります。デフォルト設定では有効になっています。

「Enable Diagnostics」: 有効にすると、OPC HDA のデータを取り込んで、保存用のログをイベントログサービスに作成できます。デフォルトでは無効に設定されています。

● **注記:** 診断を有効にすると、サーバーの実行時パフォーマンスが低下します。イベントログの作成の詳細については、[OPC 診断ビューア](#)を参照してください。

● 設定をデフォルト値/プリセット値に復元するには、「デフォルト」ボタンを使用します。

プロジェクトのプロパティ - ThingWorx

ThingWorx ネイティブインタフェースがサポートされているため、ThingWorx プラットフォームとの接続の作業が簡潔になるとともに、必要に応じて OPC およびその他に同時接続することが可能になっています。

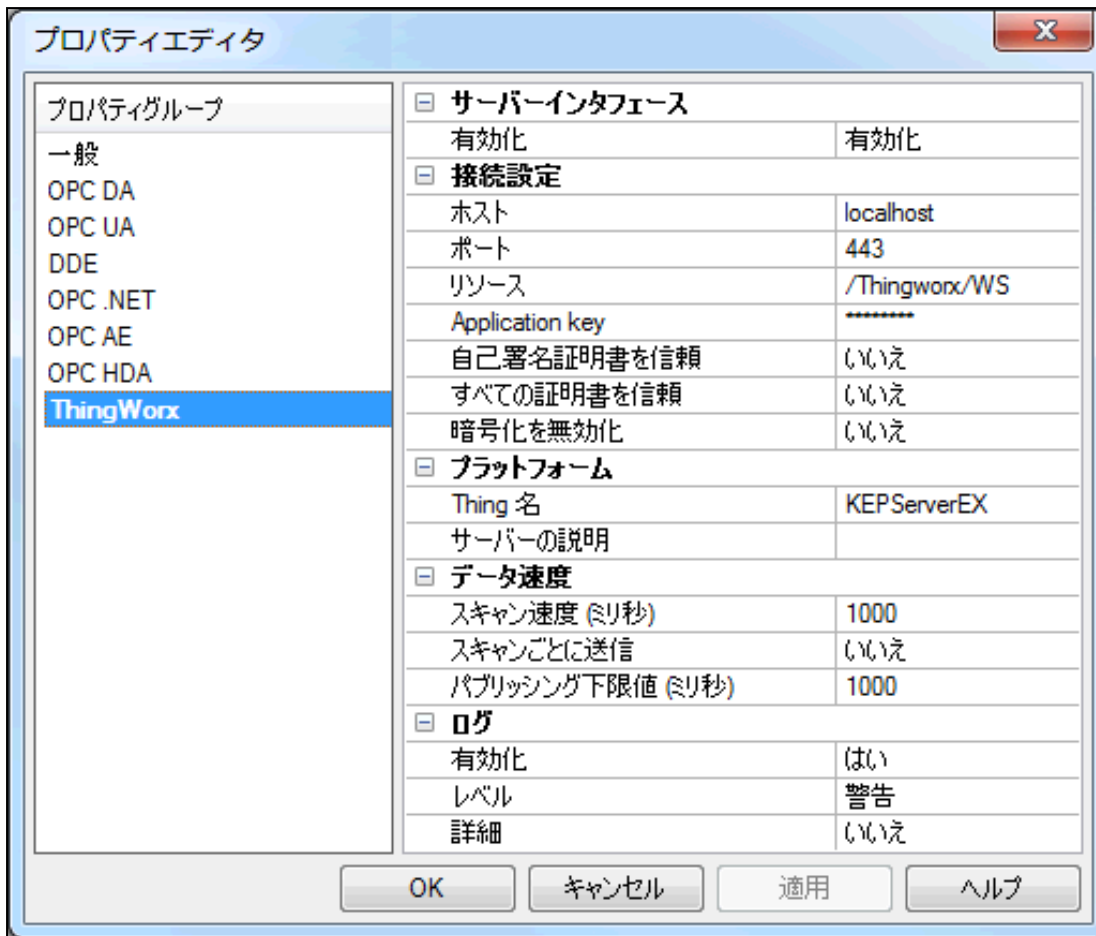
ThingWorx ネイティブインタフェースを設定する前に、"Industrial Gateway" Thing Template を使用して ThingWorx プラットフォームに Thing を作成します。Thing のデータソースを表す名前を使用することをお勧めします。Thing が作成された後、OPC サーバーが Thing の名前を使用して ThingWorx プラットフォームに接続するように設定します。新しい接続がこの Thing に自動バインドされます。

ThingWorx プラットフォームへの接続が確立された後、左側のメニューから「インダストリアル接続」を使用して、新しく作成されたサーバーインスタンスからタグを選択します。これらのタグは、ThingWorx Composer で直接選択し、新しい Thing にバインドできます。

● 詳細については、ThingWorx Composer のドキュメントを参照してください。

警告:

- 配列データ型のタグは、ThingWorx プラットフォームで「常時」プッシュタイプに設定されている必要があります。プッシュしきい値が値変更に設定されていると、プラットフォームへの更新のパブリッシングに失敗します。
- ネイティブインタフェースのほとんどは、クライアント/サーバー構成で動作するものであるのに対し、ThingWorx ネイティブインタフェースは、ThingWorx プラットフォームへの送信接続を作成し、よりクライアントに近いものとして機能します。つまり、ThingWorx ネイティブインタフェースは、標準のポートとプロトコルを使用してリモートの ThingWorx プラットフォームに接続できるため、ファイアウォールやルーティングに関して、例外的なルールを作成する必要がありません。OPC サーバーをホスティングしているマシンから ThingWorx Composer にブラウザで到達できるかぎり、サーバーは、当該のプラットフォームにネイティブインタフェースを通じてデータを渡すことができます。
- ThingWorx ドキュメントに記載されているように、ThingWorx のアプリケーションキーの構成は、安全な環境を提供するために非常に重要です。サーバーインスタンスと ThingWorx プラットフォーム間で適切にデータを交換するには、アプリケーションキーによって必要な権限が付与される必要があります。



サーバーインターフェース

「有効化」: ThingWorx ネイティブインターフェースについて「はい」を選択し、提供される情報との接続を試行します。

「レガシーモード」: Thingworx NextGen Composer (v7.4 以降) と接続しているときには、「レガシーモード」は「無効」(デフォルト) になってなければなりません。ThingWorx バージョン 7.3x またはそれ以前に接続するには、「有効化」を選択して RemoteKEPServerEXThing テンプレートシェイプの使用を強制します。このモードを適切に使用するには、ThingWorx インスタンスの RemoteKEPServerEXThing 拡張機能をインポートします。これにより、適切な ThingShape とサービス定義がこのネイティブインターフェースと連動するようになります。

●「レガシーモード」は KEPServerEX V6.0 ThingWorx ネイティブインターフェースの既存のプロジェクトおよび Thingworx Composer (v7.3x 以前) でのみ使用されます。

接続設定

「ホスト」: ThingWorx サーバーの IP アドレスまたは DNS 名を指定します。

「ポート」: ThingWorx サーバーによって使用される TCP ポートの番号を指定します。

「リソース」: ThingWorx サーバー上の URL エンドポイントを指定します。

「アプリケーションキー」: ThingWorx サーバーに接続するための認証文字列を入力するか、貼り付けます。

●警告: このプロパティは、運用モードで使用可能になっている HTTP で Config API サービスを使用して設定しないでください。セキュリティが低下します。

「自己署名証明書を信頼」: セキュリティを最大限まで高めるには、「いいえ」を選択します。開発の進行中、自己署名証明書を受け入れるには、「はい」を選択します。

●警告: この値は、本番環境では「はい」に設定しないでください。セキュリティが低下します。

「**すべての証明書を信頼**」: セキュリティを最大限まで高めるには、「いいえ」を選択します。「はい」を選択すると、TLS ライブラリでサーバー証明書が検証されなくなります。

● **警告**: この値は、本番環境では「はい」に設定しないでください。セキュリティが低下します。

「**暗号化を無効化**」: SSL で保護されていない ThingWorx プラットフォームへの接続を許可するかどうかを示します。

● **警告**: この値は、本番環境では「はい」に設定しないでください。セキュリティが低下します。

プラットフォーム

「**Thing 名**」: このデータソースを表す、ThingWorx サーバー上のエンティティ (Remote Thing) の名前を入力します。Remote Thing を作成するには、OPC サーバーのテンプレートを使用します。

データ速度

「**パブリッシング下限値**」: 更新をプラットフォームに送信する速度の最小値を指定します。0 の場合、可能なかぎり最高の頻度で更新を送信します。

ログ

「**有効化**」: ThingWorx ネイティブインタフェースに関する高度なログの作成を有効にするには、「はい」を選択します。このログは、サーバーイベントログに送信されます。このログによって、イベントログが短時間でいっぱいになる可能性があります。トラブルシューティング以外では、無効のままにしておくことをお勧めします。

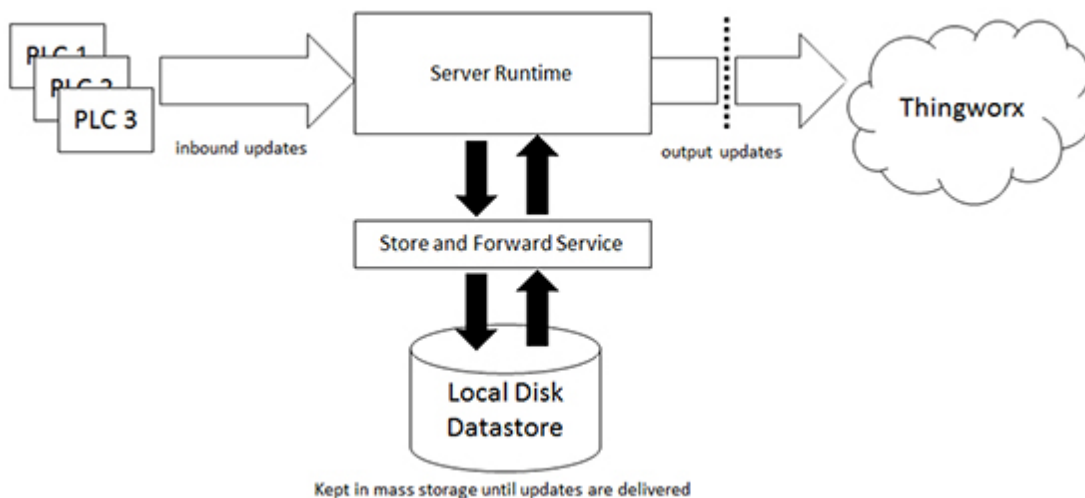
「**レベル**」: イベントログに送信するログの重要度を選択します。「トレース」には、ThingWorx ネイティブインタフェースからのすべてのメッセージが含まれます。

「**詳細**」: 「はい」を選択すると、エラーメッセージは、可能なかぎり最も詳細な内容になります。

● **関連項目**: [イベントログ](#)、[イベントログオプション](#)

ストアアンドフォワード

ThingWorx ネイティブインタフェースでは、産業用サーバーの ThingWorx プラットフォームへの接続が失われたときにプロパティ更新を保持するための「ストアアンドフォワード」データストアがサポートされています。有効な場合、「ストアアンドフォワード」により、ThingWorx ネイティブインタフェースが「更新が受信された」という確認をプラットフォームから受信するまで、受信したすべてのプロパティ更新がディスクに保持されます。プラットフォームへの接続が失われた場合、更新が保存されるディスクが最大許容量の 500MB 以内になるか、保存された更新のサイズが指定された最大サイズを超えるか、いずれか早い方に到達するまで、すべての更新が保存されてディスクに保持されます。データストアまたはディスクがいっぱいになると、受信データを格納するのに十分な空き容量ができるまで、受信した更新がドロップされます。



● **関連項目**: [補充率の例](#)

ストアアンドフォワードのプロパティ

「有効化」: 接続が中断されているか大量のデータが転送されているとき、データをローカルディスクディレクトリに保存してデータが失われないようにするには、「はい」を選択します。この設定を有効にした場合、接続が確立されてデータの受信が確認されると、データをキューに入れて前方にプッシュすることができます。

「保存場所」: データをキャッシュするディレクトリへの完全修飾パスを入力またはブラウズします。

● 注記: ストアアンドフォワードのデータストアを初期化できない場合、ThingWorx ネイティブインタフェースは更新をメモリのキューに入れます。サーバーはデータストアを初期化できるまで自動的に再試行します。具体的なエラー情報については、イベントログを参照してください。

「データストアの最大サイズ」: パージする前のデータの許容最大値をメガバイトまたはギガバイトで選択します。使用可能なデータストアのサイズ範囲は 128 MB から 16 GB です。

「フォワードモード」: 接続の回復時に ThingWorx に送信する更新を特定する方法を選択します。プラットフォームから接続解除されたときにデータを失うことなく生産データをアクティブにモニターする必要がある場合は、再接続時にストアアンドフォワードを実行するか、生産データがアクティブにモニターされていないとき (生産のダウンタイム) に保存済みの更新を転送するようにスケジュールできます。オプションには、「アクティブ」および「保留中」があります。

- **アクティブ** - 「フォワードモード」を「アクティブ」に設定すると、保存されたプロパティ更新は、先に届いたものから順にすべて ThingWorx プラットフォームに送信されます。更新はリアルタイムでプラットフォームに送信されます。ThingWorx プラットフォームの接続解除中に多くの更新が収集された場合、プロパティ更新は先に届いたものから順に送信されるため、遅延が生じることがあります。
- **保留中** - 「フォワードモード」を「保留中」に設定すると、接続解除から回復した後、最新の更新のみがプラットフォームに送信されます。これにより、ThingWorx アプリケーションは生産データをアクティブにモニターしながら最新データを取得できます。生産がアクティブにモニターされていない場合は、モードを「アクティブ」に設定し、サーバーがプラットフォームから接続解除されていた間に保存された古い更新の転送を開始できます。産業用サーバーはプロパティ更新をディスクに保存する前に最大 25,000 個までメモリ内にバッファリングできます。25,000 個の制限に達すると、プロパティ更新はディスクにプッシュされ、「フォワードモード」が「アクティブ」に設定されるまで保留されます。これにより、産業用サーバーは ThingWorx プラットフォームへの接続が回復すると最新の 25,000 個の更新を優先的に受信できます。また、更新が失われないように保留して後で転送することができます。データストアのサイズ制限に達するか、ディスクの 500 MB の制限を超えると、新しい更新はドロップされます。メモリ内バッファは通常、ThingWorx プラットフォームへの接続が失われた場合にのみ使用されます。ただし、プロパティ更新がプラットフォームに転送されるよりも速く収集される場合にも使用されます。

ストアアンドフォワードの考慮事項

- ストアアンドフォワード構成は現在、産業用サーバーの構成ツールでサポートされているほか、構成 API を使用することでサポートされます。
- ストアアンドフォワードはデフォルトで無効になっており、産業用サーバーのプロジェクトのプロパティまたは構成 API を使用して有効にする必要があります。
- ストアアンドフォワードは ThingWorx プラットフォームから構成する必要はありません。ただし、転送した更新を ThingWorx プラットフォームに保存するには、値ストリームを構成し、履歴が必要なすべてのプロパティのログ記録を有効にする必要があります。
- データストアのパス構成 (「保存場所」設定で定義) が修正された場合、既存のデータストアはディスク上に残ります。データストアのパス構成が回復されると、現在のプロジェクトに関連付けられている更新がプラットフォームに転送されます。
- ストアアンドフォワードのプロパティを変更する際にプラットフォーム接続を再初期化する必要はありません。Thingworx インタフェースは変更を適用しつつ更新の収集を続けます。

ストアアンドフォワードのパスは構成時と実行時に検証され、次の条件を満たしている必要があります。

- 3 文字から 256 文字までの範囲内である
 - システムで禁止されている文字や記号が含まれていない
 - 絶対パスである (ドライブ文字で始まっている)
 - ネットワークリソース (マッピングされたドライブ* または UNC 共有) を参照していない
 - USB ドライブ* などのリムーバブルメディアを参照していない
- * は実行時にのみ検証されるアイテムを示しています

ストアアンドフォワードのステータスとモニタリングには、次の方法でアクセスできます。

- ・産業用サーバーの[ストアアンドフォワードタグ](#)
- ・産業用サーバーのイベントログ

ストアアンドフォワードの動作に関する考慮事項

- ・「ストアアンドフォワード」の信頼性の要件により、この機能を有効にすると、ThingWorx プラットフォームに送信される前にすべての更新がディスクを介してルーティングされ、次のセットを送信する前にプラットフォームが最新の更新セットを受信したという確認を ThingWorx ネイティブインタフェースが受け取るのを待機するため、パフォーマンスが少し低下します。
- ・保存された更新はサーバーが再起動されるまで保持されます。
- ・サーバーのメジャー/マイナーアップグレードを実行すると更新は失われるため、保存したすべての更新はソフトウェアをアップグレードする前に転送する必要があります。

操作 - レガシーモード

●「レガシーモード」は KEPServerEX V6.0 ThingWorx ネイティブインタフェースの既存のプロジェクトおよび Thingworx Composer (v7.3x 以前) でのみ使用されます。

有効な情報を使用してインタフェースを設定し、有効にすると、ThingWorx プラットフォームとの接続を KEPServerEX が確立します。プロジェクトのプロパティで使用されている Thing と同じ名前で、新しい "Thing" を ThingWorx プラットフォーム上に作成する必要があります。RemoteKEPServerEXThing Thing 拡張機能を ThingWorx プラットフォームにインポートして、統合対象の新しい Thing の作成に使用する必要があります。プラットフォーム上での作成が完了すると、以下のサービスがプラットフォームを通じて呼び出される場合があります。

拡張機能のインポート方法については、ThingWorx のヘルプガイドを参照してください。

● **注記:** RemoteKEPServerEXThing 拡張機能は、オンラインの ThingWorx Marketplace で公開されているほか、次のフォルダに収められています。

- ・64 ビット版 Windows の場合: C:\Program Files (x86)\Kepware\KEPServerEX 5\Utilities\KEPServerEX Extension for the ThingWorx IoT Platform
- ・32 ビット版 Windows の場合: C:\Program Files\Kepware\KEPServerEX 5\Utilities\KEPServerEX Extension for the ThingWorx IoT Platform

「BrowseGroups」: チャネルデバイスおよびタググループのリストを返します。フィルタとパスの入力が受け付けられます。フィルタは、文字リストを保持する OPC フィルタと同じものです。パスは、"Channel1.Device1" など、チャネルとデバイスのリストです。

「BrowseItems」: ある一定のパスにあるタグのリストを返します。フィルタとパスが受け付けられます。フィルタは、文字リストを保持する OPC フィルタと同じものです。パスは、"Channel1.Device1" など、チャネルとデバイスのリストです。

「AddItems」: タグのサブスクリプション、および Thing のプロパティとしてのタグの追加を可能にします。このサービス呼び出すには、インフォテーブルが必要です。インフォテーブルは、次の情報を含んでいる必要があります。ReadOnly: Boolean、ScanRateMS (オプション): Integer、Description (オプション): String、BaseType: ThingWorx データ型、SourceType: KEPServer データ型、Persistent: Boolean、Logged: Boolean、Source: タグアドレス (channel.device.tag)、Name: タグのローカル名。

● **注記:** ThingWorx での制限事項に従って、名前フィールドでは、空白文字および特殊文字は使用できず、先頭を数字にすることもできません。ハイフン (-) およびアンダースコア (_) を使用できるほか、名前の中および末尾では数字を使用できます。

「RemoveItems」: タグからサブスクリプションを除去します。このサービス呼び出すには、インフォテーブルが必要です。インフォテーブルは、次の情報を含んでいる必要があります。Name: タグのローカル名。Boolean 値の ForceRemove を有効にすると、プロパティを削除することなく、タグがプロパティからアンバインドされます (オプション)。

「GetConfiguration」: ミリ秒単位のスキャン速度、サーバーの説明、ミリ秒単位のパブリッシング下限値が保持されたインフォテーブルを返します。

「SetConfiguration」: ミリ秒単位のスキャン速度、サーバーの説明、ミリ秒単位のパブリッシング下限値を設定します。値を空白のままにすると、現在の設定がそのまま保持されます。

● **注記:**

1. Date データ型を使用するとき、KEPServerEX から取得される値は、UTC として解釈されます。適切なタイムゾーンオフセットを考慮してください。
2. サーバーへのアイテムの追加は同期的に実行され、短時間で完了します。プラットフォームでのプロパティの自動バインドは、ある程度の時間を要する可能性があり、アイテムの追加完了後に、バックグラウンドで実行されることがあります。自動バインドのプロセスによってバインドプロセスが完了すると、イベントが開始されます。
3. RemoveItem を呼び出した場合、当該のプロパティから除去されるのはバインドのみです。RemoveItem を呼び出した後、プログラムによって、または Composer を通じて、当該のプロパティに別のタグを再バインドするか、当該のプロパティを Composer で削除してください。これらのプロパティは、再バインドまたは削除されるまで、「Remote Property Name」(リモートプロパティ名) が空になった状態で Composer に表示されます。
4. 一度の複数のアイテムを追加するとき、2 つ以上のアイテムが同じ ThingWorx 名を使用するように設定されていると、addItem 呼び出し全体が失敗します。すべてのプロパティに、それぞれ一意の名前が割り当てられていることを確認してください。
5. 例に示したコマンドは、cURL またはほかの POST/PUT/GET ユーティリティを使用して実行できます。これらは一例にすぎません。使用できるすべてのサービスとの対話については、ThingWorx のドキュメンテーションを参照してください。

●設定をデフォルト値/プリセット値に復元するには、「**デフォルト**」ボタンを使用します。

ストアアンドフォワード - 補充率の例

最大更新回数および補充率を特定するには、保存する更新のデータストアの最大サイズとデータ型を考慮する必要があります。以下の表はデータ型ごとに最大更新回数と補充率を示しています。データストアの最大サイズは 128 MB、更新は 1 秒あたり 1 回を想定しています。

データ型	最大更新回数	補充率 (バイト/秒)
Word/Short	5817792	22
DWord/Long/Float	5333076	24
Double	4571321	28
String (文字列長 = 10)	3764743	34

以下の数式と上記の表の情報を使用すると、プロジェクトのタグのデータ型に対応する補充率を合計して、特定のプロジェクトの補充率を特定できます。

総補充率 =

$$\begin{aligned} & \text{ScanRate(seconds)} * \\ & \text{プロパティ数 (Bool)} * \text{充填率 (Bool)} + \\ & \text{プロパティ数 (Word)} * \text{充填率 (Word)} + \\ & \text{プロパティ数 (Word)} * \text{充填率 (Short)} + \\ & \text{プロパティ数 (DWord)} * \text{充填率 (DWord)} + \\ & \text{プロパティ数 (Word)} * \text{充填率 (Long)} + \\ & \text{プロパティ数 (Word)} * \text{充填率 (Float)} + \\ & \text{プロパティ数 (Double)} * \text{充填率 (Double)} + \\ & \text{プロパティ数 (String)} * \text{充填率 (String)} \end{aligned}$$

以下の表は 500 個の Word プロパティ、500 個の DWord プロパティ、10 個の String プロパティ、100 個の Double プロパティで構成されたサンプルプロジェクトでデータ損失が発生する前の補充率とオフライン時間を異なるスキャン速度に基づいて示しています。データストアの最大サイズは 128 MB と想定しています。

プロパティあたりのスキャン速度 (ミリ秒)	補充率 (バイト/秒)	オフライン時間 (分)
10000	2614	816
1000	26140	81
250	104560	20

ストアアンドフォワード - システムタグ

システムタグはデータストアのステータス情報を提供し、サーバークライアントが更新を管理できるようにします。これらのシステムタグは、ストアアンドフォワードが有効になっているとき、サーバークライアントのみが使用できます。これらのタグは、クライアントブラウズツリー内の _System フォルダと同じレベルの _ThingWorx group フォルダに保管されています。

タグ	クラス	データ型	説明
_StoreAndForwardEnabled	読み取り/書き込み	Boolean	このタグを使用すると、ストアアンドフォワードをオンまたはオフにできます。このタグが False に設定されている場合、ストアアンドフォワードは無効になります。ストアアンドフォワードが無効になっている場合、すべてのデータストア関連のシステムタグは、デフォルト値である 0 を報告します。  注記: <ul style="list-style-type: none">必ずしも構成によってストアアンドフォワードの有効化/無効化状態がわかるわけではありません。使用中の構成を取得するには、_StoreAndForwardEnabledStatus システムタグを使用します。たとえば、ストアアンドフォワードを妨げるエラーが発生すると、_StoreAndForwardEnabledStatus は 0 を返します。
_StoreAndForwardEnabledStatus	読み取り/書き込み	Boolean	このタグはインタフェースでストアアンドフォワードが使用されているかどうかを示します。
_DatastoreDiskFull	読み取り専用	Boolean	このタグはデータストアに使用されているディスクが、更新を保存するために必要な 500 MB のしきい値を超えていないかどうかを示します。
_DatastoreFull	読み取り専用	Boolean	このタグは更新の保存に使用するために設定した データストアの最大サイズ にデータストアが達しているかどうかを示します。
_StoredUpdateCount	読み取り専用	DWord	このタグはデータストア内の更新数を示します。  注記: <ul style="list-style-type: none">値がゼロ以外であっても、ThingWorx の接続が失われたわけではありません。ストアアンドフォワードが有効になっている場合、更新は常にデータストアを介してルーティングされるためです。定常状態での動作中、この値は変動することが予想されますが、保存された更新数が時間とともに増加することはありません。ThingWorx プラットフォームに送信されるデータよりも収集されるデータの方が多い場合、値はゼロ以外になります。
_DeleteStoredData	読み取り/書き込み	Boolean	このタグはデータストアの内容の削除に使用できます。このタグに任意の値を書き込むと、ストアアンドフォワードのデータストア内にある保存済みの更新がすべて削除されます。
_DatastoreCurrentSizeMB	読み取り専用	Double	このタグは現在ディスク上に存在するすべての更新によって使用されている容量 (MiB 単位) を報告します。
_DatastoreRemainingSpaceMB	読み取り専用	Double	このタグは更新の保存に使用できるデータストア内の残りの容量 (MiB 単位) を報告します。これは「データストアの最大サイズ」プロパティと、使用できないディスク容量に基づいています。残りのディスク容量については、_DatastoreUsableDiskSpace

タグ	クラス	データ型	説明
			タグを参照してください。
_DatastoreUseableDiskSpaceMB	読み取り専用	Double	このタグはデータストアに使用されているディスクの、更新の保存に使用できる容量 (MiB 単位) を報告します。ストアアンドフォワードはディスク全体を使用しないように 500 MiB の安全バッファを確保します。このシステムタグは計算時にこの安全バッファを考慮します。このタグにはユーザーが指定したデータストアの残り容量は反映されません。詳細は、_DatastoreSizeRemaining を参照してください。
_DatastoreAttachError	読み取り専用	Boolean	このタグはストアアンドフォワードの使用を妨げるエラーが発生したことを示します。タグ値が True の場合は、エラーが発生しています。このエラーの情報については、サーバーイベントログを参照してください。ストアアンドフォワードのデータストアの使用を妨げるエラーを解決するには、考えられる原因/解決策を参照してください。
_DroppedUpdates	読み取り専用	Long	このタグは ThingWorx インタフェースの起動以降にドロップされた更新の総数を報告します。値は 2,147,483,647 に達すると、0 にロールオーバーされます。ThingWorx の接続が再初期化されると、値は 0 にリセットされます。
_ForwardMode	読み取り/書き込み	DWord	<p>このタグは ThingWorx ネイティブインタフェースの現在のフォワードモード構成を報告します。このタグは構成したモードを変更するための書き込みをサポートします。有効な値には、0 (「アクティブ」) および 1 (「保留中」) があります。ほかのすべての書き込み値は無視されます。</p> <p>注記:</p> <ul style="list-style-type: none"> 必ずしも構成によって使用中のストアアンドフォワードがわかるわけではありません。使用中のモードを取得するには、_ForwardModeStatus システムタグを使用します。たとえば、ストアアンドフォワードを妨げるエラーが発生すると、_ForwardModeStatus は空白を返します。
_ForwardModeStatus	読み取り専用	String	このタグはネイティブインタフェースによって使用されている現在のフォワードモードを報告します。有効な値には、「アクティブ」および「保留中」があります。ストアアンドフォワードが使用されていない場合、システムタグは空白の文字列を返します。

● **関連項目:** ThingWorx プラットフォームへのアクセスを制御する方法と、関連するデータ転送については、[ThingWorx インタフェースユーザー](#)を参照してください。

サーバーのオプション

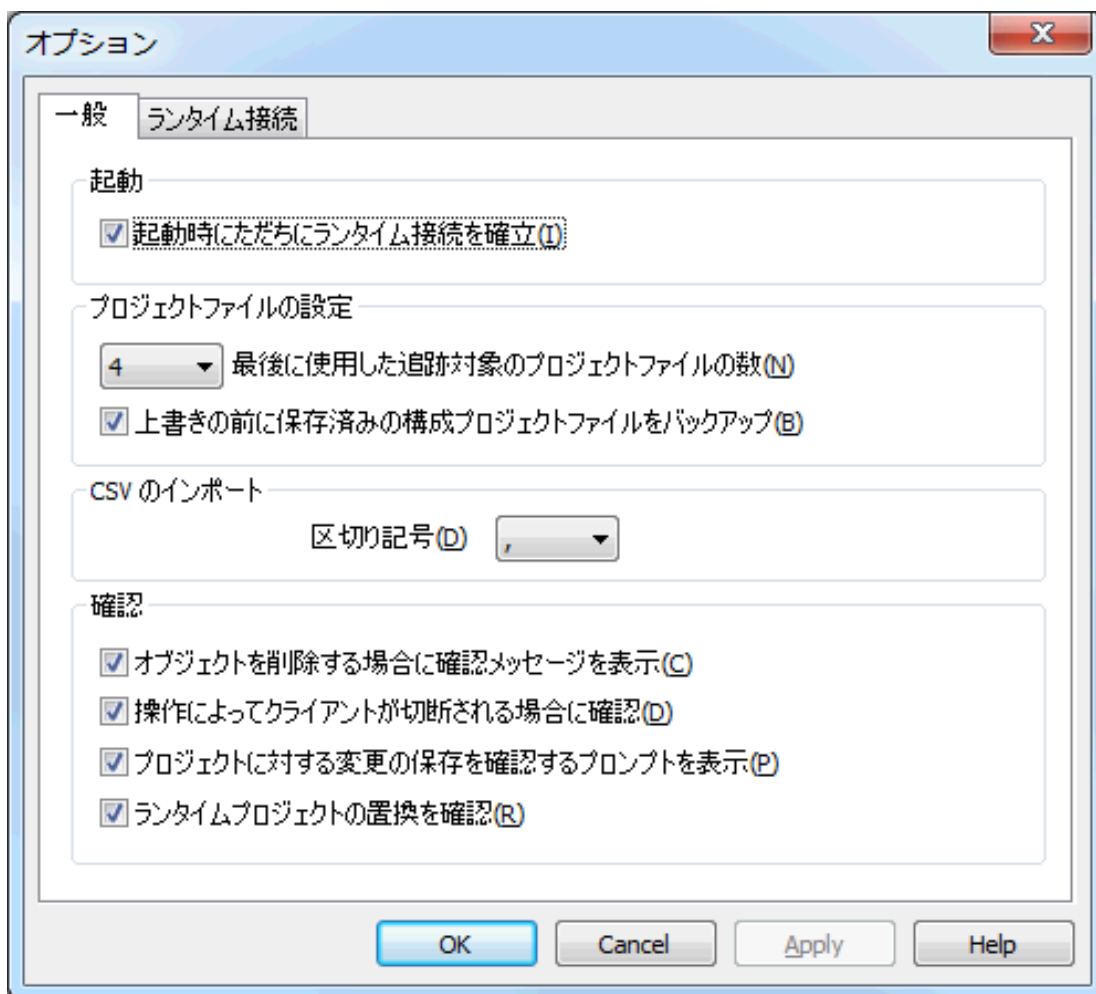
構成からサーバーオプションのグループにアクセスするには、「ツール」|「オプション」の順にクリックします。これらの設定は個別に設定します。詳細については、以下のリストからリンクを選択してください。

[オプション - 一般](#)

[オプション - ランタイム接続](#)

オプション - 一般

このダイアログボックスは、サーバーに関する一般的なオプションを指定するために使用します (ランタイムとの接続を確立するタイミング、保存済みの構成プロジェクトファイルをバックアップするタイミング、警告のポップアップを表示する条件など)。



「**起動時にただちにランタイム接続を確立**」: 構成ツールの起動時に、ランタイムに接続するかどうかを指定します。無効にする場合は、ユーザーが手動で接続する必要があります。デフォルト設定では有効になっています。

「**最後に使用した追跡対象のプロジェクトファイルの数**」: プロジェクトの MRU (最近使用した) リストに表示されるプロジェクトファイルの数を選択します。有効な範囲は 1 から 16 です。デフォルトの設定は 8 です。

「**上書きの前に保存済みの構成プロジェクトファイルをバックアップ**」: 有効にすると、構成プロジェクトファイルが新しいプロジェクトファイルによって上書きされる前に、直近の保存済み構成プロジェクトファイルのバックアップコピーが自動的に作成されます。バックアップファイルの名前と位置は、イベントログに表示されます。

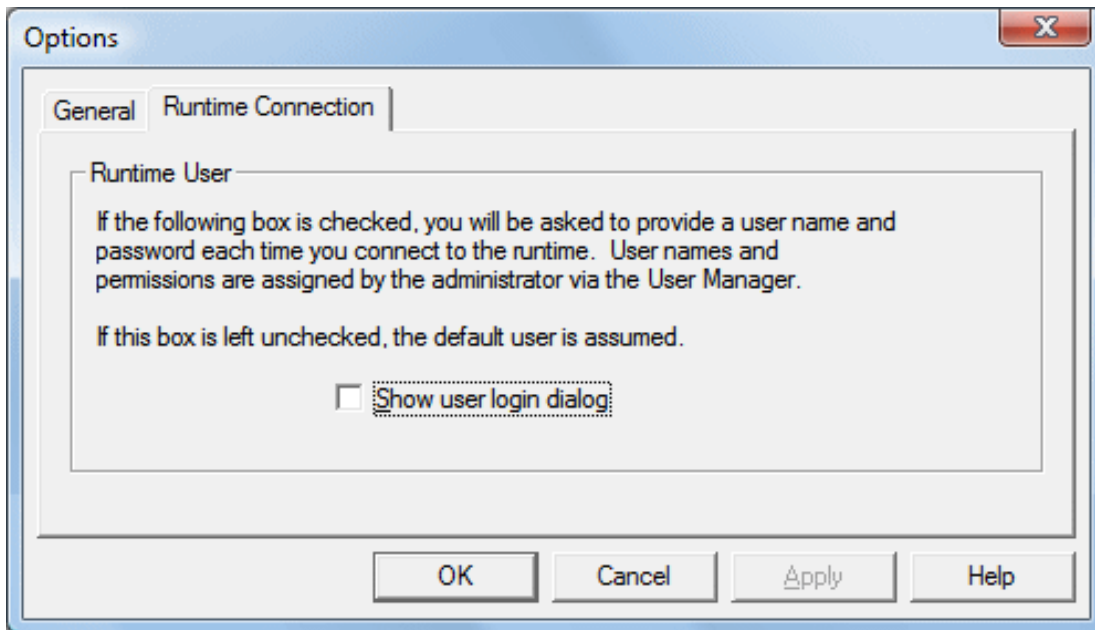
「**CSV インポート**」: 「**区切り記号**」設定では、タグデータをコンマ区切り値 (CSV) ファイルにインポートおよびエクスポートするときに使用される区切り記号を指定します。コンマ (,) またはセミコロン (;) を指定できます。デフォルトの設定はコンマです。詳細については、[タグの管理](#)を参照してください。

「確認」: 構成の実行中に、オペレータに対して警告のポップアップを表示する条件を指定します。

- 「オブジェクトを削除する場合に確認メッセージを表示」: これを有効にすると、構成削除操作を実行するたびに、削除操作を実行するかどうかの確認を求める警告ポップアップが表示されます。
- 「操作によってクライアントが切断される場合に確認」: これを有効にすると、クライアントアプリケーションをサーバーから切断する構成操作を実行するたびに、警告ポップアップが表示されます。切断シーケンスを開始するには、このポップアップで操作を了承する必要があります。
- 「プロジェクトに対する変更の保存を確認するプロンプトを表示」: これを有効にすると、プロジェクト内に未処理の変更が存在している状態でサーバーをシャットダウンしようとした場合に、ポップアップが表示されます。
- 「ランタイムプロジェクトの置換を確認」: これを有効にすると、構成の実行中にランタイムに接続している場合に、プロジェクトをオフラインで開いて編集できるという旨の警告が表示されます。

オプション - ランタイム接続

このダイアログボックスは、ランタイムへの接続をどのように管理するかを指定するために使用します。



「ユーザーログインダイアログを表示」: 有効にすると、構成をランタイムに接続してプロジェクトを編集しようとするとき、有効なユーザー名とパスワードを要求されます。デフォルトでは無効に設定されています。

● 注記: ユーザー名とアクセス許可は、管理者によって割り当てられます。詳細については、[設定 - ユーザーマネージャ](#)を参照してください。

コンポーネントと概念

特定のサーバーコンポーネントの詳細については、以下のリストからリンクを選択してください。

[チャネルとは](#)

[デバイスとは](#)

[タグとは](#)

[タググループとは](#)

[エイリアスマップとは](#)

[イベントログとは](#)

チャネルとは

チャネルは、PC から 1 つまたは複数の外部デバイスへの通信メディアを表します。チャネルを使用して、シリアルポート、PC にインストールされたカード、またはイーサネットソケットを表すことができます。

デバイスをプロジェクトに追加する前に、デバイスとの通信に使用するチャネルを定義する必要があります。チャネルとデバイスドライバは密接に関係しています。チャネルを作成した後、選択したドライバがサポートするデバイスのみをこのチャネルに追加できます。

チャネルの追加

チャネルは、チャネル定義プロセスを使用してユーザーをガイドするチャネルウィザードを使用して追加されます。最初に、チャネルを割り当てる論理名の指定を求められます。この名前は、プロジェクトで定義されているすべてのチャネルとデバイスの間で一意でなければなりません。予約文字については、[チャネル、デバイス、タグ、およびタググループに適切な名前を付ける方法](#)を参照してください。

使用するデバイスドライバの指定を求められます。システムに現在インストールされているすべてのデバイスドライバが含まれるリストボックスが表示されます。すべてのシリアルドライバは、同じプロジェクトで複数のチャネルを使用できます。

● **注記:** ハードウェアカードドライバで、複数のチャネルを単一のプロジェクトで使用できるかどうかを確認するには、ドライバのヘルプドキュメントを参照してください。サポートされているチャネル数の確認方法については、[サーバーのサマリー情報](#)を参照してください。

使用する特定の通信パラメータの指定を求められます。複数のチャネルは同じ通信パラメータを共有できません。たとえば、2 つのシリアルドライバが COM1 を使用することはできません。特定のデバイスの正しい通信パラメータについては、製造メーカーとドライバのヘルプドキュメントを参照してください。

● **注記:** シリアルドライバの「フロー制御」の設定は、コンバータを介して RS422/485 ネットワークデバイスを RS232 シリアルポートに接続する場合に主に使用されます。RS232 から RS422/485 へのほとんどのコンバータでは、フロー制御なし（「なし」）にするか、RTS 行を、PC が送信中はオンに、受信待機中はオフに設定する必要があります。

チャネルウィザードには最後に新しいチャネルのサマリーが表示されます。

チャネルの除去

チャネルをプロジェクトから除去するには、目的のチャネルを選択し、「**削除**」キーを押します。あるいは、「**編集**」メニューまたはツールバーから「**編集**」|「**削除**」の順に選択します。

チャネルのプロパティの表示

特定のチャネルのチャネルプロパティを表示するには、そのチャネルを選択し、「**編集**」メニューまたはツールバーから「**編集**」|「**プロパティ**」の順にクリックします。

● **関連項目:** [チャネルのプロパティ - 一般](#)

チャネルのプロパティ - 一般

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

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

識別

「名前」: このチャンネルのユーザー定義の識別情報。各サーバープロジェクトで、それぞれのチャンネル名が一意でなければなりません。名前は最大 256 文字ですが、一部のクライアントアプリケーションでは OPC サーバーのタグ空間をブラウズする際の表示ウィンドウが制限されています。チャンネル名は OPC ブラウザ情報の一部です。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の情報。

● 「説明」などのこれらのプロパティの多くには、システムタグが関連付けられています。

「ドライバー」: このチャンネルに選択されているプロトコルドライバー。このプロパティでは、チャンネル作成時に選択されたデバイスドライバーが示されます。チャンネルのプロパティではこの設定を変更することはできません。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。このことを念頭において、大規模なクライアントアプリケーションを開発した後はプロパティに対する変更を行わないようにします。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。

診断

「診断取り込み」: このオプションが有効な場合、チャンネルの診断情報が OPC アプリケーションに取り込まれます。サーバーの診断機能は最小限のオーバーヘッド処理を必要とするので、必要なときにだけ利用し、必要がないときには無効にしておくことをお勧めします。デフォルトでは無効になっています。

● 注記: ドライバー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。

チャンネルのプロパティ - 詳細

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

プロパティグループ	<div> <div>非正規化浮動小数点処理</div> <div>浮動小数点値</div> <div>デバイス間遅延</div> <div>デバイス間遅延 (ミリ秒)</div> </div>	
一般	ゼロで置換	
シリアル通信		
書き込み最適化		
詳細	0	
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、このプロパティは使用できません。チャンネルの浮動小数点正規化の設定に従って、リアルタイムのドライバータグ (値や配列など) が浮動小数点正規化の対象となります。たとえば、EFM データはこの設定の影響を受けません。

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

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

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

チャンネルのプロパティ - イーサネット通信

イーサネット通信を使用してデバイスと通信できます。

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

イーサネット設定

「ネットワークアダプタ」: バインドするネットワークアダプタを指定します。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。

チャンネルのプロパティ - シリアル通信

シリアル通信のプロパティはシリアルドライバーで設定でき、選択されているドライバー、接続タイプ、オプションによって異なります。使用可能なプロパティのスーパーセットを以下に示します。

クリックして[接続タイプ](#)、[シリアルポートの設定](#)、[イーサネット設定](#)、[実行動作](#)のいずれかのセクションにジャンプします。

- **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。これらのプロパティに対する変更によって通信が一時的に不通になることがあるので、サーバー機能へのアクセス権を制限するには、ユーザーマネージャを使用します。

プロパティグループ	接続タイプ	
一般	物理メディア	COM ポート
シリアル通信	共有	いいえ
書き込み最適化	シリアルポートの設定	
詳細	COM ID	3
通信シリアル化	ボーレート	19200
リンク設定	データビット	8
	パリティ	なし
	ストップビット	1
	フロー制御	なし
	実行動作	
	通信エラーを報告	有効化

接続タイプ

「物理メディア」: データ通信に使用するハードウェアデバイスのタイプを選択します。オプションには「COM ポート」、「なし」、「モデム」、「イーサネットカプセル化」があります。デフォルトは「COM ポート」です。

- ・「なし」: 物理的な接続がないことを示すには「なし」を選択します。これによって通信なしの動作セクションが表示されます。
- ・「COM ポート」: シリアルポートの設定セクションを表示して設定するには、「COM ポート」を選択します。
- ・「モデム」: 通信に電話回線を使用する場合 (モデム設定セクションで設定)、「モデム」を選択します。
- ・「イーサネットカプセル化」: 通信にイーサネットカプセル化を使用する場合に選択します。これによってイーサネット設定セクションが表示されます。
- ・「共有」: 現在の構成を別のチャネルと共有するよう接続が正しく識別されていることを確認します。これは読み取り専用プロパティです。

シリアルポートの設定

「COM ID」: チャネルに割り当てられているデバイスと通信するときに使用する通信 ID を指定します。有効な範囲は 1 から 9991 から 16 です。デフォルトは 1 です。

「ボーレート」: 選択した通信ポートを設定するときに使用するボーレートを指定します。

「データビット」: データワードあたりのデータビット数を指定します。オプションは 5、6、7、8 です。

「パリティ」: データのパリティのタイプを指定します。オプションには「奇数」、「偶数」、「なし」があります。

「ストップビット」: データワードあたりのストップビット数を指定します。オプションは 1 または 2 です。

「フロー制御」: RTS および DTR 制御回線の利用方法を指定します。一部のシリアルデバイスと通信する際にはフロー制御が必要です。以下のオプションがあります。

- ・「なし」: このオプションでは、制御回線はトグル (アサート) されません。
- ・「DTR」: このオプションでは、通信ポートが開いてオンのままになっている場合に DTR 回線がアサートされます。
- ・「RTS」: このオプションでは、バイトを転送可能な場合に RTS 回線がハイになります。バッファ内のすべてのバイトが送信されると、RTS 回線はローになります。これは通常、RS232/RS485 コンバータハードウェアで使用されます。
- ・「RTS、DTR」: このオプションは DTR と RTS を組み合わせたものです。
- ・「RTS 常時」: このオプションでは、通信ポートが開いてオンのままになっている場合に、RTS 回線がアサートされます。
- ・「RTS 手動」: このオプションでは、「RTS 回線制御」で入力したタイミングプロパティに基づいて RTS 回線がアサートされます。これは、ドライバが手動による RTS 回線制御をサポートしている場合 (またはプロパティが共有され、このサポートを提供するドライバに 1 つ以上のチャネルが属している場合) にのみ使用できます。
「RTS 手動」を選択した場合、次のオプションから成る「RTS 回線制御」プロパティが追加されます。
 - ・「事前オン」: このプロパティでは、データ転送のどれだけ前に RTS 回線を事前にオンにするかを指定します。有効な範囲は 0 から 9999 ミリ秒です。デフォルトは 10 ミリ秒です。
 - ・「遅延オフ」: このプロパティでは、データ転送後に RTS 回線をハイのままにする時間を指定します。有効な範囲は 0 から 9999 ミリ秒です。デフォルトは 10 ミリ秒です。
 - ・「ポーリング遅延」: このプロパティでは、通信のポーリングが遅延する時間を指定します。有効な範囲は 0 から 9999 です。デフォルトは 10 ミリ秒です。

● ヒント: 2 回線 RS 485 を使用している場合、通信回線上で "エコー" が発生することがあります。この通信はエコー除去をサポートしていないので、エコーを無効にするか、RS-485 コンバータを使用することをお勧めします。

実行動作

- ・「通信エラーを報告」: 低レベル通信エラーに関するレポートを有効または無効にします。オンにした場合、低レベルのエラーが発生するとイベントログに書き込まれます。オフにした場合、通常の要求の失敗は書き込まれますが、これと同じエラーは書き込まれません。デフォルトは「有効化」です。
- ・「アイドル接続を閉じる」: チャネル上のクライアントによっていずれのタグも参照されなくなった場合、接続を閉じます。デフォルトは「有効化」です。
- ・「クローズするまでのアイドル時間」: すべてのタグが除去されてから COM ポートを閉じるまでサーバーが待機する時間を指定します。デフォルトは 15 秒です。

イーサネット設定

● **注記:** すべてのシリアルドライバがイーサネットカプセル化をサポートするわけではありません。このグループが表示されない場合、機能はサポートされていません。

イーサネットカプセル化は、イーサネットネットワーク上のターミナルサーバーに接続しているシリアルデバイスとの通信を可能にします。ターミナルサーバーは基本的には仮想のシリアルポートであり、イーサネットネットワーク上の TCP/IP メッセージをシリアルデータに変換します。メッセージが変換されると、ユーザーはシリアル通信をサポートする標準デバイスをターミナルサーバーに接続可能になります。ターミナルサーバーのシリアルポートが接続先のシリアルデバイスの要件に合うように適切に設定されている必要があります。詳細については、サーバーのヘルプで「イーサネットカプセル化の使用方法」を参照してください。

- **「ネットワークアダプタ」:** このチャンネルのイーサネットデバイスがバインドするネットワークアダプタを指定します。バインド先のネットワークアダプタを選択するか、OS がデフォルトを選択可能にします。
- 一部のドライバでは追加のイーサネットカプセル化プロパティが表示されることがあります。詳細については、「チャンネルのプロパティ- イーサネットカプセル化」を参照してください。

モデム設定

- **「モデム」:** 通信に使用するインストール済みモデムを指定します。
- **「接続タイムアウト」:** 接続が確立される際に待機する時間を指定します。この時間を超えると読み取りまたは書き込みが失敗します。デフォルトは 60 秒です。
- **「モデムのプロパティ」:** モデムハードウェアを設定します。クリックした場合、ベンダー固有のモデムプロパティが開きます。
- **「自動ダイヤル」:** 電話帳内のエントリに自動ダイヤルできます。デフォルトは「無効化」です。詳細については、サーバーのヘルプで「モデム自動ダイヤル」を参照してください。
- **「通信エラーを報告」:** 低レベル通信エラーに関するレポートを有効または無効にします。オンにした場合、低レベルのエラーが発生するとイベントログに書き込まれます。オフにした場合、通常の要求の失敗は書き込まれますが、これと同じエラーは書き込まれません。デフォルトは「有効化」です。
- **「アイドル接続を閉じる」:** チャンネル上のクライアントによっていずれのタグも参照されなくなった場合、モデム接続を閉じます。デフォルトは「有効化」です。
- **「クローズするまでのアイドル時間」:** すべてのタグが除去されてからモデム接続を閉じるまでサーバーが待機する時間を指定します。デフォルトは 15 秒です。

通信なしの動作

- **「読み取り処理」:** 明示的なデバイス読み取りが要求された場合の処理を選択します。オプションには「無視」と「失敗」があります。「無視」を選択した場合には何も行われません。「失敗」を選択した場合、失敗したことがクライアントに通知されます。デフォルト設定は「無視」です。

チャンネルのプロパティ- イーサネットカプセル化

イーサネットカプセル化はワイヤレスネットワーク接続 (802.11b ネットワークや CDPD パケットネットワークなど) で使用でき、これは広範なシリアルデバイスをサポートすることも目的として開発されました。ターミナルサーバーデバイスを使用することで、RS-232 または RS-485 デバイスをプラント全体に配置すると同時に、それらのリモートマウントのデバイスに 1 台のローカライズされた PC からアクセスできます。イーサネットカプセル化では、必要に応じて各デバイスに個別のネットワーク IP アドレスを割り当てることもできます。複数のターミナルサーバーを介して、1 台の PC から数百のシリアルデバイスにアクセスできます。1 つのチャンネルはローカル PC のシリアルポートを使用するよう定義し、別のチャンネルはイーサネットカプセル化を使用するよう定義できます。

● **注記:** これらのプロパティはシリアルドライバでのみ使用できます。表示されるプロパティは、選択した通信ドライバおよびサポートされている機能によって異なります。

「ネットワークアダプタ」: このプロパティでは、ネットワークアダプタを指定します。

「デバイスアドレス」: このプロパティでは、このデバイスが接続しているターミナルサーバーの 4 つのフィールドから成る IP アドレスを指定します。IP は YYY.YYY.YYY.YYY として指定します。YYY は IP アドレスを示します。各 YYY バイトが 0 から 255 の範囲でなければなりません。チャンネルごとに独自の IP アドレスがあります。

「ポート」: このプロパティでは、リモートターミナルサーバーに接続する際に使用するイーサネットポートを設定します。有効な範囲は 1 から 65535 であり、一部の番号は予約済みです。デフォルトは 2101 です。

「プロトコル」: このプロパティでは、使用されているターミナルサーバーの特性に応じて、TCP/IP または UDP 通信を指定します。デフォルトは TCP/IP です。使用可能なプロトコルの詳細については、ターミナルサーバーのヘルプドキュメントを参照してください。

● **重要:** イーサネットカプセル化モードは実際のシリアル通信ドライバーに対して完全に透過的です。ユーザーは残りのデバイスプロパティを、これらがあたかもローカルPCのシリアルポート上で直接デバイスに接続しているかのように構成する必要があります。

「**接続タイムアウト**」: このプロパティでは、調整するリモートデバイスのソケット接続を確立するために必要な時間を指定します。多くの場合、デバイスとの接続にかかる時間は、そのデバイスに対する通常の通信要求にかかる時間よりも長くなります。有効な範囲は1から999秒です。デフォルトは3秒です。

● **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。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。

チャネルのプロパティ - 通信シリアル化

サーバーのマルチスレッドアーキテクチャにより、チャネルはデバイスとの並列通信が可能になります。これは効率的ですが、物理ネットワークに制約がある（無線イーサネットなど）場合には通信をシリアル化できます。通信シリアル化によって、仮想ネットワーク内で同時に通信可能なチャネルは1つに制限されます。

"仮想ネットワーク" という用語は、通信に同じパイプラインを使用するチャネルと関連デバイスの集合を表します。たとえば、無線イーサネットのパイプラインはマスター無線です。同じマスター無線を使用しているチャネルはすべて同じ仮想ネットワークに関連付けられています。チャネルは"ラウンドロビン"方式で1つずつ順番に通信できます。デフォルトでは、チャネルが1つのトランザクションを処理した後で、通信を別のチャネルに渡します。トランザクションには1つ以上のタグが含まれることがあります。要求に応答しないデバイスが制御チャネルに含まれている場合、そのトランザクションがタイムアウトになるまでチャネルは制御を解放できません。これによって、仮想ネットワーク内のその他のチャネルでデータ更新の遅延が生じます。

プロパティグループ	チャンネルレベルの設定	
一般	仮想ネットワーク	なし
シリアル通信	サイクルあたりのトランザクション数	1
書き込み最適化	グローバル設定	
詳細	ネットワークモード	負荷分散
通信シリアル化		

チャンネルレベルの設定

「**仮想ネットワーク**」: このプロパティでは、チャネルの通信シリアル化モードを指定します。オプションには「なし」、「ネットワーク1」、「ネットワーク500」があります。デフォルトは「なし」です。オプションの説明は次のとおりです。

- ・「**なし**」: このオプションを選択した場合、チャネルの通信シリアル化は無効になります。
- ・「**ネットワーク1**」-「**ネットワーク500**」: このオプションでは、チャネルを割り当てる仮想ネットワークを指定します。

「**サイクルあたりのトランザクション数**」: このプロパティでは、そのチャネルで実行可能な単一ブロック/非ブロック読み取り/書き込みトランザクションの数を指定します。あるチャネルが通信する機会を得ると、この数だけトランザクションが試みられます。有効な範囲は1から99です。デフォルトは1です。

グローバル設定

- ・「**ネットワークモード**」: このプロパティでは、チャネル通信を委譲する方法を制御します。「**負荷分散**」モードでは、各チャネルが1つずつ順番に通信する機会を得ます。「**優先順位**」モードでは、チャネルは次の規則（最も高い優先順位から最も低い優先順位の順）に従って通信する機会を得ます。
 - ・書き込みが保留中になっているチャネルの優先順位が最も高くなります。
 - ・（内部のプラグインまたは外部のクライアントインタフェースによって）明示的な読み取りが保留中になっているチャネルは、その読み取りの優先順位に基づいて優先順位が決まります。
 - ・スキャン読み取りおよびその他の定期的イベント（ドライバー固有）。
 デフォルトは「負荷分散」であり、すべての仮想ネットワークとチャネルに影響します。

● **非送信請求応答に依存するデバイスを仮想ネットワーク内に配置してはなりません。** 通信をシリアル化する場合、「自動格下げ」を有効にすることをお勧めします。

データを読み書きする方法はドライバによって異なるので(単一ブロック/非ブロックランザクションなど)、アプリケーションの「サイクルあたりのランザクション数」プロパティを調整する必要があります。その場合、次の要因について検討します。

- 各チャネルから読み取る必要があるタグの数
- 各チャネルにデータを書き込む頻度
- チャネルが使用しているのはシリアルドライバかイーサネットドライバか?
- ドライバは複数の要求に分けてタグを読み取るか、複数のタグをまとめて読み取るか?
- デバイスのタイミングプロパティ(「要求のタイムアウト」や「連続した x 回のタイムアウト後の失敗」など)が仮想ネットワークの通信メディアに最適化されているか?

チャネルのプロパティ - ネットワークインタフェース

イーサネットカプセル化では、現在使用可能なほぼすべてのドライバがなんらかの形式のイーサネット通信をサポートしています。ネイティブにイーサネットベースのドライバか、イーサネットカプセル化用に設定されているシリアルドライバかにかかわらず、なんらかの形式のネットワークインタフェースが使用されます。ほとんどの場合、そのインタフェースはネットワークインタフェースカード (NIC) の形式をとります。ネットワークがインストールされている PC では、これは通常、IT またはプラントフロアネットワーク (またはその両方) への接続を可能にする 1 つの NIC がインストールされていることを意味します。

一般的なネットワーク構成と負荷ではこの構成で十分です。ただし、イーサネットデバイスから定期的にデータを受信する必要がある場合には、問題が生じることがあります。プラントフロアネットワークと IT ネットワークが混在している場合、大きなバッチファイルの転送によってプラントフロアデータの間隔が完全に乱れる可能性があります。この問題に対処する最も一般的な方法としては、PC に 2 つ目の NIC をインストールします。一方の NIC は IT ネットワークへのアクセスに使用し、もう一方の NIC はプラントフロアデータへのアクセスに使用できます。これは妥当な措置だと思われるが、ネットワークを分離しようとした場合に問題が発生することがあります。複数の NIC を使用している場合、ユーザーはバインドの順序を指定する必要があります。バインドの順序によって、イーサネットネットワークの各部分にアクセスするときに使用される NIC が決まります。多くの場合、オペレーティングシステムのツールを使用してバインド設定を管理できます。

各 NIC カードで使用されるプロトコルとサービスのタイプがはっきり区別されている場合、バインド順序はオペレーティングシステムによって作成できます。特定のバインド順序を選択する明確な手段がない場合、イーサネットデバイス接続が間違ったネットワークにルーティングされる可能性があります。その場合、下に示されているネットワークインタフェースを使用して、そのイーサネットドライバで使用する特定の NIC カードを選択できます。ネットワークインタフェースの選択肢を使用して、NIC 名またはそれに現在割り当てられている IP アドレスに基づいて特定の NIC カードを選択できます。この使用可能な NIC のリストには、一意の NIC カードまたは複数の IP が割り当てられている NIC が含まれています。この選択肢には、アクティブな WAN 接続 (ダイヤルアップ接続など) が表示されます。

● **注記:** このプロパティはイーサネットドライバでのみ使用できます。

特定の NIC インタフェースを選択することで、指定した NIC を介してドライバがすべてのイーサネット通信を送信するように設定できます。NIC が選択された場合、オペレーティングシステムの標準のバインド順序は完全にバイパスされます。これにより、ネットワークの動作がユーザーの管理下に置かれ、勘に頼る操作が不要になります。

「ネットワークアダプタ」ドロップダウンメニューに表示される選択肢は、ネットワーク構成の設定、PC にインストールされている一意の NIC の数、NIC に割り当てられている一意の IP の数によって異なります。オペレーティングシステムによってバインド順序の選択肢が作成されるようにするには、ネットワークアダプタとして「デフォルト」を選択します。これにより、ドライバはオペレーティングシステムの標準のバインド順序を使用して NIC を設定できます。

● **重要:** 使用する NIC がわからない場合、デフォルトの条件を選択します。さらに、イーサネットベースのデバイスが使用されていて、この機能が製品アップグレードを介して公開されている場合、デフォルトの条件を選択します。

● **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。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。このプロパティに対する変更によって通信が一時的に不通になることがあるので注意してください。

チャネルのプロパティ - 書き込み最適化

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

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

書き込み最適化

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

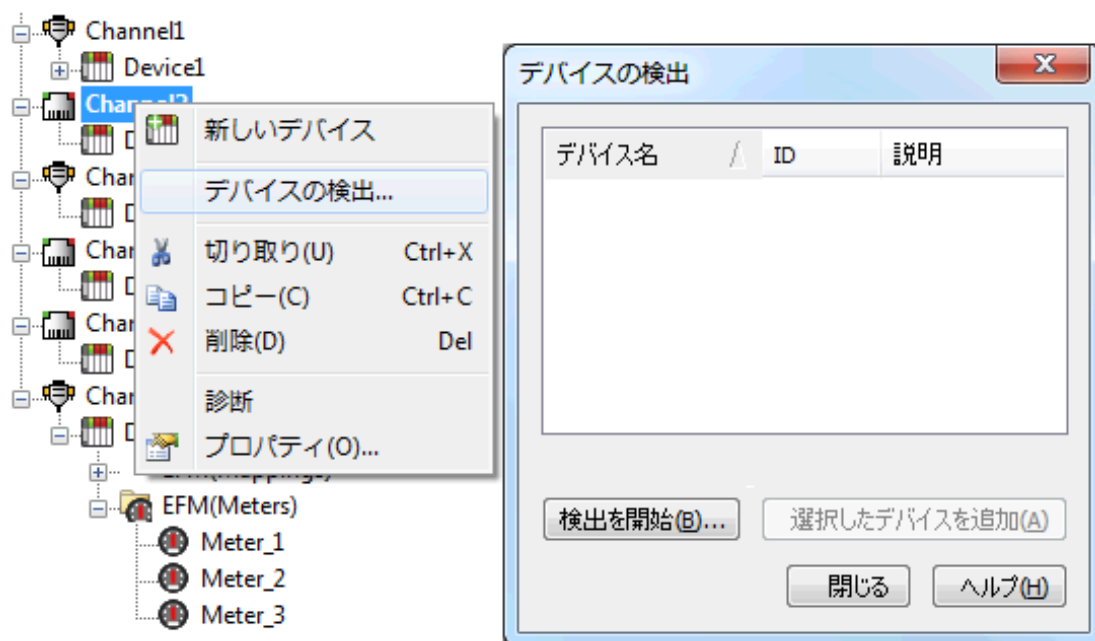
- 「すべてのタグのすべての値を書き込み」: このオプションを選択した場合、サーバーはすべての値をコントローラに書き込もうとします。このモードでは、サーバーは書き込み要求を絶えず収集し、サーバーの内部書き込みキューにこれらの要求を追加します。サーバーは書き込みキューを処理し、デバイスにできるだけ早くデータを書き込むことによって、このキューを空にしようとする。このモードでは、クライアントアプリケーションから書き込まれたすべてのデータがターゲットデバイスに送信されます。ターゲットデバイスで書き込み操作の順序または書き込みアイテムのコンテンツが一意に表示される必要がある場合、このモードを選択します。
- 「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。

デバイスの検出手順

デバイスの検出は、ネットワーク上でデバイスを特定可能なドライバーで実行できます。検出されたデバイスをチャネルに追加できます。一度に検出可能なデバイスの最大数は 65535 です。



1. デバイスを検出して追加するチャンネルを選択します。
2. チャンネルノードを右クリックして「デバイスの検出...」を選択します。
3. 「検出を開始...」ボタンをクリックして検出プロセスを開始します。
4. アドレス範囲、タイムアウト、検出範囲など、ドライバー固有の検出プロパティを指定します。
5. 「OK」をクリックします。
6. ダイアログの「名前」、「ID」、「説明」の各見出しに、検出されたデバイスの情報が表示されます。
7. 検出されたデバイスの中に目的のデバイスがある場合、そのデバイスを選択し、「選択したデバイスを追加...」をクリックします。
8. 「閉じる」をクリックします。

デバイスとは

デバイスは、サーバーが通信する PLC またはその他のハードウェアを表します。チャンネルが使用しているデバイスドライバーは、デバイスの選択を制限します。

デバイスの追加

デバイスは、初期設定時およびその後「新しいデバイス」ウィザードを使用して追加できます。これを行うには、「編集」|「新しいデバイス」の順にクリックします。ユーザー定義の論理デバイス名となるデバイス名を入力するよう求められます。これは、デバイスの割り当てられたタグにアクセスするために OPC リンクで使用するブラウザブランチ名です。予約文字については、[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。

デバイスのネットワークでデバイスを一意に識別する数値または文字列であるネットワーク ID の入力も求められます。サーバーのデータリクエストが正常にルーティングされるように、ネットワークに接続されたマルチドロップデバイスには一意の識別子が必要です。マルチドロップ以外のデバイスには ID は不要です。この設定は使用できません。

デバイスの除去

デバイスをプロジェクトから除去するには、目的のデバイスを選択して「削除」を押します。または、「編集」|「削除」の順にクリックします。

デバイスのプロパティの表示

デバイスのプロパティを表示するには、まずデバイスを選択し、「編集」|「プロパティ」の順にクリックします。

● 詳細については、[デバイスのプロパティ](#)を参照してください。

デバイスのプロパティ - 一般

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

Property Groups	<input checked="" type="checkbox"/> Identification	
General	Name	
Scan Mode	Description	
	Channel Assignment	
	Driver	
	Model	
	ID Format	Decimal
	ID	2

識別

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

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

「説明」: このデバイスに関するユーザー定義の情報。

- 「説明」などのこれらのプロパティの多くには、システムタグが関連付けられています。

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

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

「ID」: このプロパティでは、デバイスのドライバー固有のステーションまたはノードを指定します。入力する ID のタイプは、使用されている通信ドライバーによって異なります。多くの通信ドライバーでは、ID は数値です。数値 ID をサポートするドライバーでは、ユーザーは数値を入力でき、そのフォーマットはアプリケーションのニーズまたは選択した通信ドライバーの特性に合わせて変更できます。フォーマットはデフォルトではドライバーによって設定されます。オプションには「10 進数」、「8 進数」、「16 進数」があります。

- **注記**: ドライバーがイーサネットベースであるか、通常とは異なるステーションまたはノード名をサポートしている場合、デバイスの TCP/IP アドレスをデバイス ID として使用できます。TCP/IP アドレスはピリオドで区切った 4 つの値から成り、各値の範囲は 0 から 255 です。一部のデバイス ID は文字列ベースです。ドライバーによっては、ID フィールドで追加のプロパティを設定する必要があります。詳細については、ドライバーのヘルプドキュメントを参照してください。

動作モード

Property Groups	<input checked="" type="checkbox"/> Identification	
General	<input checked="" type="checkbox"/> Operating Mode	
Scan Mode	Data Collection	Enable
	Simulated	No

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

「**シミュレーション**」: このオプションは、デバイスをシミュレーションモードにします。このモードでは、ドライバ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のデータとして扱います。つまり、シミュレーションモードのデバイスに書き込まれたデータはすべて再び読み取られ、各 OPC アイテムは個別に処理されます。アイテムのメモリマップはグループ更新レートに基づきます。(サーバーが再初期化された場合などに) サーバーがアイテムを除去した場合、そのデータは保存されません。デフォルトは「いいえ」です。

● **注記:**

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

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

デバイスのプロパティ - スキャンモード

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

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、要求ポールのみ**」: このモードでは、デバイスに属するタグは定期的にポーリングされず、アクティブになった後はアイテムの初期値の読み取りは実行されません。更新のポーリングは、_DemandPoll タグに書き込むか、個々のアイテムについて明示的なデバイス読み取りを実行することによって、クライアントが行います。詳細については、サーバーのヘルプで「デバイス要求ポール」を参照してください。
- 「**タグに指定のスキャン速度を適用**」: このモードでは、静的構成のタグプロパティで指定されている速度で静的タグがスキャンされます。動的タグはクライアントが指定したスキャン速度でスキャンされます。

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

デバイスのプロパティ - 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	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 - 通信パラメータ

イーサネットカプセル化モードは、イーサネットネットワーク上のターミナルサーバーに接続しているシリアルデバイスとの通信用に設計されています。ターミナルサーバーは基本的には仮想のシリアルポートです。ターミナルサーバーはイーサネットネットワーク上の TCP/IP メッセージをシリアルデータに変換します。メッセージがシリアル形式に変換されると、ユーザーはシリアル通信をサポートする標準デバイスをターミナルサーバーに接続可能になります。

● **詳細**については、サーバーのヘルプで「イーサネットカプセル化の使用方法」を参照してください。

● **注記**: イーサネットカプセル化モードは実際のシリアル通信ドライバーに対して完全に透過的なので、ユーザーは残りのデバイスプロパティを、これらがあたかもローカル PC のシリアルポート上で直接デバイスに接続しているかのように設定する必要があります。

「IP アドレス」: このプロパティには、デバイスが接続しているターミナルサーバーの 4 つのフィールドから成る IP アドレスを入力します。IP は YYY.YYY.YYY.YYY として指定します。YYY は IP アドレスを示します。各 YYY バイトが 0 から 255 の範囲でなければなりません。各シリアルデバイスは独自の IP アドレスを持つことができますが、単一のターミナルサーバーからマルチドロップされた複数のデバイスがある場合、複数のデバイスが同じ IP アドレスを持つことがあります。

「ポート」: このプロパティでは、リモートターミナルサーバーに接続する際に使用するイーサネットポートを設定します。

「プロトコル」: このプロパティでは、TCP/IP 通信または UDP 通信を選択します。この選択は使用されているターミナルサーバーの特性によります。デフォルトのプロトコル選択は TCP/IP です。使用可能なプロトコルの詳細については、ターミナルサーバーのヘルプドキュメントを参照してください。

● **注記**:

1. 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。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。
2. IP アドレスの有効な範囲は 0.0.0.0 から 255.255.255.255 です (0.0.0.0 と 255.255.255.255 は含まれません)。

デバイスのプロパティ - イーサネットカプセル化

イーサネットカプセル化は、イーサネットネットワーク上のターミナルサーバーに接続しているシリアルデバイスとの通信用に設計されています。ターミナルサーバーは基本的には仮想のシリアルポートです。ターミナルサーバーはイーサネットネットワーク上のTCP/IPメッセージをシリアルデータに変換します。メッセージがシリアル形式に変換されると、ユーザーはシリアル通信をサポートする標準デバイスをターミナルサーバーに接続可能になります。

● 詳細については、サーバーのヘルプで「イーサネットカプセル化の使用方法」を参照してください。

● イーサネットカプセル化はドライバーに対して透過的なので、残りのプロパティを、これらがあたかもローカルシリアルポート上で直接デバイスに接続しているかのように設定します。

プロパティグループ	イーサネット設定	
一般	IP アドレス	
スキャンモード	ポート	2101
イーサネットカプセル化	プロトコル	TCP/IP

「IP アドレス」: このプロパティには、デバイスが接続しているターミナルサーバーの4つのフィールドから成るIPアドレスを入力します。IPはYYYY.YYYY.YYYY.YYYYとして指定します。YYYYはIPアドレスを示します。各YYYYバイトが0から255の範囲でなければなりません。各シリアルデバイスは独自のIPアドレスを持つことができますが、単一のターミナルサーバーからマルチドロップされた複数のデバイスがある場合、複数のデバイスが同じIPアドレスを持つことがあります。

「ポート」: このプロパティでは、リモートターミナルサーバーに接続する際に使用するイーサネットポートを設定します。

「プロトコル」: このプロパティでは、TCP/IP通信またはUDP通信を選択します。この選択は使用されているターミナルサーバーの特性によります。デフォルトのプロトコル選択はTCP/IPです。使用可能なプロトコルの詳細については、ターミナルサーバーのヘルプドキュメントを参照してください。

● 注記

1. 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。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。
2. IPアドレスの有効な範囲は0.0.0.0から255.255.255.255です(0.0.0.0と255.255.255.255は含まれません)。

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	<input type="checkbox"/> タグ生成	
一般	デバイス起動時	起動時に生成しない
スキャンモード	重複タグ	作成時に削除
タイミング	親グループ	
自動格下げ	自動生成されたサブグループを許可	有効化
タグ生成		

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

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

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

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

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

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。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 - 時刻の同期化

このグループは、デバイスのタイムゾーンと時刻同期化のプロパティを指定するときに使用します。これは主に、タイムスタンプ付きデータや、リモート位置にありデバイスの時刻がずれる (これによってタイムスタンプ付きデータに問題が生じる) 可能性があるバッテリー電源デバイスに適用されます。この問題が発生しないようにするため、ユーザーはサーバーがデバイスの時刻を同期化するように指定できます。

プロパティグループ	
一般	
スキャンモード	
タイミン	
自動格下げ	
タグ生成	
時刻の同期化	
	タイムゾーン
	タイムゾーン (UTC-05:00) Eastern Time (US & C...
	サマータイムを適用 いいえ
	同期化
	時刻同期化方法 絶対
	同期化絶対 12:00:00 AM

● **注記:** すべてのドライバーとモデルですべてのオプションがサポートされているわけではありません。

「タイムゾーン」: このプロパティでは、デバイスのタイムゾーンを指定します。タイムゾーンを無視するには、リストの先頭にある 4 つのオプション (オフセットがない) のうちの 1 つを選択します。デフォルトはローカルシステムのタイムゾーンです。

● **注記:** ドライバーは、デバイスの時刻を同期化するときと EFM タイムスタンプをデバイスの時刻から UTC 時刻に変換するときの両方でこのプロパティを使用します。

「サマータイムを適用」: デバイスの時刻を同期化するときサマータイムのオフセットに従うには、「はい」を選択します。サマータイムを無視するには、「いいえ」を選択します。サマータイム制が導入されているタイムゾーンのみが影響を受けます。デフォルトは「いいえ」(無効) です。

● **注記:** 有効な場合、サマータイムになると (春) デバイスの時間が +1 時間調整され、サマータイムが終了すると (秋) -1 時間調整されます。

「時刻同期化方法」: このプロパティを使用して、同期化の方法を指定します。オプションには、「無効」、「絶対」、「間隔」があります。デフォルトは「無効」です。オプションの説明は次のとおりです。

- 「無効」: 同期化を行いません。
- 「絶対」: 「時刻」プロパティ (「絶対」を選択した場合にのみ表示) で指定した時刻で毎日同期化します。
- 「間隔」: 起動時および「同期化間隔」プロパティ (「間隔」を選択した場合にのみ表示) で指定した分数ごとに同期化します。デフォルトは 60 分です。
- 「ポーリング時」: ポーリングが完了すると同期化します (EFM デバイスにのみ適用)。

「時刻同期化しきい値」: このプロパティを使用して、デバイス時刻とシステム時刻との間で許可される最大の差異時間を秒単位で指定します。この値よりも差異が大きくなると、デバイス時刻がシステム時刻に同期化されます。このしきい値を 0 に設定すると、常に時刻が同期化されます。デフォルト値は 0 秒です。許可される最大のしきい値は 600 秒です。

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	
一般	
スキャンモード	
タイミン	
自動格下げ	
	通信タイムアウト
	要求のタイムアウト (ミリ秒) 5000
	再試行回数 3
	タイミン
	要求間遅延 (ミリ秒) 0

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なる場合があります。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**」: このプロパティでは、ターゲットデバイスからの応答を待つのをいつやめるかを判断する際にすべてのドライバーが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 (167.6667 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**」: このプロパティでは、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、各ドライバ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。

タイミグ

「**要求間遅延**」: このプロパティでは、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。

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。

タグとは

タグは、サーバーの通信対象となる PLC またはその他のハードウェアデバイス内のアドレスを表します。サーバーでは、動的タグとユーザー定義の静的タグの両方を使用できます。動的タグは OPC クライアントに直接入力するもので、デバイスのデータを指定します。ユーザー定義の静的タグはサーバーで作成されるもので、タグのスケール変換をサポートします。これらのタグは、タグのブラウズをサポートする OPC クライアントからブラウズできます。

タグのプロパティの表示

特定のタグのタグプロパティを呼び出すには、サーバー設定のタグ選択表示枠でそのタグをダブルクリックします。

タグ名	アドレス	データ型	スキャン...	スケール...	説明
Tag1	40001	Word	100	None	
Tag2	40002	Word	100	None	
Tag3	40003	Word	100	None	
Tag4	40004	Float	100	None	
Tag5	40005	Word	100	None	
Tag6	40006	Word	100	Square Root	
Tag7	40007	Word	100	None	
Tag8	40008	Word	100	None	
Tag9	40009	Word	100	None	
Tag10	40010	Word	50	None	
Tag11	40011	Word	100	None	
Tag12	40012	Word	100	None	
Tag13	40013	Word	100	None	
Tag14	40014	Word	100	Linear	
Tag15	40015	Word	100	None	
Tag16	40016	Word	100	None	
Tag17	40017	Word	100	None	
Tag18	40018	LBCD	100	None	
Tag19	40019	Word	100	None	
Tag20	40020	Word	100	None	
Tag21	40021	Word	100	None	
Tag22	40022	Word	100	None	

タグのプロパティ - 一般

タグは、サーバーの通信対象となる PLC またはその他のハードウェアデバイスのアドレスを表します。サーバーでは、動的タグとユーザー定義の静的タグの両方を使用できます。動的タグは OPC クライアントに直接入力するもので、デバイスのデータを指定します。ユーザー定義の静的タグはサーバーで作成されるもので、タグのスケール変換をサポートします。これらのタグは、タグのブラウズをサポートする OPC クライアントからブラウズできます。

● 詳細については、[動的タグ](#)および[静的タグ \(ユーザー定義\)](#)を参照してください。

プロパティグループ	識別
一般	名前 Tag1
スケール変換	説明
	データプロパティ
	アドレス 40001
	データ型 Word
	クライアントアクセス 読み取り/書き込み
	スキャン速度 (ミリ秒) 100

「名前」: このタグを表す文字列を入力します。タグ名の長さは最大 256 文字です。通常はわかりやすい名前を使用することをお勧めしますが、一部の OPC クライアントアプリケーションでは、OPC サーバーのタグ空間をブラウズするとき、表示ウィンドウの領域が限られている可能性があります。タグ名は、OPC ブラウズデータタグ名の一部であり、特定のデバイスブランチまたはタググループブランチ内で一意でなければなりません。予約文字については、[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。

● ヒント: このアプリケーションが、同じ名前を持つタグのブロックの使用に最も適している場合は、タググループを使用してタグを分離します。詳細については、[タググループのプロパティ](#)を参照してください。

「説明」: タグにコメントを適用します。最大 255 文字の文字列を説明として入力できます。データアクセス 2.0 のタグのプロパティをサポートする OPC クライアントを使用している場合は、タグのアイテムの説明プロパティからこの説明プロパティにアクセスできます。

「**アドレス**」: ターゲットタグのドライバアドレスを入力します。アドレスのフォーマットはドライバのプロトコルに基づきます。アドレスの長さは最大 128 文字です。

● **ヒント**: アドレスを入力する方法に関するヒントを表示するには、参照 (...) ボタンをクリックします。入力されたアドレスをドライバが受け入れると、メッセージは表示されません。ポップアップがすべてのエラーを通知します。アドレス文字列ではなく、データ型の選択に関連するエラーもあります。

「**データ型**」: 物理デバイスで検出されたこのタグのデータのフォーマットを指定します。ほとんどの場合、これはクライアントに返されるデータのフォーマットでもあります。データ型の設定は、通信ドライバがデバイスにデータを読み込む/書き込む方法に関する重要な要素です。多くのドライバでは、データの特定部分のデータ型は厳密に固定されており、ドライバは、デバイスのデータの読み込み時に使用する必要があるフォーマットを認識しています。ただし、場合によっては、デバイスのデータの解釈はユーザーに大きく依存します。その例として、16 ビットのデータレジスタを使用するデバイスがあります。通常、これはデータが Short または Word であることを示します。レジスタベースのデバイスの多くは、2 つのレジスタにまたがる値もサポートします。このような場合、二重のレジスタ値を Long、DWord、または Float にできます。使用しているドライバでこのレベルの柔軟性がサポートされている場合は、このタグのデータの読み取り方法をユーザーが指定する必要があります。適切なデータ型を選択することで、1、2、4、8、または 16 個のレジスタを、またはブール値を読み取るようにドライバに指示が与られます。ドライバは、選択されているデータフォーマットを制御します。

- 「**デフォルト**」 - ドライバのデフォルトのデータ型を使用します。
- 「**Boolean**」 - 真または偽のバイナリ値
- 「**Char**」 - 符号付き 8 ビット整数データ
- 「**Byte**」 - 符号なし 8 ビット整数データ
- 「**Short**」 - 符号付き 16 ビット整数データ
- 「**Word**」 - 符号なし 16 ビット整数データ
- 「**Long**」 - 符号付き 32 ビット整数データ
- 「**DWord**」 - 符号なし 32 ビット整数データ
- 「**LLong**」 - 符号付き 64 ビット整数データ
- 「**QWord**」 - 符号なし 64 ビット整数データ
- 「**Float**」 - 32 ビット実数値 IEEE-754 標準定義
- 「**Double**」 - 64 ビット実数値 IEEE-754 標準定義
- 「**String**」 - Nul 終端 Unicode 文字列
- 「**BCD**」 - 2 バイトパック BCD の値の範囲は 0 から 9999 です。
- 「**LBCD**」 - 4 バイトパック BCD 値の範囲は 0 から 99999999 です。
- 「**Date**」 - [Microsoft® 知識ベース](#)を参照してください。

「**クライアントアクセス**」: タグが「**読み取り専用**」であるか、「**読み取り書き込み**」であるかを指定します。「**読み取り専用**」を選択すると、このタグに含まれるデータがクライアントアプリケーションによって変更されることを防止できます。「**読み取り書き込み**」を選択すると、必要に応じて、このタグの値の変更をクライアントアプリケーションに許可することができます。「**クライアントアクセス**」を選択すると、OPC クライアントのブラウズ領域にタグを表示する方法にも影響を与えます。多くの OPC クライアントアプリケーションでは、属性に基づいたタグのフィルタリングもできます。このタグのアクセス方法を変更すると、OPC クライアントのブラウズ領域にタグを表示する方法とそのタイミングも変更されることがあります。

「**スキャン速度**」: OPC 以外のクライアントで使用している場合のこのタグの更新間隔を指定します。OPC クライアントは、すべての OPC グループの一部である更新レートを使用して、データのスキャン速度を制御できます。通常は、OPC 以外のクライアントにはその機能がありません。OPC 以外のクライアントでは、サーバーを使用して、タグごとに更新レートを指定します。スキャン速度を使用すると、アプリケーションのニーズに合わせてサーバーの帯域幅要件を調整できます。たとえば、変更が非常に遅いデータを読み取る必要がある場合は、値を頻繁に読み取る理由はありません。スキャン速度を使用すると、このタグに遅い速度で読み取らせ、必要な通信チャネルを減らすことができます。有効範囲は、10 ミリ秒単位で 10 から 99999990 ミリ秒 (ms) です。デフォルトは 100 ミリ秒です。

● サーバーがオンラインで常時稼働している場合は、これらのプロパティをいつでも変更できます。タグのプロパティに加えられた変更はただちに有効になります。ただし、このタグにすでに接続されている OPC クライアントは、タグを解放し、再び取得しようとするまでは影響を受けません。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。

複数タグの生成

「複数タグの生成」ツールは、ユーザー定義のドライバーの命名法を使用して複数のタグを動的に作成します。これにより、さまざまなアドレスフォーマット（10 進数、16 進数、8 進数システムを使用する範囲など）を使用できます。「複数タグの生成」ツールには、データの重複を回避するために、ユーザー定義のデータ型によって増分する機能もあります。

特定のダイアログの詳細については、以下のリストからリンクを選択してください。

[数値範囲を追加](#)

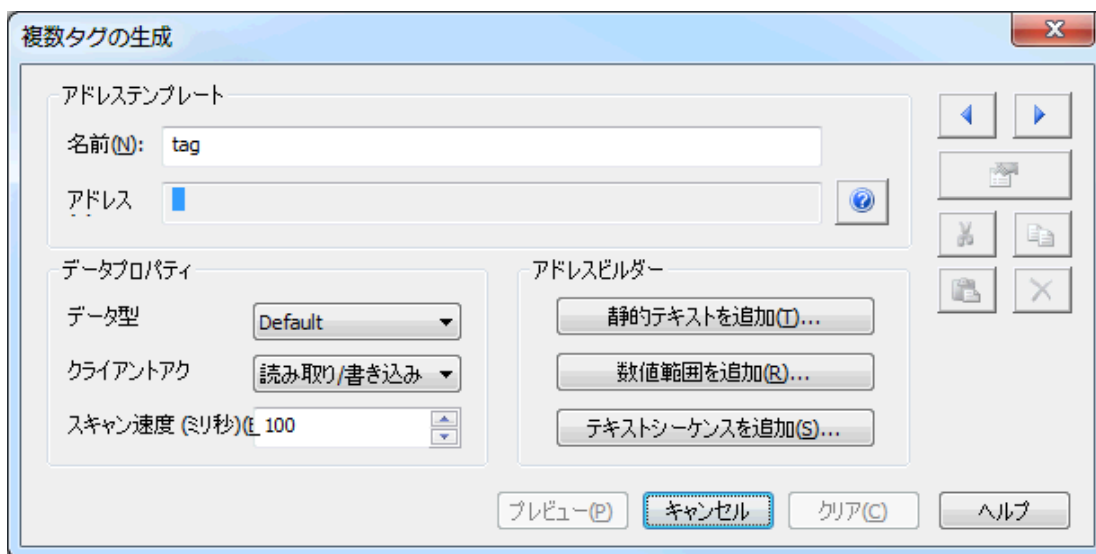
[静的テキストを追加](#)

[テキストシーケンスを追加](#)

[複数タグ生成のプレビュー](#)

[タグ名のプロパティ](#)

複数タグの生成



アドレステンプレート

「名前」: ユーザー定義のタグ名を入力します。

「アドレス」: 「アドレスビルダー」セクションで定義されているオプションを使用して生成されたタグアドレスを検証します。

データプロパティ

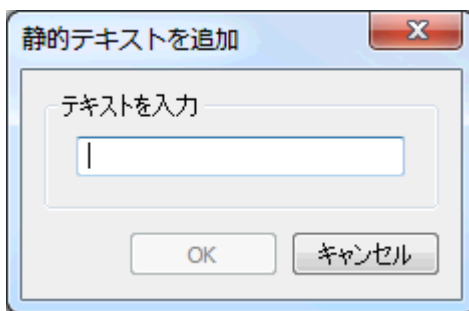
「データ型」: 生成されたすべてのタグに適用するデータ型を選択します。ドライバーでサポートされているネイティブインタフェースによっては、このデータ型が、最後の要素に適用されている「数値範囲を追加」プロパティのデフォルトの増分をオーバーライドする場合があります。デフォルト設定は「デフォルト」です。

「クライアントアクセス」: タグのアクセス許可設定を「読み取り専用」または「読み取り/書き込み」から選択します。デフォルト設定は「読み取り専用」です。

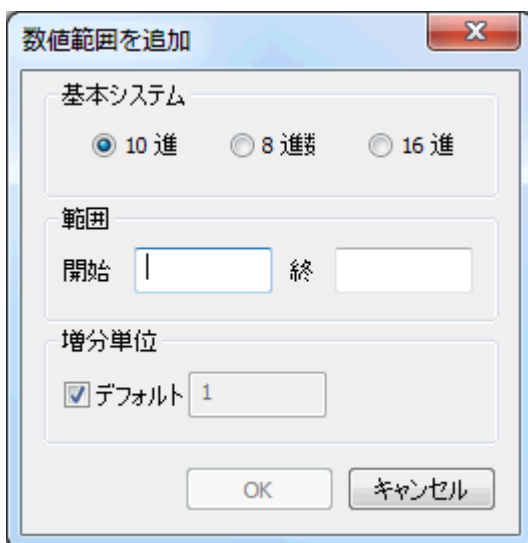
「スキャン速度」: タグがスキャンされる頻度を指定します。有効な範囲は 10 から 99999990 ミリ秒です。デフォルトの設定は 100 ミリ秒です。

アドレスビルダー

「静的テキストを追加...」: これをクリックすると、「静的テキストを追加」ダイアログが起動します。このダイアログでは、単一行のテキストを入力できます。

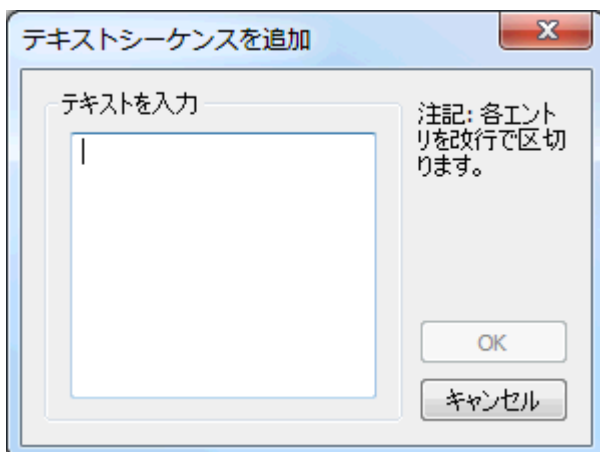


「数値範囲を追加...」: これをクリックすると、「数値範囲を追加」ダイアログが起動します。



- ・「基本システム」基本システムのフォーマットを「10進数」、「8進数」、または「16進数」から選択します。デフォルト設定は「10進数」です。
- ・「範囲」「開始」フィールドと「終了」フィールドに数値範囲の開始値と終了値を入力します。
- ・「増分単位」デフォルト (1 つずつ増分) を使用しない場合は、カスタム増分値を指定できます。範囲は、選択した基本システムに従って増分します。

「テキストシーケンスを追加...」: これをクリックすると、「テキストシーケンスを追加」ダイアログが起動します。このダイアログでは、複数の文字列を作成できます。文字列はそれぞれ、リストで指定されているほかの文字列とは切り離されて挿入されます。



■ ヒント

1. 右側にある「編集」アイコンを有効にするには、タグアドレス構文要素のセクションをハイライトします。
2. 「ヒント」アイコンをクリックすると、「アドレス説明」でヘルプファイルが開きます。

「プレビュー」: これをクリックすると、生成されたタグのテストビューが生成されます。

複数タグ生成のプレビュー

The dialog box titled "Tag List Preview" contains a table with the following data:

Tag Name	Address	Access	Data Type	Scan Rate
Tag1	K0001	Read Only	Word	10
Tag2	K0002	Read Only	Word	10
Tag3	K0003	Read Only	Word	10
Tag4	K0004	Read Only	Word	10
Tag5	K0005	Read Only	Word	10
Tag6	K0006	Read Only	Word	10
Tag7	K0007	Read Only	Word	10
Tag8	K0008	Read Only	Word	10
Tag9	K0009	Read Only	Word	10
Tag10	K0010	Read Only	Word	10
Tag11	K0011	Read Only	Word	10
Tag12	K0012	Read Only	Word	10
Tag13	K0013	Read Only	Word	10
Tag14	K0014	Read Only	Word	10
Tag15	K0015	Read Only	Word	10
Tag16	K0016	Read Only	Word	10
Tag17	K0017	Read Only	Word	10
Tag18	K0018	Read Only	Word	10
Tag19	K0019	Read Only	Word	10
Tag20	K0020	Read Only	Word	10

On the right side of the dialog, there are several controls:

- Buttons: "Generate", "Cancel", "Tag Name...", and "Help".
- Text field: "Number of tags:" with the value "21".
- Checkboxes:
 - ☐ Add as Group
 - ☒ Renumber valid tags consecutively before adding to project

「生成」: これをクリックすると、有効なすべてのタグがサーバーに送信されて挿入されます。

「キャンセル」: これをクリックすると、タグに対して加えられた変更が却下され、直前のダイアログに戻ります。

「タグ名...」: これをクリックすると、「タグ名のプロパティ」ダイアログが起動されます。

「グループとして追加」: これを有効にすると、タグが単一の組織グループに追加されます。デフォルトでは無効に設定されています。

「プロジェクトに追加する前に有効なタグに連番を再割り当て」: これを有効にすると、タグがプロジェクトに追加する前に連番に変更されます。デフォルト設定では有効になっています。

● 注記: 緑色のチェックマークで示されているタグは有効です。赤色の感嘆符 (!) で示されているタグは無効です。

タグ名のプロパティ

「複数タグの生成」ツールには、カスタム命名スキームを使用するオプションがあるため、すべてのタグに対して名前プレフィックスと数値のサフィックスの両方を指定できます。数値のサフィックスは自動的にタグごとに増分されるため、タグのカスタム名を作成して、読みやすくすることができます。割り当てられたタグ名は、生成後に変更できます。「タグ名のプロパティ」ダイアログでカスタム名が定義されてなければ、生成された各タグにはデフォルトの命名スキームが実装されます。

● **注記:**「タグ重複」ダイアログに戻る前に、アドレス指定構文を変更するために「生成」ダイアログで命名スキームを変更した場合、次回タグリストが生成されるときに使用されるように命名スキームを保存できます。

「**名前のプレフィックス**」: カスタムの名前のプレフィックス (タグ名の先頭に追加する文字) を入力します。

「**開始値**」: タグごとの増分の最初の数値を指定します。

「**デフォルトの命名スキーム**」: これを有効にすると、デフォルトの命名スキームが使用されます。デフォルトでは無効に設定されています。

● **関連項目:** [複数のタグの生成](#)

タグのプロパティ - スケール変換

このサーバーは、タグのスケール変換をサポートしています。デバイスから提供される生データのスケールを、アプリケーションにとって妥当な範囲に変換できます。

プロパティグループ	<div> <input checked="" type="checkbox"/> スケール変換 </div>	
一般		
スケール変換		
	タイプ	線形
	生データ下限	0
	生データ上限	1000
	スケール変換後のデータ型	Double
	スケール変換後の下限	0
	スケール変換後の上限	1000
	下限でクランプ	いいえ
	上限でクランプ	いいえ
	値を負数化	いいえ
	単位	

「**タイプ**」: 生データ値のスケール変換方法を選択します。「線形」、「平方根」、または「なし」(無効にする場合)を選択します。スケール変換の計算式を以下に示します。

タイプ	値のスケール変換の計算式
線形	$(((\text{ScaledHigh} - \text{ScaledLow}) / (\text{RawHigh} - \text{RawLow})) * (\text{RawValue} - \text{RawLow})) + \text{ScaledLow}$
平方根	$(\text{平方根 } ((\text{RawValue} - \text{RawLow}) / (\text{RawHigh} - \text{RawLow})) * (\text{ScaledHigh} - \text{ScaledLow})) + \text{ScaledLow}$

「**生データ下限**」: デバイスからのデータの範囲の下限を指定します。有効範囲はタグの生データのデータ型によって決まります。たとえば、生データ値が Short の場合、有効となる範囲は -32768 から 32767 です。

「**生データ上限**」: デバイスからのデータの範囲の上限を指定します。生データ上限の値は生データ下限の値より大きくなければなりません。有効範囲はタグの生データのデータ型によって決まります。

「**スケール変換後のデータ型**」: スケール変換後のタグのデータ型を選択します。データ型を、Short などの生データ型を含む任意の有効な OPC データ型や、Long データ型のエンジニアリング値に設定できます。スケール変換後のデフォルトのデータ型は、Double です。

「**スケール変換後の下限**」: スケール変換後の有効なデータ値の範囲の下限を指定します。有効範囲はタグのデータ型によって決まります。

「**スケール変換後の上限**」: スケール変換後の有効なデータ値の範囲の上限を指定します。有効範囲はタグのデータ型によって決まります。

「**下限でクランプ**」: 変換後のデータが指定した範囲の下限を下回らないようにするには、「はい」を選択します。設定されている範囲からデータが外れることを許可するには、「いいえ」を選択します。

「**上限でクランプ**」: 変換後のデータが指定した範囲の上限を上回らないようにするには、「はい」を選択します。設定されている範囲からデータが外れることを許可するには、「いいえ」を選択します。

「**値を負数化**」: 変換後の値をクライアントに渡す前に値の正負を反転するには、「はい」を選択します。値を修正せずにクライアントに渡すには、「いいえ」を選択します。

● このサーバーは、データアクセス仕様 2.0 で利用可能となった OPC タグプロパティをサポートしています。使用される OPC クライアントがこれらのプロパティをサポートしている場合、サーバーは、オブジェクトの範囲 (たとえば、ユーザー入力オブジェクトや表示) を「スケール変換」設定を使用して自動的に設定できます。権限のないオペレータによってこれらのプロパティが変更されることを防止するには、ユーザーマネージャを使用して、サーバー機能へのアクセス権を制限します。

動的タグ

動的タグアドレス指定は、タグを定義する 2 つ目の方法であり、これを使用する場合、クライアントアプリケーションでのみタグを定義します。このため、サーバーで作成された別のタグアイテムのアドレスを指定するタグアイテムをクライアントで作成する代わりに、デバイスドライバのアドレスに直接アクセスするタグアイテムをクライアントで作成する必要があるだけです。サーバーはクライアント接続でその位置の仮想タグを作成し、自動的にデータのスキャンを開始します。

オプションのデータ型を指定するには、以下のいずれかの文字列を "@" シンボルの後に追加します。

- BCD
- Boolean
- Byte
- Char
- Double
- DWord
- Float
- LBCD
- LLong
- Long
- QWord
- Short
- 文字列
- Word

データ型を省略すると、ドライバは参照されているデバイスとアドレスに基づいてデフォルトのデータ型を選択します。それぞれのドライバのヘルプドキュメントに、すべての位置のデフォルトのデータ型が記載されています。指定されているデータ型がデバイスの位置に対して有効でなければ、サーバーはタグを却下し、イベントログにエラーが出力されます。

動的アドレス指定を使用する OPC クライアントの例

Simulator デバイスで 16 ビットの位置 "R0001" をスキャンします。以下の動的タグの例は、プロジェクトが例の一部として作成されていることを前提としています。

1. OPC クライアントアプリケーションを起動し、サーバーに接続します。
2. Simulator Driver を使用してチャンネルを作成し、それに "Channel1" という名前を付けます。次に、デバイスを作成し、それに "Device1" という名前を付けます。
3. クライアントアプリケーションで、アイテム名を "Channel1.Device1.R0001@Short" として定義します。

4. クライアントプロジェクトが自動的にデータの受信を開始します。Simulator デバイスでのアドレス R0001 のデフォルトのデータ型は Word です。これをオーバーライドして、データ型として Short を選択するために、@Short が追加されています。

● **注記:** 通常、OPC クライアントアプリケーションで動的タグを使用する場合、@[Data Type] 修飾子を使用する必要はありません。OPC クライアントでは、特定のデータアイテムのリンクを登録するときに、目的のデータ型を要求の一部として指定できます。OPC クライアントによって指定されるデータ型は、通信ドライバーによってサポートされている場合に使用されます。@[Data Type] 修飾子は、通信ドライバーが 1 つのデータを必要なおりに解釈する必要がある場合に役立ちます。

OPC 以外のクライアントの例

OPC 以外のクライアントでは、@[Update Rate] を追加することによってタグごとに更新レートをオーバーライドできます。

たとえば、

<DDE サービス名>[_ddedata!Device1.R0001@500 を追加すると、更新レートのみがオーバーライドされます。

<DDE サービス名>[_ddedata!Device1.R0001@500,Short を追加すると、更新レートとデータ型の両方がオーバーライドされます。

● **ヒント:**

1. サーバーは、プロジェクト内のすべてのデバイスに対して特殊な Boolean タグを作成します。クライアントは、これを使用して、デバイスが適切に機能しているかどうかを確認します。このタグを使用するには、リンク内のアイテムを "エラー" として指定します。このタグの値は、デバイスが適切に通信している場合は 0、適切に通信していない場合は 1 になります。
2. デバイスアドレスがサーバー内のユーザー定義のタグの名前と一致するようにリンクのアイテムとして使用されていると、リンクはユーザー定義のタグが指すアドレスを参照します。
3. サーバーでデータをスケール変換するには、静的タグを使用する必要があります。

● **関連項目:**

[静的タグ\(ユーザー定義\)](#)

[プロジェクトの設定: ユーザー定義のタグの追加](#)

静的タグ(ユーザー定義)

サーバーを使用してデバイスからクライアントアプリケーションにデータを取得する最も一般的な方法には、2 つの要件があります。ユーザーはまず、割り当てられたタグ名をクライアントとサーバー間の各リンクのアイテムとして使用して、サーバー内でタグのセットを定義する必要があります。この方法を使用する主な利点は、すべてのユーザー定義のタグが大部分の OPC クライアント内で参照できることです。静的タグを作成するかどうかを決定する前に、クライアントがサーバーからタグを参照またはインポートできることを確認します。

● **ヒント:** ユーザー定義タグは、スケーリングをサポートします。

タググループとは

このサーバーでは、プロジェクトにタググループを追加できます。タググループは、アプリケーションのニーズに合わせて、論理グループに OPC データのレイアウトをカスタマイズするために使用されます。タググループによって、複数の同一タグのセットを同じデバイスに追加できます。これは、単一のデバイスでいくつかの類似したマシンセグメントを扱う場合に便利です。

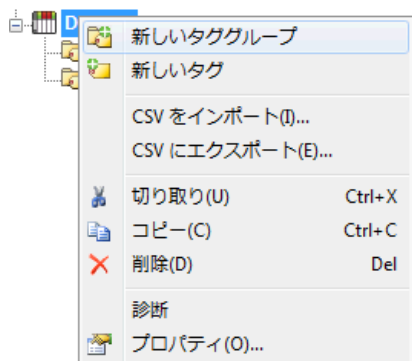
タググループのプロパティ

OPC クライアントという観点から、タググループによって OPC データを小さいタグリストに分類することで、特定のタグを簡単に検索できます。

次の図では、提供された OPC Quick Client を使用して Cell1 および Cell2 タググループを作成し、OPC クライアントのブラウズを簡略化しました。

プロパティグループ	
一般	
識別	
名前	Group1
説明	
タグ数	
グループ内のタグ	0
ブランチ内のタグ	0

新しいタググループをプロジェクトに追加するには、既存のデバイスまたはタググループブランチを右クリックし、コンテキストメニューの「新しいタググループ」を選択します。または、既存のデバイスまたはタググループブランチをクリックし、ツールバーの「新しいタググループ」アイコンをクリックします。



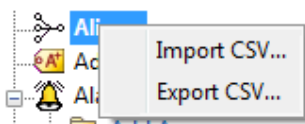
タググループは、デバイスレベルより下の任意のレベルに追加することもできます。また、アプリケーションのニーズに合わせて、複数のタググループをまとめてネストすることができます。上の「OPC Quick Client」ダイアログからわかるように、OPC アイテムの完全修飾パスは "Channel1.Device1.Machine1.Cell1.Tag1" です。この OPC アイテムでは、セグメント "Machine1" および "Cell1" はネストされたタググループです。

● **注記:** これらのプロパティは、サーバーをオンラインでフルタイム運用している状態で、いつでも変更できます。タググループに加えたすべての変更は、ただちに有効になります。名前を変更した場合、OPC アイテムリクエストの一部としてそのタググループをすでに使用している OPC クライアントは、アイテムを解放し、再び取得しようとするまで影響を受けません。プロジェクトに追加された新しいタググループによって、OPC クライアントからただちにブラウズできるようになります。サーバー機能へのアクセス権を制限してオペレータがプロパティを変更できないようにするには、ユーザーマネージャを使用します。

エイリアスマップとは

エイリアスマップは、従来のサーバーアプリケーションとの下位互換性のためのメカニズムに加えて、複雑なタグ参照に単純なエイリアス名を割り当てる方法を提供します。これは、タグアドレスパスのサイズを制限するクライアントアプリケーションで特に有効です。サーバーの最新バージョンではエイリアスマップが自動的に作成されますが、ユーザーが独自のエイリアスマップエントリを追加し、サーバーによって作成されたものを補完することもできます。サーバーにより作成されたエイリアスをフィルタし、自身が作成したものだけを表示することもできます。

ツリービューの枠でターゲットのエイリアスを右クリックすることで、エイリアスマップの要素をエクスポートおよびインポートできます。



詳細枠でターゲットのエイリアスを右クリックすることで、エイリアスマップの要素を追加、編集、削除できます。

Alias Name	Mapped To	Scan Rate
AdvancedTags	_AdvancedTags	0
Channel1_CommunicationSerialization	Channel1_CommunicationSerialization	0
Channel1_Statistics	Channel1_Statistics	0
Channel1_System	Channel1_System	0
Channel1_Device1	Channel1.Device1	0
Channel1_Device1_Statistics	Channel1.Device1_Statistics	0
Channel1_Device1_System	Channel1.Device1_System	0
Channel2_Statistics	Channel2_Statistics	0
Channel2_System	Channel2_System	0
Channel2_Device1	Channel2.Device1	0
Channel2_Device1_Statistics	Channel2.Device1_Statistics	0
Channel2_Device1_System	Channel2.Device1_System	0
Channel4_Statistics	Channel4_Statistics	0
Channel4_System	Channel4_System	0
Channel4_Device1	Channel4.Device1	0
Channel4_Device1_Statistics	Channel4.Device1_Statistics	0
Channel4_Device1_System	Channel4.Device1_System	0
Channel5_Statistics	Channel5_Statistics	0
Channel5_System	Channel5_System	0
Channel5_Device1	Channel5.Device1	0
Channel5_Device1_Statistics	Channel5.Device1_Statistics	0
Channel5_Device1_System	Channel5.Device1_System	0
Channel6_CommunicationSerialization	Channel6_CommunicationSerialization	0
Channel6_Statistics	Channel6_Statistics	0

● 注記: 有効にした場合、「自動生成されたエイリアスを表示」に、サーバーによって自動的に作成されたエイリアスマップが表示されます。

● 関連項目: [エイリアスの作成および使用方法](#)

エイリアスのプロパティ

エイリアスマップを使用すると、クライアントアプリケーションで使用できる複雑なタグ参照にエイリアス名を割り当てることができます。

エイリアスは、エイリアス名を入力し、目的のデバイス名またはグループ名をクリックすることによって作成されます。

プロパティグループ	識別	
一般	名前	Channel1_Statistics
	説明	
	エイリアスのプロパティ	
	マッピング先	Channel1_Statistics
	スキャン速度オーバーライド (ミリ秒)	0

「名前」: エイリアス名を指定します。長さは最大 256 文字です。これはエイリアスマップ内で一意である必要があります。予約文字については、[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」: このエイリアスのデータソースとレポートに関する説明を入力します (オプション)。

「マッピング先」: エイリアスの場所を指定するかブラウズします。エイリアスマップではタグアイテムをエイリアステーブルからブラウズできないため、タグへのアドレスを置き換える短いニックネームを作成します。これにより、タグをブラウズできないクライアントアプリケーションで簡単にアイテムのアドレスを設定できます。

「スキャン速度オーバーライド」: このエイリアスマップエントリを使用してアクセスされるすべての非 OPC タグに適用する更新レートを指定します。有効な範囲は 0 から 99999990 ミリ秒です。デフォルトは 0 ミリ秒です。

● **ヒント**: この設定は、DDE のみのサーバーの多くで使用されているトピック更新レートに相当します。

● **注記**: 0 ミリ秒に設定すると、サーバーは個々のタグレベルで設定されているスキャン速度を適用します。

イベントログとは

イベントログには、日付、時刻、およびエラー、警告、情報、またはセキュリティイベントの原因が表示されます。詳細については、以下のリストからリンクを選択してください。

イベントログのオプション

イベントログの設定

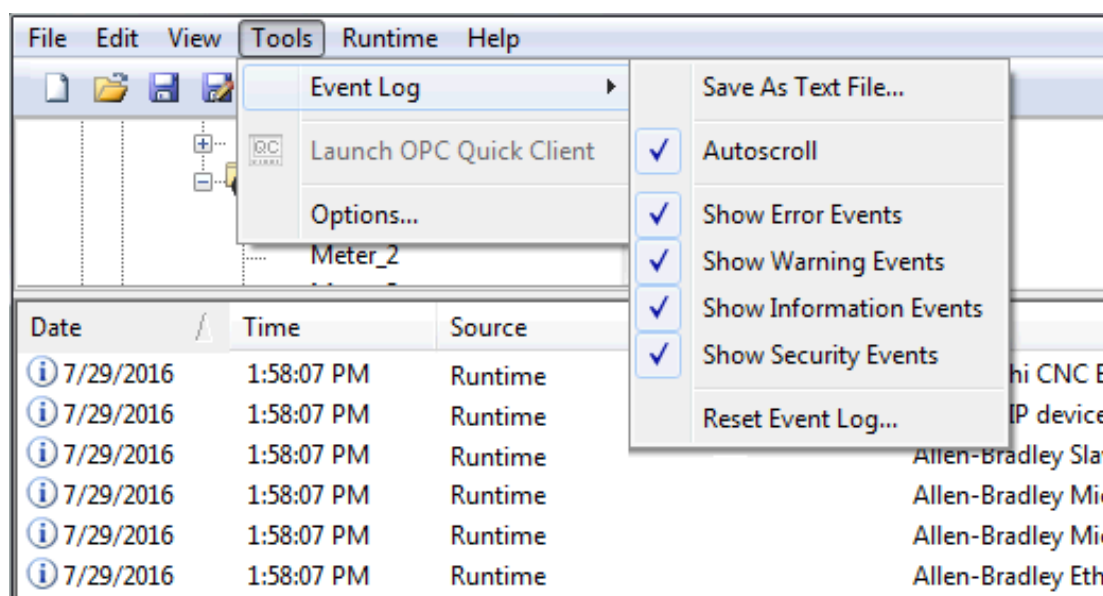
イベントログ

イベントログに表示されるイベントのタイプを指定できます。現時点では、エラーイベント、警告イベント、情報イベント、およびセキュリティイベントの 4 つのイベントのタイプを記録できます。それぞれのイベントの説明は次のとおりです。

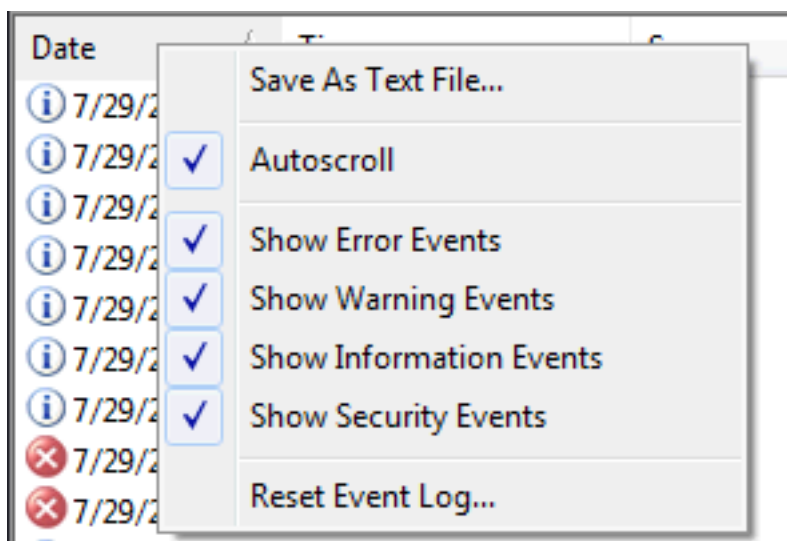
- **「情報」**: 接続やデータ収集の成功など、操作や修正を必要としないステータスやデータに関するメッセージ。
- **「セキュリティ」**: セキュリティの観点から最良事例ではない状態について注意を促すメッセージ。たとえば、有効な資格証明を持っているログイン済みのユーザーではなくデフォルトのユーザーとしてソフトウェアを実行しているなどがこれに該当します。
- **「警告」**: デバイスが応答しないなど、操作は必要としないが、予期しない結果になる可能性のある問題を示すメッセージ。
- **「エラー」**: 一般的に、調査したり、最良の結果を得るために修正したりする必要のある障害または問題についてユーザーに警告するメッセージ。

● **注記**: 構成クライアントでイベントタイプにアクセスするには、「ツール」|「イベントログ」の順にクリックします。または、イベントログの表示の任意の場所を右クリックします。

「ツール」メニュー



右クリック



● **注記:** イベントログシステムは、そのコンテンツを保護するメカニズムが存在しなければ、役に立ちません。オペレータがこれらのプロパティを変更したり、ログをリセットしたりできると、目的が失われます。オペレータがこれらの機能にアクセスするのを制限したり、これらの操作が発生するのを防止したりするには、ユーザーマネージャを使用します。

● **関連項目:** [設定 - イベントログ](#)

タグの管理

サーバーのユーザー定義のタグ管理機能により、各アプリケーションに固有の特性に合わせてタグデータベース構造を作成できます。複数のタググループを定義することにより、タグのデータをデバイスごとに分類できます。また、ドラッグ＆ドロップ編集によって多数のタグを簡単に追加することもできます。CSVのインポートとエクスポートを使用して、任意のアプリケーションでタグを編集することもできます。ほかのすべてのサーバー機能と同様に、新しいタグはいつでもアプリケーションに追加できます。

自動タグデータベース生成

選択した通信ドライバーのタグを自動的に生成する OPC サーバーの機能により、OPC テクノロジーがプラグアンドプレイ操作に一步近付きます。タグ情報はデバイスから直接読み取ることができます。また、タグを保存されたタグデータから生成することもできます。どちらの場合でも、OPC タグをサーバーに手動で入力する必要がなくなりました。

システムタグ

システムタグによって、クライアントアプリケーションに一般的なエラーフィードバックが提供され、デバイスがデータをアクティブに収集する時期を操作で制御できるようになり、さらに、チャンネルまたはデバイスの標準プロパティを OPC クライアントアプリケーションから変更できるようになります。チャンネルまたはデバイスレベルで使用可能なシステムタグの数は、使用しているドライバーの特性によって異なります。

● **注記:** システムタグは、その目的に従って、ステータスとコントロールの両方、またはプロパティ操作としてグループ化できます。

● **関連項目:** [ストアードフォワード - システムタグ](#)

プロパティタグ

プロパティタグは、プロパティ名を完全修飾タグアドレス追加することによって任意のデータアクセスクライアントからアクセスできる追加のタグです。アイテムのブラウズをサポートする OPC クライアントを使用している場合は、OPC DA 設定の「**クライアントがサーバーをブラウズする際にタグのプロパティを含める**」を有効にするとタグのプロパティを参照できます。詳細については、[プロジェクトのプロパティ - OPC DA](#) を参照してください。

統計タグ

統計タグは、サーバーのチャンネル通信の操作に関するフィードバックをクライアントアプリケーションに提供します。診断が有効な場合は、7 つの組み込みの統計タグを使用できます。詳細については、[OPC 診断ビューア](#) を参照してください。

モデムタグ

モデムタグは、モデムのプロパティを設定し、モデムのステータスを監視します。これらは、「チャンネルのプロパティ」の「**接続タイプ**」が「**モデム**」に設定されている場合にのみ使用できます。詳細については、[チャンネルのプロパティ - シリアル通信](#) を参照してください。

通信シリアル化タグ

通常、ドライバー通信は複数のチャンネルで同時に行われ、データスループットを向上させます。ただし、一部のアプリケーションでは、一度に 1 つのチャンネルのみの通信を許可するよう求められます。通信シリアル化がこれをサポートします。通信シリアル化タグは、チャンネルのシリアル化ステータスの設定と監視に使用されます。フィーチャーとそのタグの両方は、特定のドライバーのみで使用できます。詳細については、ドライバーのヘルプドキュメントを参照してください。

CSV のインポートとエクスポート

このサーバーは、タグデータをコンマ区切り変数 (CSV) ファイル形式でインポートおよびエクスポートすることで、アプリケーション内にタグを簡単に作成できます。CSV 関数は、デバイスまたはタググループが選択されているときにのみ使用できます。

● **注記:** 変数として指定する文字については、[オプション - 一般](#) を参照してください。

特定のセクションにジャンプするには、以下のリストからリンクを選択します。

[サーバータグリストのエクスポート](#)

[サーバーへのサーバータグリストのインポート](#)

[ほかの区切り記号文字の使用](#)

テンプレートの作成

CSV ファイル作成してインポートする最も簡単な方法は、テンプレートを作成することです。詳細については、次の手順を参照してください。

1. 最初に、「ファイル」|「CSV にエクスポート」の順にクリックします。プロジェクトに使用するチャネルとデバイスを定義します。
2. 各デバイスのタグを定義します。
3. 各デバイスまたはタググループを CSV ファイルとしてエクスポートします。
4. CSV ファイルをサポートするスプレッドシートアプリケーションでこのテンプレートを使用し、必要に応じてファイルを修正します。

● **注記:** 作成した CSV ファイルは、ディスクに保存して、同じ (または新しい) デバイスまたはタググループに使用するためにサーバーに再インポートできます。

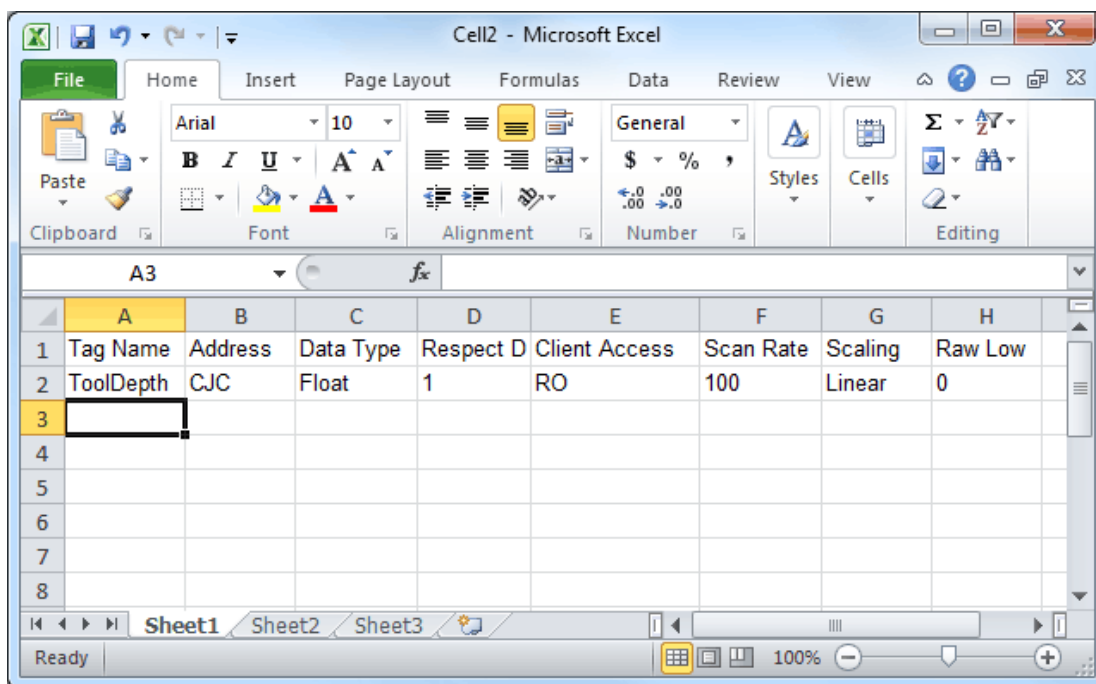
サーバータグリストのエクスポート

サーバータグリストをエクスポートすると、ヘッダーレコードに続いて、選択したデバイスまたはタググループに対して定義されている各タグのレコードが含まれている CSV テキストファイルが生成されます。ヘッダーレコードには以下のフィールドが含まれています。

- 「**タグ名**」: OPC クライアントで参照されているタグの名前。
● タグ名にグループ名がピリオドでタグ名から分離されてプレフィックスとして含まれている場合があります。たとえば、タグ名が "Group1.Tag1" であれば、"Tag1" を含んでいる "Group1" という名前のグループが作成されます。
- 「**アドレス**」: タグによって参照されているデバイスの位置。
- 「**データ型**」: サーバータグのデータ型ドロップダウンリストに表示されるタグに使用されているデータ型。
- 「**データ型を考慮**」: これを選択すると、タグが、OPC クライアント要求 (1, 0) ではなく、タグに対して定義されているデータ型に従います。
- 「**クライアントアクセス**」: 読み取り/書き込みアクセス (読み取り専用と読み取り/書き込み)。
- 「**スキャン速度**」: タグアドレスが OPC 以外のクライアントで使用される場合に (例外あり) スキャンされる速度 (ミリ秒)。
- 「**スケール変換**」: スケールモード (なし、線形、および平方根)。
- 「**生データ下限**」: 生データ下限値。
- 「**生データ上限**」: 生データ上限値。
- 「**スケール変換後の下限**」: スケール変換後の下限値。
- 「**スケール変換後の上限**」: スケール変換後の上限値。
- 「**スケール変換後のデータ型**」: スケール変換が適用された後にタグに使用されるデータ型。
- 「**下限でクランプ**」: これを選択すると、スケール変換後の値が「スケール変換後の下限」の値内に制限されます (1, 0)。
- 「**上限でクランプ**」: これを選択すると、スケール変換後の値が「スケール変換後の上限」の値内に制限されます (1, 0)。
- 「**エンジニアリング単位**」: 単位文字列。
- 「**説明**」: タグの説明。
- 「**値を負数化**」: スケール変換が適用されるときに、変換後の値がクライアントに渡される前に負数化されます (1, 0)。

● **注記:** 各タグレコードに各フィールドのデータが含まれています。

サーバーの外部で大規模なタグのグループを編集するには、Microsoft Excel が非常に便利です。テンプレート CSV ファイルのエクスポートが完了すると、それを Excel に直接ロードして編集できるようになります。次の図は、Excel にロードされた CSV ファイルを示しています。



サーバーへのサーバータグリストのインポート

タグリストの編集が完了すると、それをサーバーに再インポートできるようになります。これを行うには、「ファイル」|「CSVをインポート」の順にクリックします。このオプションは、デバイスまたはタググループが選択されているときにのみ使用できます。

ほかの区切り記号文字の使用

コンマまたはセミコロン以外の区切り記号を使用している CSV ファイルを使用する場合は、以下のいずれかを行う必要があります。

- プロジェクトを XML で保存してから、CSV を使用する代わりに、XML ファイルで全体構成を行う。
- CSV ファイルで区切り記号に対してサーチと置換を実行して、区切り記号をコンマまたはセミコロンに置き換えます。OPC サーバーで使用されている区切り記号 (コンマまたはセミコロン) を置換後の文字として設定する必要があります。

● 関連項目: [オプション - 一般](#)

システムタグ

システムタグは、クライアントアプリケーションに一般的なエラーのフィードバックを提供して、デバイスがアクティブにデータを収集するときの動作制御を可能にし、必要に応じてチャネルまたはデバイスの標準プロパティを OPC クライアントアプリケーションによって変更できるようにします。

チャネルレベルとデバイスレベルの両方で使用可能なシステムタグの数は、使用されるドライバーの性質によって異なります。さらに、アプリケーションレベルのシステムタグにより、クライアントアプリケーションがサーバーのステータスを監視できるようになります。システムタグは目的に応じて、ステータスと制御またはプロパティの操作としてグループ化することもできます。以下でこれについて説明します。

- ステータスタグ** ステータスタグは、サーバー操作に対するデータを提供する読み取り専用のタグです。
- パラメータ制御タグ** パラメータ制御タグは、サーバーアプリケーションの動作特性を修正するために使用できます。これにより、OPC アプリケーションの柔軟性が大幅に向上します。プロパティ制御タグを使用して、ユーザーは、通信リンクの切り替えまたはターゲットデバイスのデバイス ID の変更による冗長性を実装できます。特別な監視画面からタグにアクセスできるようにすることもできます。これにより、プラントエンジニアが必要に応じてサーバーの通信パラメータを変更できるようになります。

以下の表では、次の内容について説明しています。

[アプリケーションレベルのシステムタグ](#)

[シリアルポートドライバー用のチャネルレベルのシステムタグ](#)

イーサネットドライバ用 のチャンネルレベルのシステムタグ

シリアルおよびイーサネットドライバ用 のデバイスレベルのシステムタグ

アプリケーションレベルのシステムタグ

構文の例: <チャンネル名>.<デバイス名>._System._ActiveTagCount

タグ	クラス	説明
_ActiveTagCount	ステータスタグ	_ActiveTagCount タグは、サーバー内で現在アクティブなタグの数を示します。 これは読み取り専用のタグです。
_ClientCount	ステータスタグ	_ClientCount タグは、現在サーバーに接続しているクライアントの数を示します。 これは読み取り専用のタグです。
_Date	ステータスタグ	_Date タグは、サーバーが実行されているシステムの現在の日付を示します。この文字列のフォーマットは、オペレーティングシステムの日付と時刻の設定によって定義されます。 これは読み取り専用のタグです。
_DateTime	ステータスタグ	_DateTime タグは、サーバーが実行されているシステムの GMT 日付と時刻を示します。文字列のフォーマットは '2004-05-21T20:39:07.000' です。 これは読み取り専用のタグです。
_DateTimeLocal	ステータスタグ	_DateTimeLocal タグは、サーバーが実行されているシステムのローカライズされた日付と時刻を示します。文字列のフォーマットは '2004-05-21T16:39:07.000' です。 これは読み取り専用のタグです。
_Date_Day	ステータスタグ	_Date_Day タグは、サーバーが実行されているシステムの現在の日付を示します。 これは読み取り専用のタグです。
_Date_DayOfWeek	ステータスタグ	_Date_DayOfWeek タグは、サーバーが実行されているシステムの現在の曜日を示します。文字列のフォーマットは、0 (日曜日) ~ 6 (土曜日) の数字です。 これは読み取り専用のタグです。
_Date_Month	ステータスタグ	_Date_Month タグは、サーバーが実行されているシステムの現在の月を示します。文字列のフォーマットは数字 ("September" ではなく "9") です。 これは読み取り専用のタグです。
_Date_Year2	ステータスタグ	_Date_Year2 タグは、サーバーが実行されているシステムの現在の年の後ろ 2 桁を示します。 これは読み取り専用のタグです。
_Date_Year4	ステータスタグ	_Date_Year4 タグは、サーバーが実行されているシステムの現在の年を示します。 これは読み取り専用のタグです。
_ExpiredFeatures	ステータスタグ	_ExpiredFeatures タグは、使用時間が制限されているサーバー機能のうち期限切れになったものすべてのリストを提供します。これらの機能は動作していません。 これは読み取り専用のタグです。
_FullProjectName	ステータスタグ	_FullProjectName タグは、現在ロードされているプロジェクトへの完全修飾パスとファイル名を示します。

タグ	クラス	説明
	タグ	これは読み取り専用のタグです。
_IsDemo	ステータスタグ	バージョン 6.0 以上ではランタイムは時間制限モードに切り替わらないので、_IsDemo タグは使用できなくなりました。サーバー機能のステータスをモニターするには、_TimeLimitedFeatures、_LicensedFeatures、および _ExpiredFeatures の各タグを参照してください。
_LicensedFeatures	ステータスタグ	_LicensedFeatures タグは、有効なライセンスを持つ使用中のすべてのサーバー機能のリストを提供します。これらの機能に時間制限はなく、時間制限のある機能が期限切れになった後も通常の動作を続けます。 これは読み取り専用のタグです。
_OpcClientNames	ステータスタグ	_OpcClientNames タグは、サーバーに接続して IOPCCommon::SetClientName メソッドによって名前を登録しているすべての OPC クライアントの名前をリストする文字列配列です。 これは読み取り専用のタグです。
_ProductName	ステータスタグ	_ProductName タグは、基盤となっている通信サーバーの名前を示します。 これは読み取り専用のタグです。
_ProductVersion	ステータスタグ	_ProductVersion タグは、基盤となっている通信サーバーのバージョンを示します。 これは読み取り専用のタグです。
_ProjectName	ステータスタグ	_ProjectName タグは、現在ロードされているプロジェクトファイル名を示します。ただしパスの情報は含まれません。 これは読み取り専用のタグです。
_ProjectTitle	ステータスタグ	_ProjectTitle タグは、現在ロードされているプロジェクトのタイトルを示す文字列タグです。 これは読み取り専用のタグです。
_Time	ステータスタグ	_Time タグは、サーバーが実行されているシステムの現在の時刻を示します。この文字列のフォーマットは、オペレーティングシステムの日付と時刻の設定によって定義されます。 これは読み取り専用のタグです。
_Time_Hour	ステータスタグ	_Time_Hour タグは、サーバーが実行されているシステムの現在の時刻の時間 (hour) を示します。 これは読み取り専用のタグです。
_Time_Hour24	ステータスタグ	_Time_Hour24 タグは、サーバーが実行されているシステムの現在の時刻の時間 (hour) を 24 時間形式で示します。 これは読み取り専用のタグです。
_Time_Minute	ステータスタグ	_Time_Minute タグは、サーバーが実行されているシステムの現在の時刻の分 (minute) を示します。 これは読み取り専用のタグです。
_Time_PM	ステータスタグ	_Time_PM タグは、サーバーが実行されているシステムの現在の午前/午後のステータスを示します。これは Boolean のタグで、0 (False) は AM を示し、1 (True) は PM を示します。 これは読み取り専用のタグです。
_Time_Second	ス	_Time_Second タグは、サーバーが実行されているシステムの現在の時刻の秒

タグ	クラス	説明
	タス タグ	(second) を示します。 これは読み取り専用のタグです。
_TimeLimitedFeatures	ス テ ー タ ス タ グ	_TimeLimitedFeatures タグは、時間制限があるすべてのサーバー機能のリストと残り時間 (秒数) を提供します。残り時間が経過すると、その機能は動作を停止します。 これは読み取り専用のタグです。
_TotalTagCount	ス テ ー タ ス タ グ	_TotalTagCount タグは、現在アクセスされているタグの合計数を示します。これらのタグは、アクティブまたは非アクティブにできます。 ● 注記: この数はプロジェクトで設定されているタグの数を表しているわけではありません。 これは読み取り専用のタグです。

シリアルポートドライバ用のチャンネルレベルのシステムタグ

構文の例: <チャンネル名>._System._BaudRate

タグ	クラス	説明
_AvailableNetworkAdapters	ス テ ー タ ス タ グ	_AvailableNetworkAdapters タグは、使用可能な NIC を一覧しており、ここには一意の NIC カードおよび複数の IP が割り当てられている NIC の両方が含まれます。さらにこのタグには、ダイヤルアップ接続など、アクティブな WAN 接続も表示されます。このタグは文字列タグとして提供され、この PC で使用可能なネットワークアダプタを決定するために使用できます。返される文字列には、すべての NIC 名とそこに割り当てられた IP が含まれます。一意の NIC はそれぞれセミコロンで区切られ、OPC アプリケーション内で名前を解析できるようになっています。シリアルドライバでは、このタグは「イーサネットカプセル化」を選択した場合のみ使用されます。 これは読み取り専用のタグです。
_BaudRate	パラ メ ー タ 制 御 タ グ	_BaudRate タグにより、ドライバのボーレートを任意に変更できます。_BaudRate タグは Long 値として定義されているので、新しいボーレートはこのフォーマットで書き込む必要があります。有効なボーレートは、300、600 1200、2400、4800、9600、19200、38400、56000、56700、115200、128000、256000 です。 これは、読み取り/書き込みタグです。
_ComId	パラ メ ー タ 制 御 タ グ	_ComId タグにより、ドライバを任意に変更するための通信ポートを選択できます。文字列タグなので、目的の通信ポートは文字列値としてタグに書き込む必要があります。この場合に使用できる選択肢は、COM 1、COM 2、COM 3、COM 4、---、COM 16 および「イーサネットカプセル化」です。「イーサネットカプセル化」モードを選択した場合は、ユーザーがリモートターミナルサーバーの IP 番号も設定する必要があります。このタグはデバイスレベルで実行し、方法は後述します。 これは、読み取り/書き込みタグです。
_DataBits	パラ メ ー タ 制 御 タ グ	_DataBits タグにより、ドライバのデータビットを任意に変更できます。_DataBits タグは、符号付きの 8 ビット値として定義されます。データビットの有効な選択肢は、5、6、7、8 です。 これは、読み取り/書き込みタグです。

タグ	クラス	説明
_Description	ステータスタグ	<p>_Description タグは、参照するチャンネルの現在のユーザー定義のテキスト説明を示します。</p> <p>これは読み取り専用のタグです。</p>
_EnableDiagnostics	パラメータ制御タグ	<p>_EnableDiagnostics タグにより、ドライバーの診断システムを有効/無効にできます。診断システムを有効にすると、ドライバーにかかる負荷が少し追加されます。このためサーバーでは、ドライバーのパフォーマンスを向上させるために、診断を有効または無効にできます。無効にすると、診断タグは使用できません。詳細については、統計タグを参照してください。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationPort	パラメータ制御タグ	<p>_EncapsulationPort タグは、イーサネット接続の宛先ポートを制御します。有効な範囲は 0 から 65535 です。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationProtocol	パラメータ制御タグ	<p>_EncapsulationProtocol タグは、イーサネット接続に使用されるプロトコルを制御します。オプションには TCP/IP と UDP があります。</p> <p>これは、読み取り/書き込みタグです。</p>
_FloatHandlingType	パラメータ制御タグ	<p>_FloatHandlingType タグにより、現在のチャンネルレベルの float 処理を変更できます。このタグは、チャンネルレベルの _System フォルダ内に存在します。詳細については、チャンネルのプロパティ - 詳細を参照してください。</p> <p>これは、読み取り/書き込みタグです。</p>
_FlowControl	パラメータ制御タグ	<p>_FlowControl タグにより、ドライバーのフロー制御の設定を任意に変更できます。文字列タグなので、目的のフロー制御の設定は、このフォーマットでタグに書き込む必要があります。フロー制御には次のような選択肢があります:「なし」、「DTR」、「RTS」、「DTR、RTS」、「RTS 常時」、「RTS 手動」。すべてのドライバーが「RTS 手動」モードの操作をサポートしているわけではありません。</p> <p>これは、読み取り/書き込みタグです。</p>
_InterDeviceDelayMS	パラメータ制御タグ	<p>_InterDeviceDelayMS タグは、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、チャンネルが次のデバイスに要求を送信するまでの遅延時間を指定します。有効な範囲は 0 から 60000 ミリ秒です。デフォルトの設定は 0 です。</p> <p>● 注記: このタグは「デバイス間遅延」を利用するプロトコルを使用するチャンネルでのみ使用できます。</p> <p>これは、読み取り/書き込みタグです。</p>
_NetworkAdapter	パラメータ制御タグ	<p>_NetworkAdapter タグにより、ドライバーが使用している現在の NIC アダプタを任意に変更できます。文字列タグなので、新しく必要とする NIC アダプタの名前は、文字列形式でこのタグに書き込む必要があります。変更を有効にするには、書き込まれる文字列が目的の NIC の説明と正確に一致する必要があります。NIC 名は前述の _AvailableNetworkAdapters タグから入手できます。シリアルドライバーでは、このタグは「イーサネットカプセル化」を選択した場合のみ使用されます。</p>

タグ	クラス	説明
		<p>● 注記: NIC の選択を変更すると、ドライバーが現在のすべてのデバイス接続を強制的に解除し、再接続します。</p> <p>これは、読み取り/書き込みタグです。</p>
_Parity	パラメータ制御タグ	<p>_Parity タグにより、ドライバーのパリティを任意に変更できます。文字列タグなので、目的のパリティ設定は文字列値としてタグに書き込む必要があります。この場合に使用できる選択肢は、「なし」、「奇数」、「偶数」です。</p> <p>これは、読み取り/書き込みタグです。</p>
_ReportComErrors	パラメータ制御タグ	<p>_ReportComErrors タグにより、パリティエラーやフレーミングエラーなどの低レベル通信エラーのレポートを有効/無効にできます。このタグは、Boolean タグとして定義され、True または False に設定できます。True の場合、ドライバーはすべての低レベル通信エラーをサーバーイベントシステムにレポートします。False の場合、ドライバーは低レベル通信エラーを無視し、レポートしません。ただし、エラーが含まれていれば、ドライバーが通信トランザクションを拒否します。電気ノイズが多く含まれる環境の場合、この機能を無効にすることで、イベントログがエラーメッセージで満杯になるのを防ぐことができます。</p> <p>これは、読み取り/書き込みタグです。</p>
_RtsLineDrop	パラメータ制御タグ	<p>_RtsLineDrop タグにより、ドライバーがメッセージの転送を試みた後、ユーザー選択の期間だけ RTS 回線を低くすることができます。このタグは「手動 RTS」モードをサポートするドライバーに対してのみ有効になります。_RtsLineDrop は Long 値として定義されます。有効な範囲は 0 から 9999 ミリ秒です。「手動 RTS」モードはラジオモデムで使用するよう設計されています。</p> <p>これは、読み取り/書き込みタグです。</p>
_RtsLinePollDelay	パラメータ制御タグ	<p>_RtsLinePollDelay タグにより、各メッセージがドライバーから送信された後に、ユーザー設定可能な一時停止を実行できます。このタグは「手動 RTS」モードをサポートするドライバーに対してのみ有効になります。_RtsLinePollDelay は Long 値として定義されます。有効な範囲は 0 から 9999 ミリ秒です。「手動 RTS」モードはラジオモデムで使用するよう設計されています。</p> <p>これは、読み取り/書き込みタグです。</p>
_RtsLineRaise	パラメータ制御タグ	<p>_RtsLineRaise タグにより、ドライバーがメッセージの転送を試みる前に、ユーザー選択の期間だけ RTS 回線を上昇させることができます。このタグは「手動 RTS」モードをサポートするドライバーに対してのみ有効になります。_RtsLineRaise は Long 値として定義されます。有効な範囲は 0 から 9999 ミリ秒です。「手動 RTS」モードはラジオモデムで使用するよう設計されています。</p> <p>これは、読み取り/書き込みタグです。</p>
_SharedConnection	ステータスタグ	<p>_SharedConnection タグは、ポートの設定が別のチャネルと共有されていることを示します。</p> <p>これは読み取り専用のタグです。</p>
_StopBits	パラメータ制御タグ	<p>_StopBits タグにより、ドライバーのストップビットを任意に変更できます。_StopBits タグは、符号付きの 8 ビット値として定義されます。データビットの有効な選択肢は、1 および 2 です。</p> <p>これは、読み取り/書き込みタグです。</p>

タグ	クラス	説明
_UnsolicitedEncapsulationPort	パラメータ制御タグ	<p>_UnsolicitedEncapsulationPort タグは、接続を許可するためにすでに開いているイーサネットポートを制御します。有効な範囲は 0 から 65535 です。</p> <p>これは、読み取り/書き込みタグです。</p>
_UnsolicitedEncapsulationProtocol	パラメータ制御タグ	<p>_UnsolicitedEncapsulationProtocol タグは、非送信請求カプセル化ポートに接続するために使用するイーサネットプロトコルを制御します。オプションには TCP/IP と UDP があります。</p> <p>これは、読み取り/書き込みタグです。</p>
_WriteOptimizationDutyCycle	パラメータ制御タグ	<p>_WriteOptimizationDutyCycle タグにより、読み取り対書き込み比率のデューティサイクルを任意に変更できます。デューティサイクルは、ドライバが 1 回の読み取りに対して実行する書き込みの回数を制御します。_WriteOptimizationDutyCycle は、符号なしの Long 値として定義されます。有効な範囲は、1 回の読み取りあたり 1 ～ 10 回の書き込みです。詳細については、チャンネルのプロパティ - 書き込み最適化を参照してください。</p> <p>これは、読み取り/書き込みタグです。</p>

イーサネットドライバ用のチャンネルレベルのシステムタグ

構文の例: <チャンネル名>._System._NetworkAdapter

タグ	クラス	説明
_AvailableNetworkAdapters	ステータスタグ	<p>_AvailableNetworkAdapters タグは、使用可能な NIC を一覧しており、ここには一意の NIC カードおよび複数の IP が割り当てられている NIC の両方が含まれます。さらにこのタグには、ダイヤルアップ接続など、アクティブな WAN 接続も表示されます。このタグは文字列タグとして提供され、この PC で使用可能なネットワークアダプタを決定するために使用できます。返される文字列には、すべての NIC 名とそこに割り当てられた IP が含まれます。一意の NIC はそれぞれセミコロンで区切られ、OPC アプリケーション内で名前を解析できるようになっています。シリアルドライバでは、このタグは「イーサネットカプセル化」を選択した場合のみ使用されます。</p> <p>これは読み取り専用のタグです。</p>
_Description	ステータスタグ	<p>_Description タグは、参照するチャンネルの現在のユーザー定義のテキスト説明を示します。</p> <p>これは読み取り専用のタグです。</p>
_EnableDiagnostics	パラメータ制御タグ	<p>_EnableDiagnostics タグにより、ドライバの診断システムを有効/無効にできます。診断システムを有効にすると、ドライバにかかる負荷が少し追加されます。このためサーバーでは、ドライバのパフォーマンスを向上させるために、診断を有効または無効にできます。無効にすると、診断タグは使用できません。詳細については、統計タグを参照してください。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationPort	パラメータ制御	<p>_EncapsulationPort タグは、イーサネット接続に使用されるポートを制御します。有効な範囲は 0 から 65535 です。</p> <p>これは、読み取り/書き込みタグです。</p>

タグ	クラス	説明
	御 タグ	
_EncapsulationProtocol プロパティ	パラ メータ 制御 タグ	<p>_EncapsulationProtocol タグは、イーサネット 接続に使用されるプロトコルを制御します。オプションには TCP/IP と UDP があります。</p> <p>これは、読み取り/書き込みタグです。</p>
_FloatHandlingType	パラ メータ 制御 タグ	<p>_FloatHandlingType タグにより、現在のチャンネルレベルの float 処理を変更できます。このタグは、チャンネルレベルの _System フォルダ内に存在します。詳細については、チャンネルのプロパティ - 詳細を参照してください。</p> <p>これは、読み取り/書き込みタグです。</p>
_InterDeviceDelayMS	パラ メータ 制御 タグ	<p>_InterDeviceDelayMS タグは、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、チャンネルが次のデバイスに要求を送信するまでの遅延時間を指定します。有効な範囲は 0 から 60000 ミリ秒です。デフォルトの設定は 0 です。</p> <p>● 注記: このタグは「デバイス間遅延」を利用するプロトコルを使用するチャンネルでのみ使用できます。</p> <p>このタグは、読み取り/書き込みタグです。</p>
_NetworkAdapter	パラ メータ 制御 タグ	<p>_NetworkAdapter タグにより、ドライバーが使用している現在の NIC アダプタを任意に変更できます。文字列タグなので、新しく必要とする NIC アダプタの名前は、文字列形式でこのタグに書き込む必要があります。変更を有効にするには、書き込まれる文字列が目的の NIC の説明と正確に一致する必要があります。NIC 名は前述の _AvailableNetworkAdapters タグから入手できます。シリアルドライバーでは、このタグは「イーサネットカプセル化」を選択した場合のみ使用されます。</p> <p>● 注記: NIC の選択を変更すると、ドライバーが現在のすべてのデバイス接続を強制的に解除し、再接続します。</p> <p>これは、読み取り/書き込みタグです。</p>
_UnsolicitedEncapsulationPort	パラ メータ 制御 タグ	<p>_UnsolicitedEncapsulationPort タグは、接続を許可するためにすでに開いているイーサネットポートを制御します。有効な範囲は 0 から 65535 です。</p> <p>これは、読み取り/書き込みタグです。</p>
_UnsolicitedEncapsulationProtocol	パラ メータ 制御 タグ	<p>_UnsolicitedEncapsulationProtocol タグは、非送信請求カプセル化ポートに接続するために使用するイーサネットプロトコルを制御します。オプションには TCP/IP と UDP があります。</p> <p>これは、読み取り/書き込みタグです。</p>
_WriteOptimizationDutyCycle	パラ メータ 制御 タグ	<p>_WriteOptimizationDutyCycle タグにより、読み取り対書き込み比率のデューティサイクルを任意に変更できます。デューティサイクルは、ドライバーが 1 回の読み取りに対して実行する書き込みの回数を制御します。_WriteOptimizationDutyCycle は、符号なしの Long 値として定義されます。有効な範囲は、1 回の読み取りあたり 1 ~ 10 回の書き込みです。詳細については、チャンネルのプロパティ - 書き込み最適化を参照してください。</p>

タグ	クラス	説明
		これは、読み取り/書き込みタグです。

シリアルおよびイーサネットドライバ用のデバイスレベルのシステムタグ

構文の例: <チャンネル名>.<デバイス名>._System._Error

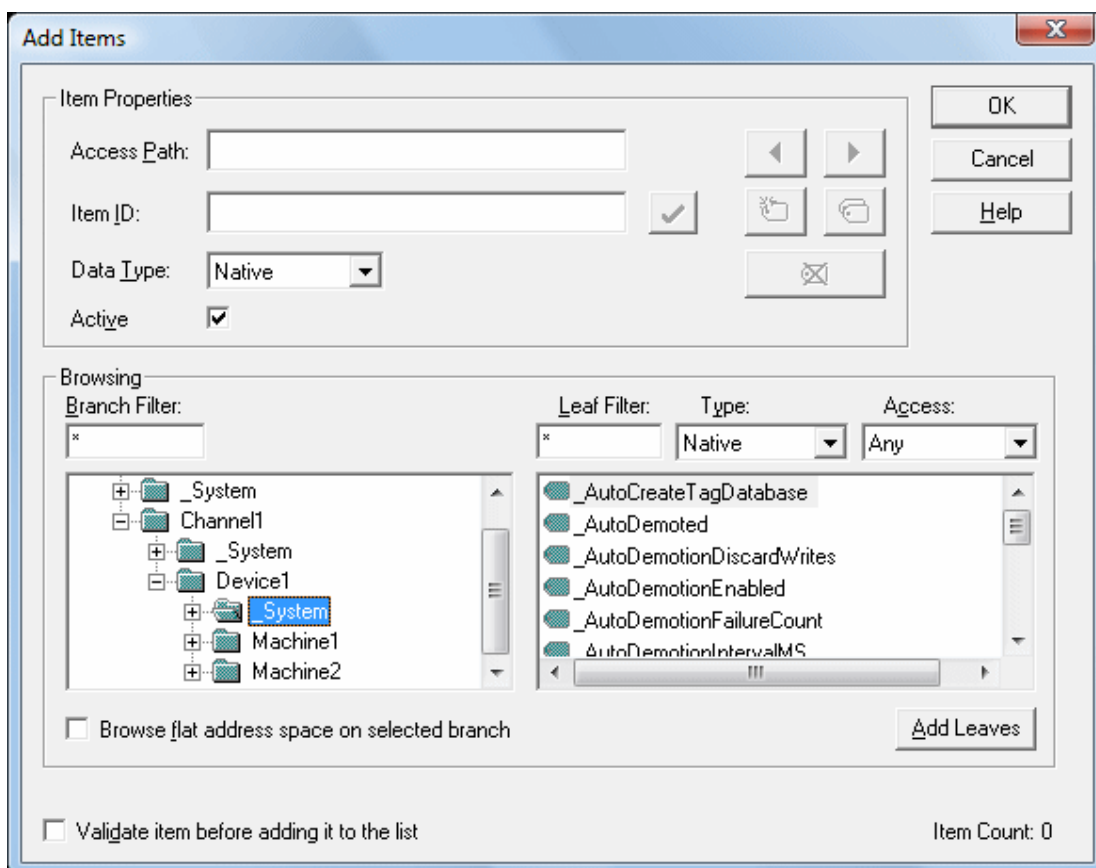
タグ	クラス	説明
_AutoCreateTagDatabase	パラメータ制御タグ	<p>_AutoCreateTagDatabase タグは、このタグの接続先のデバイスに対してこのドライバの自動 OPC タグデータベース関数を開始するために使用する Boolean タグです。このタグを True に設定すると、通信ドライバはこのデバイスに対して OPC タグデータベースを自動生成しようとします。このタグは、自動 OPC タグデータベース生成をサポートしないドライバは表示されません。</p> <p>これは、読み取り/書き込みタグです。</p>
_AutoDemoted	ステータスタグ	<p>_AutoDemoted タグは、デバイスの現在の自動格下げ状態を返す Boolean タグです。False の場合、デバイスは格下げされず、ドライバによってスキャンされます。True に設定すると、デバイスは格下げ状態になり、ドライバによってスキャンされません。</p> <p>これは読み取り専用のタグです。</p>
_AutoDemotionDiscardWrites	パラメータ制御タグ	<p>_AutoDemotionDiscardWrites タグは、格下げ期間の書き込み要求を破棄するかどうかを指定する Boolean タグです。このタグが False に設定されている場合、_AutoDemoted の状態に関係なく、すべての書き込み要求が実行されます。このタグが True に設定されている場合、格下げ期間の書き込み要求はすべて破棄されます。</p> <p>これは、読み取り/書き込みタグです。</p>
_AutoDemotionEnabled	パラメータ制御タグ	<p>_AutoDemotionEnabled タグは、デバイスが応答しない場合に、デバイスを特定の期間だけ自動的に格下げできるようにする Boolean タグです。このタグが False に設定されている場合、デバイスは格下げされません。このタグが True に設定されている場合、_AutoDemotedFailureCount に到達したときに、デバイスが格下げされます。</p> <p>これは、読み取り/書き込みタグです。</p>
_AutoDemotedFailureCount	パラメータ制御タグ	<p>_AutoDemotedFailureCount タグは、デバイスの格下げまでに連続して発生する障害の数を指定します。_AutoDemotedFailureCount は、Long データとして定義されます。有効な範囲は 1 から 30 です。このタグに書き込みができるのは、_AutoDemotionEnabled が True に設定されている場合のみです。</p> <p>これは、読み取り/書き込みタグです。</p>
_AutoDemotionIntervalMS	パラメータ制御タグ	<p>_AutoDemotionIntervalMS タグは、デバイスとの通信を再試行するまでにデバイスが格下げされている期間を、ミリ秒単位で指定します。_AutoDemotionIntervalMS は、Long データとして定義されます。有効な範囲は 100 から 3600000 ミリ秒です。このタグに書き込みができるのは、_AutoDemotionEnabled が True に設定されている場合のみです。</p> <p>これは、読み取り/書き込みタグです。</p>
_ConnectTimeout	パラメータ制御タグ	<p>_ConnectTimeout タグにより、デバイスへの IP 接続の確立に関連付けられているタイムアウトを任意に変更できます。このタグが使用できるのは、ネイティブイーサネットドライバを使用中の場合か、シリアルドライバが「イーサネットカプセル化」モードである場合です。_ConnectTimeout は、Long データとして定義されます。有効な範囲は 1 から 30 秒です。</p> <p>これは、読み取り/書き込みタグです。</p>

タグ	クラス	説明
_DemandPoll	ステータス制御タグ	<p>_DemandPoll タグは、デバイスに関連付けられているすべてのアクティブなクライアントアイテムへのデバイス読み取りを発行します。これは、これらのアイテムに対する非同期のデバイス読み取りを実行するクライアントと同等です。アクティブにスキャンされているアイテムに対して発生するスケジュールされた読み取りよりも優先されます。</p> <p>書き込みが行われると、_DemandPoll タグは True (1) になります。これは、読み取り要求が完了したことを最後のアクティブなタグが示すと False (0) を返します。_DemandPoll タグへのそれ以降の書き込みは、タグの値が False に戻るまで失敗します。要求ポールは、チャンネルの読み取り/書き込み使用率を考慮します。</p> <p>これは、読み取り/書き込みタグです。</p>
_Description	ステータスタグ	<p>_Description タグは、参照するデバイスの現在のユーザー定義のテキスト説明を示します。</p> <p>これは読み取り専用のタグです。</p>
_Deviceld	パラメータ制御タグ	<p>_Deviceld タグにより、デバイスの ID を任意に変更できます。_Deviceld のデータ形式は、デバイスのタイプによって異なります。ほとんどのシリアルデバイスでは、このタグは Long データです。イーサネットドライバーでは、_Deviceld は文字列タグとしてフォーマットされ、IP アドレスを入力できるようになっています。どちらの場合も、このタグに新規のデバイス ID を書き込むと、ドライバーがターゲットフィールドのデバイスを変更します。このことは、このタグに書き込まれたデバイス ID が正しくフォーマットされており、指定されたドライバーの有効な範囲内にある場合のみ発生します。</p> <p>これは、読み取り/書き込みタグです。</p>
_Enabled	パラメータ制御タグ	<p>_Enabled タグは、デバイスのアクティブ状態をオン/オフにできる Boolean タグです。このタグを False に設定すると、このデバイスのその他すべてのユーザー定義のタグとデータが無効にマークされ、デバイスに対する書き込みは受け入れられません。このタグを True に設定すると、デバイスによる正常な通信が実行されます。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationIp	パラメータ制御タグ	<p>_EncapsulationIp タグにより、リモートターミナルサーバーの IP を任意に指定および変更できます。このタグは、デバイスのプロパティ - イーサネットカプセル化 モードをサポートするシリアルドライバーでのみ使用できます。_EncapsulationIp は文字列データタイプとして定義されており、IP アドレス番号を入力できます。サーバーは、無効な IP アドレスのエントリを却下します。このタグは「イーサネットカプセル化」モードのシリアルドライバーでのみ有効です。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationPort	パラメータ制御タグ	<p>_EncapsulationPort タグにより、リモートターミナルサーバーのポート番号を指定および変更できます。_EncapsulationPort は、Long データとして定義されます。有効な範囲は 0 から 65535 です。イーサネットカプセル化を正しく実行するためには、このタグに入力するポート番号が、目的のリモートターミナルサーバーと一致する必要があります。このタグは「イーサネットカプセル化」モードのシリアルドライバーでのみ有効です。</p> <p>これは、読み取り/書き込みタグです。</p>
_EncapsulationProtocol	パラメータ制御タグ	<p>_EncapsulationProtocol タグにより、イーサネットカプセル化に使用する IP プロトコルを指定および変更できます。_EncapsulationProtocol は文字列データタイプとして定義されます。タグに「TCP/IP」または「UDP」と書き込むことで IP プロトコルを指定します。イーサネットカプセル化を正しく実行するためには、使用されるプロトコルがリモートターミナルサーバーのプロトコルと一致す</p>

タグ	クラス	説明
		<p>する必要があります。このタグは「イーサネットカプセル化」モードのシリアルドライバーでのみ有効です。</p> <p>これは、読み取り/書き込みタグです。</p>
_Error	ステータスタグ	<p>_Error タグは、デバイスの現在のエラー状態を返す Boolean タグです。False の場合、デバイスは適切に動作しています。True の場合、このデバイスとの通信時に、ドライバーがエラーを検出しています。デバイスは、要求タイムアウトのサイクルを完了し、応答なしで再試行した場合に、エラー状態に入ります。</p> <p>● 注記: 詳細については、デバイスのプロパティ - タイミングを参照してください。</p> <p>これは読み取り専用のタグです。</p>
_FailedConnection	ステータスタグ	<p>_FailedConnection タグは、接続の失敗を指定します。特定のドライバーに対してのみ使用できます。</p> <p>これは読み取り専用のタグです。</p>
_InterRequestDelay	パラメータ制御タグ	<p>_InterRequestDelay タグにより、デバイスのトランザクションの時間間隔を任意に変更できます。_InterRequestDelay は、Long データとして定義されます。有効な範囲は 0 から 30000 ミリ秒です。このタグは、この機能をサポートするドライバーにのみ適用されます。</p> <p>これは、読み取り/書き込みタグです。</p>
_RequestAttempts	パラメータ制御タグ	<p>_RequestAttempts タグにより、通信試行回数を任意に変更できます。_RequestAttempts は Long 値として定義されます。有効な範囲は 1 から 10 回です。このタグは、すべてのドライバーに均等に適用されます。</p> <p>これは、読み取り/書き込みタグです。</p>
_RequestTimeout	パラメータ制御タグ	<p>_RequestTimeout タグにより、データ要求に関連付けられているタイムアウトを任意に変更できます。_RequestTimeout タグは Long 値として定義されます。有効な範囲は 100 から 30000 ミリ秒です。このタグは、すべてのドライバーに均等に適用されます。</p> <p>これは、読み取り/書き込みタグです。</p>
_NoError	ステータスタグ	<p>_NoError タグは、デバイスの現在のエラー状態を返す Boolean タグです。True の場合、デバイスは適切に動作しています。False の場合、このデバイスとの通信時に、ドライバーがエラーを検出しています。デバイスは、要求タイムアウトのサイクルを完了し、応答なしで再試行した場合に、エラー状態に入ります。</p> <p>● 注記: 詳細については、デバイスのプロパティ - タイミングを参照してください。</p> <p>これは読み取り専用のタグです。</p>
_ScanMode	ステータスタグ	<p>_ScanMode タグにより、更新に使用する方法をクライアントが指示できます。このタグは文字列の値として定義され、(デバイスのプロパティにある) ユーザー指定の「スキャンモード」の設定に対応しています。「クライアント固有のスキャン速度を適用」には値「UseClientRate」、「指定した速度 x 以下でデータを要求」には値「UseFloorRate」、「すべてのデータを指定した速度 x で要求」には値「ForceAllToFloorRate」が対応します。デフォルト設定は「クライアント固有のスキャン速度を適用」です。</p> <p>これは読み取り専用のタグです。</p>
_ScanRateMs	ステータスタグ	<p>_ScanRateMs タグは _ScanMode タグに対応しており、スキャンモードが「指</p>

タグ	クラス	説明
	タスタグ	<p>定したスキャン速度以下でデータを要求」または「すべてのデータを指定したスキャン速度で要求」に設定されている場合に使用されます。このタグは、DWord タグとして定義されます。デフォルトの設定は 1000 ミリ秒です。</p> <p>これは読み取り専用のタグです。</p>
_SecondsInError	ステータスタグ	<p>_SecondsInError タグは、デバイスがエラー状態に入ってから秒数を表示する DWord タグです。デバイスがエラー状態にない場合、このタグは 0 を表示します。</p> <p>これは読み取り専用のタグです。</p>
_Simulated	パラメータ制御タグ	<p>_Simulated タグは、現在のデバイスのシミュレーション状態に関するフィードバックを提供する Boolean タグです。True の場合、このデバイスはシミュレーションモードです。シミュレーションモードでは、サーバーはこのデバイスの適切なデータを返しますが、実際の物理的なデバイスとは通信しません。このタグが False の場合、物理的なデバイスとの通信がアクティブになっています。タグの値を変更すると、クライアントはシミュレーションモードを有効/無効にできます。</p> <p>これは、読み取り/書き込みタグです。</p>

OPC クライアントを使用する場合、システムタグは、特定のデバイスのサーバーブラウズ領域の _System ブランチの下にあります。次の図は、付属の OPC Quick Client から取得したもので、OPC クライアントでのシステムタグの表示方法を示しています。



DeviceName ブランチの下にある _System ブランチは、常に使用できます。上記の例で DDE がデフォルト設定されており、DDE アプリケーションからシステムタグを参照している場合、リンクは "<DDE service name>[_ddedata!Channel1.Device1._System._Error]" と表示されます。

_Enabled タグは、OPC アプリケーションを非常に柔軟に制御する手段を提供します。場合によっては (特にモデムアプリケーションの場合)、現在モデムに接続されているデバイス以外のすべてのデバイスを無効にすると便利です。さらに、物理デバイスのサービス実行中に _Enable タグを使用してアプリケーションで特定のデバイスをオフにできます。これによって無害に排除できますが、サーバーのイベントログには不要な通信エラーが記録されます。

● 関連項目:

[プロパティタグ](#)

[モデムタグ](#)

[統計タグ](#)

[ストアアンドフォワードのタグ](#)

プロパティタグ

プロパティタグは、クライアントアプリケーションのタグプロパティに対して、読み取り専用でアクセスするために使用します。タグプロパティにアクセスするには、サーバーのタグデータベースに定義されている完全修飾タグアドレスに、プロパティの名前を付加します。詳細については、[タグのプロパティ - 一般](#)を参照してください。

完全修飾タグアドレスが "Channel1.Device1.Tag1" である場合、説明にアクセスするには、説明のプロパティを "Channel1.Device1.Tag1._Description" という形で付加します。

サポートされているプロパティタグ名

タグ名	説明
_Name	参照しようとしているタグの現在の名前。
_Address	参照しようとしているタグの現在のアドレス。
_Description	参照しようとしているタグの現在の説明。
_RawDataType	参照しようとしているタグの生データ型。
_ScalingType	参照しようとしているタグのスケール変換のタイプ (「なし」、「線形」、または「平方根」)。
_ScalingRawLow	参照しようとしているタグの生の下限。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingRawHigh	参照しようとしているタグの生の上限。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingScaledDataType	参照しようとしているタグの、スケール変換先のデータ型。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingScaledLow	参照しようとしているタグの、スケール変換後の下限。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingScaledHigh	参照しようとしているタグの、スケール変換後の上限。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingClampLow	参照しようとしているタグの、スケール変換後の下限値をクランプするかどうか。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingClampHigh	参照しようとしているタグの、スケール変換後の上限値をクランプするかどうか。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。
_ScalingUnits	参照しようとしているタグの、スケール変換の単位。スケール変換が「なし」に設定されている場合、この値には、スケール変換が適用された場合のデフォルト値が保持されます。

● 関連項目:

[統計タグ](#)

[モデムタグ](#)

[システムタグ](#)

統計タグ

統計タグを使用して、クライアントアプリケーションに、サーバー内のチャンネル通信の操作に関するフィードバックを提供します。統計タグは診断が有効な場合のみ使用できます。詳細については、[チャンネル診断](#)および[OPC 診断のビューア](#)を参照してください。

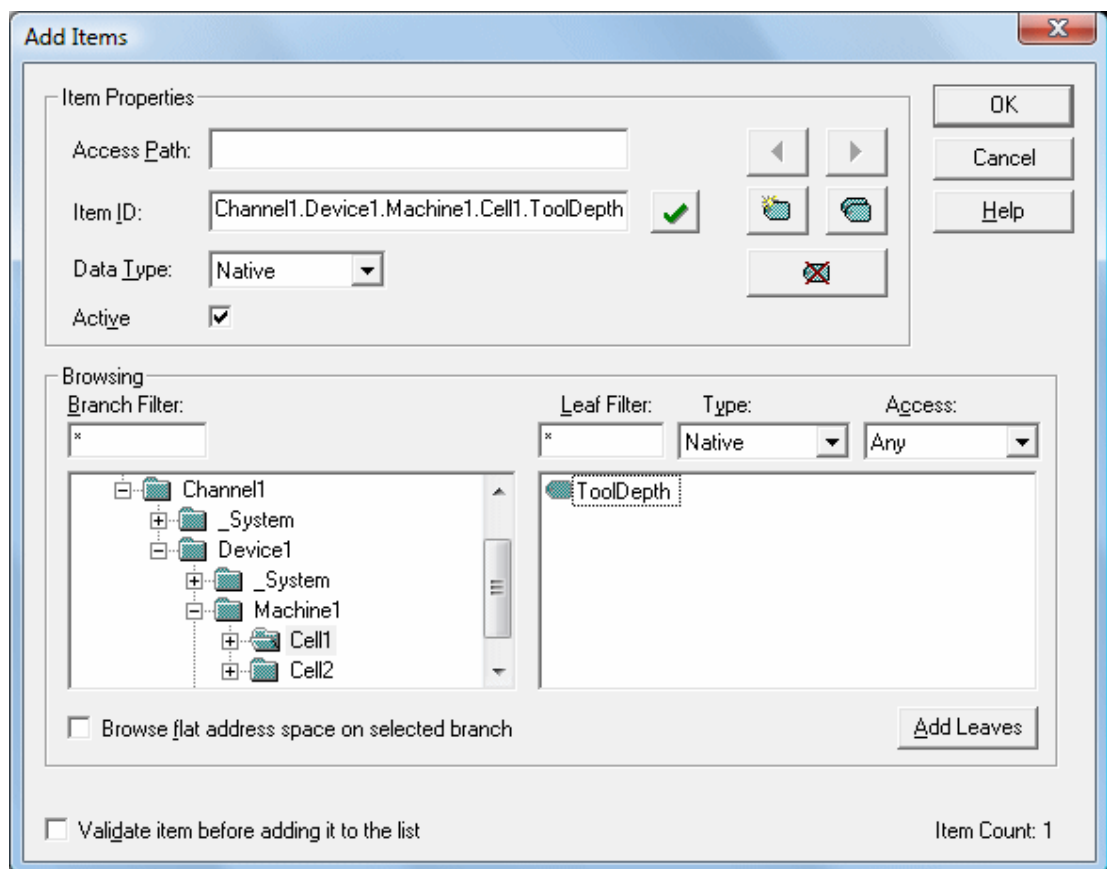
構文の例: <チャンネル名>._Statistics._FailedReads

サポートされている統計タグ名

タグ名	説明
_SuccessfulReads	_SuccessfulReads タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが正常に完了した読み取りの回数が含まれます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_SuccessfulWrites	_SuccessfulWrites タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが正常に完了した書き込みの回数が含まれます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_FailedReads	_FailedReads タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが完了できなかった読み取りの回数が含まれます。この回数は、デバイスに対して設定されたタイムアウトと再試行回数に基づいて、チャンネルが要求に失敗した後にのみ増分されます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_FailedWrites	_FailedWrites タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが完了できなかった書き込みの回数が含まれます。この回数は、デバイスに対して設定されたタイムアウトと再試行回数に基づいて、チャンネルが要求に失敗した後にのみ増分されます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_RxBytes*	_RxBytes タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが接続先のデバイスから受信したバイト数が含まれます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_TxBytes	_TxBytes タグには、アプリケーションの開始以降、または _Reset タグの最後の呼び出し以降、このチャンネルが接続先のデバイスに送信したバイト数が含まれます。このタグは符号なしの 32 ビット整数としてフォーマットされ、最終的にはロールオーバーします。このタグは読み取り専用です。
_Reset	_Reset タグを使用して、すべての診断カウンタをリセットできます。_Reset タグは、Boolean タグとしてフォーマットされます。_Reset タグにゼロ以外の値を書き込むと、診断カウンタがリセットされます。このタグは読み取り/書き込み可能です。
_MaxPendingReads	_MaxPendingReads タグには、アプリケーションの開始 (または _Reset タグの呼び出し) 以降の、チャンネルの保留中の読み取り要求の最大数が含まれます。このタグは、符号なしの 32 ビット整数としてフォーマットされます。このタグは読み取り専用です。
_MaxPendingWrites	_MaxPendingWrites タグには、アプリケーションの開始 (または _Reset タグの呼び出し) 以降の、チャンネルの保留中の書き込み要求の最大数が含まれます。このタグは、符号なしの 32 ビット整数としてフォーマットされます。このタグは読み取り専用です。
_NextReadPriority	_NextReadPriority は、チャンネルの読み取り保留中キューに次の読み取りの優先度を反映する、チャンネルレベルのシステムタグです。使用可能な値は次のとおりです: 「-1」 - 読み取り保留中にはありません。「0」 - 次の読み取りは、スケジュールレベルの要求ポールの結果、またはクライアントからの明示的な読み取りの結果になります。「1 - n」 - 次の読み取りは、スケジュールされた読み取りの結果になります。このタグは読み取り専用です。
_PendingReads	_PendingReads タグには、チャンネルの現在保留中の読み取り要求の数が含まれます。このタグは、符号なしの 32 ビット整数としてフォーマットされます。このタグは読み取り専用です。
_PendingWrites	_PendingWrites タグには、チャンネルの現在保留中の書き込み要求の数が含まれます。このタグは、符号なしの 32 ビット整数としてフォーマットされます。このタグは読み取り専用です。

* この統計アイテムはシミュレーションモードでは更新されません ([デバイスのプロパティ](#)を参照)。

統計タグは診断が有効な場合のみ使用できます。OPC クライアントからアクセスするには、指定したチャンネルのサーバーブラウズ領域の `_Statistics` ブランチから診断タグをブラウズできます。次の図は OPC Quick Client から取得されたもので、OPC クライアントでの診断タグの表示方法を示しています。



(チャンネルブランチの下にある) `_Statistics` ブランチは、チャンネルに対する診断が有効な場合にのみ表示されます。上記の例で DDE がデフォルト設定されている場合、DDE アプリケーションから診断タグを参照するときのリンクは次のようになります: `"<DDE サービス名>|_ddedata!Channel1._Statistics._SuccessfulReads"`。

診断タグの値は、通信診断ビューアを使用してサーバーでも表示できます。「チャンネルのプロパティ」で「診断取り込み」が有効になっている場合、そのチャンネルを右クリックして「診断」を選択します。

● 関連項目:
[システムタグ](#)
[プロパティタグ](#)

モデムタグ

モデムの使用が選択されると、以下のタグがチャンネルに対して自動的に作成されます。

構文の例: `<チャンネル名>.<デバイス名>._Modem._Dial`

サポートされているモデムタグ名

タグ名	説明	アクセス
<code>_Dial</code>	このタグに値を書き込むと、現在の電話番号のダイヤルが開始されます。この書き込みは、現在のステータスが 3 (アイドル状態) ではないかぎり無視されます。現在の電話番号が初期化されていない場合は、エラーが報告されます。モードタグが 2 (着信のみ) に設定されているときにダイヤルコマンドを発行しようとすると、エラーが生成されます。	読み取り

タグ名	説明	アクセス
		書き込み
_DialNumber	_DialNumber タグは、ダイヤルプリファレンスの翻訳 (市外局番の追加など) が適用された後に、実際にダイヤルされる電話番号を示します。このタグは、デバッグに使用されることを目的としています。これは、電話番号が手動で入力される場合に、オペレータに有益なフィードバックを提供します。	読み取り専用
_Hangup	このタグに値を書き込むと、現在の接続がハングアップします。_Hangup タグは、外部デバイスがサーバーを呼び出すと、現在の接続を終了します。_Hangup タグへの書き込みは、ステータスが 3 (アイドル状態) 以下であれば (つまり、現在開かれている接続がなければ) 無視されます。	読み取り/書き込み
_LastEvent	ステータスが変更されるたびに、変更の理由がこのタグで番号として設定されます。イベント番号のリストと意味については、 最終イベント値 を参照してください。	読み取り専用
_Mode	これにより、発信用のみ、応答用のみ、またはその両方の回線を構成できます。 _Mode タグに 1 と書き込むと、回線は発信用のみに設定され、このモードのときには着信に応答しません。_Mode タグに 2 と書き込むと、回線は着信用のみ設定され、発信ダイヤルの要求 (ダイヤルタグへの書き込み) は無視されます。デフォルト設定は 0 であり、これは発信用と着信用の両方に使用できることを意味します。 この値は、ステータスが 3 (アイドル状態) 以下である場合にのみ変更できます。	読み取り/書き込み
_PhoneNumber	これは、ダイヤルする現在の電話番号です。この値への書き込みはいつでもできますが、変更はステータスが 3 (アイドル状態) 以下である場合にのみ有効です。ステータスが 3 より大きいときに電話番号に書き込むと、その番号はキューに入ります。ステータスが 3 以下になるとすぐに新しい番号がタグに転送されます。キューのサイズは 1 であるため、最後に書き込まれた電話番号のみが保持されます。 ダイヤルプリファレンスを適用するには、電話番号が正準形式である必要があります。正準形式が使用される場合、ダイヤルされる番号を _DialNumber として表示できます (ダイヤルプリファレンスが適用された後に)。 正準形式は次のとおりです。 +<国番号>[スペース]<市外局番>[スペース]<電話番号> 例: +1 (207) 846-5881 ● 注記: アメリカ合衆国の国番号は 1 です。 番号が正準形式でなければ、ダイヤルプリファレンスは適用されません。番号は入力されたとおりにダイヤルされます。電話番号の代わりに電話帳タグ名を入力することもできます。この場合、電話帳タグの現在の値が使用されます。	読み取り/書き込み
_Status	これは、チャンネルに割り当てられているモデムの現在のステータスです。ステータス値のリストと意味については、 ステータス値 を参照してください。	読み

タグ名	説明	アクセス
		取り専用
_StringLastEvent	これには、最終イベントタグの値のテキスト表現が含まれています。イベント番号のリストと意味については、 最終イベント文字列値 を参照してください。	読み取り専用
_StringStatus	これには、ステータスタグの値のテキスト表現が含まれています。イベント番号のリストと意味については、 ステータス文字列値 を参照してください。	読み取り専用

ステータス値

現時点では、32 ビットのステータス変数の最下位の 5 ビットが使用されています。

ビット	意味
0	TAPI で初期化されている
1	回線は開かれている
2	接続されている
3	呼び出している
4	応答している

整数として読み込まれる場合、ステータスタグの値は常に次のいずれかになります。

値	意味
0	初期化されていない、チャネルは使用できない
1	初期化されている、開かれている回線はない
3	回線は開かれていて、アイドル状態である
7	接続されている
11	呼び出している
19	応答している

ステータス文字列値

ステータス値	ステータス文字列テキスト
0	初期化されていない、チャネルは使用できない
1	初期化されている、開かれている回線はない
3	アイドル
7	接続されている
11	呼び出している
19	応答している

最終イベント値

最終イベント	変更理由
-1	<空白> [まだイベントは発生していない]
0	TAPI で初期化されている
1	回線が閉じた
2	回線が開かれた
3	回線が接続された
4	ユーザーによって回線が切断された
5	リモートサイトで回線が切断された
6	応答がない
7	回線がビジー状態である
8	発信音がない
9	着信が検出された
10	ユーザーがダイヤルした
11	電話番号が無効である
12	回線でハードウェアエラーが発生して回線が閉じた

最終イベント文字列値

最終イベント	最終イベント文字列
-1	<空白> [まだイベントは発生していない]
0	TAPI で初期化されている
1	回線が閉じた
2	回線が開かれた
3	回線が接続された
4	ユーザーによって回線が切断された
5	リモートサイトで回線が切断された
6	応答がない
7	回線がビジー状態である
8	発信音がない
9	着信が検出された
10	ユーザーがダイヤルした
11	電話番号が無効である
12	回線でハードウェアエラーが発生して回線が閉じた
13	ダイヤルできない

通信シリアル化タグ

構文の例: <チャンネル名>._CommunicationSerialization._VirtualNetwork

タグ	説明
_NetworkOwner クラス: ステータスタグ	_NetworkOwner タグは、チャンネルが現在ネットワーク上の通信の制御を所有しているかどうかを示します。変更の頻度は、チャンネルに制御が付与される頻度を反映します。 このタグは読み取り専用です。
_Registered クラス: ステータスタグ	_Registered タグは、チャンネルが現在仮想ネットワークに登録されているかどうかを示します。_VirtualNetwork を設定すると、チャンネルは現在登録されているネットワーク(_RegisteredTo で示される) から登録解除されます(可能な場合)。言い換えると、チャンネルが切り替え中に制御を所有している場合、それが制御を解放するまで登録解除できません。チャンネルは登録解除において新しい仮想ネットワークに登録されま

タグ	説明
	<p>す。_VirtualNetwork が「なし」であれば、このタグは「FALSE」です。</p> <p>このタグは読み取り専用です。</p>
<p>_RegisteredTo</p> <p>クラス: ステータスタグ</p>	<p>_RegisteredTo タグは、チャンネルが現在登録されている仮想ネットワークを示します。_VirtualNetwork を設定すると、チャンネルは現在登録されているネットワークから登録解除されます (可能な場合)。言い換えると、チャンネルが切り替え中に制御を所有している場合、それが制御を解放するまで登録解除できません。チャンネルは登録解除において新しい仮想ネットワークに登録されます。このタグは、一定時間にわたって _VirtualNetwork と _RegisteredTo が異なるためネットワークの切り替えが遅延しているかどうかを示します。_VirtualNetwork が「なし」であれば、このタグは「N/A」です。</p> <p>このタグは読み取り専用です。</p>
<p>_StatisticAvgNetworkOwnershipTimeSec</p> <p>クラス: ステータスタグ</p>	<p>_StatisticAvgNetworkOwnershipTimeSec タグは、アプリケーションが起動された時点 (または最後に _StatisticsReset に書き込みがあった時点) からチャンネルが制御を所有していた時間の平均を示します。このタグは、ビジーチャンネル/ボトルネックを識別するために役立ちます。このタグは、32 ビット浮動小数点としてフォーマットされ、最終的にはロールオーバーする可能性があります。</p> <p>このタグは読み取り専用です。</p>
<p>_StatisticNetworkOwnershipCount</p> <p>クラス: ステータスタグ</p>	<p>_StatisticNetworkOwnershipCount タグは、アプリケーションが起動された時点 (または最後に _StatisticsReset に書き込みがあった時点) からチャンネルに通信の制御が付与された回数を示します。このタグは、符号なし 32 ビット整数としてフォーマットされ、最終的にはロールオーバーする可能性があります。</p> <p>このタグは読み取り専用です。</p>
<p>_StatisticNetworkOwnershipTimeSec</p> <p>クラス: ステータスタグ</p>	<p>_StatisticNetworkOwnershipTimeSec タグは、アプリケーションが起動された時点 (または最後に _StatisticsReset に書き込みがあった時点) からチャンネルが制御を所有していた時間を秒単位で示します。このタグは、32 ビット浮動小数点としてフォーマットされ、最終的にはロールオーバーする可能性があります。</p> <p>このタグは読み取り専用です。</p>
<p>_StatisticsReset</p>	<p>_StatisticsReset タグは、すべての統計カウンタをリセットするために使用できます。_StatisticsReset タグは、Boolean タグとしてフォーマットされます。_StatisticsReset タグに 0 以外の値を書き込むと、統計カウンタはリセットされます。</p> <p>このタグは読み取り/書き込み可能です。</p>
<p>_TransactionsPerCycle</p>	<p>_TransactionsPerCycle タグは、仮想ネットワーク内のほかのチャンネルと交代するときにチャンネルで発生する読み取り/書き込みトランザクションの数を示します。これにより、チャンネルレベルの設定をクライアントアプリケーションから変更できます。このタグは、符号付き 32 ビット整数 (Long) としてフォーマットされます。有効な範囲は 1 から 99 です。デフォルトの設定は 1 です。</p> <p>このタグは読み取り/書き込み可能です。</p>
<p>_VirtualNetwork</p> <p>クラス: パラメータタグ</p>	<p>_VirtualNetwork タグは、仮想ネットワークを選択してチャンネルを即座に変更できるようにします。これは文字列タグであるため、目的の仮想ネットワークは「なし」、「ネットワーク 1」、「ネットワーク 2」のように (最大値は「ネットワーク 500」) 使用可能なオプションを使用して文字列値としてタグに書き込まれる必要があります。通信シリアル化を無効にするには、「なし」を選択します。</p>

タグ	説明
	このタグは読み取り/書き込み可能です。

通信管理

自動格下げ

自動格下げのプロパティを使用して、デバイスが応答していない場合にドライバがデバイスを一時的にスキャン停止にできるようにします。反応していないデバイスをオフラインにして、反応していないデバイスとの通信を特定の時間にわたって停止することにより、ドライバは引き続き同じチャンネル上のほかのデバイスとの通信を最適化できます。特定の時間が経過すると、ドライバは反応していないデバイスと再び通信しようとして、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。

● 詳細については、[デバイスのプロパティ - 自動格下げ](#)を参照してください。

ネットワークインタフェース選択

イーサネットカプセル化モードで実行されているイーサネットドライバまたはシリアルドライバで使用する NIC カードは選択できます。ネットワークインタフェース機能は、NIC 名またはそれに現在割り当てられている IP アドレスに基づいて特定の NIC カードを選択するために使用されます。使用可能な NIC のリストには、一意の NIC カードと、複数の IP が割り当てられている NIC の両方が含まれています。選択すると、アクティブであると想定されるすべての WAN 接続 (ダイヤルアップ接続など) が表示されます。

イーサネットカプセル化

イーサネットカプセル化モードは、イーサネットネットワーク上のターミナルサーバーに接続しているシリアルデバイスとの通信用に設計されています。ターミナルサーバーは本質的には仮想シリアルポートであり、イーサネットネットワーク上の TCP/IP メッセージはターミナルサーバーによってシリアルデータに変換されます。メッセージがシリアル形式に変換されると、ユーザーはシリアル通信をサポートする標準デバイスをターミナルサーバーに接続可能になります。ターミナルサーバーデバイスを使用することで、1 台のローカル化された PC がリモートでマウントされているデバイスにアクセスできる状態で、RS-232 および RS-485 デバイスをプラント設備全体に配置できます。さらに、イーサネットカプセル化モードでは、必要に応じて各デバイスに個別のネットワーク IP アドレスを割り当てることができます。複数のターミナルサーバーを使用することで、1 台の PC から何百台ものシリアルデバイスにイーサネットネットワーク経由でアクセスできます。

● 詳細については、[操作方法](#)と[デバイスのプロパティ - イーサネットカプセル化](#)を参照してください。

モデムのサポート

このサーバーでは、リモートデバイスへの接続にモデムを使用できます。これは、ダイヤルアップ接続が作成されるとチャンネルレベルで使用できるようになる特殊なモデムタグを使用することによって確立されます。このようなチャンネルレベルのモデムタグは、リモートデバイスへのダイヤル、接続中のモデムのステータスのモニター、および完了時のコールの終了に使用できます。

● **注記:** すべてのシリアルドライバでモデムの使用がサポートされているわけではありません。モデムのサポートを確認するには、使用するドライバのヘルプドキュメントを参照してください。

[モデムシステムタグ](#)にアクセスするとき、ベースグループまたはトピック名としてチャンネル名を使用できます。モデムを使用するには、オペレーティングシステムでコントロールパネル設定を使用してモデムを構成する必要があります。モデムが適切にインストールされると、[チャンネルのプロパティ](#)で「物理メディア」として「**モデム**」を選択することによって有効にできます。

● 具体的なセットアップ情報については、Windows およびモデムのドキュメントを参照してください。

● **重要:** 最近の商用モデムの多くが、ネットワークサーバー接続をダイヤルアップし、最も速く、クリアな信号をネゴシエートするように設計されています。シリアル自動化デバイスと通信する場合、モデムは特定ボー (bps) およびパリティで接続する必要があります。このため、外部モデム (特定のボーレートおよびパリティ設定を使用してダイヤルするように構成できる) の使用を強くお勧めします。特定のアプリケーションに最適なモデムを確認するには、テクニカルサポートまでご連絡ください。プロジェクトでモデムを使用する方法の例については、[サーバープロジェクトでのモデムの使用](#)を参照してください。

サーバープロジェクトでのモデムの使用

モデムは、RS-232 ポートからのシリアルデータを、電話回線で転送できる信号レベルに変換します。これを行うため、モデムはシリアルデータの各バイトをビットに分解して信号を生成し、転送します。ほとんどのモデムが、送信されるデータのバイトにつき最大で 10 ビットの情報を変換できます。デバイスがモデムを介して通信するには、使用できるビット数が 10 ビット以下である必要があります。特定のデバイスによって使用されているビット数を確認するには、次の計算式を使用します。

スタートビット + データビット + パリティ + ストップビット = 合計ビット数

たとえば、Modbus RTU ドライバは、8 データビット、偶数パリティ、1 ストップビット、および 1 スタートビットを使用するように構成されています。これを計算式に当てはめると、1 + 8 + 1 + 1 = 11 ビットになります。通常のモデムは、この

Modbus デバイスにデータを転送できません。「パリティ」を「なし」に変更すると、これは $1 + 8 + 0 + 1 = 10$ ビットになります。通常のモデムは、この Modbus デバイスにデータを転送できます。

一部のドライバーは、10 ビット以下のデータフォーマットを使用するように構成できないため、標準のモデムを使用できません。代わりに、11 データビットの転送を処理できるモデムを必要とします。このカテゴリに分類されるドライバーを使用する場合は、推奨される適切なモデムベンダーについて、デバイスの製造メーカーにお問い合わせください。ドライバーがモデム操作をサポートしているかどうかに関係なく、モデム操作はすべてのシリアルドライバーで有効になります。

発信モデムの構成

このサーバーは、Windows TAPI インタフェースを使用して、PC に接続されているモデムにアクセスします。TAPI インタフェースは、PC に存在するさまざまなモデムがアクセスできる共通のインタフェースを Windows プログラムに提供するために設計されました。モデムの製造メーカーによって提供されている一連の Windows OS 用ドライバーがインストールされていなければ、サーバーはプロジェクトでモデムを使用できません。Windows のコントロールパネルを使用して新しいモデムをインストールできます。

● モデムのインストールとセットアップについては、Windows とモデムのドキュメントを参照してください。

モデムが適切にインストールされると、サーバープロジェクトで使用できるようになります。受信側（デバイスのモデム）が適切に構成されていなければ、その使用を開始できません。受信モデムが、ドライバーによって提供されるプロファイルと一致していることを確認する必要があります。

ケーブル

受信モデムとデバイス間でケーブル接続が構成されていなければ、プロジェクトを使用できません。直接接続のための既存のデバイス通信ケーブル、NULL モデムアダプタ、および NULL モデムケーブルの 3 本のケーブルが必要です。NULL モデムケーブルは、モデムに接続され、ケーブルの両端ですべてのピンが同じピンに接続されます。デバイス通信ケーブルは、ターゲットデバイスに接続するために使用され、通常はピン 2 とピン 3 が逆です。デバイスと通信するために使用される直接接続のためのケーブルは、この時点ではすでに機能しているため、NULL モデムアダプタを接続することによって受信モデムで使用できます。同様に、PC モデムケーブルは、PC から発信モデムに配線されます。ケーブルが適切に配置されると、アプリケーションでモデムを使用できるようになります。

● 注記: NULL モデムアダプタは、ほとんどのコンピュータ販売店で入手できます。

例: サーバー側モデムの構成

モデムが構成され、インストールされると、サーバーで使用できるようになります。

1. まず、直接接続プロジェクトをロードし、チャンネル名をダブルクリックします。「**チャンネルのプロパティ**」で、「**シリアル通信**」グループを開きます。
2. 「**物理メディア**」ドロップダウンメニューで「**モデム**」を選択します。
3. 「**モデム設定**」で、コンピュータで使用できるモデムを選択します。

● 注記: コンピュータに使用可能なモデムがない場合は、「物理メディア」ドロップダウンメニューで「モデム」を選択できません。これに該当する場合は、サーバーを終了し、オペレーティングシステムで提供されているモデム構成ツールを使用して、モデムの再インストールを試みてください。

4. 発信モデムの特性を構成するには、「**モデム設定**」のプロパティを使用します。詳細については、「**チャンネルのプロパティ - シリアル通信**」を参照してください。
5. 終了後、「**適用**」をクリックします。次に、「**OK**」をクリックして「チャンネルのプロパティ」を終了します。

アプリケーションでのモデムの使用

モデム操作を有効にすると、定義済みのタグのリストがデータクライアントで使用できるようになります。これらの**モデムタグ**は、チャンネル名に含まれており（モデムタグにアクセスするためのアクティブな OPC アクセスパスになりました）、接続されているモデムを制御およびモニターします。サーバーは、アプリケーションがモデムの制御に必要な要件をほとんど把握していないため、どのようなタイプの制御も示唆しません。定義済みのモデムタグを使用することにより、アプリケーションのスクリプト作成機能を適用して、サーバーが特定のモデムをどのように使用するかを制御できます。

電話帳

電話帳は電話帳タグ（電話番号）の集合であり、モデムシステムタグ内の "_PhoneNumber" タグに書き込まれる電話番号を指定する代わりに使用できます。電話帳は「**物理メディア**」が「**モデム**」に設定されているすべてのチャンネルに自動的に作成されます。電話帳タグに関連付けられているデータは、サーバーによってダイヤルされる電話番号です。クライアントが電話帳タグに書き込むと、サーバーはそのタグに関連付けられている電話番号をダイヤルします。

データ型	権限
文字列	読み取り/書き込み

電話帳タグは電話帳に新しいエントリを作成することによって作成されます。新しい電話帳エントリを追加するには、プロジェクトツリーで電話帳ノードをクリックし、「新しい電話番号」アイコンをクリックします。

これによって「電話番号」プロパティエディタが開きます。

「名前」: 電話番号エントリの名前を指定します。これは "_Phonebook" システムタググループの OPC ブラウズデータの一部になります。長さは最大 256 文字です。通常はわかりやすい名前を使用することをお勧めしますが、一部の OPC クライアントアプリケーションでは、OPC サーバーのタグ空間をブラウズするとき、表示ウィンドウの領域が限られている可能性があります。電話番号の名前は電話帳内で一意である必要があります。

「番号」: 関連付けられている電話帳タグが OPC クライアントアプリケーションから呼び出された場合にダイヤルする電話番号を指定します。最大 64 桁の文字列を入力できます。

「説明」: 電話番号エントリに添付するコメントのテキストを入力します。長さは最大 255 文字です。

● **注記:** サーバーがオンラインで常時稼働している場合、これらのパラメータをいつでも変更できます。プロパティに加えた変更は、ただちに有効になります。ただし、当該のタグにすでに接続している OPC クライアントは、そのタグを解放して再取得するまで影響を受けません。

自動ダイヤルの優先順位

チャネルで自動ダイヤルが有効になっている場合、初期接続リクエストは、電話帳で最初に見つかったエントリにダイヤルしようとするにより開始されます。その試行が失敗すると、電話帳の次の番号へのダイヤルが試行されます。このシーケンスは、モデム接続が確立されるまで、またはチャネルが提供できるデータへのすべての参照をクライアントが解放するまで継続されます。自動ダイヤルで使用される優先順位はユーザー定義であり、電話帳エントリを選択して以下に示すいずれかの優先順位変更アイコンをクリックすることによって変更できます。選択したエントリのコンテキストメニューを開くことによって変更できます。

例

"Site1" という名前で作成された電話帳エントリ:

構文の例: <チャネル名>._Phonebook.Site1

自動ダイヤル

自動ダイヤルは、サーバープロジェクト内でモデムの使用が指定されている場合にクライアントで必要とされる操作を自動化します。自動ダイヤルを使用しない場合、これらの操作（接続、切断、電話番号の割り当てなど）は外部のクライアントアプリケーションによってチャネルレベルのモデムタグを介して行われます。たとえば、接続を確立するプロセスを開始するには、クライアントは "<>" という名前のダイヤル文字列を書き込みます。_Modem_電話番号" に書き込み、値を "<チャネル名>" に書き込みます。_Modem_Dial" に書き込みます。リモートデバイスからのデータが不要になると、クライアントは "<チャネル名>._Modem_Hangup" に書き込むことによってコールを終了します。

自動ダイヤルは、接続の確立を試行する際に、電話帳で定義されている電話番号に自動的にダイヤルするため、クライアントはこれらの操作を行う必要がありません。モデム接続に依存しているタグへのクライアント参照がなくなると、接続は自動的に切断されます。自動ダイヤルプロパティにアクセスするには、「**チャネルのプロパティ**」|「**シリアル通信**」の順にクリックします。

● 詳細については、[チャネルのプロパティ - シリアル通信](#)を参照してください。

モデムの接続と切断

モデム接続を確立するプロセスは、クライアントがサーバーランタイムに接続し、自動ダイヤルが有効になっているチャネルにデバイス接続からのデータを要求すると開始されます。最初の接続要求は、電話帳で最初に検出された電話番号へのダイヤルを試行することによって開始されます。その試行が失敗すると、電話帳の次の番号へのダイヤルが試行されます。このシーケンスは、モデム接続が確立されるまで、またはチャネルが提供できるデータへのすべての参照をクライアントが解放するまで継続されます。

● **注記:** 接続を再確立するとき、最後に正常に接続した電話帳エントリが使用されます。正常に接続したエントリが電話帳にない場合（またはそのエントリが削除されている場合）、ユーザー定義の電話番号のシーケンスが使用されます。サーバーの再初期化中または再起動中、再ダイヤルに使用された番号は保持されません。

● 関連項目: [電話帳](#)

タイミング

タイミング設定 (次の番号に進むまで接続を待つ時間など) は、特定のモデム自動ダイヤル設定ではなく、TAPI モデム構成によって決定されます。

● **注記:** ドライバーによっては、シリアルポートが一度開かれた後に閉じることが許可されていない場合があります。これらのドライバーを使用して確立された接続は、すべてのクライアント参照が解放されるまで切断されません (一定のアイドル時間が経過した後に切断するように TAPI 設定が構成されていないかぎり)。

クライアントアクセス

モデムタグを使用してクライアントレベルでモデムを制御できます。ただし、モデム自動ダイヤルが有効になっていると、1つのフォームのアクセスのみが可能であるため、モデムタグへの書き込みアクセスが制限されます。モデムタグの値は、クライアントがモデムを制御しているかのように更新されます。

構成からの自動ダイヤル設定の変更

ランタイムは、以下の規則に従って、設定の変更に反応します。

- クライアントがモデムにダイヤルし、接続を確立した後で自動ダイヤルが有効になると、変更はモデムが切断されるまで無視されます。切断時にクライアントがまだチャネルからのデータを要求している場合、最初の接続シーケンスが開始されます。
- モデム接続が存在せず、クライアントによってチャネルからのデータが要求されているときに自動ダイヤルが有効になると、最初の接続シーケンスが開始されます。
- 既存の自動ダイヤル接続が存在するときに自動ダイヤルが無効になると、処理は一切行われず、接続は切断されます。

● 関連項目: [チャネルのプロパティ - シリアル通信](#)

プロジェクトの設計

以下の例は、サーバーに付属の Simulator Driver を使用してプロジェクトを作成、構成、および実行するプロセスを示しています。Simulator Driver は、デモの目的で静的なデータと変化するデータの両方を提供するメモリベースのドライバーです。これはほかの通信ドライバーで使用されている構成オプションをすべてサポートしているわけではないため、一部の例では特定の製品機能を説明するためにほかのドライバーの図が使用されています。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。

[サーバーの実行](#)

[新しいプロジェクトの開始](#)

[チャネルの追加と構成](#)

[デバイスの追加と構成](#)

[ユーザー定義のタグの追加](#)

[複数のタグの生成](#)

[タグスケール変換の追加](#)

[プロジェクトの保存](#)

[暗号化されたプロジェクトを開く](#)

[プロジェクトのテスト](#)

● ソフトウェアおよびハードウェア要件については、[システム要件](#)を参照してください。

サーバーの実行

このサーバーは、サービスとしてもデスクトップアプリケーションとしても実行できます。サービスとしてデフォルト設定で実行する場合、サーバーは常にオンラインです。デスクトップアプリケーションとして実行する場合、OPC クライアントは自動的にサーバーを起動して接続し、データを収集しようとします。いずれの場合も、プロセスが正常に機能するには、まずプロジェクトを作成して構成する必要があります。開始時、サーバーは最後に使用したプロジェクトを自動的にロードします。

最初に、ユーザーが手動でサーバーを起動する必要があります。これを行うには、デスクトップのアイコンをダブルクリックするか、システムトレイにある管理メニューから「**構成**」を選択します。インタフェースの外観は、ユーザーが加えた変更によって異なります。

サーバーの実行が開始されると、プロジェクトを作成できるようになります。

● サーバー要素の詳細については、[基本的なサーバーコンポーネント](#)を参照してください。ユーザーインタフェースの詳細については、[ユーザーインタフェースのナビゲーション](#)を参照してください。

新しいプロジェクトの開始

操作中に提供されるコンテンツを決定するようにサーバーを構成する必要があります。サーバープロジェクトにはチャネル、デバイス、タググループ、およびタグの定義が含まれています。これらの要素は、プロジェクトファイルのコンテキストにあります。多くのアプリケーションと同様に、多数のプロジェクトファイルを定義、保存、およびロードできます。

一部の構成オプションはグローバルであり、すべてのプロジェクトに適用されます。これらのグローバルオプションは「**ツール**」|「**オプション**」ダイアログで構成します。このダイアログには「一般」オプションと「ランタイム接続」オプションの両方が含まれています。これらの設定は、インストール中に選択された Application Data ディレクトリに保存されている "settings.ini" と呼ばれる Windows INI ファイルに保存されます。グローバルオプションは通常は Windows レジストリに保存されますが、INI ファイルを使用すると、これらのグローバル設定をマシン間でコピーできます。

ソフトウェアは最初はデフォルトのプロジェクトが開いた状態で開きます。そのファイルをその他のファイルと同様に編集、保存、クローズできます。

1. 新しいプロジェクトを定義するには、「**ファイル**」|「**新規**」の順に選択します。
2. オフラインでクローズ、保存、編集するよう求められた場合。
3. 「**ファイル**」|「**名前を付けて保存**」を選択します。
4. 暗号化されたプロジェクトファイルをセキュリティで保護するためのパスワードを入力します。
5. ファイルの保存場所を選択します。


6. 「保存」をクリックします。
7. [チャンネルの追加](#)によってプロジェクトファイルの設定を開始します。

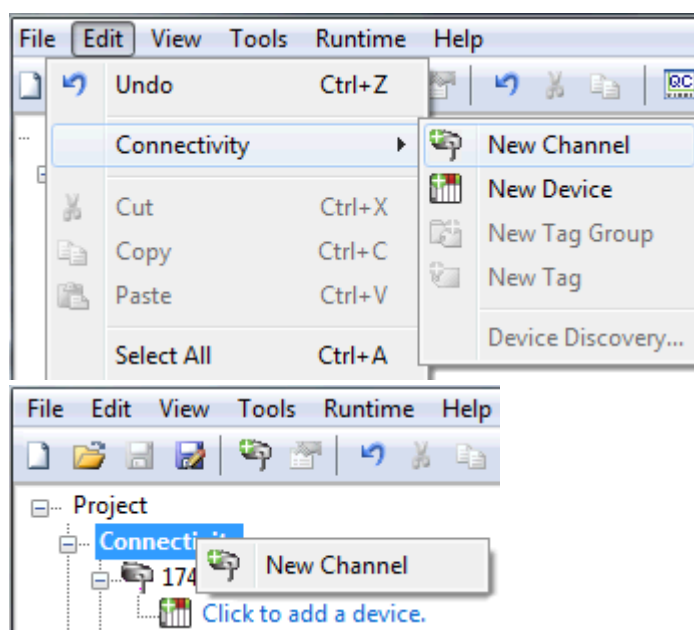
● 関連項目: [オプション - 一般](#)、[プロジェクトの保存](#)

チャンネルの追加と構成

新しいプロジェクトを作成するとき、まずアプリケーションが必要とする通信ドライバーを決定する必要があります。サーバーではこれをチャンネルと呼びます。チャンネル数は、インストールされているドライバーに応じて、単一のプロジェクト内で定義できます。詳細については、次の手順を参照してください。

1. まず、プロジェクトに新しいチャンネルを追加します。
「編集」|「接続性」|「新しいチャンネル」の順にクリックする、

ツールバーの「新しいチャンネル」アイコン  をクリックする、または
ツリーで「接続」ノードを右クリックして「新しいチャンネル」を選択する方法があります。



2. [チャンネルウィザード](#)では、チャンネル名をそのデフォルト設定 "Channel1" のままにしておきます。次に、「次へ」をクリックします。
3. 「デバイスドライバー」で、チャンネルに適用される通信ドライバーを選択します。次に、「次へ」をクリックします。この例では、Simulator Driver が使用されています。
4. Simulator Driver の場合、次のページは「チャンネルのサマリー」です。その他のデバイスのチャンネルウィザードには、ほかのプロパティ (通信ポート、ボーレート、パリティなど) を構成できる追加のページがあります。詳細については、[チャンネルのプロパティ - シリアル通信](#)を参照してください。
5. 完了した後、「終了」をクリックします。

● 関連項目: [サーバープロジェクトを最適化する方法](#)、[サーバーのサマリー情報](#)

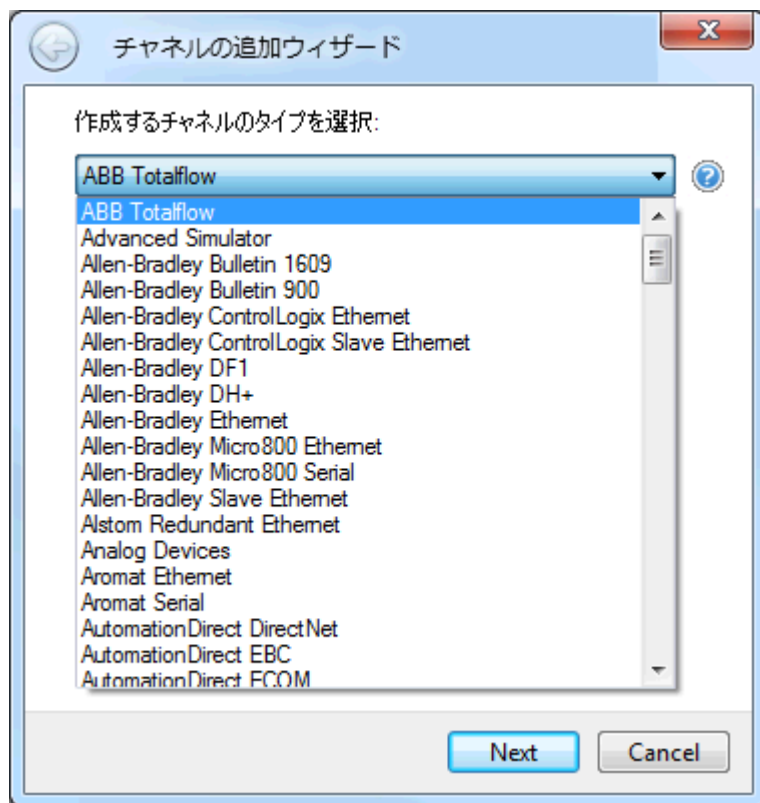
チャンネル作成ウィザード

チャンネル作成ウィザードのステップに従って、(使用しているプロトコルによって定義される) チャンネルを設定できます。チャンネルを定義した後は、そのチャンネルに割り当てられているすべてのデバイスでそのプロパティと設定が使用されます。一部のプロパティは選択しているプロトコルまたはドライバーによって異なります。

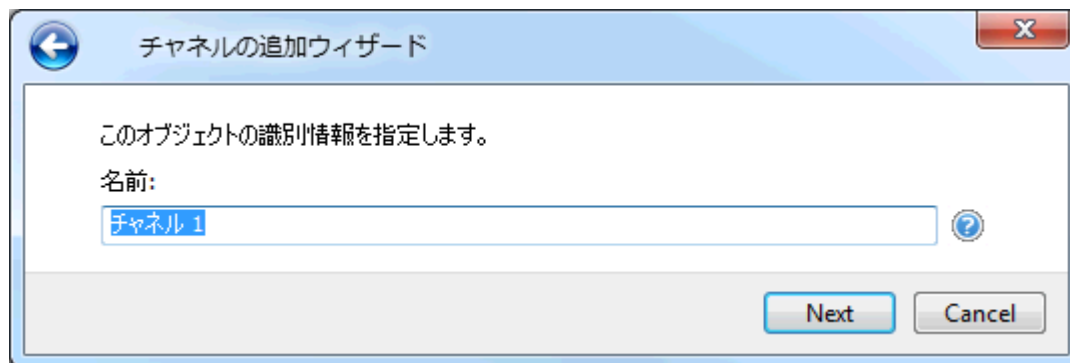
1. ツリービューで、「接続性」ノードを右クリックし、「新しいチャンネル」を選択します (または「編集」|「接続性」|「新しいチャンネル」の順に選択します)。



2. 使用可能なドライバーのドロップダウンリストから、作成するチャネルのタイプを選択します。



3. 「次へ」をクリックします。
4. チャネルを容易に区別できるような名前を入力します (タグのパス、イベントログメッセージ、エイリアスで使用されます)。



5. 「次へ」をクリックします。
6. オプションと環境に応じて チャンネルのプロパティ を設定します。

7. 新しいチャンネルのサマリーを確認し、「戻る」を選択して変更を行うか、「完了」を選択して閉じます。

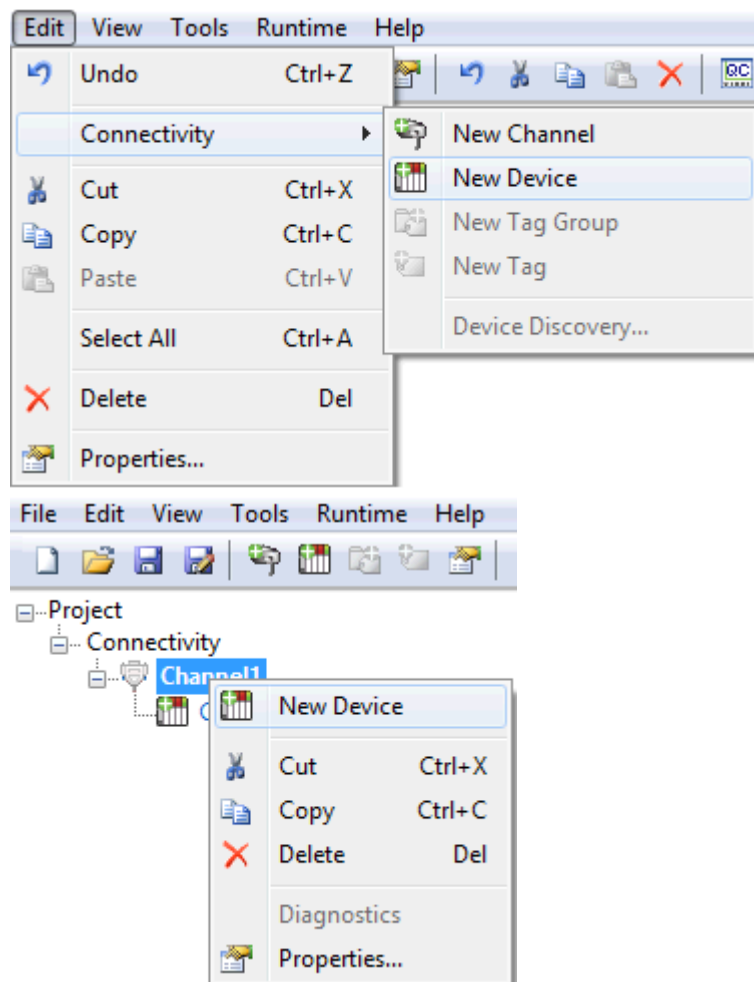
デバイスの追加と構成

チャンネルの定義が完了すると、デバイスを追加できるようになります。デバイスによって通信リンクの物理ノードまたはステーションが識別されるため、デバイスは接続の定義をアプリケーション内の特定の注目点にフレームする 1 つの方法として考えることができます。この点において、デバイスは、データベースオブジェクトへの接続を説明するには適切な用語です。したがって、「デバイス」は、ネットワーク上の特定のデバイスを指し、複数のデバイスノードをサポートするため、ユーザーはネットワーク化されたデバイスをシミュレートできます。

● **注記:** この例では、Simulator Driver が使用されています。デバイスウィザードのオプションは、ドライバーによって異なります。

1. まず、デバイスを追加するチャンネルを選択します。
2. まず、プロジェクトに新しいデバイスを追加します。
「編集」|「接続性」|「新しいデバイス」の順にクリックする、

ツールバーの「新しいデバイス」アイコン  をクリックする、または
ツリーで「接続」ノードを右クリックして「新しいデバイス」を選択する方法があります。



3. [デバイスウィザード](#)で、名前をそのデフォルト設定 "Device1" のままにし、「次へ」をクリックします。
4. 「モデル」で、シミュレートするデバイスのレジスタサイズを 8 ビットまたは 16 ビットから選択し、「次へ」をクリックします。

● **注記:** デバイスドライバーによっては、代わりにデバイスモデルを選択する必要があります。この例では、16 ビットレジスタサイズが選択されています。

5. 「ID」で、デバイス ID (実際の通信プロトコルが必要とする一意の識別子) を選択します。次に、「次へ」をクリックします。

● **注記:** デバイス ID のフォーマットとスタイルは、使用されている通信ドライバーによって異なります。Simulator Driver の場合、デバイス ID は数値です。

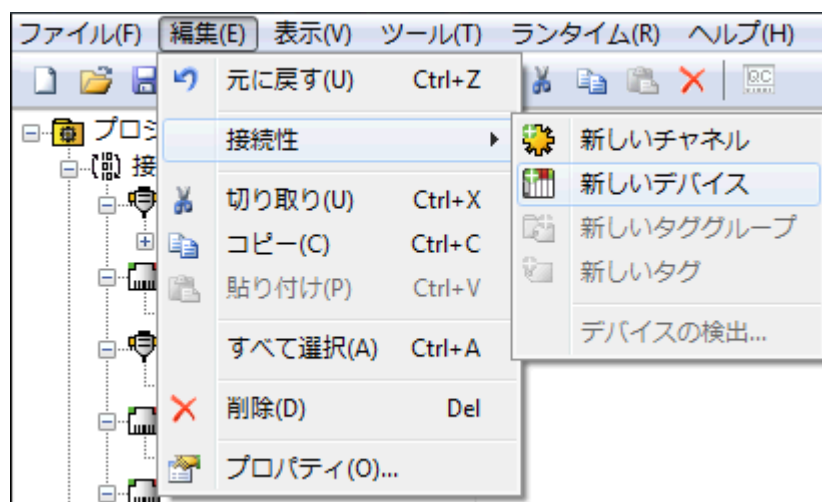
6. 「スキャンモード」で、デバイスのスキャン速度を指定します。次に、「次へ」をクリックします。
7. Simulator Driver の場合、次のページは「**デバイスのサマリー**」です。その他のドライバーのデバイスウィザードには、ほかのプロパティ(タイミングなど)を構成できる追加のページがあります。詳細については、[デバイスのプロパティ](#)を参照してください。
8. 完了した後、「終了」をクリックします。

● **注記:** サーバーがオンラインで常時稼働している場合、サーバーはただちに OPC データの提供を開始できます。ただし、この時点では、プロジェクトが保存されていないため、構成が失われる可能性があります。保存する前に、サーバーにタグを追加できます。詳細については、[ユーザー定義のタグの追加](#)を参照してください。

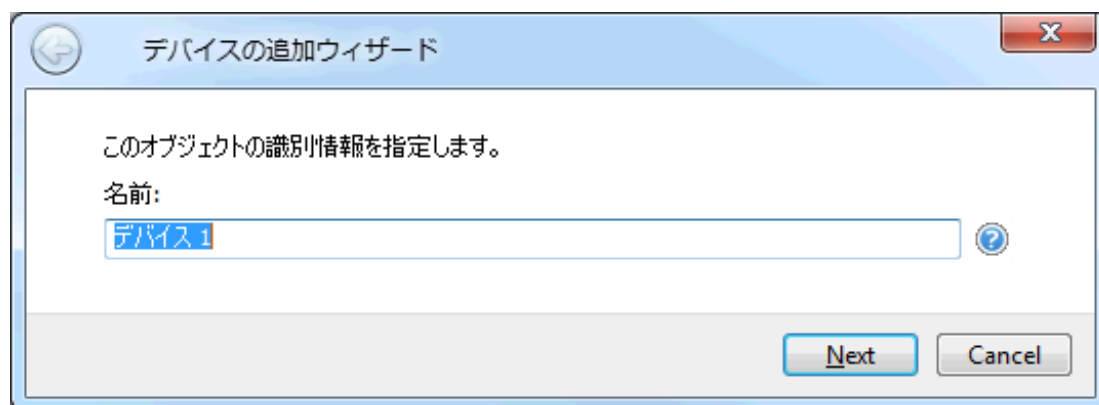
デバイス作成ウィザード

デバイス作成ウィザードのステップに従って、通信とデータ収集を行うデバイスを設定できます。一部のプロパティは選択しているプロトコルまたはドライバーによって異なります。

1. ツリービューで、デバイスを追加するチャンネルを見つけて選択します。
2. 右クリックして「新しいデバイス」を選択するか、「編集」|「接続性」|「新しいデバイス」の順に選択します。



3. デバイスを容易に区別できるような名前を入力します (タグのパス、イベントログメッセージ、エイリアスで 사용됩니다)。



4. 「次へ」をクリックします。
5. オプションと環境に応じて[デバイスのプロパティ](#)を設定します。
6. 新しいデバイスのサマリーを確認し、「Back」を選択して変更を行うか、「Finish」を選択して閉じます。

ユーザー定義のタグの追加 (例)

サーバーは、2 つの方法でデバイスからクライアントアプリケーションにデータを取得できます。最も一般的な方法では、サーバープロジェクトで一連のタグを定義し、各タグに割り当てられていた名前をクライアントとサーバー間の各リンクのアイテムとして使用する必要があります。この方法により、ユーザー定義のすべてのタグを OPC クライアント内でブラウズできます。

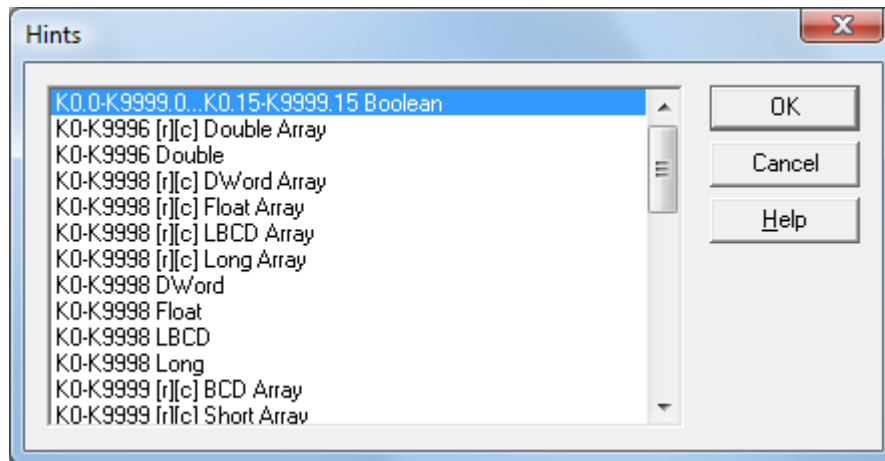
- ユーザー定義のタグは、スケール変換をサポートしています。詳細については、[タグスケール変換の追加](#)を参照してください。
- 状況によっては、複数のタグのブラウズと選択がサポートされています。詳細については、[タグのブラウズ](#)を参照してください。

1. 最初に、接続性ツリーノードからデバイス名を選択します。この例で選択されているデバイスは "Device1" です。
2. 「編集」|「接続性」|「新しいタグ」の順にクリックします。または、デバイスを右クリックし、「新しいタグ」を選択します。
3. 「タグのプロパティ - 一般」で、以下のようにプロパティを編集します。

- ・「タグ名」 MyFirstTag
- ・「アドレス」 R000
- ・「説明」(オプション) 最初の Simulator タグ
- ・「データ型」 Word
- ・「クライアントアクセス」 読み取り/書き込み
- ・「スキャン速度」 100 ミリ秒。このプロパティは、OPC タグには適用されません。

- 詳細については、[タグのプロパティ - 一般](#)を参照してください。

4. 必要に応じて、「ヒント」を使用してドライバーの正しい設定を特定します。ヒントを起動するには、「タグのプロパティ」にある疑問符アイコンをクリックします。



● **注記** 「アドレス」、「データ型」、および「クライアントアクセス」フィールドは、通信ドライバーによって異なります。たとえば、Simulator Driver では "R000" が有効なアドレスであり、これはデータ型 Word をサポートし、読み取り/書き込みアクセスを持ちます。

5. 追加の情報を確認するには、「ヘルプ」をクリックします。これにより、ドライバーのヘルプドキュメントで "アドレス説明" トピックが呼び出されます。
6. 「適用」を押してタグをサーバーにコミットします。これでタグがサーバーに表示されるようになります。
7. この例では**タグのプロパティ - スケール変換**で使用する 2 つ目のタグを追加する必要があります。これを行うには、「タグのプロパティ - 一般」にある「新規」アイコンをクリックします。これにより、プロパティがデフォルト設定に戻ります。
8. 以下のように入力します。
 - ・「タグ名」 MySecondTag
 - ・「アドレス」 K000
 - ・「説明」 2 つ目のスケール変換済みのタグ
 - ・「データ型」 Short
 - ・「クライアントアクセス」 読み取り書き込み
9. 次に、「適用」を押して新しいタグをサーバーにコミットします。これでタグがサーバーに表示されるようになります。

エラーメッセージ

タグ情報を入力しているときに、サーバーまたはドライバーからエラーメッセージが表示されることがあります。サーバーは、ユーザーが既存のタグと同じ名前のタグを追加しようとすると、エラーメッセージを生成します。通信ドライバーがエラーを生成する原因として、以下の 3 つが考えられます。

1. アドレスのフォーマットまたはコンテンツにエラーが入力された (特定のデバイス固有のデータアイテムの範囲内を含む)。
2. 選択したデータ型はアドレスに使用できない。
3. 選択したクライアントアクセスレベルはアドレスに使用できない。

● 特定のエラーメッセージの詳細については、[エラーの説明](#)を参照してください。

動的タグアドレス指定

動的タグアドレス指定は、クライアントアプリケーションでのみタグを定義します。サーバーで作成された別のタグアイテムのアドレスを指定するタグアイテムをクライアントで作成する代わりに、デバイスアドレスに直接アクセスするタグアイテムをクライアントで作成する必要があるだけです。サーバーはクライアント接続でその位置の仮想タグを作成し、自動的にデータのスキャンを開始します。

● 詳細については、[動的タグ](#)を参照してください。

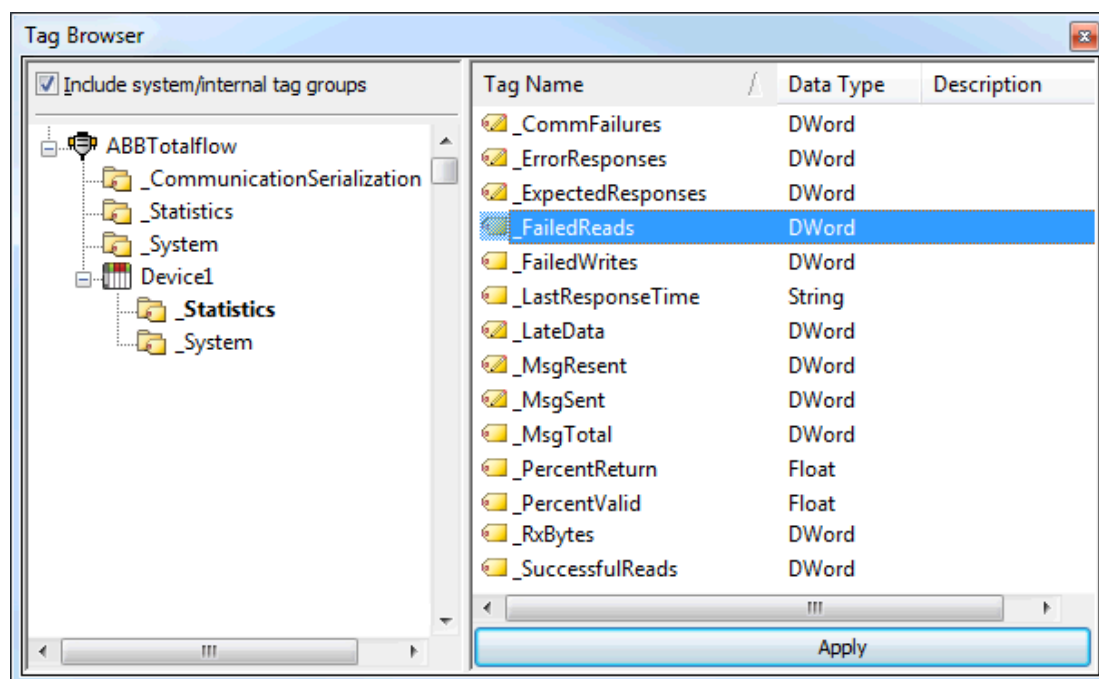
● **ヒント:**

1. サーバーは、プロジェクト内のすべてのデバイスに対して特殊な Boolean タグを作成します。クライアントは、これを使用して、デバイスが適切に機能しているかどうかを確認します。このタグを使用するには、リンク内のアイテムを "エラー" として指定します。このタグは、デバイスが適切に通信している場合は 0、適切に通信していない場合は 1 になります。
2. データ型を省略すると、ドライバーは参照されているデバイスとアドレスに基づいてデフォルトのデータ型を選択します。ドライバーのヘルプドキュメントに、すべての位置のデフォルトのデータ型が記載されています。指定されているデータ型がデバイスの位置に対して有効でなければ、サーバーはタグを却下し、イベントログにエラーが出力されます。
3. デバイスアドレスがリンクのアイテムとして使用されていると (アドレスがサーバー内のユーザー定義のタグの名前と一致するように)、リンクはユーザー定義のタグが指すアドレスを参照します。サーバーがオンラインで常時稼働している場合は、この時点で、このプロジェクトの OPC クライアントでの使用を開始できます。

タグのブラウズ

サーバーでは、使用可能なタグのブラウズがサポートされており、場合によっては複数のタグを選択してプロジェクトに追加できます。

1. 「タグブラウザ」ダイアログボックスにアクセスします。



2. 「システム/内部タググループを含める」を使用できる場合は、それを有効にして、これらのグループを選択できるようにします。
3. 「ブランチレベルのタグ選択」を使用できる場合は、それを有効にして、ブランチノードを左側のツリービューで選択できるようにします (これにより、関連付けられているすべてのタグが右側で選択されます)。
4. 左側のツリーを移動して、追加するタグを含んでいるブランチを見つけます。
5. 「ブランチレベルのタグ選択」が有効になっていない場合は、右側でタグを選択します。複数のタグを追加できる場合は、標準のキーボード機能 (Shift, Ctrl) を使用して複数のタグを選択できます。
6. 「適用」をクリックします。

● 関連項目: [ユーザー定義のタグの追加](#)

複数のタグの生成

「複数タグの生成」ツールは、ユーザー定義のドライバーの命名法を使用してタグを動的に作成します。このツールの使用方法については、次の手順を参照してください。

● そのプロパティの詳細については、[複数タグの生成](#)を参照してください。

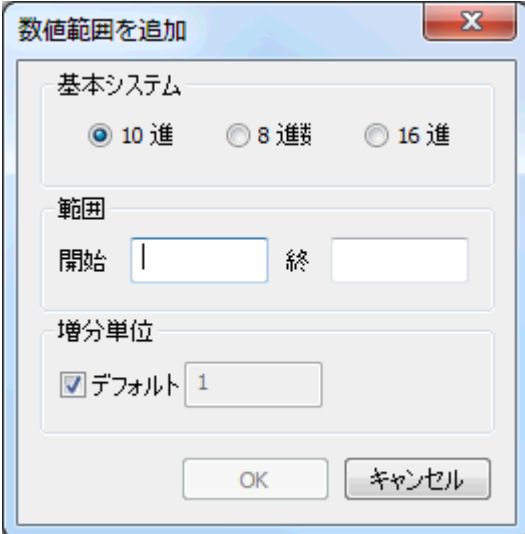
1. まず、デバイスを選択し、「編集」|「接続」|「新しいタグ」の順にクリックします。または、デバイスを右クリックし、「新しいタグ」を選択します。
2. 「タグのプロパティ」で、「複数タグの生成」アイコン（「識別」プロパティの右下にある）を選択します。



3. 「複数タグの生成」で、タグ名を定義し、必要に応じて「データプロパティ」のプロパティを構成します。

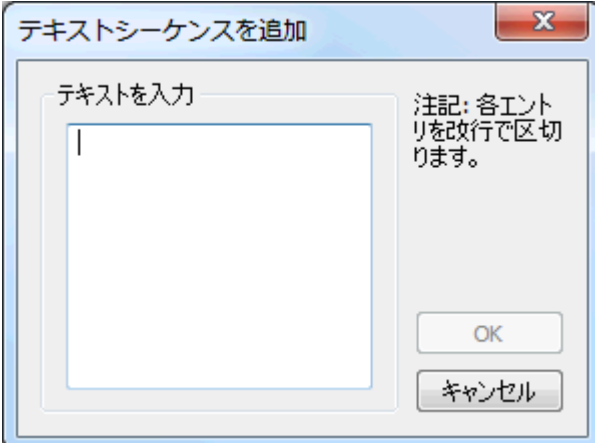
4. 「静的テキストを追加」をクリックします。このグループで、必要に応じてテキストを入力します。終了後、「OK」を押します。

5. 「数値範囲を追加」をクリックします。このグループで、基本システム、範囲、および増分を入力します。終了後、「OK」を押します。



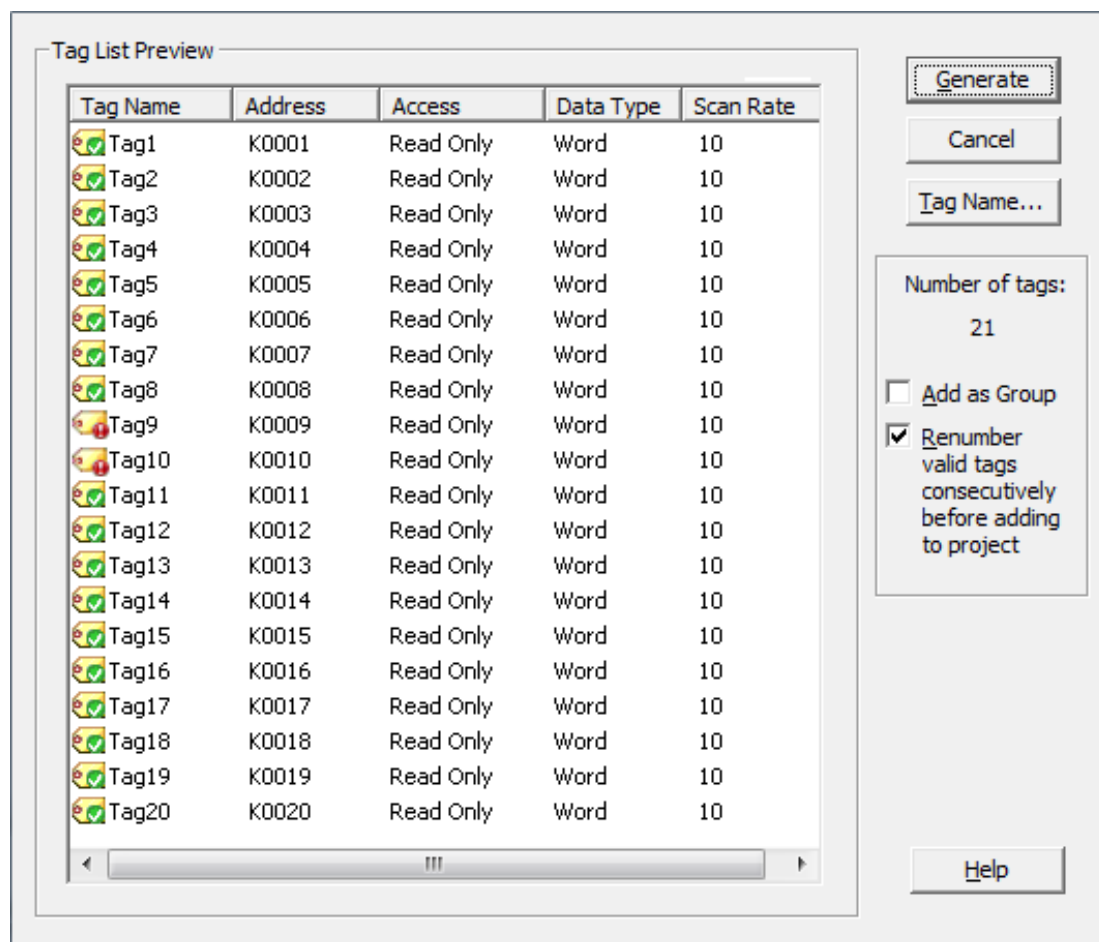
The dialog box is titled "数値範囲を追加" (Add Numerical Range). It contains three sections: "基本システム" (Basic System) with radio buttons for "10進" (selected), "8進" (octal), and "16進" (hexadecimal); "範囲" (Range) with "開始" (Start) and "終" (End) input fields; and "増分単位" (Increment Unit) with a checked "デフォルト" (Default) checkbox and a value of "1". At the bottom are "OK" and "キャンセル" (Cancel) buttons.

6. 「テキストシーケンスを追加」をクリックします。このグループで、必要に応じてテキストを入力します。エントリを区切るには改行を使用します。終了後、「OK」を押します。



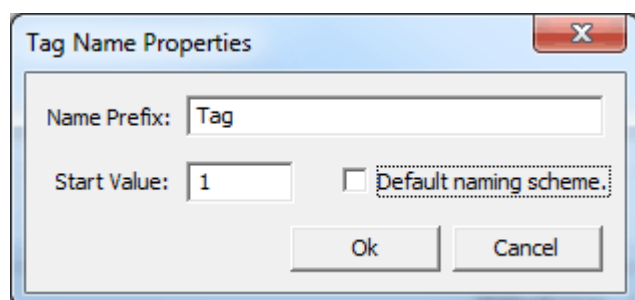
The dialog box is titled "テキストシーケンスを追加" (Add Text Sequence). It features a large text input area labeled "テキストを入力" (Enter text). To the right of the input area is a note: "注記: 各エントリを改行で区切ります。" (Note: Separate each entry with a line break). At the bottom right are "OK" and "キャンセル" (Cancel) buttons.

7. 「プレビュー」をクリックします。



● **注記:** 有効なタグには緑色のチェックマークが表示されます。無効なタグには赤色の x が表示されます。

8. タグをグループとして追加するには、「**グループとして追加**」を使用します。
9. タグの名前または開始値を変更するには、「**タグ名**」を選択します。終了後、「**OK**」をクリックします。



10. タグを生成するには、「**生成**」をクリックします。生成が正常に行われると、「複数タグの生成」ダイアログに戻ります。
11. 「**閉じる**」をクリックします。次に、「**OK**」をクリックします。生成されたタグがタグ表示ウィンドウに表示されます。

● **関連項目:** [複数タグの生成](#)

タグスケール変換の追加

サーバーで新しいタグを作成するとき、タグスケール変換を適用できます。これにより、デバイスからの生データをアプリケーションに適切な範囲にスケール変換できます。スケール変換には線形と平方根の2つのタイプがあります。詳細については、[タグのプロパティ - スケール変換](#)を参照してください。

1. まず、タグの「タグのプロパティ」を開きます。
2. 「スケール変換」グループを開きます。
3. タイプとして、「線形」または「平方根」を選択します。
4. デバイスから予想されるデータ範囲を上限/下限の値とクランプによって指定します。また、「スケール変換後のデータ型」で、スケール変換後の値がどのように OPC クライアントアプリケーションに提示されるかも指定できます。

プロパティグループ	
一般	
スケール変換	

スケール変換	
タイプ	線形
生データ下限	0
生データ上限	1000
スケール変換後のデータ型	Double
スケール変換後の下限	0
スケール変換後の上限	1000
下限でクランプ	いいえ
上限でクランプ	いいえ
値を負数化	いいえ
単位	

5. 「単位」で、スケール変換後のエンジニアリング値のフォーマットまたは単位を OPC クライアントに対して説明する文字列を指定します。「単位」フィールドを使用するには、Data Access 2.0 のタグのプロパティデータにアクセスできる OPC クライアントが必要です。クライアントがこれらの機能をサポートしていない場合は、このフィールドを構成する必要はありません。
6. これまでに説明したようにデータを入力した後、「OK」をクリックします。

プロジェクトの保存

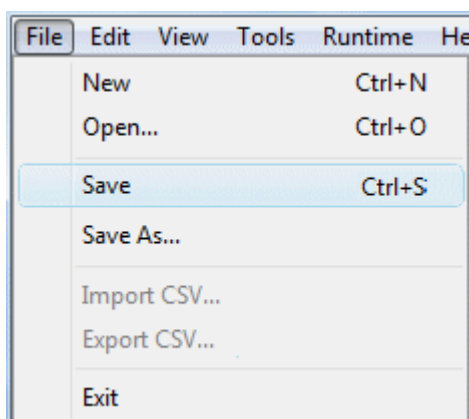
これまでの手順により、保存可能なユーザー定義タグを使用してプロジェクトが構成されました。プロジェクトの保存方法は、プロジェクトがランタイムプロジェクトであるかオフラインプロジェクトであるかによって異なります。

サーバーがオンラインで常時稼働している状態でランタイムプロジェクトを編集すると、プロジェクトがディスクに保存された直後に、クライアントからタグにアクセスすることができます。変更は実際のプロジェクトに適用されるので、ユーザーは「ファイル」|「保存」の順にクリックすることで保存できます。

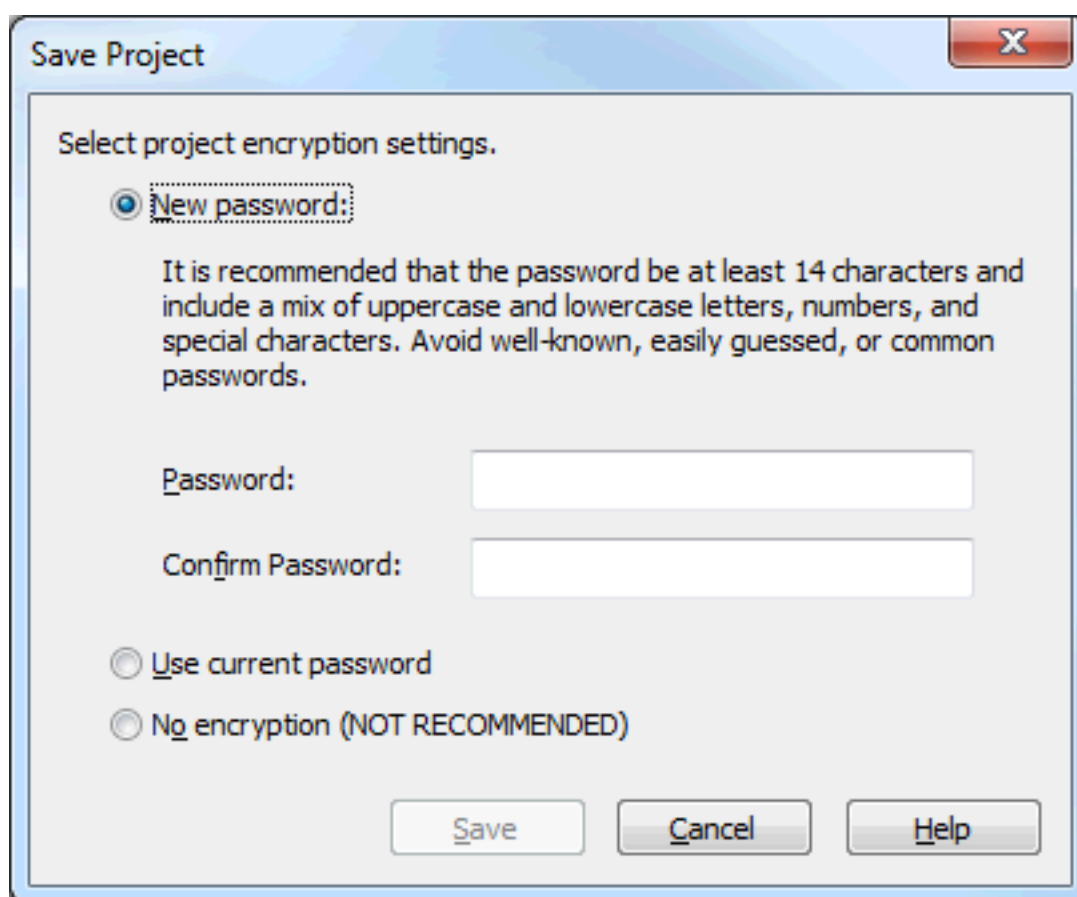
プロジェクトファイルでは、.OPF、.SOPF、.JSON など、いくつかのファイルフォーマットがサポートされています。.OPF フォーマットは、暗号化されていないバイナリプロジェクトファイルフォーマットです。.JSON (JavaScript Object Notation) フォーマットは、人間が読むことができるテキストベースの便利なフォーマットですが、安全性が多少低いので、セキュリティを確保する方法が導入されている場合にのみ使用してください。プロジェクトは .SOPF ファイルとして保存することをお勧めします。このファイルは暗号化されているため、最も安全性の高い方法でプロジェクトファイルを保存できます。

既存のプロジェクトを上書きしたり、編集内容を新しいプロジェクトとして保存したり、新しいプロジェクトをデフォルトのランタイムプロジェクトとしてロードしたりすることができます。

「ファイル」|「開く」の順に選択してプロジェクトファイルを見つけて選択することで、保存済みのプロジェクトを開きます。



オフラインプロジェクトを編集する場合、同じプロジェクトを保存することも、新しいプロジェクトとして保存することもできます。完了した後、「ランタイム」|「接続」の順にクリックし、新しいプロジェクトをデフォルトのランタイムプロジェクトとしてロードします。



プロジェクトファイルの暗号化を有効にして新しいプロジェクトを保存する場合（デフォルトではオン）、パスワードを設定する必要があります。パスワードを入力するか、「暗号化なし」(推奨されません)を選択し、「保存」をクリックします。「プロジェクトのプロパティ」|「一般」|「プロジェクトファイルの暗号化」で、パスワードを修正したり、プロジェクトの暗号化をオンまたはオフにできます。プロジェクトを保存せずに中止するには、「キャンセル」をクリックします。

● **ヒント:** パスワードの長さは少なくとも 14 文字で、大文字と小文字の両方、数字、および特殊文字を含めることをお勧めします。広く知られたパスワード、簡単に推測できるパスワード、一般的なパスワードは避けてください。パスワードを指定して暗号化ファイルとして保存したプロジェクトは、.SOPF ファイルとして保存されます。暗号化されたプロジェクトの場合、.JSON ファイルと .OPF ファイルはサポートされません。

● **注記:** OPC クライアントアプリケーションは、クライアントがデータを必要とするときに自動的に OPC サーバーを起動できます。ただし、この方法で起動される場合、実行するプロジェクトを OPC サーバーが認識している必要があります。サー

バーは、最後にロードまたは構成されたプロジェクトをロードします。サーバーがどのプロジェクトをロードするかを確認するには、「ファイル」にある「最近使用したファイル」リストを参照します。ロードされたプロジェクトが最初のプロジェクトファイルとしてリストに表示されます。

プロジェクトファイルはデフォルトで以下のディレクトリに保存されます。

64 ビット OS バージョンの場合、プロジェクトファイルは(デフォルトで)以下のディレクトリに保存されます。
C:\Users\<ユーザー名>\Documents\Kepware\KEPServerEX\V6

32 ビット OS バージョンの場合、プロジェクトファイルは(デフォルトで)以下のディレクトリに保存されます。
C:\Users\<ユーザー名>\Documents\Kepware\KEPServerEX\V6

プロジェクトのコピーが以下のディレクトリに自動的に保存されます。

64 ビット OS バージョンの場合、プロジェクトファイルは(デフォルトで)以下のディレクトリに保存されます。
C:\ProgramData\Kepware\KEPServerEX\V6

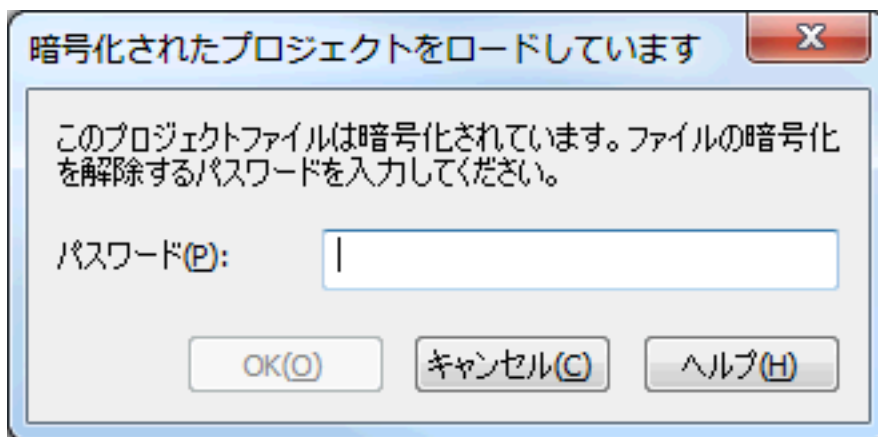
32 ビット OS バージョンの場合、プロジェクトファイルは(デフォルトで)以下のディレクトリに保存されます。
C:\ProgramData(x86)\Kepware\KEPServerEX\V6

● **ヒント:** 別の場所にファイルが保存されている場合は、*.opf、*.sopf、または*.json をサーチして、有効なプロジェクトファイルを見つけください。

● **関連項目:** 現在の環境でセキュリティを確保する必要がある場合は、[Secure KEPServerEX® Deployment](#) ガイドを参照してください。

暗号化されたプロジェクトを開く

プロジェクトファイルの暗号化が有効になっている状態で保存されたプロジェクトファイルを開く場合、パスワードの入力画面が表示されます。

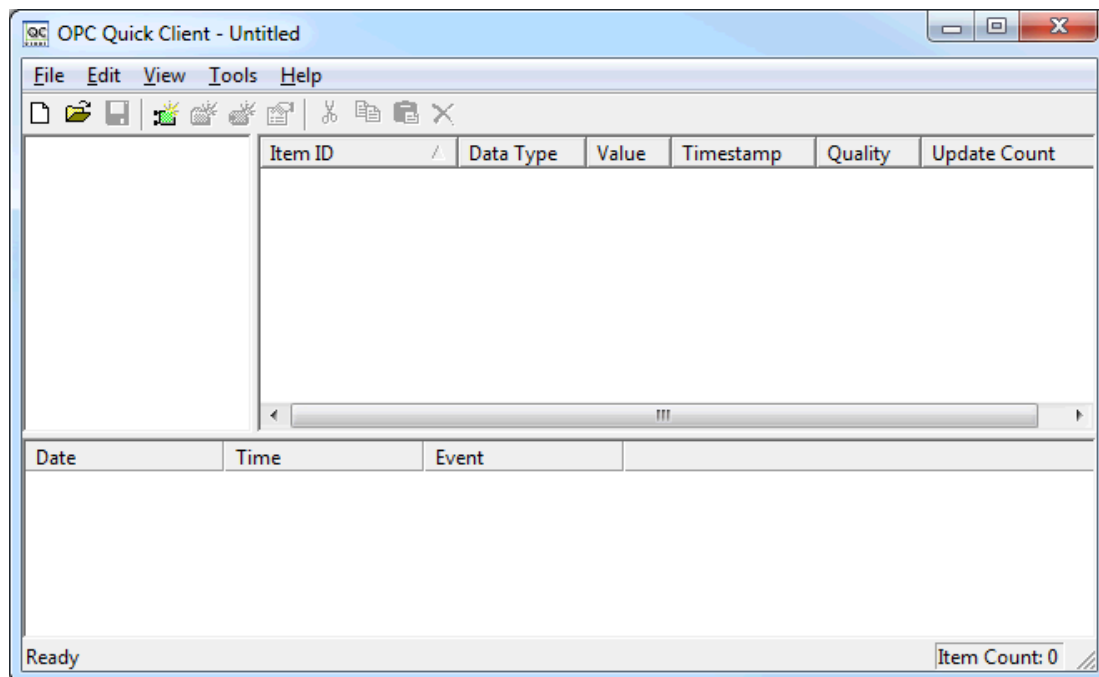


プロジェクトファイルの暗号化に使用するパスワードを入力し、「OK」をクリックします (または「キャンセル」をクリックしてファイルを開く操作を中止します)。

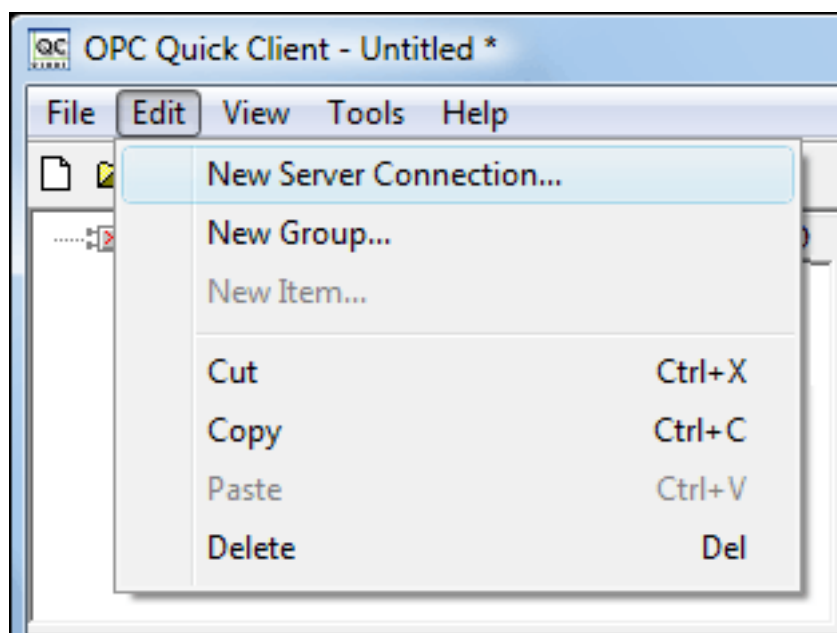
プロジェクトのテスト

サーバーには、OPC クライアントアプリケーションで使用できるすべての操作をサポートしているフル機能の OPC Quick Client が含まれています。Quick Client は、サーバーアプリケーションで使用できるすべてのデータにアクセスでき、データの読み取りと書き込み、構造体テストスイートの実行、およびサーバーのパフォーマンスのテストを行うために使用されます。また、サーバーによって返される OPC エラーについて詳細なフィードバックを提供します。

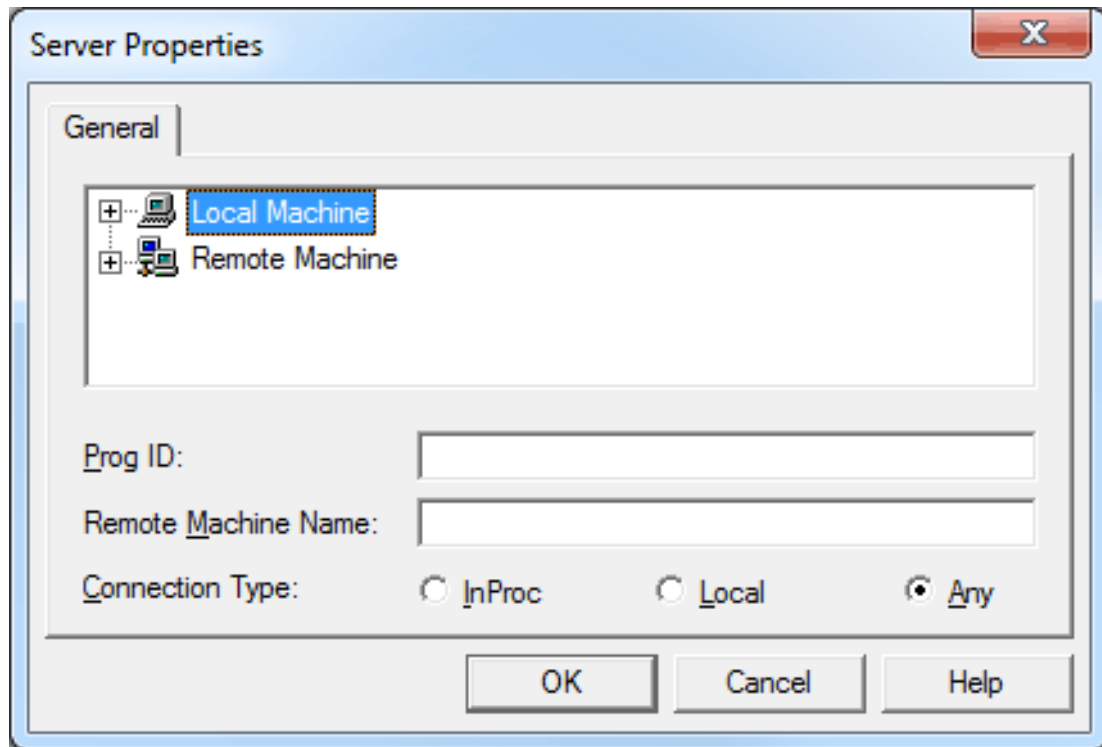
1. まず、サーバーと同じプログラムグループで OPC Quick Client プログラムを見つけます。次に、OPC Quick Client を実行します。



2. 「編集」|「New Server Connection」の順にクリックして接続を確立します。

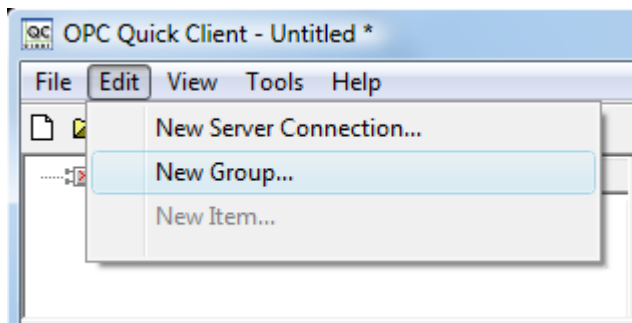


3. 「サーバーのプロパティ」で、DCOM を介したローカルまたはリモートでの OPC サーバーとの接続を作成します。このダイアログは、デフォルトでは、サーバーのプログラム ID (OPC クライアントが特定の OPC サーバーを参照するために使用する) を使用して事前構成されています。



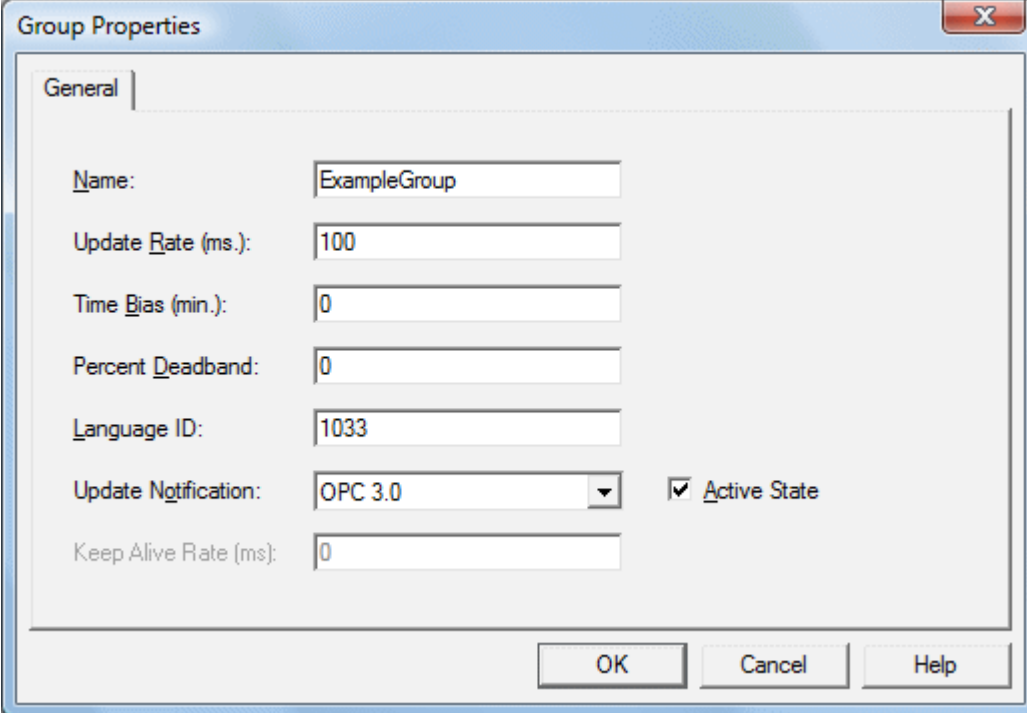
● **注記:** 接続を作成すると、以下の2つが発生します。サーバーが実行されている場合は、OPC Quick Client がサーバーに接続します。サーバーが実行されていない場合は、サーバーが自動的に起動します。

4. 接続にグループを追加します。これを行うには、サーバー接続を選択し、「編集」|「新しいグループ」の順にクリックします。



● **注記:** グループはサーバーからアクセスされるタグのコンテナとして機能し、これによりタグの更新方法を制御できます。すべての OPC クライアントがグループを使用して OPC サーバーデータにアクセスします。グループには多数のプロパティが関連付けられており、これによって OPC クライアントはデータがどのくらいの頻度でタグから読み取られるか、タグがアクティブかどうか、デッドバンドが適用されるかどうかなどを決定できます。OPC クライアントは、これらのプロパティを使用して、OPC サーバーの動作を制御します。グループのプロパティの詳細については、OPC Quick Client のヘルプドキュメントを参照してください。

5. この例を使用して説明を進めるため、次の図のようにグループのプロパティを編集します。



● **注記:**「更新レート」、「パーセントデッドバンド」、および「アクティブな状態」プロパティによって、グループのタグのデータが返されるかどうか、およびいつ返されるかが制御されます。プロパティの説明は次のとおりです。

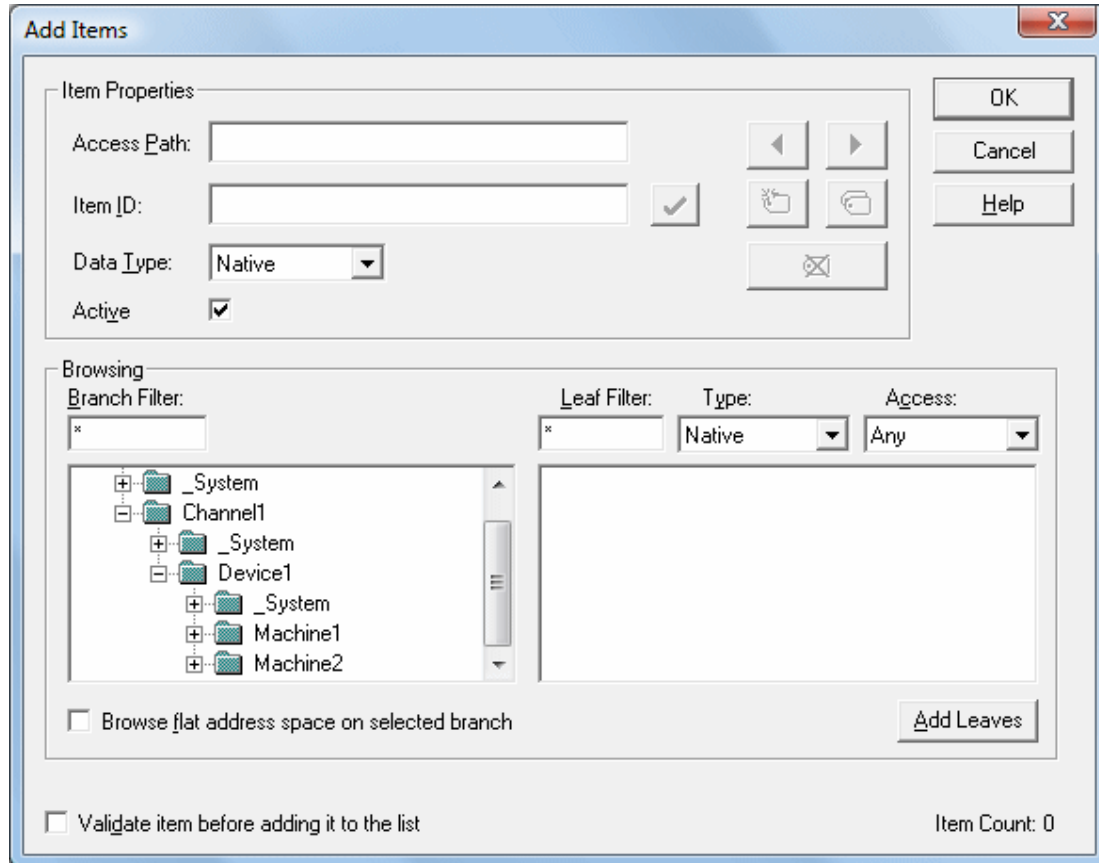
- ・ **「名前」:** このプロパティは、クライアントからの参照に使用され、実際には空白のままにできます。
- ・ **「更新レート」:** データがどのくらいの頻度で実際のデバイスからスキャンされるか、およびそのスキャンの結果としてデータがどのくらいの頻度で OPC クライアントに返されるかを指定します。
- ・ **「パーセントデッドバンド」:** このプロパティは、要求された進捗状況変更を超えた変更を検出することによってのみ、データ内のノイズコンテンツを除去または削減します。進捗状況変更は、指定されたタグのデータ型の要素です。
- ・ **「アクティブな状態」:** このプロパティでは、このグループ内のすべてのタグをオフまたはオフにします。

6. 完了後、「OK」をクリックします。

タグへのアクセス

OPC サーバータグは、グループに追加するまでアクセスできません。OPC データアクセス仕様では、タグブラウズインタフェースは、OPC クライアントが OPC サーバー内の使用可能なタグに直接アクセスして表示できるようにするものとして定義されています。OPC クライアントアプリケーションが OPC サーバーのタグ空間をブラウズできるようにすることで、タグが、クリックすると自動的にグループに追加されます。

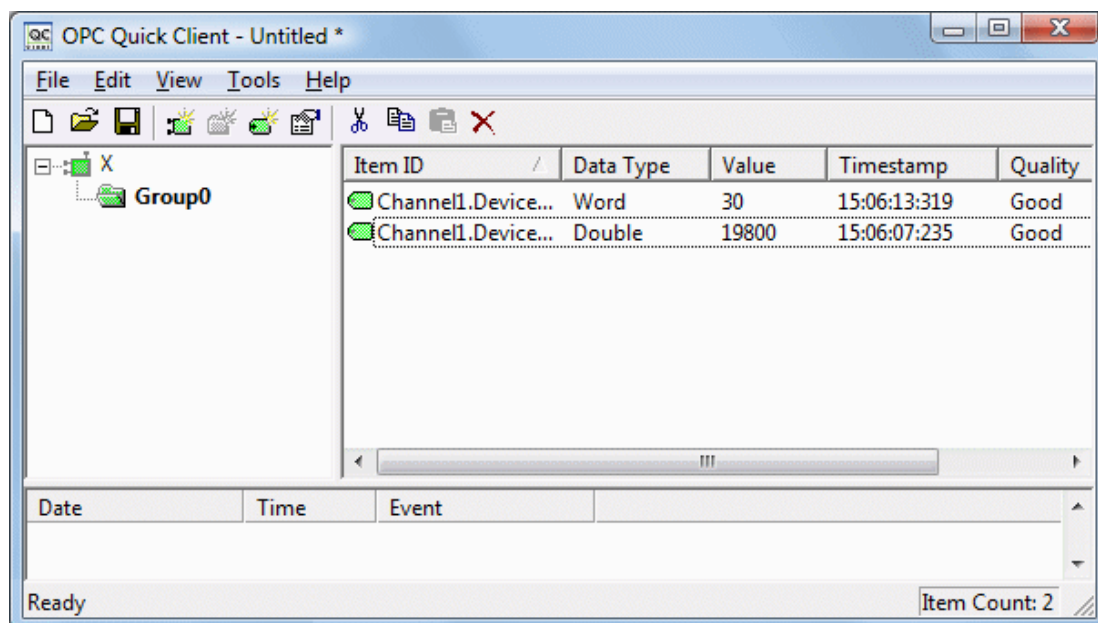
1. まず、タグが配置されるグループを選択します。「編集」|「新規アイテム」の順にクリックします。



● **注記:**「アイテムを追加」ダイアログにも「ブラウズ」セクションのツリービューが表示され、これを使用して OPC サーバーをブラウズし、サーバーで構成されているタグを見つけることができます。「Example1」プロジェクトを使用する場合、ビューのブランチを展開することによって、すでに定義されているタグにアクセスできます。

2. 上の図に示されているツリー階層の部分が表示されれば、タグ名をダブルクリックすることによって、そのタグを OPC グループに追加できます。タグがグループに追加されると、「アイテムを追加」ダイアログの下部にある「**アイテム数**」が増え、アイテムが追加されたことを示します。「MyFirstTag」と「MySecondTag」の両方が追加されると、アイテム数は 2 になります。
3. 完了後、「OK」をクリックします。

● **注記:**これで、定義された 2 つのタグを使用してサーバーからのデータにアクセスできるようになりました。

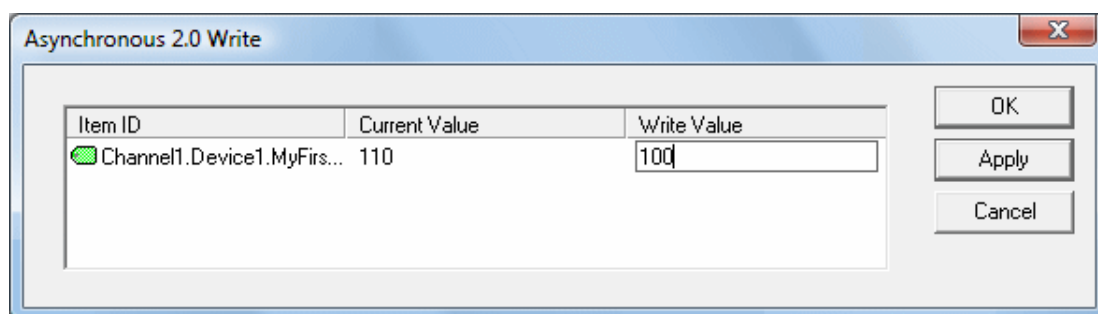


● **注記:** 最初のタグ "MyFirstTag" には変動値が含まれています。2 つ目のタグは、この時点では 0 です。OPC アイテムの読み取りのみをテストする必要がある場合、それはこれで終了です。OPC アイテムを変更する場合は、次に説明する書き込み方法のいずれかを使用して、OPC アイテムに新しいデータを送信できます。

OPC サーバーへのデータの書き込み

OPC Quick Client では、OPC サーバーにデータを書き込む方法として、同期書き込みと非同期書き込みの 2 つがサポートされています。同期書き込みは、OPC サーバーに対して書き込み操作を行い、それが完了するのを待ちます。非同期書き込みは、OPC サーバーに対して書き込みを行います、それが完了するのを待ちません。OPC アイテムにデータを書き込む際には、どちらの方法も選択できます。OPC クライアントアプリケーションの設計上の都合で、2 つの異なる書き込み方法が用意されています。

1. まず、アイテムを選択します。次に、それを右クリックし、「同期」または「非同期書き込み」を選択します。この例を使用して説明を進めるため、「MyFirstTag」を右クリックし、「非同期書き込み」を選択します。



● **注記:** 「非同期 2.0 書き込み」ダイアログが表示されますが、値は引き続き更新されます。

2. このアイテムの新しい値を入力するには、「値を書き込む」をクリックし、別の値を入力します。
3. 「適用」をクリックしてデータを書き込みます。これにより、引き続き新しい値を書き込むことができます。「OK」をクリックすると、新しい値が書き込まれてからダイアログが閉じます。
4. 「OK」をクリックします。

● **注記:** 新しいデータを入力しなかった場合は、「OK」をクリックしてもデータはサーバーに送信されません。

まとめ

ここでは、OPC プロジェクトの構築とテストに関連する基本的なステップのすべてについて説明しました。サーバーと OPC Quick Client のさまざまな機能をより深く理解し、把握するために、引き続きテストを行うことをお勧めします。OPC Quick Client の詳細については、そのヘルプドキュメントを参照してください。

これで OPC アプリケーションの開発を開始できるようになりました。Visual Basic を使用している場合は、付属のプロジェクト例を参照してください。これら 2 つのプロジェクトは、Visual Basic アプリケーションで直接 OPC テクノロジーを使用する方法の簡潔な例と複雑な例を示しています。

操作方法

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。

[デスクトップとの対話を許可する方法](#)

[エイリアスの作成および使用](#)

[サーバープロジェクトの最適化](#)

[プロセス配列データ](#)

[チャネル、デバイス、タグ、およびタググループに適切な名前を付ける方法](#)

[サーバーに接続されている DNS/DHCP デバイスの電源を切断してから再投入したときによく発生する問題を解決する方法](#)

[適切なネットワークケーブルの選択](#)

[エイリアスを使用してプロジェクトを最適化する方法](#)

[サーバーで DDE を使用する方法](#)

[動的タグアドレス指定を使用する方法](#)

[イーサネットカプセル化の使用法](#)

[非正規化浮動小数点値を使用する方法](#)

デスクトップ対話の許可

一部の通信インターフェースは、サーバーがデスクトップと対話することが許可されていることを要件としています。たとえば、Windows メッセージングレイヤーは DDE および FastDDE によって使用されます。デスクトップとの通信方法を選択するときにオペレーティングシステムを考慮に入れることは重要です。

Windows Vista、Windows Server 2008、およびそれ以降のオペレーティングシステム

Windows Vista、Windows Server 2008、およびそれ以降のオペレーティングシステムでは、サービスは、コンソールにログオンしているユーザーがアクセスできない隔離されたセッションで実行されます。これらのオペレーティングシステムを使用する場合、プロセスモードが「対話型」に設定されている必要があります。これにより、ランタイムを現在のユーザーと同じユーザーアカウントで実行できます。プロセスモードを変更する方法については、[設定 - ランタイムプロセス](#)を参照してください。

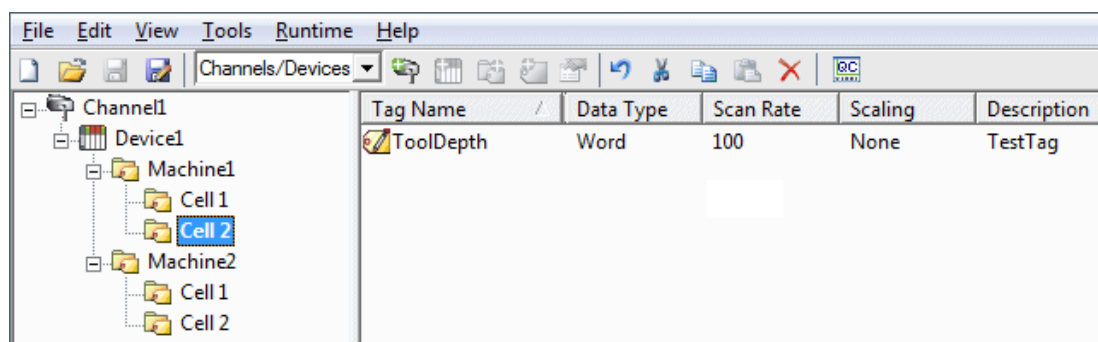
● 関連項目:

[管理メニューへのアクセス](#)

エイリアスの作成および使用

複雑なタグ参照の例

次の図は、サーバー内の複雑なタグ参照を示しています。



たとえば、「ToolDepth」タグのアプリケーションへの DDE リンクを作成するには、DDE リンクを "<DDE サービス名>|ddedata|Channel1.Device1.Machine1.Cell2.ToolDepth" と入力する必要があります。

DDE リンクの <アプリケーション>|<トピック>|<アイテム> フォーマットはまだ存在しますが、オプションのタググループとチャネル名がトピックの一部として必要であれば、コンテンツはより複雑になります。エイリアスマップを使用することにより、DDE クライアントアプリケーションで短いバージョンの参照を使用できます。

● 詳細については、[エイリアスマップとは](#)を参照してください。

複雑なアドレスパスのエイリアスの作成

エイリアスを作成して複雑なタグアドレスパスを簡略化するには、次の手順に従います。

- ツリービューで、編集するエイリアスを選択し、ダブルクリックしてそのエイリアスノードを開きます。
- 詳細ビューで、右クリックして「**新規エイリアス**」を選択します (または「**編集**」|「**エイリアス**」|「**新しいエイリアス**」の順に選択します)。

Alias Name	Mapped To	Scan Rate
AdvancedTags	_AdvancedTags	0
Channel1_CommunicationSerialization	Channel1_CommunicationSerialization	0
Channel1_Statistics	Channel1_Statistics	0
Channel1_System	Channel1_System	0
Channel1_Device1	Channel1.Device1	0
Channel1_Device1_Statistics	Channel1.Device1_Statistics	0
Channel1_Device1_System	Channel1.Device1_System	0
Channel2_Statistics	Channel2_Statistics	0
Channel2_System	Channel2_System	0
Channel2_Device1	Channel2.Device1	0
Channel2_Device1_Statistics	Channel2.Device1_Statistics	0
Channel2_Device1_System	Channel2.Device1_System	0
Channel4_Statistics	Channel4_Statistics	0
Channel4_System	Channel4_System	0
Channel4_Device1	Channel4.Device1	0
Channel4_Device1_Statistics	Channel4.Device1_Statistics	0
Channel4_Device1_System	Channel4.Device1_System	0
Channel5_Statistics	Channel5_Statistics	0
Channel5_System	Channel5_System	0
Channel5_Device1	Channel5.Device1	0
Channel5_Device1_Statistics	Channel5.Device1_Statistics	0
Channel5_Device1_System	Channel5.Device1_System	0
Channel6_CommunicationSerialization	Channel6_CommunicationSerialization	0
Channel6_Statistics	Channel6_Statistics	0

- 参照されるアイテムを含んでいるグループまたはデバイスをブラウズします。

プロパティグループ	識別
一般	名前
	説明
	エイリアスのプロパティ
	マッピング先
	スキャン速度オーバーライド (ミリ秒)

- 複雑なタグ参照を表すエイリアス名を入力します。このエイリアス名をクライアントアプリケーションで使用してサーバー内にあるタグのアドレスを指定できます。予約文字については、[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。
- 複雑なトピックおよびアイテム名 "_ddedata! Channel1.Device1.Machine1.Cell2" をエイリアス "Mac1Cell2" に置き換えることができます。これを上の例に適用すると、アプリケーション内の DDE リンクは "<DDE サービス名>|Mac1Cell2!ToolDepth" と入力できます。

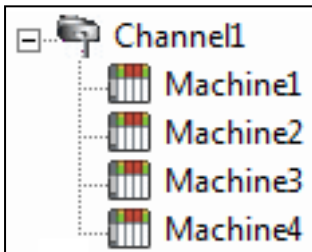
● **注記:** チャンネルと同じ名前のエイリアスを作成することは可能ですが、お勧めしません。クライアントのアイテムが同じ名前を使用して動的アドレスを参照している場合はエラーが発生します。たとえば、"Channel1" というエイリアスが "Channel1.Device1" にマッピングされている場合、"Channel1.Device1.<アドレス>" を参照しているクライアント内のアイテムは無効です。クライアントの参照が正常に行われるようにするには、このエイリアスを除去または名前変更する必要があります。

● **関連項目:** [エイリアスのプロパティ](#)

サーバープロジェクトの最適化

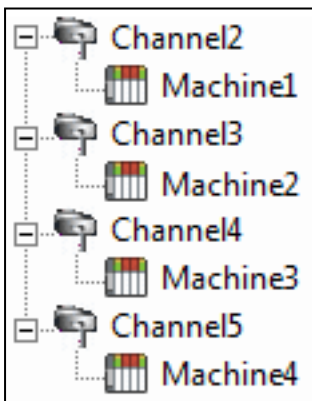
このサーバーのほぼすべてのドライバーが少なくとも 100 個のチャンネルをサポートしています (つまり、イーサネット通信の場合は、少なくとも 100 個の COM/シリアルポートまたはソースソケット)。各デバイスで使用できる、サポートされているチャンネルの数を確認するには、[サーバーのサマリー情報](#) の「ドライバー情報」を参照してください。

このサーバーは、通信プロトコルをチャンネルとして参照します。アプリケーションで定義されている各チャンネルは、サーバーでの個々の実行パスを表します。チャンネルが定義された後、そのチャンネルの下に一連のデバイスを定義する必要があります。これらのデバイスはそれぞれ、データが収集される単一のデバイスを表しています。このアプリケーションを定義するアプローチでは、高いレベルのパフォーマンスが達成されますが、ドライバーまたはネットワークが最大限に活用されません。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。



デバイスそれぞれが単一のチャンネルの下に表示されます。この構成では、ドライバーは効果的な速度で情報を収集するために、できるだけ速やかにあるデバイスから次のデバイスに移動する必要があります。さらにデバイスが追加されたり、1 つのデバイスからより多くの情報が要求されたりするにしがたい、全体的な更新レートが低下していきます。

ドライバーが単一のチャンネルのみを定義できる場合は、上に示した例が唯一使用できるオプションになります。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。



ここではそれぞれのデバイスが各自のチャンネルの下に定義されています。この新しい構成では、各デバイスからのデータ収集タスクごとに 1 つの実行パスが割り当てられます。アプリケーションのデバイスの数がより少ない場合、まさにここで示したように最適化できます。

アプリケーションのデバイス数がチャンネル数を超える場合でもパフォーマンスは改善されます。チャンネルごとに 1 つのデバイスを配置することが理想的ですが、アプリケーションの性能はチャンネルの追加によって向上します。デバイスの負荷をすべてのチャンネルに分散してもサーバーはデバイスを切り替えますが、単一のチャンネルで処理するデバイスの数ははるかに少なくなります。

● このプロセスは、1 つのイーサネットデバイスに対して複数の接続を確立するためにも使用できます。OPC サーバーでは、ほとんどのデバイスに 100 個のチャンネルを使用できますが、許可される接続の数は最終的にはデバイスによって決定されます。この制約は、ほとんどのデバイスでサポートされている接続の数が制限されているという事実に基づいています。デバイスに対して確立される接続が多いほど、それぞれの接続で要求が処理される時間は短くなります。これは、接続が追加されると、パフォーマンスに対しては反比例のトレードオフが提供される可能性があることを意味します。

プロセス配列データ

このサーバーで使用できる多くのドライバーでは、クライアントが配列フォーマットのデータにアクセスすることが可能になっています。配列を利用すると、クライアントアプリケーションから、一連の連続的な個別データを 1 回のリクエストで要求できます。配列は、それ自体が独自のデータ型です。Word データ型および DWord データ型を組み合わせる配列を使用することはできません。また、配列には 1 回のトランザクションで書き込みが実行されます。サーバーで配列を使用するには、少なくとも配列データの読み取りをクライアントアプリケーションがサポートしている必要があります。

DDE クライアントでの配列データの処理

配列データをクライアントが使用できるのは、クリップボードのフォーマットとして CF_TEXT または Advanced DDE を使用している場合のみです。

Advanced DDE を使用しているクライアントアプリケーションの場合、配列の要素の数は、SPACKDDE_DATAHDR_TAG 構造体で指定します。このプロトコルでは、1 次元配列のみがサポートされています。サーバーに配列データを送出するときは、この構造体を使用してください。

CF_TEXT を使用しているクライアントの場合は、1 次元または 2 次元の配列がサポートされます。行内の各データはタブ (0x09) 文字で区切り、それぞれの行は、CR (0x0d) 文字および LF (0x0a) 文字で終端します。クライアントからデータ値の配列を送出する場合、書き込まれるテキスト文字列は、この区切りフォーマットで記述されている必要があります。

どちらのフォーマットで Array タグに送出的場合も、配列全体が記述されている必要はありませんが、開始位置は固定です。配列として宣言されていないデータを配列フォーマットでタグに送出した場合は、配列の最初の値しか書き込まれません。タグの配列サイズを上回る量のデータを送出した場合、書き込まれるのは、タグの配列サイズ分のデータのみになります。一部のデータ値を空白にしたままデータを送出した場合、サーバーは、デバイスへの書き戻しの際に、直近の既知の値を当該の配列要素に使用します。当該のレジスタの値が変更されていても、サーバー内で更新されていない場合は、古い値で上書きされます。したがって、データを配列に書き込む際は慎重を期してください。

OPC クライアントでの配列データの処理

配列をサポートしている OPC クライアントでは、OPC アイテムのデータ値は、実際にはバリエーションデータ型の配列です。配列の要素データを解析するのは、OPC クライアントです。クライアントによっては、表示を目的としてサブタグを作成することがあります。たとえば、OPC クライアントがデータベースに 'Process' という名前で作成し、関連する OPC アイテムが 5 要素の 1 次元配列であった場合は、'Process_1' や 'Process2' といった 5 つのタグが作成されます。データをコンマ区切り値 (CSV) として表示するクライアントもあります (OPC Quick Client など)。

チャンネル、デバイス、タグ、およびタググループの適切な名前の指定

チャンネル、デバイス、タグ、またはタググループに名前を付けるとき、以下の文字は予約または制限されているため使用できません。

- ピリオド
- 二重引用符
- 先頭のアンダースコア
- 先頭または末尾のスペース

● **注記:** 制限されている文字の一部は、特定の状況では使用できます。詳細については、以下のリストを参照してください。

1. ピリオドは、エイリアス名で元のチャンネル名とデバイス名を区切るために使用されます。たとえば、有効な名前は "Channel1.Device1" です。
2. アンダースコアは、先頭の文字の後では使用できます。たとえば、有効な名前は "Tag_1" です。
3. スペースは、先頭の文字の後および末尾の文字の前では使用できます。たとえば、有効な名前は "Tag 1" です。

サーバーに接続された DNS/DHCP デバイスで電源サイクルを行ったときの通信の問題の解決

特定のドライバーでは接続性のために DNS/DHCP による解決がサポートされています。これにより、識別することを目的として、一意のドメイン/ネットワーク名を割り当てることができます。ネットワークを起動して接続するとき、デバイスはネットワーク DNS サーバーから IP アドレスを要求します。この接続性のためにドメイン名を IP アドレスに解決するプロセスには時間がかかります。その速度を上げるため、オペレーティングシステムは解決されたすべての IP/ドメイン名をキャッシュして再利用します。解決された名前は、デフォルトでは 2 時間キャッシュに保持されます。

● デバイスのドメイン/ネットワークに関連付けられている IP アドレスの名前が変更されると、サーバーはデバイスへの再接続に失敗します。この変更が、デバイスの電源を切断してから再投入したことによる結果である場合、それは新しい IP を取得します。この変更が、デバイスで IP を手動で変更したことによる結果である場合もあります。どちらの場合も、使用されていた IP アドレスが存在しなくなります。

サーバーが自動的に30秒間隔でキャッシュをフラッシュするため、IPは強制的に解決されます。これで問題が解決しない場合は、PCのコマンドプロンプトに"ipconfig / flushdns"と入力して手動でキャッシュをフラッシュできます。

● 詳細については、Microsoft サポートの記事 [クライアント側のDNS キャッシュを無効にする方法](#)を参照してください。

適切なネットワークケーブルの選択

イーサネット対応デバイスの取り扱い経験、またはイーサネットコンバータへのシリアル接続の経験がないユーザーは、適切なネットワークケーブルを選択する際に、判断に迷う場合があります。適切なケーブル構成を判断するには、一般に2つの方法があります。ネットワークハブまたはスイッチを経由してデバイスまたはコンバータ接続する場合は、**パッチケーブル**が必要です。パッチケーブルは、電話交換手が利用していたのと同じ形状の基板を使用して、デバイスを互いに一時接続または常時接続していた時代に命名されたものです。これに対して、PCからデバイスに直接接続する場合は、**クロスオーバーケーブル**が必要です。どちらのケーブルも、電器店またはPC サプライ用品店で購入できます。

エイリアスを使用したプロジェクトの最適化

プロジェクトのパフォーマンスを最適化するために、各デバイスをそれぞれのチャンネルに配置することをお勧めします。プロジェクトを作成した後に、その通信方法を最適化する場合、新しいアイテム名を参照するようにクライアントアプリケーションを変更することが困難な場合があります。エイリアスマップを使用することによって、クライアントが新しい構成に対してレガシー要求を行うことができるようになります。まず、次の手順に従います。

1. まず、デバイスごとに新しいチャンネルを作成します。デバイスを新しいチャンネルに配置し、元のチャンネルを削除します。
2. ツリービューの「エイリアス」の下で、**エイリアスマップ**内の各デバイスに**新規エイリアス**を作成します。元のチャンネルとデバイス名がピリオドで区切られたものがエイリアス名になります ("Channel1.Device1" など)。

● 予約文字については、[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。

Alias Name	Mapped To	Scan Rate
AdvancedTags	_AdvancedTags	0
Channel1_CommunicationSerialization	Channel1_CommunicationSerialization	0
Channel1_Statistics	Channel1_Statistics	0
Channel1_System	Channel1_System	0
Channel1_Device1	Channel1.Device1	0
Channel1_Device1_Statistics	Channel1.Device1_Statistics	0
Channel1_Device1_System	Channel1.Device1_System	0
Channel2_Statistics	Channel2_Statistics	0
Channel2_System	Channel2_System	0
Channel2_Device1	Channel2.Device1	0
Channel2_Device1_Statistics	Channel2.Device1_Statistics	0
Channel2_Device1_System	Channel2.Device1_System	0
Channel4_Statistics	Channel4_Statistics	0
Channel4_System	Channel4_System	0
Channel4_Device1	Channel4.Device1	0
Channel4_Device1_Statistics	Channel4.Device1_Statistics	0
Channel4_Device1_System	Channel4.Device1_System	0
Channel5_Statistics	Channel5_Statistics	0
Channel5_System	Channel5_System	0
Channel5_Device1	Channel5.Device1	0
Channel5_Device1_Statistics	Channel5.Device1_Statistics	0
Channel5_Device1_System	Channel5.Device1_System	0
Channel6_CommunicationSerialization	Channel6_CommunicationSerialization	0
Channel6_Statistics	Channel6_Statistics	0

● **注記:** サーバーは、まずアイテムの要求をエイリアスマップに照らし合わせて検証し、アイテムが存在しない場合は、それを示すエラーとともにクライアントアプリケーションに回答を返します。

サーバーでの DDE の使用

アプリケーションでの DDE の使用

動的データ交換 (DDE) は、Windows オペレーティングシステムで実行されているアプリケーション間でデータを交換する方法を提供する Microsoft 通信プロトコルです。DDE クライアントプログラムは、DDE サーバーアプリケーションへのチャンネルを開き、アプリケーション (サービス) 名、トピック名、およびアイテム名の階層を使用してアイテムデータを要求します。

● DDE クライアントがサーバーインターフェースに接続するには、ランタイムがデスクトップと対話することが許可されている必要があります。

● 詳細については、[デスクトップとの対話を許可する方法](#)を参照してください。

例 1: ローカルでのレジスタへのアクセス (デフォルトのトピックを使用)

構文は <アプリケーション>|<トピック>|<アイテム> です。説明は次のとおりです。

- <アプリケーション> DDE サービス名
- <トピック> _ddedata*
- <アイテム> Modbus.PLC1.40001

* これは、エイリアスマップのエントリを使用しないすべての DDE データのデフォルトのトピックです。

● **注記:** 構文の例は "MyDDE|_ddedata!Modbus.PLC1.40001" です。

例 2: ローカルでのレジスタへのアクセス (エイリアス名をトピックとして使用)

構文は <アプリケーション>|<トピック>|<アイテム> です。説明は次のとおりです。

- <アプリケーション> DDE サービス名
- <トピック> ModPLC1*
- <アイテム> 40001

* これは、エイリアスマップのエントリを使用するトピックです。

● **注記:** 構文の例は "MyDDE|ModPLC1|40001" です。その他の考えられる構文については、DDE クライアントの特定のヘルプドキュメントを参照してください。

● 関連項目:

[プロジェクトのプロパティ - DDE](#)

[プロジェクトのプロパティ - FastDDE & SuiteLink](#)

[エイリアスマップとは](#)

動的タグアドレス指定の使用

このサーバーは、サーバーから物理デバイスのデータアドレスを動的に参照するために使用することもできます。このサーバーは、要求されたアイテムのタグを動的に作成します。1 つのクライアントによって動的に追加されたタグを別のクライアントからブラウズすることはできません。タグを動的に追加する前に、以下に注意する必要があります。

- 正しいデータアドレスの構文が使用される必要があります。特定のドライバーの構文の詳細については、そのヘルプドキュメントを参照してください。
- 要求されたアイテムのデータ型が指定されていない場合、それはアプリケーションによってデフォルト設定に設定されます。特定のドライバーでサポートされているデータ型の詳細については、そのヘルプドキュメントを参照してください。

● **注記:** 次の例では、Simulator Driver が "Channel1" というチャネル名と "Device1" というデバイス名で使用されています。

例 1: OPC 以外のクライアントでの動的タグアドレス指定の使用

シミュレートされているデバイス内のレジスタ "K0001" からデータを取得するには、"Channel1.Device1.K001" というアイテム ID を使用します。このレジスタのデフォルトのデータ型は Short です。OPC 以外のクライアントはサーバーに対して更新レートを指定しないため、動的タグのデフォルトの更新レートは 100 ミリ秒です。データ型と更新レートは、動的要求が送信された後にオーバーライドできます。

タグのデフォルトをオーバーライドするには、アイテムの末尾に単価記号 ("@") を追加します。レジスタを DWord (符号なし 32 ビット) データ型として追加する場合は、"Channel1.Device1.K0001@DWord" というアイテム ID を使用します。デフォルトの更新レートを 1000 ミリ秒に変更するには、"Channel1.Device1.K0001@1000" を使用します。両方のデフォルトを変更するには、"Channel1.Device1.K0001@DWord,1000" を使用します。

● **注記:** クライアントアプリケーションのアドレス空間で "@" などの特殊文字を使用できる必要があります。

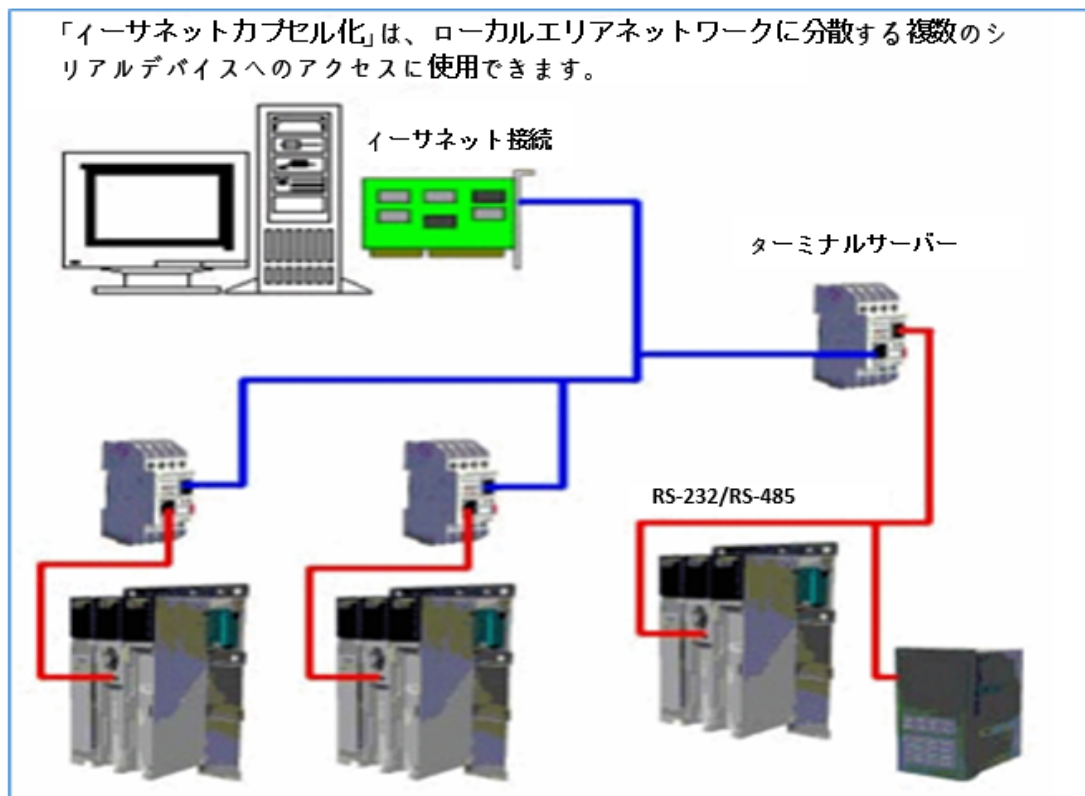
例 2: OPC クライアントでの動的タグアドレス指定の使用

OPC クライアントでは、OPC アイテムが追加されるときにデータ型を指定する方法がクライアントアプリケーションにない場合には、同じ構文を使用してデータ型をオーバーライドできます。アイテムの更新レートは OPC では使用されないため、それをオーバーライドする必要はありません。

● **注記:** クライアントアプリケーションのアドレス空間で "@" などの特殊文字を使用できる必要があります。

イーサネットカプセル化の使用

イーサネットカプセル化モードは、イーサネットネットワーク上のターミナルサーバーに接続されているシリアルデバイスとの通信に設計されています。ターミナルサーバーは基本的には仮想のシリアルポートであり、イーサネットネットワーク上の TCP/IP メッセージをシリアルデータに変換します。メッセージがシリアル形式に変換されると、シリアル通信をサポートする標準デバイスをターミナルサーバーに接続できるようになります。次の図に、イーサネットカプセル化モードの使用方法を示します。



● **注記:** イーサネットカプセル化をサポートする非送信請求ドライバーでは、ユーザーがチャンネルレベルでポートおよびプロトコル設定を構成する必要があります。これにより、指定されたポートにドライバーをバインドし、複数のデバイスからの受信要求を処理できます。チャンネルはすべてのデバイスからの受信要求を受け入れるため、IP アドレスはチャンネルに入力されません。

イーサネットカプセル化は、ワイヤレスネットワーク接続 (802.11b および CDPD パケットネットワークなど) で使用でき、幅広いシリアルデバイスをサポートするために開発されました。ターミナルサーバーデバイスを使用することで、RS-232 および RS-485 デバイスをプラントオペレーション全体に配置しながら、1 つのローカライズ PC にマウントデバイスへのリモートアクセスを許可できます。さらに、イーサネットカプセル化モードでは、必要に応じて個々のネットワーク IP アドレスを各デバイスに割り当てできます。複数のターミナルサーバーを使用しながら、単一の PC から何百ものシリアルデバイスにアクセスできます。

イーサネットカプセル化モードの設定

イーサネットカプセル化モードを有効にするには、「チャンネルのプロパティ」を表示し、「シリアル通信」グループを選択します。「接続タイプ」ドロップダウンメニューの「イーサネットカプセル化」を選択します。

● **注記:** このオプションを選択できるのは、イーサネットカプセル化をサポートするドライバーだけです。

プロパティグループ	<input type="checkbox"/> 接続タイプ <input checked="" type="checkbox"/> 物理メディア <input type="checkbox"/> イーサネット設定 <input type="checkbox"/> ネットワークアダプタ	
一般	イーサネットカプセル化	
シリアル通信	なし	
書き込み最適化	COM ポート	
詳細	モデム	
通信シリアル化	イーサネットカプセル化	

● **注記:** サーバーによる複数チャンネルのサポートにより、ドライバープロトコルごとに最大で 16 のチャンネルを許可できます。これにより、1 つのチャンネルでローカル PC シリアルポートを、別のチャンネルでイーサネットカプセル化モードを使用するように指定できます。

● イーサネットカプセル化モードを選択した場合、シリアルポート設定 (ボーレート、データビット、パリティなど) は使用できません。チャンネルをイーサネットカプセル化モード用に設定した後、イーサネット操作用のデバイスを設定する必要があります。

ます。新しいデバイスがチャンネルに追加されたら、イーサネットカプセル化の設定を使用して、イーサネット IP アドレス、イーサネットポート番号、およびイーサネットプロトコルを選択できます。

● **注記:** 使用しているターミナルサーバーでは、そのターミナルサーバーに接続するシリアルデバイスの要件に一致するようにシリアルポートを設定する必要があります。

非正規化浮動小数点値の使用

非正規化浮動小数点値は、無限大、非数 (NaN)、または非正規化数として定義されます。詳細については、以下の表を参照してください。

用語	定義
非正規化浮動小数点値	以下のいずれかに当てはまる IEEE-754 浮動小数点数。 <ul style="list-style-type: none"> 負の無限大～負のクワイエット NaN。 正の無限大～正のクワイエット NaN。 負の非正規化値。 正の非正規化値。
NaN	浮動小数点として表すことができる範囲の外側にある数。NaN 表現にはクワイエットとシグナリングの 2 つのタイプがあります。*
非正規化数	Float (単精度) または Double (倍精度) で表すことができる、絶対値が最小 IEEE 754-2008 値の絶対値よりも小さい 0 以外の浮動小数点数。 <ul style="list-style-type: none"> Float (単精度) の場合は、-1.175494E-38 と -1.401298E-45 の間の数 (負の非正規化) および 1.401298E-45 と 1.175494E-38 の間の数 (正の非正規化) がこれに該当します。 Double (倍精度) の場合は、-2.225074E-308 と -4.940657E-324 の間の数 (負の非正規化) および 4.940657E-324 と 2.225074E-308 の間の数 (正の非正規化) がこれに該当します。

* Float (単精度) または Double (倍精度) データ型が使用される場合、シグナリング NaN 範囲内の浮動小数点値は、クライアントに転送される前にクワイエット NaN に変換されます。この変換を回避するには、単一要素の浮動小数点配列を使用します。

非正規化 IEEE-754 浮動小数点値の処理

チャンネルのプロパティ - アドバンスにある「非正規化値の処理」プロパティを使用して、ドライバーが非正規化 IEEE-754 浮動小数点値を処理する方法を指定できます。「未修正」が選択されていると、すべての値が修正されることなくクライアントに転送されます。たとえば、32 ビット浮動小数点値 0xFF800000 (負の無限大) を読み取るドライバーは、この値を "そのまま" クライアントに転送します。「ゼロで置換」が選択されていると、特定の値がクライアントに転送される前に 0 に置き換えられます。たとえば、32 ビット浮動小数点値 0xFF800000 (負の無限大) を読み取るドライバーは、この値をクライアントに転送する前に 0 に置き換えます。

● **注記:** クライアントに転送される前に 0 に置き換えられる値については、次の表を参照してください。

32 ビット浮動小数点値の IEEE-754 範囲

名前	16 進範囲	10 進範囲
負のクワイエット NaN	0xFFFFFFFF ~ 0xFFC00001	N/A
正のクワイエット NaN	0x7FC00000 ~ 7FFFFFFF	N/A
不定	0xFFC00000	N/A
負のシグナリング NaN	0xFFBFFFFFF ~ 0xFF800001	N/A
正のシグナリング NaN	0x7F800001 ~ 7FBFFFFFF	N/A
負の無限大 (負のオーバーフロー)	0xFF800000	-3.4028235677973365E+38 以下

正の無限大 (正のオーバーフロー)	0x7F800000	3.4028235677973365E + 38 以上
負の正規化 -1.m × 2(e-127)	0xFF7FFFFFFF ~ 0x80800000	-3.4028234663852886E+38 ~ -1.1754943508222875E-38
負の非正規化 -0.m × 2(-126)	0x807FFFFFFF ~ 0x80000001	-1.1754942106924411E-38 ~ -1.4012984643248170E-45(-7.0064923216240862E-46)
正の非正規化 0.m × 2(-126)	0x00000001 ~ 0x007FFFFFFF	(7.0064923216240862E-46) * 1.4012984643248170E-45 ~ 1.1754942106924411E-38
正の正規化 1.m × 2(e-127)	0x00800000 ~ 0x7F7FFFFFFF	1.1754943508222875E-38 ~ 3.4028234663852886E+38

64 ビット浮動小数点値の IEEE-754 範囲

名前	16 進範囲	10 進範囲
負のクワイエット NaN	0xFFFFFFFFFFFFFFFF ~ 0xFF80000000000001	N/A
正のクワイエット NaN	0x7FF8000000000000 ~ 0x7FFFFFFFFFFFFFFF	N/A
不定	0xFF80000000000000	N/A
負のシグナリング NaN	0xFFF7FFFFFFFFFFFFFF ~ 0xFF00000000000001	N/A
正のシグナリング NaN	0x7FF0000000000001 ~ 0x7FF7FFFFFFFFFFFFFF	N/A
負の無限大 (負のオーバーフロー)	0xFF00000000000000	-1.7976931348623158E+308 以下
正の無限大 (正のオーバーフロー)	0x7FF0000000000000	1.7976931348623158E+308 以上
負の正規化 -1.m × 2(e-1023)	0xFFEFFFFFFFFFFFFFFF ~ 0x8010000000000000	-1.7976931348623157E+308 ~ -2.2250738585072014E-308
負の非正規化 -0.m × 2(-1022)	0x800FFFFFFFFFFFFFFF ~ 0x8000000000000001	-2.2250738585072010E-308 ~ -4.9406564584124654E-324 (-2.4703282292062328E-324)
正の非正規化 0.m × 2(-1022)	0x0000000000000001 ~ 0x000FFFFFFFFFFFFFFF	(2.4703282292062328E-324) * 4.9406564584124654E-324 ~ 2.2250738585072010E-308
正の正規化 1.m × 2(e-1023)	0x0010000000000000 から 0x7FEFFFFFFFFFFFFFFF	2.2250738585072014E-308 から 1.7976931348623157E+308

デバイス要求ポール

デバイス要求ポールは、クライアントアプリケーションからのデバイスのポーリングを完全に制御する必要がある顧客にとって便利な機能です。これは、著しい通信の遅延が発生することのある石油/ガス、上水/下水、電気などの SCADA 業界にとっては特に便利です。

クライアント側 SCADA システムの多くで、スキャン速度を構成できなかったり、スキャン速度の最小値が、SCADA オペレータが必要とするデータ更新には長すぎたりします。サーバーで利用できるデバイス要求ポールタグへの書き込みを SCADA システムで行うことにより、この制限を回避できます。このシナリオでは、サーバー内の各デバイスが DemandPoll タグを公開します。これによって、デバイス上の参照されているすべてのタグがクライアントによって書き込ま

れるときにポーリングされます。ポーリング中、_DemandPoll タグは True (1) になります。これは、読み取り要求が完了したことを最後のアクティブなタグが示すと False (0) を返します。タグの値が False に戻るまで、後続の _DemandPoll タグへの書き込みは失敗します。要求ポーリングは、チャンネルの読み取り/書き込み使用率を考慮します。_DemandPoll タグへの書き込みを行ってポーリングを発生させるクライアント側 SCADA スクリプト (「再表示」ボタンスクリプトなど) を開発できます。ポーリング結果はクライアントアプリケーションに渡されます。詳細については、[システムタグ](#)を参照してください。

● **注記:** 上記の手順は OPC 準拠の動作ではありません。これが問題である場合は、通信を 2 つのデバイスに分離することをお勧めします。1 台のデバイスでは従来の OPC 更新間隔を使用でき、もう 1 台のデバイスでは「スキャンモード」を「スキャンしない、要求ポーリングのみ」に設定して、_DemandPoll タグに書き込まれているときにのみポーリングを行うことができます。

デバイス要求ポーリングが使用されているかどうかに関係なく、タグのスキャン速度によって制限されているクライアントでは、サーバーが OPC クライアントのグループ更新レートに準拠しているため、オペレータが待機する時間が発生する可能性があります。この OPC 準拠の動作を回避するには、「グループ更新レートを無視し、データを使用できるようになるとすぐに返す」設定を構成します。これにより、ポーリング結果はすぐに返され、更新間隔は無視されます。詳細については、[プロジェクトのプロパティ - OPC DA 準拠](#)を参照してください。

● **関連項目:** [デバイスのプロパティ - スキャンモード](#)

iFIX アプリケーションからの構成

IGS I/O アドレスを参照するようにプロセスデータベースブロックを構成する方法については、以下のリストからリンクを選択してください。

[概要: iFIX アプリケーション内でのデータブロックの作成](#)

[IGS の設定オプション](#)

[iFIX データベースマネージャでのドライバー情報の入力](#)

[デバイスフィールドでの I/O ドライバーの指定](#)

[iFIX データベースマネージャでの I/O アドレスの指定](#)

[iFIX データベースマネージャでの信号条件の指定](#)

[I/O 信号条件のオプション](#)

[アナログレジスタとデジタルレジスタ \(AR/DR\) でのオフセットフィールドの使用](#)

[iFIX アプリケーションのプロジェクトの起動](#)

概要: iFIX アプリケーション内でのデータブロックの作成

IGS ドライバー構成プログラムを使用してすべての IGS ドライバータグを作成する必要はありません。正しい情報を使用するかぎり、iFIX データベースマネージャでデータベースを構成しているときに IGS ドライバータグを追加できます。これを行うには以下の情報が必要です。

- 3 文字で構成されるドライバーの頭字語。IGS ドライバーの場合、その頭字語は "IGS" です。
- データを収集するチャンネル、デバイス、およびタグの名前 (チャンネル、装置、および元のデータ) が収集されます (IGS ドライバー構成プログラムで定義されている)。
- タグに関するその他の情報 (ビットオフセットの配列要素など)。

● 自動的にデータブロックを作成するためにデータベースマネージャでデータを入力する方法の詳細については、[iFIX データベースマネージャでのドライバー情報の入力](#)を参照してください。

iFIX データベースマネージャでの I/O ドライバーの指定

データベースブロックがアクセスする I/O ドライバーを特定するには、データベースマネージャで **Driver** プロパティを見つけます。ドライバーの 3 文字の頭字語を指定します。IGS ドライバーを使用するには、「IGS」と入力します。

デフォルトのドライバーを検索するには、「**System Configuration Utility (SCU)**」を開き、「**SCADA Configuration**」をクリックします。デフォルトのドライバーは、設定済みの「**Configured I/O Driver**」リストボックスの最初に表示されるドライバーです。

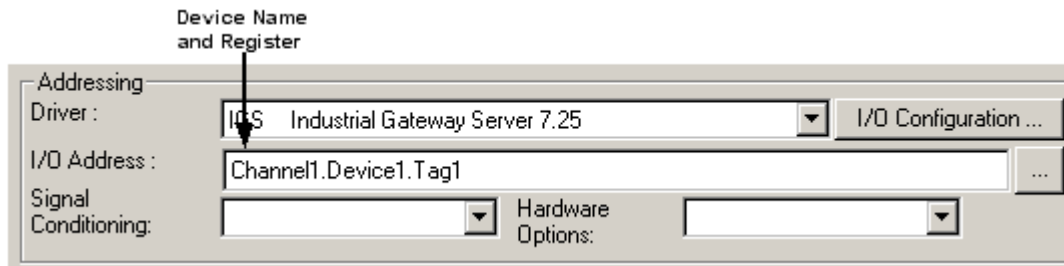
The screenshot shows the 'Addressing' dialog box in iFIX. It has four main sections: 'Driver', 'I/O Address', 'Signal Conditioning', and 'Hardware Options'. The 'Driver' dropdown menu is currently selected and shows 'IGS Industrial Gateway Server 7.25'. An arrow points to this dropdown with the text 'Three-letter Driver Acronym'. The 'I/O Address' field contains the text 'Channel1.Device1.Tag1'. The 'Signal Conditioning' dropdown is empty. The 'Hardware Options' section has two empty dropdown menus.

● **注記:** 入力された頭文字をデータベースマネージャが認識している場合は、SCU の「**Configured I/O Driver**」リストボックスに必ず表示されます。

iFIX データベースマネージャでの I/O アドレスの指定

アクセスするデータブロックのアドレスを指定するには、データベースマネージャで「**I/O Address**」プロパティを見つけます。I/O アドレスを入力します。このフィールドでは大文字と小文字は区別されません。IGS ドライバーでは、I/O アドレスは通常、チャンネル、デバイス、タグの名前で構成され、ドライバーに固有です。

● **注記:** IGS サーバーでは複数のブロックが同じ I/O アドレスを使用することがあります。



ドライバーの I/O アドレスは、次のフォーマットになります: *Channel_Name.Device_Name.Tag_Name*

ここで、

- **Channel_Name:** IGS サーバープロジェクトで使用されるプロトコルまたはドライバーの名前です。IGS 構成内のチャンネル名と一致する必要があります。
- **Device_Name:** サーバーが通信する PLC またはその他のハードウェアの名前です。IGS 構成内の指定されたチャンネルのデバイス名と一致する必要があります。
- **Tag_Name:** サーバーが通信する PLC またはその他のハードウェアデバイス内のアドレス名です。IGS 構成内の指定されたチャンネルおよびデバイスのタグ名と一致する必要があります。

● **注記:** タグが Controllogix L5K ファイルからインポートされた場合は、タグ名へのフルパスが含まれる必要があります。

ビットアドレス指定

ビットアドレス指定は、次の 2 つの方法のいずれかを使用して実行できます。

1. デジタルレジスタ (DR) ブロックを使用している場合は、整数データ内のビット (または Boolean 配列データ内のビット) は、番号付きのフィールド F_0、F_1、F_2 などによって指定できます。詳細については、アナログレジスタとデジタルレジスタ (AR/DR) でのオフセットフィールドの使用を参照してください。
2. DR ブロックが使用されていない場合、タグアドレス内で指定された目的のビットによって、IGS サーバープロジェクトでタグを設定する必要があります。または、ブロックの I/O アドレス内の適切なビットアドレスを指定して、タグを動的に作成できるようにします。詳細については、IGS デバイスドライバーのヘルプドキュメントを参照してください。

● **注記:**

1. ユーザーは DA および DI ブロックの I/O アドレス内で整数タグを指定することもできます。ただし、これらのブロックタイプでは、その整数の最下位ビットしか読み取りまたは書き込みできません。
2. タグを L5K ファイルからインポートする場合は、ビットアドレス指定がサポートされないため、ユーザーが IGS サーバー構成プログラムで、追加ビットアドレスとそれに関連するタグ名を手動で追加する必要があります。たとえば、グローバルコントローラタグ "ValveArea3" が L5K インポートファイル内で Short データ型として設定されているとします。iFIX PDB でこのタグのビット 1 のアドレスを指定するには、まず IGS サーバー構成プログラムで、ビット 1 のアドレスと対応するタグ名を手動で追加する必要があります。この例では、"ValveArea3_1" がビット 1 アドレスに対して指定されたタグ名です。iFIX PDB のビットアドレスに対する I/O アドレス指定は、"Channel1.Device1.Global.ValveArea3_1" です。

配列アドレス指定

IGS サーバーのデバイスドライバーの多くは、配列をサポートします。ユーザーは、アナログレジスタ (AR) ブロックと番号付きのフィールド F_0、F_1、F_2 などを使用して、配列タグの個別の要素にアクセスできます。Boolean または整数型配列の要素内のビットにアクセスするには、デジタルレジスタ (DR) を使用します。詳細については、アナログレジスタとデジタルレジスタ (AR/DR) でのオフセットフィールドの使用を参照してください。

テキスト形式で配列全体にアクセスするには、TX ブロックを使用します。その他の方法を使用した配列内の個別の要素またはビットへのアクセスは、現在サポートされていません。その他のタイプのブロックを使用する場合、個別のタグによってデータのアドレス指定を行う必要があります。配列アドレス指定のサポートおよび構文の詳細については、IGS デバイスドライバーのヘルプドキュメントを参照してください。

iFIX データベースマネージャでの信号条件の指定

IGS ドライバーは、データに信号条件を適用できます。ユーザーは、iFIX データベースマネージャで定義されているブロックごとに、信号条件オプションを設定できます。詳細については、次の手順を参照してください。

1. 「Signal Conditioning」で、目的のアルゴリズムを指定します。信号条件がない場合は、「なし」を選択します。

Addressing

Driver : IGS Industrial Gateway Server 7.25b I/O Configuration ...

I/O Address : Channel1.Device1.Tag1 ...

Signal Conditioning : 8BN Hardware Options: ...

Select Algorithm

2. 条件付けされたデータの「エンジニアリング単位 (EGU)」の範囲を指定します。

Engineering Units

Low Limit : 0.00

High Limit : 100.00

Units :

● 注記: サポートされている信号条件アルゴリズムの詳細については、[iFIX 信号条件のオプション](#)を参照してください。

iFIX 信号条件のオプション

iFIX データベースマネージャでは、以下の信号条件のオプションを使用できます。

- 3BCD
- 4BCD
- 8AL
- 8BN
- 12AL
- 12BN
- 13AL
- 13BN
- 14AL
- 14BN
- 15AL
- 15BN
- 20P
- TNON

● 注記: 線形および対数表示スケール変換は、静的タグの場合にのみサーバーを介して使用できます。詳細については、[タグのプロパティ - スケール変換](#)および[静的タグ \(ユーザー定義\)](#)を参照してください。

3BCD 信号条件

説明	3 桁の 2 進化 10 進 (BCD) 値
入力範囲	0-999
スケール変換	3 桁の 2 進化 10 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	3 桁の BCD レジスタから読み取ります。次に、Raw_value がスケール変換される前に 3 つのニブル (4 ビット) に分割されます。それぞれのニブルに 9 (A ~ F の 16 進) よりも大きい値があるかどうかを検証されます。A ~ F の 16 進値が見つかった場合は、その値が BCD 範囲内でないことを示す範囲アラームが生成されます。見つからなかった場合、その値は次のアルゴリズムを使用してスケール変換されます。 Result=((Raw_value/999) * Span_egu) + Lo_egu

説明	3 桁の 2 進化 10 進 (BCD) 値
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 3 桁の BCD レジスタに書き込みます。 $Result = (((InputData - Lo_egu) / Span_egu) * 999) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

4BCD 信号条件

説明	4 桁の 2 進化 10 進 (BCD) 値
入力範囲	0-9999
スケール変換	4 桁の 2 進化 10 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	4 桁の BCD レジスタから読み取ります。次に、Raw_value がスケール変換される前に 4 つのニブル (4 ビット) に分割されます。それぞれのニブルに 9 (A ~ F の 16 進) よりも大きい値があるかどうかを検証されます。A ~ F の 16 進値が見つかった場合は、その値が BCD 範囲内にないことを示す範囲アラームが生成されます。見つからなかった場合、その値は次のアルゴリズムを使用してスケール変換されます。 $Result = ((Raw_value / 9999) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 4 桁の BCD レジスタに書き込みます。 $Result = (((InputData - Lo_egu) / Span_egu) * 9999) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

8AL 信号条件

説明	8 ビットの 2 進数
入力範囲	0-255
スケール変換	8 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	8BN と同じアルゴリズムを使用して 16 ビットのレジスタから読み取り、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = ((Raw_value / 255) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	8BN と同じアルゴリズムを使用して 16 ビットのレジスタに書き込み、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = (((InputData - Lo_egu) / Span_egu) * 255) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。

説明	8 ビットの 2 進数
	Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

8BN 信号条件

説明	8 ビットの 2 進数
入力範囲	0-255
スケール変換	8 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。最上位のバイトを無視します。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $Result = ((Raw_value / 255) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 8 ビットレジスタに書き込みます。 $Result = (((InputData - Lo_egu) / Span_egu) * 255) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

12AL 信号条件

説明	12 ビットの 2 進数
入力範囲	0-4095
スケール変換	12 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	12BN と同じアルゴリズムを使用して 16 ビットのレジスタから読み取り、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = ((Raw_value / 4095) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	12BN と同じアルゴリズムを使用して 16 ビットのレジスタに書き込み、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = (((InputData - Lo_egu) / Span_egu) * 4095) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

12BN 信号条件

説明	12 ビットの 2 進数
入力範囲	0-4095
スケール変換	12 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。最上位のニブル (4 ビット) を無視します。範囲外の値は 12 ビットの値として扱われます。たとえば、最上位の 4 つのビットが無視されるため、4096 は 0 として扱われます。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $Result = ((Raw_value / 4095) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。

説明	12 ビットの 2 進数
	Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 4095) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

13AL 信号条件

説明	13 ビットの 2 進数
入力範囲	0-8191
スケール変換	13 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	13BN と同じアルゴリズムを使用して 16 ビットのレジスタから読み取り、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $\text{Result} = ((\text{Raw_value} / 8191) * \text{Span_egu}) + \text{Lo_egu}$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	13BN と同じアルゴリズムを使用して 16 ビットのレジスタに書き込み、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 8191) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

13BN 信号条件

説明	13 ビットの 2 進数
入力範囲	0-8191
スケール変換	13 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。最上位の 3 ビットを無視します。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $\text{Result} = ((\text{Raw_value} / 8191) * \text{Span_egu}) + \text{Lo_egu}$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 8191) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

14AL 信号条件

説明	14 ビットの 2 進数
入力範囲	0-16383
スケール変換	14 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	14BN と同じアルゴリズムを使用して 16 ビットのレジスタから読み取り、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $\text{Result} = ((\text{Raw_value} / 16383) * \text{Span_egu}) + \text{Lo_egu}$

説明	14 ビットの 2 進数
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	14BN と同じアルゴリズムを使用して 16 ビットのレジスタに書き込み、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = (((InputData - Lo_egu) / Span_egu) * 16383) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

14BN 信号条件

説明	14 ビットの 2 進数
入力範囲	0-16383
スケール変換	14 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。最上位の 2 ビットを無視します。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $Result = ((Raw_value / 16383) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $Result = (((InputData - Lo_egu) / Span_egu) * 16383) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

15AL 信号条件

説明	15 ビットの 2 進数
入力範囲	0-32767
スケール変換	15 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。
読み取りアルゴリズム	15BN と同じアルゴリズムを使用してアラームとともに 16 ビットのレジスタから読み取り、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = ((Raw_value / 32767) * Span_egu) + Lo_egu$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	15BN と同じアルゴリズムを使用してアラームとともに 16 ビットのレジスタに書き込み、値が範囲外でアラーム状態であるかどうかを示すステータスを返します。 $Result = (((InputData - Lo_egu) / Span_egu) * 32767) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

15BN 信号条件

説明	15 ビットの 2 進数
入力範囲	0-32767

説明	15 ビットの 2 進数
スケール変換	15 ビットの 2 進値をデータベースブロックの EGU 範囲にスケール変換します。最上位のビットを無視します。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $\text{Result} = ((\text{Raw_value} / 32767) * \text{Span_egu}) + \text{Lo_egu}$
読み取りアルゴリズムの変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 32767) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

20P 信号条件

説明	6400-32000 クランプ
入力範囲	6400-32000
スケール変換	2 進値をデータベースブロックの EGU 範囲にスケール変換します。クランプの値の範囲は 6400 ~ 32000 です。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $\text{Result} = (((\text{Raw_value} - 6400) / 25600) * \text{Span_egu}) + \text{Lo_egu}$
読み取りアルゴリズム変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 25600) + 6400.5$
書き込みアルゴリズム変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

TNON 信号条件

説明	0-32000 クランプ
入力範囲	0-32000
スケール変換	2 進値をデータベースブロックの EGU 範囲にスケール変換します。クランプの値の範囲は 0 ~ 32000 です。
読み取りアルゴリズム	次のアルゴリズムを使用して 16 ビットレジスタから読み取ります。 $\text{Result} = ((\text{Raw_value} / 32000) * \text{Span_egu}) + \text{Lo_egu}$
読み取りアルゴリズム変数	Lo_egu - データベースブロックの低工学値。 Span_egu - 工学値の範囲。 Raw_value - フィールドデバイスのレジスタに保存されている値。 Result - データベースブロックに保存されるスケール変換後の値。
書き込みアルゴリズム	次のアルゴリズムを使用して 16 ビットのレジスタに書き込みます。 $\text{Result} = (((\text{InputData} - \text{Lo_egu}) / \text{Span_egu}) * 32000) + .5$
書き込みアルゴリズムの変数	Lo_egu - 低工学値。 Span_egu - 工学値の範囲。 InputData - データベースブロックの現在の値。 Result - プロセスハードウェアに送信される値。

iFIX アプリケーションのプロジェクトの起動

サーバーの iFIX インタフェースの機能が拡張されたことで、iFIX ユーザーの体感する起動パフォーマンスが向上しています。この拡張機能は、これまで、起動時に不適切に初期化されていたアナログ出力 (AO)、デジタル出力 (DO)、またはアラーム値を使用している iFIX アプリケーションに適用されます。サーバーは、iFIX クライアントによってアクセスされるアイテムがすべて含まれた、特殊な iFIX コンフィギュレーションファイルをデフォルトのサーバープロジェクト用に維持管理しています。このコンフィギュレーションファイルは、iFIX によってアイテムデータがリクエストされる前に、スキャンを自動的に開始するために使用されます。したがって、AO や DO など、1 回のみリクエストされるデータ更新は、iFIX によってリクエストされたときには初期値を保持しています。この機能を既存の iFIX プロジェクトに使用する方法については、以下の説明を参照してください。

1. 最初に、PDB データベースを iFIX データベースマネージャからエクスポートします。
2. エクスポートしたファイルをもう一度インポートして、データベース内の各アイテムがサーバーで再評価されるようにします。
3. **タグの置換を確定する**メッセージボックスで、「すべてはい」を選択します。

● **注記:** デフォルトのサーバープロジェクトファイルと同じフォルダに、新しいコンフィギュレーションファイルが "default_FIX.ini" という名前で作成されます。

4. プロジェクトに含まれているアイテムすべての初期値の読み取りにかかる時間によっては、SAC 処理の開始を遅らせることが必要になる場合もあります。開始を遅らせることで、iFIX クライアントがデータをサーバーにリクエストする前に、サーバーが十分に時間の余裕を持ってすべての初期更新を取得できます。個別の iFIX バージョンの詳細については、iFIX のドキュメンテーションを参照してください。

5. iFIX アプリケーションとサーバーの両方を再起動すると、変更内容が有効になります。

● **注記:** 新しいプロジェクトの場合 (または、既存の iFIX データベースにアイテムを追加登録する場合)、上で説明したステップを実行する必要はありません。アイテムは、データベースへの登録時にサーバーによって検証されます。アイテムが有効である場合、コンフィギュレーションファイルに登録されます。

ストアアンドフォワードサービス

ストアアンドフォワードサービスを使用すると、各種サーバーコンポーネントは一定期間中、データをローカルディスクに保存できます。このサービスは、ストアアンドフォワード機能を必要とするコンポーネントとともにインストールされます。ストアアンドフォワードサービスは、ストアアンドフォワードをサポートする機能に基づいて自動的に起動および停止します。

●関連項目:

[ThingWorx のプロジェクトのプロパティ](#)

[ストアアンドフォワードの構成設定](#)

[ストアアンドフォワード - システムタグ](#)

[ThingWorx のアクセス権](#)

構成 API サービス

構成 API を使用すると、HTTP RESTful クライアントがサーバー内のチャネル、デバイス、タグなどのオブジェクトの追加、編集、読み取り、および削除を行うことができます。構成 API には以下の機能があります。

- 人間が読み取れる標準の JSON データフォーマットのオブジェクト定義
- サーバー内の一部のオブジェクトに対するトリガーおよびモニター操作のサポート
- HTTP 基本認証と HTTP over SSL (HTTPS) によるセキュリティ
- ユーザーマネージャとセキュリティポリシープラグインに基づくユーザーレベルアクセスのサポート
- 詳細度と保持期間を構成できるトランザクションログ

● **注記:** このドキュメントは、HTTP 通信と REST の概念に精通していることを前提としています。

初期化 - 構成 API は、Windows サービスとしてインストールされ、システムとともに自動的に起動されます。

操作 - 構成 API は、サーバーと REST クライアント間の接続とコマンドをサポートします。

シャットダウン - 構成 API を停止する必要がある場合は、Windows のサービスコントロールマネージャを使用して構成 API サービスを終了します。

セキュリティ

REST クライアントから構成 API へのアクセスには HTTP 基本認証が使用される必要があります。ユーザー資格証明はサーバーの ユーザーマネージャ で定義されています。

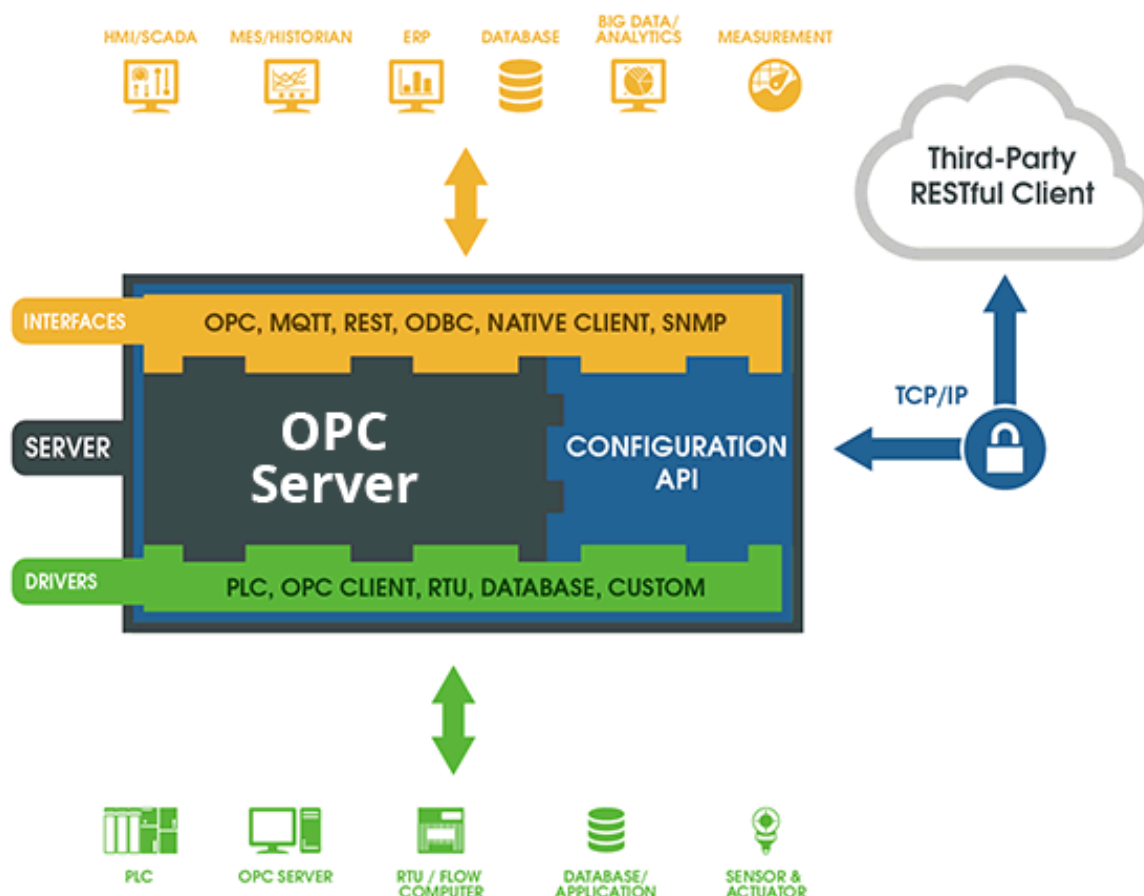
ドキュメンテーション

● プロパティ、データ範囲、エンドポイントマッピングスキーム、および各エンドポイントで許容される操作などの追加の情報については、デフォルトの構成が記載されている構成 API のトップページ (<http://localhost:57412/config/> (デフォルトの構成)) を参照してください。

● トップページから提供されるドキュメンテーションはデフォルトで HTML エンコーディングされています。JSON エンコードされているドキュメンテーションを取得するには、"application/json" に "Accept" 要求ヘッダーを含めます。

Config API サービス - アーキテクチャ

次の図は、コンポーネントのレイアウトを示しています。構成 API サービスはサーバーと同じマシンにインストールされます。



Config API サービス - 同時クライアント

構成 API は、複数のクライアントに同時に使用できます。クライアントが古い構成を編集できないようにするため、サーバーランタイムで数値のプロジェクト ID を維持します。オブジェクトが構成 API またはローカル構成クライアントを介して編集されるたびにプロジェクト ID は変更されます。現在のプロジェクト ID は、各 GET 応答で返されます。現在のプロジェクト ID は、すべての PUT 要求でクライアントによって指定される必要があります。

GET 要求を発行し、現在のプロジェクト ID を保存して、その ID を後続の PUT 要求で使用することをお勧めします。使用されるクライアントが 1 つだけの場合は、クライアントが PUT 要求ボディでプロパティ "FORCE_UPDATE": true を設定して、構成 API サーバーがプロジェクト ID を無視するように強制することができます。

Config API サービス - コンテンツの取得

コンテンツは HTTP GET 要求を送信することによってサーバーから取得します。この要求では次のいずれかの場所をターゲットの URI として指定できます。

1. オンラインドキュメンテーション (例: /config/v1/doc や /config/v1/doc/drivers)
2. イベントログエントリ (例: /config/v1/event_log)
3. トランザクションログエントリ (例: /config/v1/transaction_log)
4. プロジェクト構成 (例: /config/v1/project や /config/v1/project/channels/Channel1)

プロジェクト構成をターゲットにする場合、REST クライアントは返されるコンテンツのタイプを指定できます。この場合、"コンテンツ" という用語は、コレクションまたはオブジェクトインスタンスに関するデータの 1 つまたは複数のカテゴリを指します。

デフォルトでは、コレクションを識別するエンドポイントを使用して GET 要求が送信されると、サーバーはそのコレクション内の各インスタンスの値が 1 つ含まれる JSON 配列を返します (各値はそのインスタンスのプロパティを含む JSON オブジェクト)。

デフォルトでは、オブジェクトインスタンスを識別するエンドポイントを使用して GET 要求が行われると、サーバーはそのインスタンスのプロパティを含む JSON オブジェクトを返します。

これらの要求のデフォルトの動作は、`http://<ホスト名>:<ポート>/config/v1/project?content=children` などのように 1 つ以上の "コンテンツ" 照会パラメータを指定することによって変更できます。以下の表に、使用可能なコンテンツタイプおよび各エンドポイントタイプへのその適用の可否を示します。

コンテンツタイプ	コレクションのエンドポイント	オブジェクトインスタンスのエンドポイント
properties	yes	yes
property_definitions	no	yes
property_states	no	yes
type_definition	yes	yes
children	yes	yes

以下の表に、各コンテンツタイプに対する JSON 応答の構造体を示します。

GET 要求 URI	JSON 応答の構造体
<code>/config/v1/project?content=properties</code>	<pre>{ <property name>: <value>, <property name>: <value>, ... }</pre>
<code>/config/v1/project?content=property_definitions</code>	<pre>[{<property definition>}, {<property definition>}, ...]</pre>
<code>/config/v1/project?content=property_states</code>	<pre>{ "allow": { <property name>: true/false, <property name>: true/false, ... }, "enable": { <property name>: true/false, <property name>: true/false, ... } }</pre>
<code>/config/v1/project?content=type_definition</code>	<pre>{ "name": <type name>, "collection": <collection name>, "namespace": <namespace name>, "can_create": true/false, "can_delete": true/false, "can_modify": true/false, "auto_generated": true/false, "requires_driver": true/false, "access_controlled": true/false, "child_collections": [<collection names>] }</pre>
<code>/config/v1/project?content=children</code>	<pre>{ <collection name>: [</pre>

GET 要求 URI	JSON 応答の構造体
	<pre>{ "name": <object instance name>, "href": <object instance uri> }, ...], <collection name>: [{ "name": <object instance name>, "href": <object instance uri> }, ...], ... }</pre>

コンマで区切ることで、同じ要求内に複数のコンテンツタイプを指定できます。たとえば、`http://<ホスト名>:<ポート>/config/v1/project?content=children,type_definition` のように指定します。複数のタイプを指定した場合、JSON 応答には、以下のように、要求した各コンテンツタイプのメンバーから成る単一のオブジェクトが含まれます。

```
{
  "properties": <プロパティ応答の構造体>,
  "property_definitions": <プロパティ定義応答の構造体>,
  "property_states": <プロパティ状態応答の構造体>,
  "type_definition": <タイプ定義応答の構造体>,
  "children": <子応答の構造体> }
```

タイプ定義

以下の表で、タイプ定義 JSON オブジェクトの各メンバーについて説明します。

メンバー	タイプ	説明
name	文字列	オブジェクトタイプの名前。
collection	文字列	コレクションの名前。このタイプのオブジェクトが存在するコレクションを示します。この名前が、REST インタフェースを使用してアドレス指定可能な有効なエンドポイントを構成します。
namespace	文字列	このオブジェクトタイプを実装する名前空間。サーバーによって実装されているオブジェクトは "servermain" 名前空間に存在します。その他の名前空間は、ドライバ、プラグイン、クライアントインタフェースなどのオプションのコンポーネントによって定義されます。
can_create	ブール	このタイプのインスタンスをエンドユーザーが作成可能かどうかを示します。たとえば、"プロジェクト" は作成できないタイプなので、これが false になります。
can_delete	ブール	このタイプのインスタンスをエンドユーザーが削除可能かどうかを示します。同様に、"プロジェクト" は削除できないタイプです。
can_modify	ブール	このタイプのインスタンスをエンドユーザーが修正可能かどうかを示します。たとえば、サーバーには、子コレクションを作成するためだけに存在し、それ自身には修正可能なプロパティがない、自動生成されたオブジェクトがいくつか存在します。
auto_generated	ブール	true の場合、このタイプのインスタンスはサーバーによって自動生成されています。通常、このタイプのオブジェクトでは上記の 3 つのメンバーは "false" として定義されます。
requires_driver	ブール	インストールされているドライバの名前を指定しないとこのタイプのインスタンスを作成できない場合、true になります。
access_controlled	ブール	このタイプのインスタンスに対して実行可能な CRUD 操作に対してサーバーがグループレベルのアクセス制御を提供する場合、true になります。 (ユーザーマネージャを参照) 。
child_collections	配列	このタイプのオブジェクトで子としてサポートされているコレクション名の配列。たとえば、タイプの "child_collections" に "devices" が含まれている場合、そのタイプのオブジェクトインスタンスは 1 つ以上の "デバイス" インスタンスを子としてサポートします。

プロパティの定義

プロパティ定義は、そのプロパティの特性 (サポートするデータのタイプ、適用可能な範囲、デフォルト値など) を示します。プロパティ定義オブジェクトの JSON 構造体は以下のように定義します。

メンバー	タイプ	説明
symbolic_name	文字列	<名前空間>.<プロパティ名> という形式の正準名によってプロパティを識別します。
display_name	ローカライズされた文字列	サーバー構成プロパティエディタに表示されるときのプロパティの名前。そのサーバーで現在設定されている言語で値が返されます。
display_description	ローカライズされた文字列	サーバー構成プロパティエディタに表示されるときのプロパティの説明。そのサーバーで現在設定されている言語で値が返されます。
read_only	ブール	このプロパティが情報提供のみを目的とし、最初に定義された後は変更できない場合、true になります。
type	文字列	プロパティ値のデータ型を決定します (下の「プロパティタイプ」を参照)。
minimum_value	数値または null (数値タイプに適用)	そのプロパティが有効と見なされる最小の値。null の場合、最小値はありません。
maximum_value	数値または null (数値タイプに適用)	そのプロパティが有効と見なされる最大の値。null の場合、最大値はありません。
minimum_length	数値 (文字列のみに適用)	文字列値の最小長さ。0 の場合、最小値はありません。
maximum_length	数値 (文字列のみに適用)	文字列値の最大長さ。-1 の場合、最大値はありません。
hints	文字列の配列 (文字列のみに適用)	プロパティ値に割り当てることが可能な選択肢の配列。このメンバーは、hints が存在しない場合は含まれません。
enumeration	オブジェクト (列挙のみに適用)	列挙プロパティの場合、このオブジェクトはその列挙に格納できる有効な名前/値のペアを示します。構造体は以下のとおりです。 <pre>{ <name>: number, <name>: number, ... }</pre>
allow	オブジェクトの配列	このプロパティが関連するかどうかを決める、1 つ以上のその他のプロパティへの条件付き依存を定義します。許可されていないプロパティは、サーバー設定プロパティエディタには表示されません (以下の「許可および有効化条件」を参照)。
enable	オブジェクトの配列	このプロパティをクライアントが変更可能かどうかを決める、1 つ以上のその他のプロパティへの条件付き依存を定義します。無効になっているプロパティは、サーバー設定プロパティエディタで暗色表示されます (以下の「許可および有効化条件」を参照)。

特定のエンドポイントのプロパティ定義に関する特定の情報を取得するには、get リクエストの URL の末尾に "? content = property_definitions" を追加します。

たとえば、ローカルホスト上で実行されているサーバーとの Channel1 という名前のチャネルのプロパティ定義を取得するには、get リクエストを次のように送信します。

```
http://localhost:57412/config/v1/channels/Channel1?content=property_definitions
```

返された JSON ブロックは次のようになります。


```
[
  {
    "symbolic_name": "共通.ALLTYPES_NAME ",
    "表示名": "name",
    "display_description": "このオブジェクトの id を指定してください。",
    "read_only": false,
    "type": "String",
    "default_value": null,
    "minimum_length": 1,
    "maximum_length": 256
  },
  {
    "symbolic_name": "共通.ALLTYPES_DESCRIPTION ",
    "表示名": "Description",
    "display_description": "このオブジェクトまたはその使用法の概要を指定してください。",
    "read_only": false,
    "type": "String",
    "default_value": null,
    "minimum_length": 0,
    "maximum_length": 255
  },
  ...
]
```

プロパティのタイプ

以下の表では、プロパティ定義で "type" メンバーに使用可能な各値について説明します。"値のタイプ" は、そのプロパティの値がとるべき JSON タイプを示します。

タイプ名	値のタイプ	説明
AllowDeny	ブール	"Allow"=true と "Deny"= false の選択肢があるドロップダウンリストが表示されるプロパティ。
EnableDisable	ブール	"Enable"=true と "Disable"= false の選択肢があるドロップダウンリストが表示されるプロパティ。
YesNo	ブール	"Yes"=true と "No"= false の選択肢があるドロップダウンリストが表示されるプロパティ。
String	文字列	一般的な文字列。このタイプのプロパティには指定子 minimum_length および maximum_length が含まれます。
StringArray	配列	文字列の配列。このタイプのプロパティには、配列の長さではなく、文字列に適用される minimum_length および maximum_length 指定子が含まれます。
Password	文字列	パスワードが含まれている難読化された文字列。このタイプのプロパティの値を変更する場合、プレーンテキストのパスワードを入力する必要があります。パスワードの値の変更はセキュリティ保護された接続を介してのみ行います。
LocalFileSpec	文字列	ローカル Windows ファイルシステムにおける完全修飾によるファイル指定。
UncFileSpec	文字列	ネットワーク上の場所における完全修飾によるファイル指定。
LocalPathSpec	文字列	ローカル Windows ファイルシステムにおける完全修飾によるパス指定。
UncPathSpec	文字列	ネットワーク上の場所における完全修飾によるパス指定。

タイプ名	値のタイプ	説明
StringWithBrowser	文字列	文字列値をとるプロパティ。通常、この値は動的に生成された文字列の集まりから選択されます。
Integer	数値	符号なし 32 ビット 整数値。
Hex	数値	16 進表記で表示/編集する符号なし 32 ビット 整数値。
Octal	数値	8 進表記で表示/編集する符号なし 32 ビット 整数値。
SignedInteger	数値	符号付き 32 ビット 整数値。
Real4	数値	単精度浮動小数点値。
Real8	数値	倍精度浮動小数点値。
Enumeration	数値	プロパティ定義の "enumeration" メンバーのうち、使用可能な数値の 1 つ。
PropArray	オブジェクト	それぞれが固定長の値の配列を持つメンバーから成る構造体。
TimeOfDay	数値	特定の時刻を定義する、午前 0 時からの経過秒数が含まれている整数値。
Date	数値	特定の日付の午前 0 時を示す UNIX の時刻値。
DateAndTime	数値	特定の日付の特定の時刻を示す UNIX の時刻値。
Blob	配列	データの不透過コレクションを表すバイト値の配列。このタイプのデータはサーバーで作成されたものであり、修正を防止するためにハッシュ化されています。

allow 条件と enable 条件

allow 条件、enable 条件、またはその両方が含まれている定義の JSON 構造体は以下のとおりです。

```
<条件>: [ { "depends_on": <プロパティ名> "operation": "==" or "!="
           "value": <値> }, ... ]
```

各条件には、依存する別のプロパティと、どのように依存するか (そのプロパティの値と等しいか等しくないか) を指定します。同じプロパティまたは別のプロパティに、複数の依存が存在できます。複数の存在する場合、"演算" は常に同じになります。複数の依存が存在する場合に条件の状態を特定するための式の評価は、"==" では論理 "or" であり、"!=" では論理 "and" です。

"content=property_states" を使用している場合、返される JSON は、各プロパティのこれらの条件 (存在する場合) の評価結果を表しています。

言語の仕様

サーバーは、複数の言語をサポートします。設定された言語にローカライズされたテキストをクライアントに返します。クライアントは、要求ヘッダー内の「Accept-language」フィールドを指定することにより、GET 要求で設定されている言語をオーバーライドできます。

● 詳細については <https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> を参照してください。

例として、英語のサーバーが設定されているがクライアントをドイツ語で使用する場合、要求ヘッダーを「Accept-language: de」のように指定します。

● 注記: サーバーによってサポートされていない言語がクライアントで指定されている場合、現在設定されている言語が使用されます。

Config API サービス - データ

構成 API サービスは、標準の JSON フォーマットで REST クライアントから要求を受信します。これらの要求はサーバーによって消費され、作成、読み取り、更新、または削除コマンドに分類されます。

● プロパティ、データ範囲、エンドポイントマッピングスキーム、および各エンドポイントで許可される操作などの追加の情報については、デフォルトの構成が記載されている構成 API のトップページ (<http://localhost:57412/config/> (デフォルトの構成)) を参照してください。

● トップページから提供されるドキュメンテーションはデフォルトで HTML エンコーディングされています。JSON エンコードされているドキュメンテーションを取得するには、"application/json" に "Accept" 要求ヘッダーを含めます。

● スペースや、URL フォーマットで許可されていないその他の文字が含まれているオブジェクト名は、構成 API によって正しく解釈されるためにパーセントエンコーディングされている必要があります。パーセントエンコーディングによって、許可されていない文字がその 16 進数表現に置き換えられます。たとえば、"default object" という名前のオブジェクトは default%20object にパーセントエンコーディングされます。次の文字は URL では許可されておらず、エンコードする必要があります。

スペース	!	#	\$	&	'	()	*	+	,	/	:	;	=	?	@	[]
%20	%21	%23	%24	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

● 先頭と末尾のスペースはすべて、サーバーによって検証される前にオブジェクト名から除去されます。これによって、サーバー内のオブジェクト名とユーザーが構成 API を介して指定したオブジェクト名が一致なくなることがあります。ユーザーは PUT/POST を送信した後で親オブジェクトに対して GET を送信することで、サーバー内の新規または修正されたオブジェクト名が API を介して送信されたオブジェクト名と一致するかどうかを確認できます。

● 構成 API で PhonePriority オブジェクトの servermain.PHONEBOOK_PRIORITY プロパティに書き込むことで、電話帳内のダイヤルエントリの優先順位を変更できます。この文字列プロパティは、引用符で囲まれた電話帳エントリ名のコンマ区切りリストです。

たとえば、エンドポイント `http://{ローカルホスト:番号}/config/v1/project/channels/{チャネル名}/phonebooks/phonebook/phonePriorities/` への GET 要求では以下が返されます。

```
{
  "PROJECT_ID": 1270990535,
  "common.ALLTYPES_DESCRIPTION": "",
  "common.ALLTYPES_NAME": "PhonePriority",
  "servermain.PHONEBOOK_PRIORITY": "\"Phone3\", \"Phone1\", \"Phone2\""
}
```

servermain.PHONEBOOK_PRIORITY プロパティを変更するためのエンドポイント `http://{ローカルホスト:番号}/config/v1/project/channels/{チャネル名}/phonebooks/phonebook/phonePriorities/PhonePriority` への PUT 要求の例を以下に示します。

```
{
  "PROJECT_ID": 1270990535,
  "servermain.PHONEBOOK_PRIORITY": "\"Phone1\", \"Phone2\", \"Phone3\""
}
```

● パーセントエンコーディングは名前が有効であることを保証しません。有効な名前の値を調べるには、作成するオブジェクトのドキュメントを参照してください。

● 文字列プロパティ内の電話帳エントリ名を囲む引用符は円記号 (") によってエスケープする必要があります。モデムを選択および構成するには、API ではなくサーバー構成アプリケーションを使用することをお勧めします。

● API を介して非管理者ユーザーとして POST/PUT/DELETE を実行しようとしたときに、ユーザーがサーバー構成を開いている場合、その試みは失敗します。このエラーのステータスコードは 401 (権限なし) です。ランタイムに同時に書き込

むことができるユーザーは 1 人だけです。資格証明が不十分な場合、API はサーバー構成からアクセス許可を取り上げることができません。

オブジェクトを作成する

オブジェクトは、HTTP POST 要求を構成 API に送信することによって作成できます。新しいオブジェクトを作成するときには、オブジェクトに必要なプロパティが JSON に含まれている必要がありますが (各オブジェクトに名前が付けられている必要があるなど) が、すべてのプロパティが含まれている必要はありません。JSON に含まれていないプロパティはすべて、作成時にデフォルト値に設定されます。POST JSON ボディの例を次に示します。

```
{
  "<プロパティ 1 の名前>": <値>,
  "<プロパティ 2 の名前>": <値>,
  "<プロパティ 3 の名前>": <値>
}
```

複数のオブジェクトの作成

配列内で JSON プロパティオブジェクトを含めることによって、特定のコレクションに複数のオブジェクトを追加できます。POST JSON ボディの例を次に示します。

```
[{"< Property1_Name >": < 値 >, "< Property2_Name >": < 値 >, "< Property3_Name >": < 値 >,
  "< Property1_Name >": < 値 >, "< Property2_Name >": < 値 >, "< Property3_Name >": < 値 >}]
```

POST に複数のオブジェクトが含まれており、プロパティ検証エラーまたはその他のエラーが原因で 1 つ以上のオブジェクトを処理できない場合、HTTP ステータスコード 207 (複数のステータス) と、要求内の各オブジェクトのステータスを含む JSON オブジェクトの配列が返されます。たとえば、要求に 2 つのオブジェクトが含まれている場合、2 番目のオブジェクトには最初のオブジェクトと同じ名前が指定されます。

```
[ { "code": 201, "message": "Created" }, { "code": 400,
  "message": "行 7 でオブジェクト定義のプロパティ common.ALLTYPES_NAME に対する検証に失敗しました: 名前 'Channel1' はすでに使用されています。" } ]
```

子階層でのオブジェクトの作成

その下にすべての子オブジェクト階層を持つオブジェクトを作成できます。そのためには、JSON プロジェクトファイルに保存する際に表示されるのと同様に、階層を POST 要求に含めます。たとえば、その下にデバイスを含むチャンネルを作成するには、次の JSON を使用できます。

```
[{"common.ALLTYPES_NAME ":" Channel1 ", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER ":" シミュレータ ", " デバイス ": [{"common.ALLTYPES_NAME ":" Device1 ", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER ":" シミュレータ ", "servermain.DEVICE_MODEL ":" 0"}]}]
```

オブジェクトを読み取る

オブジェクトは、HTTP GET 要求を構成 API に送信することによって読み取ることができます。すべての GET 要求ですべてのオブジェクトプロパティが返され、各オブジェクトに Project_ID が含まれています。Project_ID プロパティは、構成の変更を追跡するために使用され、構成 API またはサーバー構成クライアントから変更が通知されるたびに更新されます。このプロパティは、古いデータの改ざんを防止するために、すべての PUT 要求で保存され、使用される必要があります。応答ボディの例を次に示します。

```
{"< Property1_Name >": < 値 >, "< Property2_Name >": < 値 >, "PROJECT_ID": 12345678}
```

● 関連項目: [コンテンツの取得](#)

オブジェクトを編集する

オブジェクトは、HTTP PUT 要求を構成 API に送信することによって編集できます。PUT 要求は、JSON ボディに Project_ID または Force_Update プロパティを必要とします。Force_Update を True に設定すると、Project_ID 検証は無視されます。PUT ボディの例を次に示します。

```
{
  "<プロパティ 1 の名前>": <値>,
  "<プロパティ 2 の名前>": <値>,
  "PROJECT_ID": 12345678,
  "FORCE_UPDATE": true
}
```

通常、PUT 要求が成功してすべてのプロパティが適切に割り当てられた場合、クライアントに返される応答ボディはなく、成功を示す 200 ステータスコードのみが表示されます。サーバーランタイムによってオブジェクトインスタンスに割り当てられていない PUT 要求にプロパティが含まれている場合には、応答ボディが次のように生成されます。

```
{, "not_applied":, {, "servermain.CHANNEL UNIQUE ID ": 2466304381}, " code ": 200, "
message ":" すべてのプロパティが適用されていません。アクティブなクライアント参照があるか、プロパティが許可
されていない/無効/読み取り専用である可能性があります。"}
```

1 つまたは複数のプロパティが、実際に使用されている値を含む各オブジェクトインスタンスに適用されなかったことを、応答は示しています。プロパティの値を適用できなかった理由として、下記のようないくつかの可能性があります。

- ・プロパティが読み取り専用で変更できない。
- ・更新できるプロパティを制限するオブジェクト上のクライアント参照がある。
- ・この条件に依存するその他のプロパティの値に基づき、プロパティを使用できない。
- ・この条件に依存するその他のプロパティの値に基づき、プロパティが無効になっている。
- ・何らかの方法 (切り上げ/切り捨てなど) で値が変換されている。

オブジェクトを削除する

オブジェクトは、HTTP DELETE 要求を構成 API に送信することによって削除できます。構成 API は、同じレベルの複数のアイテムを単一の要求で削除すること (1 つのチャンネルのすべてのデバイスを削除するなど) はサポートしていませんが、ツリー全体を削除すること (1 つのデバイスを削除すると、そのすべての子タグが削除されるなど) はできます。

エラー

構成 API サービスの要求はすべて、JSON フォーマットでエラーを返します。例:

```
{
  "code": 400,
  "message": "無効なプロパティ: 'NAME' ."
}
```

● 関連項目: [トラブルシューティング](#)

Config API サービス - チャンネルの作成

Config API サービスを介してチャンネルを作成するには、最小プロパティセットのみが必要です。その他すべてがデフォルト値に設定されます。チャンネルを定義した後は、そのチャンネルに割り当てられているすべてのデバイスでそのプロパティと設定が使用されます。一部のプロパティは選択しているプロトコルまたはドライバーによって異なります。

Postman、Insomnia、または Curl などの REST ベースの API ツールを使用します。チャンネルの終点への POST リクエストを作成します。

次の例では、ローカルホスト上で実行されているサーバー上でシミュレートドライバーを使用する Channel1 という名前のチャンネルを作成します。

```
ポスト http://localhost:57412/config/v1/project/channels

{
  "common.ALLTYPES_NAME": "Channel1",
```

```
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator"
}
```

ドライバー固有のヘルプドキュメントを参照して、そのドライバー用のチャンネルを作成するために必要なプロパティについて調べてください。

Config API サービス - チャンネルの更新

チャンネルのプロパティまたはプロパティのコレクションを更新するには、最初に、プロジェクト ID を取得するために更新するために GET 要求を終点に送信する必要があります。

プロジェクト ID の詳細については、同時クライアントセクションを参照してください。

次の例では、更新されているチャンネルが Channel1 になっています。

```
http://localhost:57412/config/v1/project/channels/Channel1 を取得
```

GET 要求は以下のような JSON BLOB を返します。

```
{
  "PROJECT_ID": 1988502408,
  "common.ALLTYPES_NAME": "Channel1",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator",
  "servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false,
  "servermain.CHANNEL_UNIQUE_ID": 2154899492,
  "servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2,
  ...
}
```

チャンネルのプロパティを更新または変更するには、プロジェクト ID と新しいプロパティ値が定義されているチャンネルに PUT 要求が送信されます。次の例では、チャンネル名が Channel1 (上から) からシミュレータに変わります。

```
http://localhost:57412/config/v1/project/channels/Channel1 を配置
```

```
{
  "PROJECT_ID": 1988502408,
  "common.ALLTYPES_NAME": "Simulator"
}
```

PUT に続くと、チャンネルの終点に GET を送信して、プロパティが変更されたことを検証できます。この場合は名前が変更されているのでエンドポイントも変更されます。GET 要求は以下のようになります。

● 注記: 一部のプロパティはクライアントによって制限されており、クライアントが接続しているときに変更できません。

```
http://localhost:57412/config/v1/project/channels/Simulator を取得
```

GET リクエストからの応答には、プロパティ値が変更されたことが示されます。上記の GET への応答は以下のようになります。

```
{
  "PROJECT_ID": 2899413628,
  "common.ALLTYPES_NAME": "Simulator",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator",
  "servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false,
  "servermain.CHANNEL_UNIQUE_ID": 2154899492,
  "servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2,
  ...
}
```

Config API サービス - チャネルの除去

チャネルを除去するには、除去するチャネルの終点に削除コマンドを送信します。これにより、チャネルとそのすべての子が除去されます。

以下の例では、チャネルシミュレータが除去されます。

```
http://localhost:57412/config/v1/project/channels/Simulator を削除
```

これは除去したエンドポイントに GET を送信することで確認できます。サーバーはエラーを返します。また、「チャネル」終点に GET を使用して検証することもできます。除去されたチャネルは、GET リクエストから返されたチャネルのリストには表示されません。

Config API サービス - デバイスの作成


Config API サービスを介してデバイスを作成するには、最小プロパティセットのみが必要です。その他すべてがデフォルト値に設定されます。一部のプロパティは選択しているプロトコルまたはドライバーによって異なります。

Postman、Insomnia、または Curl などの REST ベースの API ツールを使用します。チャネルの下のデバイスのエンドポイントに POST リクエストを作成します。

以下の例では、ローカルホスト上で実行されているサーバー上でシミュレータドライバーを使用する Device1 という名前のデバイスを Channel1 に作成します。

```
ポスト http://localhost:57412/config/v1/project/channels/Channel1/devices
```

```
{
  "common.ALLTYPES_NAME": "Device1",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator"
}
```

ドライバー  固有のヘルプドキュメントを参照して、そのドライバー用のデバイスを作成するために必要なプロパティについて調べてください。

Config API サービス - デバイスの更新

デバイスのプロパティまたはプロパティのコレクションを更新するには、最初に、プロジェクト ID を取得するために更新するために GET 要求を終点に送信する必要があります。

 プロジェクト ID の詳細については、[同時クライアント](#) セクションを参照してください。

次の例では、更新されるデバイスは Channel1 の下に Device1 でいます。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1 を取得
```

GET 要求は以下のような JSON BLOB を返します。

```
{
  "PROJECT_ID": 1342460467,
  "common.ALLTYPES_NAME": "Device1",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator",
  "servermain.DEVICE_MODEL": 0,
  "servermain.DEVICE_UNIQUE_ID": 3671734680,
  "servermain.DEVICE_CHANNEL_ASSIGNMENT": "Channel1",
  ...
}
```

デバイスのプロパティを更新または変更するには、プロジェクト ID と新しいプロパティ値が定義されているデバイスに PUT 要求が送信されます。次の例では、デバイス名が Device1 (上から) からシミュレータに変わります。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1 を配置
```



```
{
  "PROJECT_ID": 1342460467,
  "common.ALLTYPES_NAME": "Simulator"
}
```

PUT の後、GET をデバイスのエンドポイントに送信して、プロパティが変更されたことを確認できます。この場合は名前が変更されているのでエンドポイントも変更されます。GET 要求は以下のようになります。

● **注記:** 一部のプロパティはクライアントによって制限されており、クライアントが接続しているときに変更できません。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Simulator を取得
```

GET 要求の応答にはプロパティ値が変更されたことが示されます。上記の GET への応答は以下のようになります。

```
{
  "PROJECT_ID": 2135279368,
  "common.ALLTYPES_NAME": "Simulator",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Simulator",
  "servermain.DEVICE_MODEL": 0,
  "servermain.DEVICE_UNIQUE_ID": 3671734680,
  "servermain.DEVICE_CHANNEL_ASSIGNMENT": "Channel1",
  ...
}
```

Config API サービス - デバイスの除去

デバイスを除去するには、削除するデバイスのエンドポイントに削除を送信します。これにより、デバイスとそのすべての子が除去されます。

次の例では、デバイスシミュレータが除去されます。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Simulator を削除
```

これは除去したエンドポイントに GET を送信することで確認できます。サーバーはエラーを返します。また、デバイスのエンドポイントに get を使用して確認することもできます。また、除去されたデバイスは、get リクエストから返されたデバイスのリストには表示されません。

Config API サービス - タグの作成

Config API サービスを介してタグを作成するには、最小プロパティセットのみが必要です。その他すべてがデフォルト値に設定されます。一部のプロパティは選択しているプロトコルまたはドライバによって異なります。

Postman、Insomnia、または Curl などの REST ベースの API ツールを使用します。デバイスの下のタグエンドポイントにポストリクエストを作成します。

次の例では、ローカルホスト上で実行されているサーバー上でシミュレータドライバを使用する Channel1/Device1 で address R5 という名前のタグを作成します。

```
ポスト
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags
```

```
{
  "common.ALLTYPES_NAME": "MyTag",
  "servermain.TAG_ADDRESS": "R5"
}
```

タググループ内にタグを作成することもできます。タググループを追加するプロセスは同じですが、URL が tag_group の終点とグループ名を含むように変更される点が異なります。

次の例では、タググループ RampTags がすでに存在し、mytag という名前のタグが、アドレス R5 の下に作成されます。

● タググループの作成の詳細については、タググループの作成セクションを参照してください。

ポスト

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_group/RampTags/tags
```

```
{
  "common.ALLTYPES_NAME": "MyTag",
  "servermain.TAG_ADDRESS": "R5"
}
```

ドライバー固有のヘルプドキュメントを参照して、そのドライバーのタグを作成するために必要なプロパティについて調べてください。

Config API サービス - タグの更新

タグのプロパティを 1 つ以上更新するには、まず更新するエンドポイントに GET 要求を送信して、プロジェクト ID を取得する必要があります。

プロジェクト ID の詳細については、[同時クライアント](#) セクションを参照してください。

次の例では、更新されているタグは、Channel1/Device1 の下の mytag です。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/My-Tag を取得
```

GET 要求は以下のような JSON BLOB を返します。

```
{
  "PROJECT_ID": 1473736741,
  点 .ALLTYPES_NAME ":" mytag ",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.TAG_ADDRESS": "R0005",
  "servermain.TAG_DATA_TYPE": 5,
  "servermain.TAG_READ_WRITE_ACCESS": 1,
  "servermain.TAG_SCAN_RATE_MILLISECONDS": 100,
  ...
}
```

タグプロパティを更新または変更するには、プロジェクト ID と新しいプロパティ値が定義されているタグに PUT 要求が送信されます。

次の例では、タグ名が mytag (上から) から Tag1 に変更されています。

```
PUT
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/My-Tag
```

```
{
  "PROJECT_ID": 1473736741,
  点 .ALLTYPES_NAME ":" Tag1 "
}
```

次に、GET a をタグの終点に送信して、プロパティが変更されていることを確認します。この場合は名前が変更されているのでエンドポイントも変更されます。GET 要求は以下ようになります。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/Tag1 を取得
```

GET 要求の応答にはプロパティ値が変更されたことが示されます。上記の GET への応答は以下ようになります。

```
{
  "PROJECT_ID": 1239878456,
  点 .ALLTYPES_NAME ":" Tag1 ",
  "common.ALLTYPES_DESCRIPTION": "",
}
```

```
"servermain.TAG_ADDRESS": "R0005",
"servermain.TAG_DATA_TYPE": 5,
"servermain.TAG_READ_WRITE_ACCESS": 1,
"servermain.TAG_SCAN_RATE_MILLISECONDS": 100,
...
```

Config API サービス - タグの除去

タグを除去するには、削除するタグの終点に削除を送信します。これにより、タグとそのすべての子が除去されます。以下の例では、タグ Tag1 が除去されます。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/Tag1 を削除
```

これは除去したエンドポイントに GET を送信することで確認できます。サーバーはエラーを返します。これは、タグの取得エンドポイントを使用して確認することもでき、削除されたタグは get 要求から返されたタグのリストには含まれません。

Config API サービス - タググループの作成

Config API サービスを介してタググループを作成するには、グループ名のみが必要です。

Postman、Insomnia、または Curl などの REST ベースの API ツールを使用します。デバイスの下に tag_groups の終点への POST リクエストを作成します。

次の例では、ローカルホスト上で実行されているサーバー上でシミュレートドライバを使用する Channel1/Device1 の下に RampTags という名前のタググループを作成します。

```
ポスト
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_groups
```

```
{
 点 .ALLTYPES_NAME ":" RampTags "
}
```

タググループにはタグがあり、その下にタググループがネストされている場合があります。タグを追加するには、[タグの作成](#) セクションを参照してください。

別のグループ内にタググループをネストするには、既存のグループ名と tag_groups 終点を URL の末尾に追加するために、別の POST 操作が必要です。

前述の例を続けると、新しいリクエストは次のようになります。

```
ポスト
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_groups/RampTags/tag_groups
```

```
{
 点 .ALLTYPES_NAME ":" 1-10 "
}
```

Config API サービス - タググループの更新

タグのプロパティを 1 つ以上更新するには、まず更新するエンドポイントに GET 要求を送信して、プロジェクト ID を取得する必要があります。

● プロジェクト ID の詳細については、[同時クライアント](#) セクションを参照してください。

次の例では、更新されるタググループは Channel1/Device1 の下に RampTags でいます。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_
groups/RampTags を取得
```

GET 要求は以下のような JSON BLOB を返します。

```
{
  "PROJECT_ID": 3115312875,
  "common.ALLTYPES_NAME": "RampTags",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.TAGGROUP_LOCAL_TAG_COUNT": 0,
  "servermain.TAGGROUP_TOTAL_TAG_COUNT": 0,
  "servermain.TAGGROUP_AUTOGENERATED": false
}
```

タググループのプロパティを更新または変更するには、プロジェクト ID と新しいプロパティ値が定義されているタググループに PUT 要求が送信されます。

次の例では、タググループ名が RampTags (上から) から RampGroup に変わります。

```
PUT
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/My-
Tag
```

```
{
  "PROJECT_ID": 3115312875,
  点.ALLTYPES_NAME ":" RampGroup "
}
```

PUT の後、GET をタググループの終点に送信して、プロパティが変更されたことを確認できます。この場合は名前が変更されているのでエンドポイントも変更されます。GET 要求は以下のようになります。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_
groups/RampGroup を取得
```

GET 要求の応答にはプロパティ値が変更されたことが示されます。上記の GET への応答は以下のようになります。

```
{
  "PROJECT_ID": 4224221764,
  "common.ALLTYPES_NAME": "RampTags",
  "common.ALLTYPES_DESCRIPTION": "",
  "servermain.TAGGROUP_LOCAL_TAG_COUNT": 0,
  "servermain.TAGGROUP_TOTAL_TAG_COUNT": 0,
  "servermain.TAGGROUP_AUTOGENERATED": false
}
```

Config API サービス - タググループの除去

タググループを除去するには、除去するタググループの終点に削除を送信します。これにより、タググループとそのすべての子が除去されます。下の例では、タググループ RampGroup が除去されます。

```
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tag_
groups/RampGroup を削除
```

これは除去したエンドポイントに GET を送信することで確認できます。サーバーはエラーを返します。これは tag_groups の get を使用して確認することもできます。また、除去されたタググループは、get リクエストから返されたタググループのリストには含まれません。

Config API サービス - サービス

標準 CRUD (作成、読み込み、更新、削除) 操作の範囲外でオブジェクト上で起動できる操作がある場合、オブジェクトがサービスを提供することがあります。サービスにより、リモートクライアントがこれらの操作をトリガーおよびモニターでき

る、非同期プログラマティックインタフェースが使用できます。サービスは、それが動作するオブジェクトの下にある「services」と呼ばれるコレクションにあります。たとえば、プロジェクトロードサービスは、プロジェクトで動作する /config/v1/project/services/ProjectLoad エンドポイントにあります。どのオブジェクトもサービスを提供できるため、サービスコレクションが存在する場合、コレクションに対してクエリーを実行し、利用可能なサービスを確認します。

サービスアーキテクチャ

サービスは、それが動作するオブジェクトとのステートレスな対話を提供するように設計されています。サービスは、サービスとジョブの2つのコンポーネントで構成されます。ジョブは非同期的に作業を実行し、クライアントが完了したジョブまたはその操作中に発生したエラーのジョブをモニターできるメカニズムを提供します。ジョブが完了した後は、サーバーによって自動的に削除されるようスケジュールされるため、完了後にクライアントがジョブをクリーンアップする必要はありません。

サービス

サービスは、操作が呼び出されるインタフェースです。サービスは、呼び出し中に指定できるすべてのパラメータをプロパティとして公開します。使用可能なパラメータを表示するには、サービスエンドポイントで HTTP GET を実行します。サービスの名前と説明以外のすべてのプロパティは、サービスの呼び出し時に含めることができるパラメータです。サービスによって、一部のパラメータが必要になる場合や、すべてのパラメータが必要になる場合があります。

サービスの呼び出しは、要求のボディで指定したパラメーターを使用して、サービスエンドポイントで HTTP PUT 要求を実行することによって実現されます。サービスは、同時呼び出しの総数を制限することがあります。同時呼び出しの最大数に達した場合、要求は "HTTP 429 Too Many Requests" 応答で却下されます。制限に達していない場合、サーバーは、"HTTP 202 Accepted" 応答と、新しく作成されたジョブへのリンクを含む応答のボディを返します。

成功した PUT 応答の例:

```
{
  "code": 202,
  "message": "受け付けました",
  "href": "/config/v1/project/services/ProjectLoad/jobs/job1"
}
```

ビジー状態の PUT 応答の例:

```
{
  "code": 429,
  "message": "サーバーがビジー状態です。操作を後で再試行してください。"
}
```

ジョブ

ジョブは、サーバーによって受け入れられる特定の要求を表します。ジョブの状態を確認するには、ジョブエンドポイントで HTTP GET 要求を実行します。**servermain.JOB_COMPLETE** プロパティは、ジョブの現在の状態を Boolean として表します。このプロパティの値は、ジョブの実行が完了するまでは false のままです。何らかの理由でジョブが実行に失敗した場合、クライアントでは **servermain.JOB_STATUS_MSG** プロパティに適切なエラーメッセージが表示されます。

ジョブのクリーンアップ

ジョブは、設定可能な時間が経過すると、サーバーによって自動的に削除されます。デフォルトでは、ジョブが完了した後、クライアントでジョブが削除される前に 30 秒間操作可能な時間があります。クライアントが必要とする時間が長い場合、またはクライアントが低速の接続で動作している場合は、クライアントはサービスの起動時に **servermain.JOB_TIME_TO_LIVE_SECOND** パラメータを使用して、最大 5 分まで生存時間を増加することができます。各ジョブには独自の生存時間があり、ジョブが作成された後に変更することはできません。クライアントは、サーバーから手動でジョブを削除することは許可されていないので、削除前のジョブからの情報取得を損なうことのない最短生存時間の選択をお勧めします。

サービスの自動タグ生成

自動タグ生成サービスは、自動タグ生成をサポートするドライバーのデバイスのエンドポイントで動作します。デバイスの自動タグ生成をサポートするプロパティは、自動タグ生成を開始する前に設定する必要があります。関連するプロパティについては、ドライバー固有のドキュメンテーションを参照してください。

タグの自動生成を開始すると、指定された空のペイロードを持つ taggeneration エンドポイントに PUT が送信されます。次の例では、自動タグ生成が開始されています。
Channel1/Device1。

```
/config/v1/project/channels/Channel1/devices/Device1/services/TagGeneration を配置
```

応答は次のようになります。

```
{
  "code": 202,
  "message": "受け付けました",
  "href":
    "/config/v1/project/channels/Channel1/devices/Device1/services/TagGeneration/jobs/-
    job1"
}
```

これは要求が受け付けられてジョブが job1 として作成されたことを意味します。ジョブのステータスはジョブをクエリーすることによって確認できます。これはジョブのエンドポイントに GET を送信することで行います。GET 要求は以下のようになります。

```
/config/v1/project/channels/Channel1/devices/Device1/services/TagGeneration/jobs/j-
ob1 を取得
```

待機時間が経過すると、ジョブは自動的にクリーンアップされます。この待機時間は設定できます。詳細については、[ジョブのクリーンアップ](#) セクションを参照してください。

● **注記:** すべてのドライバーが自動タグ生成をサポートしているわけではありません。

サービスプロジェクトのロード

プロジェクトは、projectload エンドポイントの projectload サービスと対話することによって読み込むことができます。最初に get リクエストを送信して、後で PUT リクエストで使用するようプロジェクト ID を取得する必要があります。

● **注記:** このサービスは Linux プラットフォームではサポートされていません。

GET 要求は以下のようになります。

```
GET /config/v1/project/ProjectLoad
```

サーバーは以下のように応答します。

```
{
  "PROJECT_ID": 3531905431,
  "common.ALLTYPES_NAME": "ProjectLoad",
  "servermain.JOB_TIME_TO_LIVE_SECONDS": 30,
  "servermain.PROJECT_FILENAME": "",
  "servermain.PROJECT_PASSWORD": ""
}
```

プロジェクトのロードを開始するには、プロジェクトファイル、プロジェクトファイルのパスワード、およびプロジェクト ID への絶対パスを使用して、サーバーに PUT リクエストが送信されます。プロジェクトにパスワードがない場合、そのフィールドは必須ではありません。プロジェクトのロードでは、SOPF、opf、および JSON ファイルタイプがサポートされています。要求は以下のようになります。

```
PUT
http://localhost:57412/config/v1/project/channels/Channel1/devices/Device1/tags/My-
Tag
```

```
{
  "PROJECT_ID": 3531905431,
  "servermain.PROJECT_FILENAME": "/Absolute/Path/To/MyProject.json ",
  "servermain.PROJECT_PASSWORD": ""
}
```

サーバーは以下のように応答します。

```
{
  "code": 202,
  "message": "受け付けました",
  "href": "/config/v1/project/services/ProjectLoad/jobs/job1"
}
```

これは要求が受け付けられてジョブが job1 として作成されたことを意味します。ジョブのステータスはジョブをクエリーすることによって確認できます。これはジョブのエンドポイントに GET を送信することで行います。GET 要求は以下ようになります。

```
GET /config/v1/project/ProjectLoad/jobs/job1
```

待機時間が経過すると、ジョブは自動的にクリーンアップされます。この待機時間は設定できます。詳細については、[ジョブのクリーンアップ](#)セクションを参照してください。

サービスプロジェクトの保存

プロジェクトをロードするには、projectsave endpoint で projectsave service を操作します。後で PUT リクエストで使用するようにプロジェクト ID を取得するには、get リクエストを送信する必要があります。GET 要求は以下ようになります。

```
/config/v1/project/ProjectSave を取得
```

サーバーは以下のように応答します。

```
{
  "PROJECT_ID": 2401921849,
  "common.ALLTYPES_NAME": "ProjectSave",
  "servermain.JOB_TIME_TO_LIVE_SECONDS": 30,
  "servermain.PROJECT_FILENAME": ""
}
```

プロジェクトを開始するには、プロジェクトファイルのパスと拡張子 (SOPF、opf、または JSON) を含むファイルの名前、およびファイルを暗号化するためのパスワードとプロジェクト ID を使用して、PUT リクエストが送信されます。SOPF ファイルには password プロパティが必要ですが、それ以外の場合は無視されます。このパスは、アプリケーションデータフォルダを基準にしています。PUT 要求は以下ようになります。

```
PUT /config/v1/project/services/ProjectSave
```

```
{
  "PROJECT_ID": 2401921849,
  "servermain.PROJECT_FILENAME": "Projects/MyProject ", "servermain.PROJECT_PASSWORD": "MyPassword "}
```

サーバーは以下のように応答します。

```
{
  "code": 202,
  "message": "受け付けました",
  "href": "/config/v1/project/services/ProjectSave/jobs/job1"
}
```

これは要求が受け付けられてジョブが job1 として作成されたことを意味します。ジョブのステータスはジョブをクエリーすることによって確認できます。これはジョブのエンドポイントに GET を送信することで行います。GET 要求は以下ようになります。

```
GET /config/v1/project/ProjectLoad/jobs/job1
```

待機時間が経過すると、ジョブは自動的にクリーンアップされます。この待機時間は設定できます。詳細については、[ジョブのクリーンアップ](#)セクションを参照してください。

Config API サービス - 応答コード

以下の応答コードのいずれかが、REST 要求から返される可能性があります。可能な場合には、応答のボディに具体的なエラーメッセージが含まれています。これはエラーの原因と考えられる解決策を特定するために役立ちます。

- HTTP/1.1 200 OK (OK です)
- HTTP/1.1 201 Created (作成されました)
- HTTP/1.1 202 Accepted (受理されました)
- HTTP/1.1 207 Multi-Status (複数のステータスがあります)
- HTTP/1.1 400 Bad Request (不正な要求です)
- HTTP/1.1 401 Unauthorized (権限がありません)
- HTTP/1.1 403 Forbidden (禁止されています)
- HTTP/1.1 404 Not Found (見つかりません)
- HTTP/1.1 429 Too Many Requests (要求が多すぎます)
- HTTP/1.1 500 Internal Server Error (内部サーバーエラー)
- HTTP/1.1 503 Server Runtime Unavailable (サーバーランタイムを使用できません)
- HTTP/1.1 504 Gateway Timeout (ゲートウェイタイムアウト)
- HTTP/1.1 520 Unknown Error (不明なエラー)

● [構成 API サービスのイベントログメッセージ](#)を参照してください。

組み込み診断

通信の問題が発生した場合、問題の原因を特定するために OPC 診断とチャネル診断の両方を使用できます。これらのビューは、サーバーレベルとドライバレベルの両方の診断を提供します。診断はパフォーマンスに影響を与える可能性があるため、デバッグまたはトラブルシューティングを行うときにのみ使用することをお勧めします。詳細については、以下のリストからリンクを選択してください。

OPC 診断ビューア

チャネル診断

OPC 診断ビューア

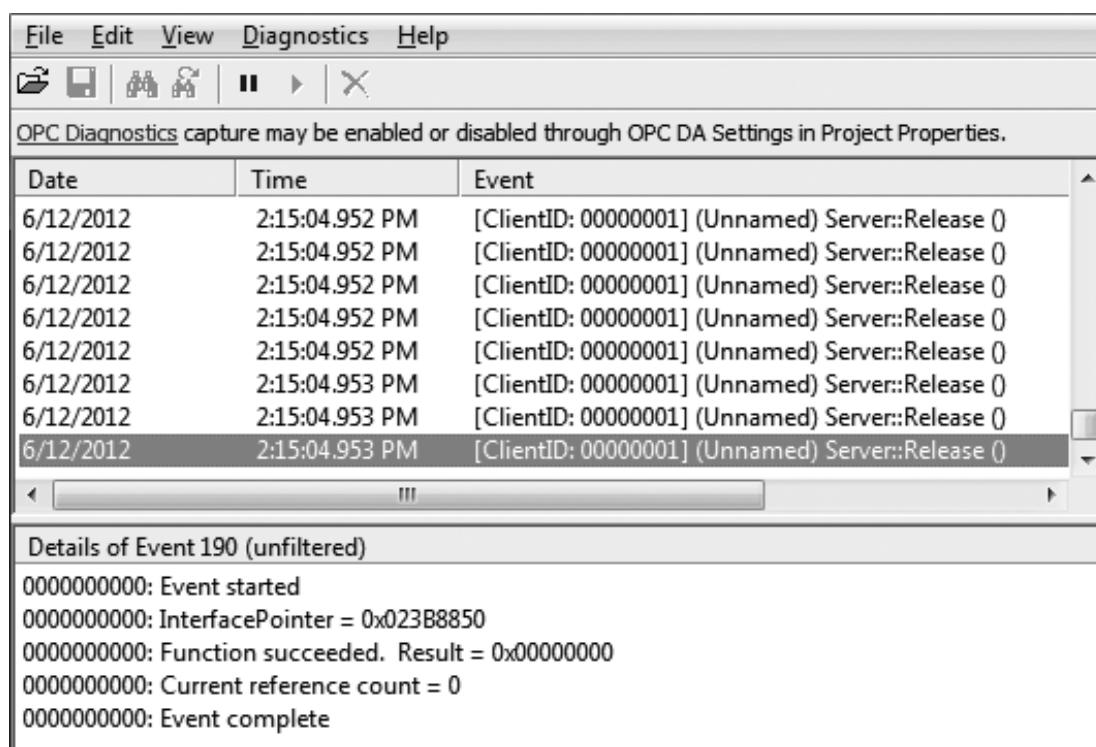
OPC 診断ビューアは、OPC クライアントとサーバーの間で発生する OPC イベントについて、リアルタイムのビューと履歴ビューを提供します。イベントとは、クライアントがサーバーに対して行うメソッド呼び出し、またはサーバーがクライアントに対して行うコールバックです。

OPC 診断ビューアへのアクセス

OPC 診断ビューアは、メインのサーバー設定ウィンドウからは分離されています。OPC 診断ビューアにアクセスするには、「表示」|「OPC 診断」の順にクリックします。

● **注記:** データの取り込みが無効になっている場合もビューアにアクセスすることはできますが、有効にするまで診断は表示されません。

● OPC の診断を有効にする方法については、[プロジェクトのプロパティ - OPC DA](#)、[プロジェクトのプロパティ - OPC UA 設定](#)、および [プロジェクトのプロパティ - OPC HDA](#) を参照してください。



● ログ設定のプロパティについては、[設定 - イベントログ](#)を参照してください。

ライブデータモード

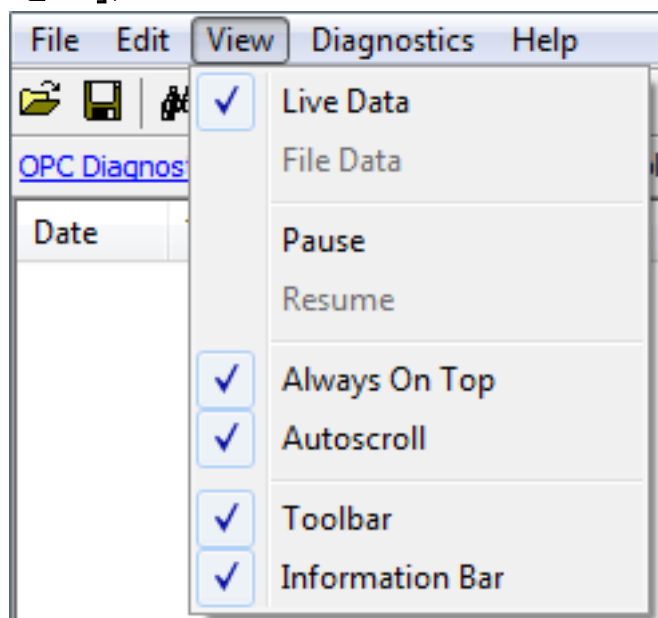
OPC 診断ビューアは、ライブデータモードで開かれます。このモードでは、イベントログから現時点で取得できる永続的な OPC 診断データが表示されます。ビューアの表示はリアルタイムで更新されます。表示を一時停止するには、「表示」|「一時停止」の順にクリックするか、「一時停止」アイコンを選択します。データは引き続き取り込まれていますが、表示は更新されなくなります。

● OPC 診断ファイルを保存するには、「ファイル」|「名前を付けて保存」の順にクリックし、「OPC 診断ファイル (*.opcdiag)」を選択します。

ファイルデータモード

OPC 診断ビューアでは、保存されている OPC 診断ファイルを開いて表示できます。保存済みのファイルを開くと、ビューアがファイルデータモードに切り替わり、ロードされたファイルの名前とデータが表示されます。モードは、「ビュー」メニューで切り替えることができます。ファイルを閉じると、ビューがライブデータに切り替わり、ファイルデータビューは、別のファイルをロードするまで利用できなくなります。

「ビュー」メニュー

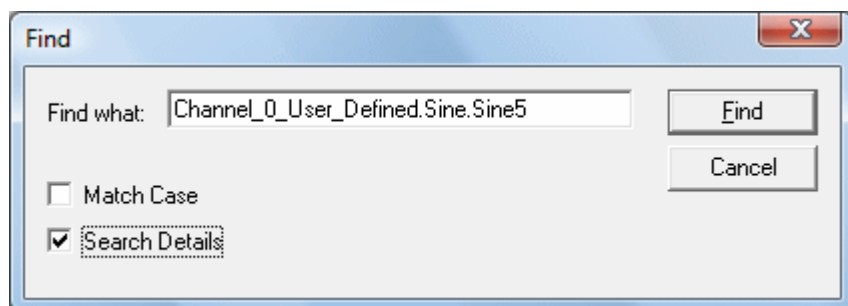


オプションの説明は次のとおりです。

- 「Live Data」有効にすると、現時点でイベントログから取得できる永続的な OPC 診断データが表示されるようになります。デフォルト設定では有効になっています。詳細については、[ライブデータモード](#)を参照してください。
- 「File Data」有効にすると、保存済みの OPC 診断ファイルに含まれているデータが表示されます。デフォルトでは無効に設定されています。詳細については、[ファイルデータモード](#)を参照してください。
- 「Always on Top」有効にすると、OPC 診断のウィンドウが、ほかのどのアプリケーションのウィンドウよりも手前に表示されたままになります。デフォルト設定では有効になっています。
- 「Autoscroll」有効にすると、新しいイベントの受信時に、表示内容がスクロールされ、常に直近のイベントが見えている状態になります。ユーザーが手動でイベントを選択すると(または、「検索」や「次を検索」でイベントが選択されると)、自動スクロールはオフになります。
- 「Toolbar」有効にすると、アイコンを取りまとめたツールバーが表示され、「ファイル」、「編集」、「表示」メニューで利用できるオプションに素早くアクセスできます。デフォルト設定では有効になっています。
- 「Information Bar」有効にすると、OPC 診断データの上部に、情報表示用のバーが表示されます。デフォルト設定では有効になっています。

検索

このダイアログでは、クライアントとサーバー間で転送された主要な情報をサーチできます。たとえば、このサーチ機能を使用して、特定のアイテム ID やグループ名に対する操作をすべて表示できます。



プロパティの説明は次のとおりです。

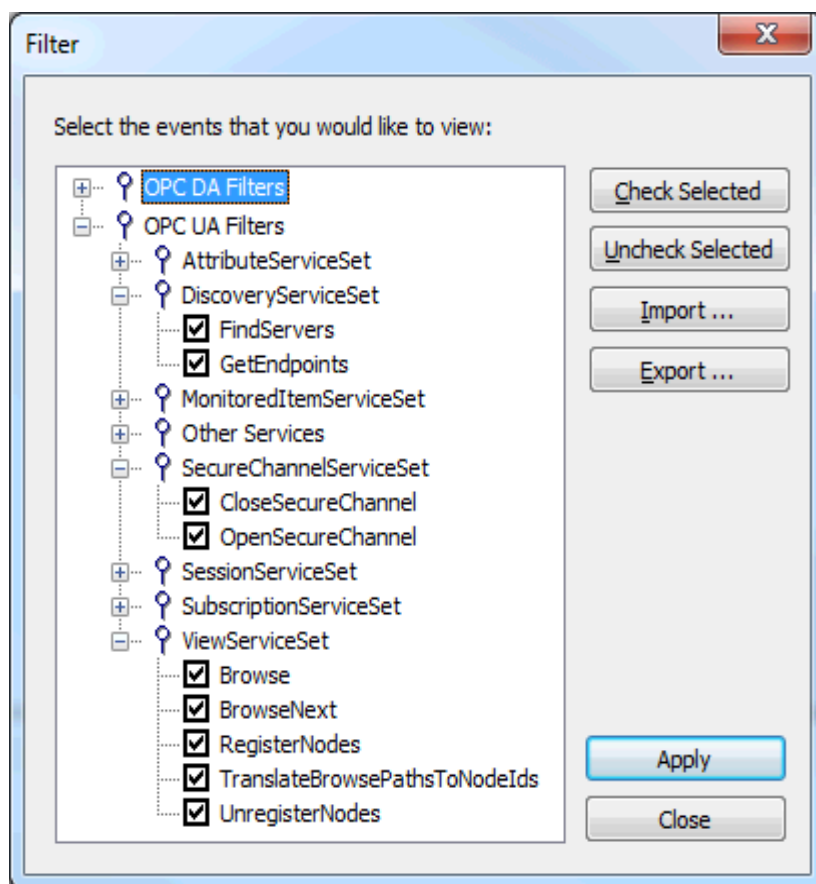
- 「Find What」このフィールドにサーチ基準を入力します。
- 「Match Case」有効にすると、サーチ基準として入力する文字列で、大文字と小文字が区別されます。
- 「Search Details」有効にすると、サーチ基準で詳細情報を指定できます。

● **注記:** 指定したテキストと一致するイベントまたは詳細が見つかったら、そのテキストが含まれている行がハイライトされます。「次を検索」操作を実行する(指定したテキストの次の一致を検索する)には、F3 キーを押します。基準に一致する最後のエンタリに達すると、メッセージボックスにその旨が表示されます。サーチ基準は、いつでも Ctrl + F キーを押して変更できます。

フィルタ

このダイアログボックスでは、OPC 診断ビューアに表示されるイベントを指定します。たとえば、ほとんどのクライアントは、GetStatus 呼び出しをサーバーに対して持続的に発行することで、サーバーが今も利用可能であるかどうかを判定しています。このイベントをフィルタすると、診断データのみを調査できるようになります。フィルタの適用対象は、取り込まれるデータではなくビューです。フィルタの設定にかかわらず、すべてのタイプのイベントが取り込まれています。また、このダイアログボックスが開かれている間にフィルタを適用できるため、設定を変更し、個別に適用できます。このダイアログボックスを閉じてもう一度開くことなく、変更を実行できます。

● **注記:** サーバーでサポートされているすべての OPC Data Access 1.0、2.0、3.0 インタフェースの各メソッドを、フィルタとして利用できます ("IOPCCommon" や "GetErrorString" など)。



オプションの説明は次のとおりです。

- 「Check Selected」: クリックすると、選択しているアイテムのすべてのイベントの表示が有効になります。デフォルトでは、すべてのインタフェースのすべてのメソッドが選択されています。
● 詳細については、[OPC DA イベント](#)と[OPC UA サービス](#)を参照してください。
- 「Uncheck Selected」: クリックすると、選択しているアイテムのすべてのイベントタイプおよびメソッドが無効になります。
- 「インポート」: クリックすると、フィルタにインポートする INI ファイルを選択できます。
- 「エクスポート」: クリックすると、フィルタを INI ファイルとしてエクスポートできます。

● 注記:

1. フィルタの設定は OPC 診断ビューアを閉じている間も引き続き適用されているため、後ほど、OPC 診断ファイルをもう一度開いて閲覧できます。ファイルデータモードで開いたファイルには、フィルタが適用されている場合があります。OPC 診断ビューアからファイルを保存するとき、保存されるのは、フィルタを適用した結果、表示されているイベントのみです。フィルタされていないデータファイルが必要な場合は、フィルタをオフにしてからファイルを保存してください。
2. 診断情報の取り込みは、クライアントとサーバーの間で通信が発生する付加的な処理レイヤーであるため、実行中は、サーバーのパフォーマンスが影響を受けます。また、OPC 診断のログを拡張データストア永続モードで作成すると、ディスク容量を大量に消費する可能性があります。関連するエラーは、Windows イベントビューアで報告されます。永続モードについては、[設定 - イベントログ](#)を参照してください。

OPC DA のイベント

個々の OPC 診断イベントの詳細については、次のリストに記載したリンク先を参照してください。

[IClassFactory](#)
[Server](#)
[IOPCCommon](#)
[IOPCServer](#)
[IConnectionPointContainer \(サーバー\)](#)
[IConnectionPoint \(サーバー\)](#)
[IOPCBrowse](#)
[IOPCBrowseServerAddressSpace](#)
[IOPCItemProperties](#)
[IOPCItemIO](#)
[Group](#)
[IOPCGroupStateMgt](#)
[IOPCGroupStateMgt2](#)
[IOPCItemMgt](#)
[IOPCItemDeadbandMgt](#)
[IOPCItemSamplingMgt](#)
[IOPCSyncIO](#)
[IOPCSyncIO2](#)
[IOPCAsyncIO](#)
[IDataObject](#)
[IAdviseSink](#)
[IAsyncIO2](#)
[IAsyncIO3](#)
[IConnectionPointContainer \(グループ\)](#)
[IConnectionPoint \(グループ\)](#)
[IOPCDataCallback](#)
[IEnumOPCItemAttributes](#)

IClassFactory

IClassFactory インタフェースには、オブジェクトクラス全体を取り扱うためのいくつかのメソッドが含まれています。特定のオブジェクトクラスのクラスオブジェクトに実装され、CLSID によって識別されます。

- **QueryInterface:** クライアントは、IConnectionPointContainer の QueryInterface を呼び出すことで、発信インタフェースをサポートしているかどうかをオブジェクトに問い合わせることができます。オブジェクトが、有効なポインタを返すことで "はい" と回答した場合、クライアントは、接続の確立を試行できることがわかります。
- **AddRef:** オブジェクトのインタフェースの参照カウントを増やします。所与のオブジェクトのインタフェースへのポインタの新しいコピーごとに、毎回呼び出されます。
- **Release:** インタフェースの参照カウントを 1 減らします。

- **CreateInstance:** 初期化されていないオブジェクトを作成します。
- **LockServer:** クラスオブジェクトのクライアントによって呼び出されたとき、インスタンスを短時間で作成し、サーバーをメモリ内でオープンに維持できます。

サーバー

クライアントは、CoCreateInstance を呼び出すことでサーバーオブジェクトと初期インタフェースを作成します。

- **QueryInterface:** クライアントは、IConnectionPointContainer の QueryInterface を呼び出すことで、発信インタフェースをサポートしているかどうかをオブジェクトに問い合わせることができます。オブジェクトが、有効なポインタを返すことで "はい" と回答した場合、クライアントは、接続の確立を試行できることがわかります。
- **AddRef:** オブジェクトのインタフェースの参照カウントを増やします。所与のオブジェクトのインタフェースへのポインタの新しいコピーごとに、毎回呼び出されます。
- **Release:** インタフェースの参照カウントを 1 減らします。

IOPCCCommon

このインタフェースは、すべての OPC サーバータイプで使用されます (DataAccess、Alarm&Event、履歴データなど)。個別のクライアント/サーバーセッションに対して有効になるロケール ID を設定し、クエリーする機能を提供します。あるクライアントの操作が、ほかのクライアントに影響を及ぼすことはありません。

- **GetErrorString:** サーバー固有のエラーコードについて、エラー文字列を返します。これには、Win32 エラー (RPC エラーなど) の取り扱いも含めることが推奨されます。
- **GetLocaleID:** このサーバー/クライアントセッションのデフォルトのロケール ID を返します。
- **QueryAvailableLocaleIDs:** このサーバー/クライアントセッションで利用できるロケール ID を返します。
- **SetClientName:** クライアントが、クライアント名をサーバーに登録できます (オプション)。このメソッドは、主にデバッグを目的として用意されています。ユーザーは、ここでノード名と EXE 名を設定することが推奨されます。
- **SetLocaleID:** このサーバー/クライアントセッションのデフォルトのロケール ID を設定します。このロケール ID は、このインタフェースの GetErrorString メソッドによって使用されます。サーバーのデフォルト値は LOCALE_SYSTEM_DEFAULT です。

IOPCServer

これは OPC サーバーのメインインタフェースです。OPC サーバーは、この仕様書のインストールおよび登録の章で取り上げた内容に沿って、オペレーティングシステムに登録されます。

- **AddGroup:** サーバーにグループを追加します。グループは、データアイテムを整理および操作するためのクライアントの論理コンテナです。
- **CreateGroupEnumerator:** サーバーによって提供されるグループの各種列挙子を作成します。
- **GetErrorString:** サーバー固有のエラーコードについて、エラー文字列を返します。
- **GetGroupByName:** (同一のクライアントが以前に作成した) プライベートグループの名前が指定されている場合、追加のインタフェースポインタを返します。パブリックグループに接続するには、GetPublicGroupByName を使用します。この関数を使用すると、インタフェースポインタがすでにすべてリリースされているプライベートグループに再接続できます。
- **GetStatus:** サーバーの現在のステータス情報を返します。
- **RemoveGroup:** グループを削除します。サーバー自体がグループへの参照を維持しているため、すべてのクライアントインタフェースがリリースされてもグループは削除されません。すべてのインタフェースがリリースされた後も、クライアントは引き続き GetGroupByName を呼び出すことができます。RemoveGroup() では、グループへの最後の参照がサーバーによってリリースされ、結果としてグループが削除されます。

IConnectionPointContainer (サーバー)

このインタフェースは、IOPCShutdown の接続ポイントへのアクセスを提供します。

- **EnumConnectionPoints:** OPC グループとクライアントの間でサポートされる接続ポイントの列挙子を作成します。OPC Servers は、IOPCShutdown を含む列挙子を返す必要があります。追加のベンダー固有コールバックを使用できます。
- **FindConnectionPoint:** OPC サーバーとクライアントの間でサポートされる個別の接続ポイントをサーチします。OPC Servers は、IID_IOPCShutdown をサポートしている必要があります。追加のベンダー固有コールバックを使用できます。

IConnectionPoint (サーバー)

このインタフェースは、クライアントへのコールバックを確立します。

- **Advise:** 接続ポイントと呼び出し元のシンクオブジェクトの間に、アドバイザリ接続を確立します。
- **EnumConnections:** この接続ポイントへの接続に対するイテレーション用に、列挙子オブジェクトを作成します。
- **GetConnectionInterface:** この接続ポイントによって管理されている発信インタフェースの IID を返します。
- **GetConnectionPointContainer:** 接続ポイントを概念上保有している接続可能オブジェクトへの IConnectionPointContainer インタフェースポインタを取得します。
- **Unadvise:** Advise メソッドによって確立されたアドバイザリ接続を終了します。
- **ShutdownRequest** サーバーが、すべてのクライアントをサーバーから切断することをリクエストできます。

IOPCBrowse

IOPCBrowse インタフェースは、サーバーアドレス空間のブラウズおよびアイテムプロパティの取得に関して、以前よりも優れたメソッドを提供します。

- **GetProperties:** アイテム ID ごとに 1 つずつ、OPCITEMPROPERTIES の配列を返します。
- **Browse:** アドレス空間の単一のブランチをブラウズし、0 個以上の OPCBROWSEELEMENT 構造体を返します。

IOPCBrowseServerAddressSpace

このインタフェースは、サーバー内の利用可能データアイテムをクライアントがブラウズするための手段を提供します。ユーザーには、アイテム ID に関する有効な定義のリストが提示されます。フラット構造または階層構造のアドレス空間が考慮され、ネットワーク全体にわたって適切に機能するよう設計されています。また、クライアントは、サーバーベンダー固有のアイテム ID 構文から解放されます。

- **BrowseAccessPaths:** アイテム ID について、使用可能な AccessPaths をブラウズするための手段を提供します。
- **BrowseOPCItemIDs:** 渡されたプロパティによって特定されるアイテム ID のリストについて、IENUMString を返します。ブラウズの実行開始位置は、ChangeBrowsePosition で設定できます。
- **ChangeBrowserPosition:** 階層空間を上下移動する、または階層空間に入るための手段を提供します。
- **GetItemID:** 完全修飾のアイテム ID を階層空間に作成するための手段を提供します。このメソッドが必要になるのは、ブラウズ関数で返されるのはアイテム ID を構成するコンポーネントまたはトークンのみであり、それらのトークンを区切るために使用される区切り記号は、返されないためです。また、各ポイントでは、現在のノードの下にある名前をブラウズしているにすぎません (たとえば、セルに含まれているユニット)。
- **QueryOrganization:** 基盤となっているシステムがフラット構造と階層構造のどちらであるのか、サーバーがアドレス空間の情報をクライアントにどのように提示するのかを特定するための手段を提供します。フラットな構造の空間と階層構造の空間は、挙動がやや異なります。結果がフラットである場合、クライアントは、Branch フラグまたは Leaf フラグを BrowseOPCItem ID に引き渡す必要がないこと、または ChangeBrowsePosition を呼び出す必要がないことがわかります。

IOPCItemProperties

このインタフェースを使用すると、アイテム ID に関連付けられている使用可能プロパティをブラウズすることに加え、プロパティの現在の値を読み取ることができます。

- **GetItemProperties:** 渡された ID コードについて、現在のデータ値のリストを返します。
- **LookupItemIDs:** 渡された各 ID コードに関して、使用可能なものがあれば、アイテム ID のリストを返します。これらは、OPC グループに追加して、アイテムプロパティに対応するデータに対し、より効率的にアクセスするために使用できるアイテム ID です。
- **QueryAvailableProperties:** このアイテム ID で取得できるプロパティに関して、ID コードと説明のリストを返します。このリストは、アイテム ID ごとにそれぞれ異なる場合があります。このリストは、アイテム ID ごとに見ると変化が比較的少ないものと考えられますが、基盤となっているシステムの構成が変更された場合は影響を受ける可能性があります。アイテム ID がこの関数に渡されるのは、サーバーが、アイテム ID に応じてそれぞれ異なる一連のプロパティを返すことを許可されているためです。

IOPCItemIO

このインタフェースの目的は、基本的なアプリケーションに対して、OPC データを簡単に取得するための手段を提供することです。

- **Read:** 指定したアイテムの 1 つ以上の値、品質、タイムスタンプを読み取ります。機能上は、IOPCSyncIO::Read メソッドに似ています。
- **WriteVQT:** 指定したアイテムについて、1 つ以上の値、品質、タイムスタンプを書き込みます。機能上は、関連付けられたグループが存在しないことを除くと、IOPCSyncIO2::WriteVQT メソッドに似ています。クライアントが

VQ、VT、またはVQTを書き込もうとする場合、クライアントは、サーバーがすべてを書き込むか、一切書き込まないことを期待します。

グループ

クライアントは、CoCreateInstance を呼び出すことでサーバーオブジェクトと初期インタフェースを作成します。

- **QueryInterface:** クライアントは、IConnectionPointContainer の QueryInterface を呼び出すことで、発信インタフェースをサポートしているかどうかをオブジェクトに問い合わせることができます。オブジェクトが、有効なポインタを返すことで "はい" と回答した場合、クライアントは、接続の確立を試行できることがわかります。
- **AddRef:** オブジェクトのインタフェースの参照カウントを増やします。所与のオブジェクトのインタフェースへのポインタの新しいコピーごとに、毎回呼び出されます。
- **Release:** インタフェースの参照カウントを 1 減らします。

IOPCGroupStateMgt

IOPCGroupStateMgt を使用すると、クライアントでグループの状態全般を管理できます。主に、この管理によってグループの更新頻度とアクティブ状態に変更が加えられます。

- **CloneGroup:** グループの 2 番目のコピーを一意的な名前で作成します。
- **GetState:** グループの現在の状態を取得します。通例、この関数は、SetState を呼び出す前に、この情報の現在の値を取得するために呼び出します。この情報は、グループの作成時に、すべてクライアントによって提供されたか、クライアントに返されたものです。
- **SetName:** プライベートグループの名前を変更します。名前は一意でなければなりません。パブリックグループの名前を変更することはできません。グループ名は、クライアントからサーバーへの個々の接続に関して、一意なものにする必要があります。
- **SetState:** グループのさまざまなプロパティを設定します。これは、元のグループとは無関係の新しいグループを表します。

IOPCGroupStateMgt2

このインタフェースは、既存の IOPCGroupStateMgt インタフェースの機能を拡張するために追加されたものです。

- **SetKeepAlive:** レポートの対象となる新たなイベントが存在していない場合に、サブスクリプションに対するクライアントコールバックがサーバーから提供されます。クライアントは、GetStatus() への呼び出しを使用してサーバーに ping を実行することなく、サーバーとサブスクリプションの状態を確認できます。
- **GetKeepAlive:** 現在有効になっている、サブスクリプションのキープアライブ時間を返します。

IOPCItemMgt

このインタフェースを使用すると、グループへのアイテムの追加、グループに含まれているアイテムの除去、グループに含まれているアイテムの挙動の制御をクライアントが実行できます。

- **AddItems:** 1 つ以上のアイテムをグループに追加します。同じアイテムをグループに複数回追加すると、一意の ServerHandle を持つ 2 番目のアイテムを生成できます。
- **CreateEnumerator:** グループに含まれているアイテムの列挙子を作成します。
- **RemoveItems:** アイテムをグループから除去します。アイテムをグループから除去しても、サーバーまたは物理デバイスのアドレス空間には影響しません。グループは、クライアントがそれらの個々のアイテムに関心を持っているかどうかを示します。
- **SetActiveState:** グループに含まれている 1 つ以上のアイテムをアクティブまたは非アクティブに設定します。これにより、有効なデータをそれらのアイテムの読み取りキャッシュから取得できるかどうか、さらに、グループに対する IAdvise サブスクリプションにそれらのアイテムを含めるかどうかを制御します。非アクティブなアイテムに関してはコールバックそのものが発生しないため、アイテムを非アクティブにしてもコールバックは発生しません。一般に、アイテムをアクティブにすると、次の UpdateRate 期間中に IAdvise コールバックが発生します。
- **SetClientHandles:** グループに含まれている 1 つ以上のアイテムのクライアントハンドルを変更します。一般に、クライアントがアイテムの追加時にクライアントハンドルを設定し、以後はクライアントハンドルを変更しないことをお勧めします。
- **SetDataTypes:** グループに含まれている 1 つ以上のアイテムについて、リクエストしたデータ型を変更します。一般に、クライアントがアイテムの追加時にリクエストしたデータ型を設定し、以後はデータ型を変更しないことをお勧めします。
- **ValidateItems:** アイテムが有効なものであり、エラーなく追加できるかどうかを特定します。正規データ型など、アイテムに関する情報も返します。グループに対しては一切作用しません。

IOPCItemDeadbandMgt

グループに含まれているすべてのアクティブなアイテムに関して、変更されているかどうかにかかわらず、IOPCDataCallback::OnDataChange へのコールバックを強制します。非アクティブなアイテムはコールバックに含まれません。データが取得されるかどうかは、MaxAge 値によって決まります。MaxAge 値は 1 つしか存在せず、その値によって、グループに含まれているすべてのアクティブなアイテムの MaxAge が決まります。つまり、キャッシュから取得される値が存在する一方で、キャッシュ内のデータの "鮮度" によっては、デバイスから取得される値も存在します。

- **SetItemDeadband:** グループに対して指定されているデッドバンドをアイテムごとにオーバーライドします。
- **GetItemDeadband:** リクエストした各アイテムのデッドバンド値を取得します。
- **ClearItemDeadband:** 個々のアイテムの PercentDeadband をクリアし、事実上、グループで設定されているデッドバンド値に戻します。

IOPCItemSamplingMgt

このオプションインタフェースを使用すると、グループに含まれている個々のアイテムを、基盤となっているデバイスから取得する頻度をクライアントで操作できます。OnDataChange のコールバックに関して、グループで設定されている更新頻度には影響しません。

- **SetItemSamplingRate:** 個々のアイテムのサンプル採取頻度を設定します。この値によって、基盤となっているデバイスからの収集に関するかぎり、グループで設定されている更新頻度がオーバーライドされます。個々のアイテムに関連付けられている更新頻度は、コールバック期間には影響しません。
- **GetItemSamplingRate:** SetItemSamplingRate で設定された、個々のアイテムのサンプル採取頻度を取得します。
- **ClearItemSamplingRate:** SetItemSamplingRate で設定された、個々のアイテムのサンプル採取頻度をクリアします。アイテムに対して設定されていた頻度は、グループで設定されている更新頻度に戻ります。
- **SetItemBufferEnable:** グループで設定されている更新頻度よりも更新頻度が高いアイテムについて収集される、識別済みアイテムのデータに関して、Enable プロパティの値に応じてバッファ処理をサーバーでオン/オフすることをリクエストします。
- **GetItemBufferEnable:** リクエストしたアイテムに関して、サーバーによるバッファ処理の現在の状態をクエリーします。

IOPCSyncIO

IOPCSyncIO を使用すると、クライアントからサーバーに対して、同期的な読み取り操作および書き込み操作を実行できます。操作は完了するまで実行されます。

- **Read:** グループに含まれている 1 つ以上のアイテムについて、値、品質、タイムスタンプの情報を読み取ります。読み取りを最後まで実行してから結果が返されます。データは、グループで設定されている UpdateRate およびデッドバンド率に収まっていて正確なものとなる場合、キャッシュから読み取ることができます。物理デバイスからの実際の読み取りを実行する必要がある場合は、デバイスから読み取ることができます。キャッシュおよびデバイスからの読み取りを実行する実装そのものは、仕様では定義されていません。
- **Write:** グループに含まれている 1 つ以上のアイテムに値を書き込みます。関数は完了するまで実行されます。値の書き込み先はデバイスです。つまり、デバイスが実際にデータを受け入れまたは拒否したことを関数が検証するまで、結果は返されません。書き込みは、グループまたはアイテムがアクティブな状態であっても影響を受けません。

IOPCSyncIO2

このインタフェースは、既存の IOPCSyncIO インタフェースの機能を拡張するために追加されたものです。

- **ReadMaxAge:** 指定したアイテムについて、1 つ以上の値、品質、タイムスタンプを読み取ります。機能上は、読み取り元 (デバイスまたはキャッシュ) が指定されないことを除くと、OPCSyncIO::Read メソッドに似ています。情報をデバイスまたはキャッシュのどちらから取得するかは、サーバーが決定します。この決定の基準となるのは、MaxAge プロパティです。キャッシュに保持されている情報が MaxAge 以内のものである場合、データはキャッシュから取得されます。その他の場合、サーバーは、リクエストされた情報をデバイスにアクセスして取得する必要があります。
- **WriteVQT:** 指定したアイテムについて、1 つ以上の値、品質、タイムスタンプを書き込みます。機能上は、品質とタイムスタンプが書き込まれる場合があることを除くと、IOPCSyncIO::Write に似ています。クライアントが VQ、VT、または VQT を書き込もうとする場合、クライアントは、サーバーがすべてを書き込むか、一切書き込まないことを期待します。

IOPCAsyncIO

IOPCAsyncIO を使用すると、クライアントからサーバーに対して、非同期での読み取り操作と書き込み操作を実行できます。クライアントが動作を継続できるよう、操作はキューに格納され、関数はただちに結果を返します。各操作はトランザクションとして扱われ、トランザクション ID が関連付けられます。操作が完了すると、クライアントの IAdvise Sink にコー

ルバックが実行されます (この接続が確立されている場合)。コールバックに含まれている情報は、トランザクション ID とエラー結果を示しています。慣例上、0 は無効なトランザクション ID です。

- **Cancel:** 未処理のトランザクションをサーバーがキャンセルするようリクエストします。
- **Read:** グループに含まれている 1 つ以上のアイテムを読み取ります。結果は、IDataObject を使用して確立された IAdvise Sink 接続を通じて返されます。キャッシュの読み取りの場合、データは、グループとアイテムの両方がアクティブになっている場合に限り、有効なものになります。デバイスの読み取りは、グループやアイテムのアクティブ状態の影響を受けることはありません。
- **Refresh:** グループに含まれているすべてのアクティブなアイテムについて、変更の有無にかかわらずコールバックを強制します。非アクティブなアイテムはコールバックに含まれません。
- **Write:** 1 つ以上のアイテムをグループに書き込みます。結果は、IDataObject を使用して確立された IAdviseSink 接続を通じて返されます。

IDataObject

IDataObject は、個々のアイテムではなく OPCGroup に実装されます。したがって、クライアントとグループの間で OPC データストリームフォーマットを使用してアドバイザリ接続を作成し、データ転送を効率化できます。

- **DAdvise:** OPC グループとクライアントの間で、特定のストリームフォーマットの接続を作成します。
- **DUnadvise:** OPC グループとクライアントの間の接続を終了します。

IAdviseSink

クライアントは、OnDataChange の完全な実装を提供するだけで済みます。

- **OnDataChange:** このメソッドは、例外ベースのデータ変更、非同期の読み取りと更新、非同期の書き込みの完了に関して、OPC グループからの通知を処理するクライアントによって提供されます。

IAsyncIO2

このインタフェースは IOPCAsync (OPC 1.0) に似たもので、IOPCAsyncIO を置き換えることを目的としています。OPC 2.05 で追加されました。

- **Cancel2:** サーバーが未処理のトランザクションをキャンセルするようリクエストします。
- **GetEnable:** SetEnable で設定された、直近の Callback Enable 値を取得します。
- **Read:** グループに含まれている 1 つ以上のアイテムを読み取ります。結果は、サーバーの IConnectionPointContainer を使用して確立された、クライアントの IOPCDataCallback 接続を通じて返されます。読み取り元はデバイスであり、グループやアイテムのアクティブ状態の影響を受けることはありません。
- **Refresh2:** グループに含まれているすべてのアクティブなアイテムについて、変更の有無にかかわらず IOPCDataCallback::OnDataChange へのコールバックを強制します。非アクティブなアイテムはコールバックに含まれません。
- **SetEnable:** OnDataChange のオペレーションを制御します。Enable を False に設定すると、トランザクション ID 0 を使用しての (更新の結果ではない) OnDataChange コールバックが無効になります。グループ作成時のこの変数の初期値は、True です。つまり、OnDataChange コールバックはデフォルトで有効になっています。
- **Write:** 1 つ以上のアイテムをグループに書き込みます。結果は、サーバーの IConnectionPointContainer を使用して確立された、クライアントの IOPCDataCallback 接続を通じて返されます。

IAsyncIO3

このインタフェースは、既存の IOPCAsyncIO2 インタフェースの機能を拡張するために追加されたものです。

- **ReadMaxAge:** 指定したアイテムについて、1 つ以上の値、品質、タイムスタンプを読み取ります。機能上は、非同期であり、読み取り元 (デバイスまたはキャッシュ) が指定されないことを除くと、OPCSyncIO::Read メソッドに似ています。情報をデバイスまたはキャッシュのどちらから取得するかは、サーバーが決定します。この決定の基準となるのは、MaxAge プロパティです。キャッシュに保持されている情報が MaxAge 以内のものである場合、データはキャッシュから取得されます。その他の場合、サーバーは、リクエストされた情報をデバイスにアクセスして取得する必要があります。
- **WriteVQT:** 指定したアイテムについて、1 つ以上の値、品質、タイムスタンプを書き込みます。結果は、サーバーの IConnectionPointContainer を使用して確立された、クライアントの IOPCDataCallback 接続を通じて返されます。機能上は、品質とタイムスタンプが書き込まれる場合があることを除くと、IOPCAsyncIO2::Write に似ています。クライアントが VQ、VT、または VQT を書き込もうとする場合、クライアントは、サーバーがすべてを書き込むか、一切書き込まないことを期待します。
- **RefreshMaxAge:** グループに含まれているすべてのアクティブなアイテムについて、変更の有無にかかわらず、IOPCDataCallback::OnDataChange へのコールバックを強制します。非アクティブなアイテムはコールバックに含まれません。データが取得されるかどうかは、MaxAge 値によって決まります。MaxAge 値は 1 つしか存在せず、

その値によって、グループに含まれているすべてのアクティブなアイテムの MaxAge が決まります。つまり、キャッシュから取得される値が存在する一方で、キャッシュ内のデータのタイプによっては、デバイスから取得される値も存在します。

IConnectionPointContainer (グループ)

このインタフェースは IDataObject に似た機能を提供しますが、より実装が容易で、理解しやすいものになっています。また、IDataObject インタフェースにはない機能も提供します。クライアントは、このインタフェースを使用して確立された接続を通じて通信する場合、新しい IOPCAsyncIO2 インタフェースを使用する必要があります。従来の IOPCAsync 接続は、以前と同様に、IDataObject 接続を通じた通信を実行します。

- **EnumConnectionPoints**: OPC グループとクライアントの間でサポートされる接続ポイントの列挙子を作成します。
- **FindConnectionPoint**: OPC グループとクライアントの間に存在する特定の接続ポイントをサーチします。

IConnectionPoint (グループ)

このインタフェースは、クライアントへのコールバックを確立します。

- **Advise**: 接続ポイントと呼び出し元のシンクオブジェクトの間に、アドバイザリ接続を確立します。
- **EnumConnections**: この接続ポイントへの接続に対するイテレーション用に、列挙子オブジェクトを作成します。
- **GetConnectionInterface**: この接続ポイントによって管理されている発信インタフェースの IID を返します。
- **GetConnectionPointContainer**: 接続ポイントを概念上保有している接続可能オブジェクトへの IConnectionPointContainer インタフェースポインタを取得します。
- **Unadvise**: Advise メソッドによって確立されたアドバイザリ接続を終了します。

IOPCDataCallback

クライアントで接続ポイントを使用するには、IUnknown インタフェースと IOPCDataCallback インタフェースの両方をサポートするオブジェクトを作成する必要があります。

- **OnDataChange**: このメソッドは、例外ベースのデータ変更と更新に関して、OPC グループからの通知を処理するクライアントによって提供されます。
- **OnReadComplete**: このメソッドは、非同期の読み取りの完了に関して、OPC グループからの通知を処理するクライアントによって提供されます。
- **OnWriteComplete**: このメソッドは、AsyncIO2 による書き込みの完了に関して、OPC グループからの通知を処理するクライアントによって提供されます。
- **OnCancelComplete**: このメソッドは、非同期のキャンセルの完了に関して、OPC グループからの通知を処理するクライアントによって提供されます。

IEnumOPCItemAttributes

IEnumOPCItemAttributes を使用すると、グループのコンテンツおよびそれらのアイテムの属性をクライアントでサーチできます。返される情報のほとんどは、AddItem の呼び出し時にクライアントによって提供されたか、クライアントに返されたものです。

- **Clone**: 列挙子の 2 番目のコピーを作成します。新しい列挙子は、当初、現在の列挙子と同じ状態になっています。
- **Next**: 次の 'celt' アイテムをグループからフェッチします。
- **Reset**: 列挙子をリセットして最初のアイテムに戻します。
- **Skip**: 次の 'celt' 属性をスキップします。

● 接続ポイントに関する一般原則の詳細については、Microsoft 社のドキュメンテーションを参照してください。

OPC UA サービス

個々の OPC 診断イベントの詳細については、次のリストに記載したリンク先を参照してください。

[AttributeServiceSet](#)

[DiscoveryServiceSet](#)

[MonitoredItemServiceSet](#)

[OtherServices](#)

[SecureChannelServiceSet](#)

[SessionServiceSet](#)

[SubscriptionServiceSet](#)

[ViewServiceSet](#)

AttributeServiceSet

このサービスセットは、ノードの一部である属性にアクセスするサービスを提供します。

- **Read:** このサービスは、1 つ以上のノードの 1 つ以上の属性を読み取る際に使用します。配列などの、インデックス付きの要素から成る構造型属性値の場合、クライアントはこのサービスを使用することで、インデックス付きの値のセット全体を複合要素として読み取ったり、個々の要素を読み取ったり、複合要素の特定の範囲の要素を読み取ったりすることができます。
- **Write:** このサービスは、1 つ以上のノードの 1 つ以上の属性に値を書き込む際に使用します。配列などの、インデックス付きの要素から成る構造型属性値の場合、クライアントはこのサービスを使用することで、インデックス付きの値のセット全体を複合要素として書き込んだり、個々の要素を書き込んだり、複合要素の特定の範囲の要素を書き込んだりすることができます。

DiscoveryServiceSet

このサービスセットは、サーバーによって実装されているエンドポイントを検出したり、それらのエンドポイントのセキュリティ構成を読み取ったりする際に使用します。

- **FindServers:** このサービスは、サーバーまたは検出サーバーによって認識されているサーバーを返します。
- **GetEndpoints:** このサービスは、サーバーによってサポートされているエンドポイントと、安全なチャネルとセッションを確立するために必要なすべての構成情報を返します。

MonitoredItemServiceSet

このサービスセットを使用することで、クライアントはデータとイベントをサブスクリプションするモニターアイテムを定義できます。モニターアイテムごとに、モニターするアイテムと、通知の送信に使用するサブスクリプションを指定します。モニターするアイテムはどのようなノード属性でも構いません。

- **CreateMonitoredItems:** このサービスは、1 つ以上のモニターアイテムを作成してサブスクリプションに追加するときに使用します。サブスクリプションが削除されると、モニターアイテムはサーバーによって自動的に削除されます。
- **DeleteMonitoredItems:** このサービスは、サブスクリプションの 1 つ以上のモニターアイテムを除去するときに使用します。モニターアイテムを削除すると、そのトリガーアイテムリンクも削除されます。
- **ModifyMonitoredItems:** このサービスはサブスクリプションのモニターアイテムを修正するときに使用します。モニターアイテムの設定に対する変更はサーバーによってただちに適用されます。
- **SetMonitoringMode:** このサービスは、サブスクリプションの 1 つ以上のモニターアイテムのモニターモードを設定するときに使用します。モードを無効に設定すると、キューに入っているすべての通知が削除されます。
- **SetTriggering:** このサービスは、トリガーアイテムのトリガーリンクを作成および削除するときに使用します。デフォルトでモニターモードによって許可されていない場合、トリガーアイテムとそのリンクによってモニターアイテムがサンプルをレポートするようになります。

OtherServices

OtherServices はさまざまなサービスと通知を表します。

- **ServiceFault:** サービスが失敗するといつでもこの応答が返されます。
- **Unsupported:** これらのサービスはこのサーバーではサポートされていません。

SecureChannelServiceSet

このサービスセットは通信チャネルを開くときに使用するサービスを定義し、サーバーとの間で交換されるすべてのメッセージの機密性と整合性を確保します。

- **CloseSecureChannel:** このサービスはセキュアチャネルを閉じる際に使用します。
- **OpenSecureChannel:** このサービスはセキュアチャネルを開いたり更新したりするときに使用し、これを使用することでセッション中のメッセージ交換の機密性と整合性を確保できます。このサービスでは、メッセージが送受信される際に各種セキュリティアルゴリズムをメッセージに適用するため、通信スタックが必要になります。

SessionServiceSet

このサービスセットは、セッションのコンテキストでアプリケーションレイヤー接続を確立するためのサービスを定義します。

- **ActivateSession:** クライアントはこのサービスを使用して、セッションに関連付けるユーザーの識別情報を指定します。
- **Cancel:** このサービスは、未処理のサービス要求をすべてキャンセルするときに使用します。正常にキャンセルされたサービス要求は `Bad_RequestCancelledByClient ServiceFaults` を返します。
- **CloseSession:** このサービスはセッションを終了するときに使用します。
- **CreateSession:** クライアントはこのサービスを使用してセッションを作成し、サーバーはそのセッションを一意に識別する 2 つの値を返します。1 つ目の値は、サーバーのアドレス空間でそのセッションを識別する `sessionId` です。2 つ目は、受信する要求をセッションに関連付ける `authenticationToken` です。

SubscriptionServiceSet

サブスクリプションは、モニターアイテムからクライアントに通知をレポートするときに使用されます。

- **CreateSubscription:** このサービスはサブスクリプションを作成するときに使用します。サブスクリプションは一連のモニターアイテムの通知をモニターし、パブリッシング要求に応答して通知をクライアントに返します。
- **DeleteSubscriptions:** このサービスは、クライアントのセッションに属する 1 つ以上のサブスクリプションを削除するときに呼び出します。このサービスが正常に完了すると、そのサブスクリプションを使用しているモニターアイテムがすべて削除されます。
- **ModifySubscription:** このサービスはサブスクリプションを修正するときに使用します。
- **Publish:** このサービスは 2 つの目的に使用します。1 つ目として、1 つ以上のサブスクリプションの通知メッセージを受信したことを確認するときに使用します。2 つ目として、通知メッセージまたはキープアライブメッセージを返すようサーバーに要求するときに使用します。パブリッシング要求は特定のサブスクリプションに宛てたものではなく、すべてのサブスクリプションで使用できます。
- **Republish:** このサービスは、サブスクリプションに対して、通知メッセージをその再転送キューから再パブリッシングするよう要求します。
- **SetPublishingMode:** このサービスは、1 つ以上のサブスクリプションで通知の送信を有効または無効にするときに使用します。
- **TransferSubscriptions:** このサービスは、サブスクリプションとそのモニターアイテムをあるセッションから別のセッションに転送するときに使用します。

ViewServiceSet

クライアントはこのサービスセットのブラウズサービスを使用してアドレス空間をナビゲートします。

- **Browse:** このサービスは、指定したノードの参照を検出するときに使用します。ブラウズサービスでは基本的なフィルタ機能もサポートされます。
- **BrowseNext:** このサービスは、Browse または BrowseNext の応答情報が大きすぎて 1 回の応答では送信できない場合に、次の情報セットを要求するときに使用します。この場合の "大きすぎる" とは、サーバーがそれ以上大きな応答を返すことができないか、返される結果の数が、最初のブラウズ要求でクライアントが指定した返される結果の最大数を超過していることを意味しています。
- **RegisterNodes:** クライアントはこのサービスを使用して、繰り返しアクセスする (Write、Read など) ことがわかっているノードを登録できます。これによって、サーバーはアクセス処理をさらに効率化するのに必要な設定を行うことができます。
- **TranslateBrowsePathsToNodeIds:** このサービスは、サーバーが 1 つ以上のブラウズパスをノード ID に変換するよう要求するときに使用します。各ブラウズパスは開始ノードと相対パスから成ります。指定した開始ノードから相対パスが開始します。相対パスには参照タイプとブラウズ名が連続して含まれています。
- **UnregisterNodes:** このサービスは、RegisterNodes サービスから取得されたノード ID を登録解除するときに使用します。

● 接続ポイントに関する一般原則の詳細については、Microsoft 社のドキュメンテーションを参照してください。

通信診断

サーバーの診断機能は、通信ドライバーのパフォーマンスに関するリアルタイム情報を提供します。すべての読み取り/書き込み操作を診断ビューアで表示したり、組み込み診断タグを使用して OPC クライアントアプリケーションで直接追跡したりできます。診断ビューアには、主要な通信パラメータ設定 (ボーレート、パリティ、デバイス ID など) を変更するときに便利なリアルタイムプロトコルビューも表示されます。変更の影響がリアルタイムで表示されます。正しい通信およびデバイス設定が設定されると、デバイスとのデータ交換が表示されるようになります。

通信診断の有効化

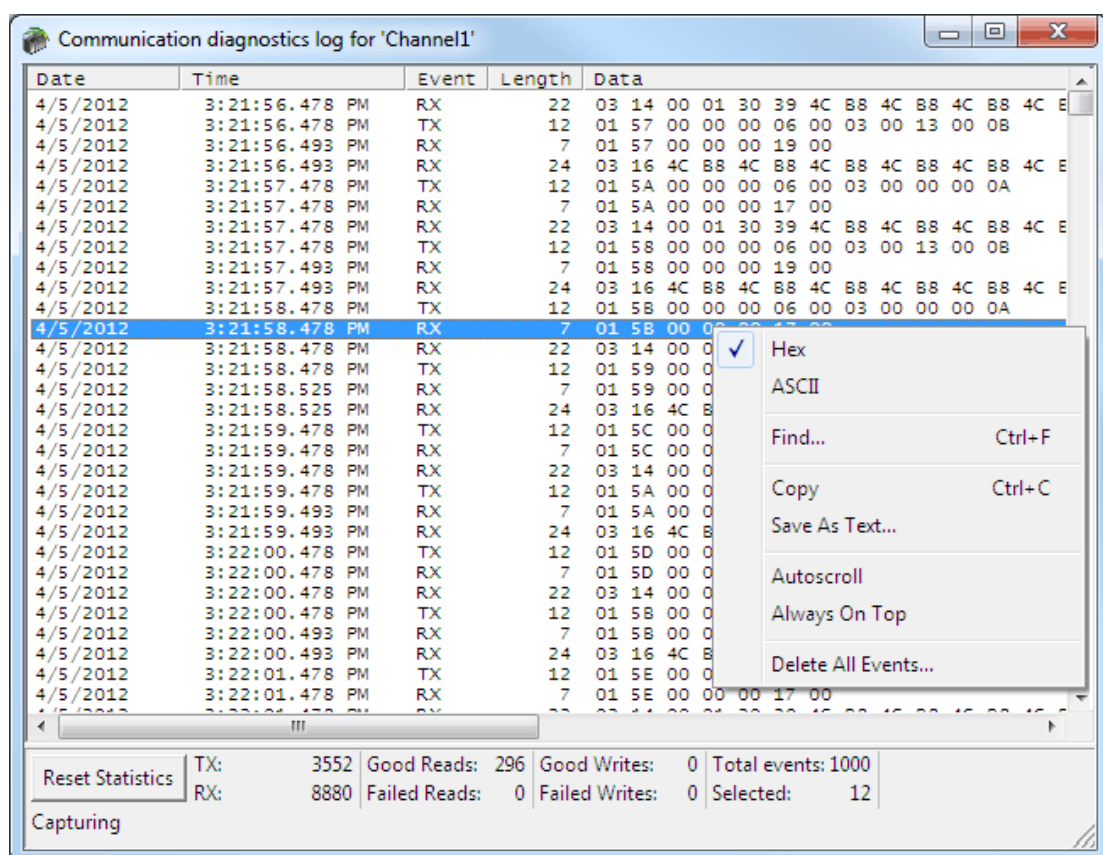
通信診断を有効にするには、Project View でチャンネルを右クリックし、「プロパティ」|「診断を有効にする」の順にクリックします。または、チャンネルをダブルクリックし、「Enable Diagnostics」を選択します。診断を有効にできるのはチャンネルの作成後です。

● 関連項目: [チャンネルのプロパティ - 一般](#)

通信診断ビューアへのアクセス

通信診断ビューアにアクセスするには、Project View でチャンネルまたはデバイスを右クリックし、「診断」を選択します。あるいは、チャンネルまたはデバイスを選択し、「表示」|「通信診断」の順にクリックします。通信診断ビューアはモードなしのフォームで動作するため、サーバーでほかのダイアログが開かれていても使用できます。このビューアを開くと、リアルタイムプロトコルデータの取り込みが開始されます。通信が適切に行われていれば、サーバーとデバイス間の通信メッセージのストリームが表示されます。TX および RX イベントだけではなくイベント合計数も確認できます。

● 注記: 取り込みが無効になっているときに通信診断ビューアを開くことはできますが、取り込みを有効にするまで診断は行われません。有効になっているときには、ビューアに「取り込んでいます」と表示されます。無効になっているときには、ビューアに「診断の取り込みが無効になっています」と表示されます。



統計をリセット

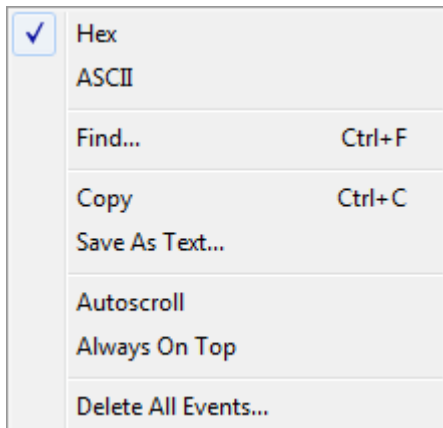
「統計をリセット」をクリックすると、「TX」、「RX」、「成功した読み取り数」、「失敗した読み取り数」、「成功した書き込み数」、および「失敗した書き込み数」の数が0に設定されます。「イベント合計」は、ビューア内の実際のイベントの数を指定するため、0に設定されません。

● ログ設定については、[設定 - イベントログ](#)を参照してください。

コンテキストメニューへのアクセス

通信が正常に動作していないようであれば、チャンネルのプロパティにアクセスして、通信パラメータを修正できます。診断ウィンドウはチャンネルのプロパティが表示された後も表示されたままになるため、プロパティを変更して、その効果をモニターできます。診断ウィンドウが表示されていないければ、どのダイアログにもアクセスできません。

通信の問題が解決しない場合は、診断ウィンドウを右クリックしてコンテキストメニューを起動します。次に、使用可能なオプションを選択して診断ウィンドウの操作を調整します。

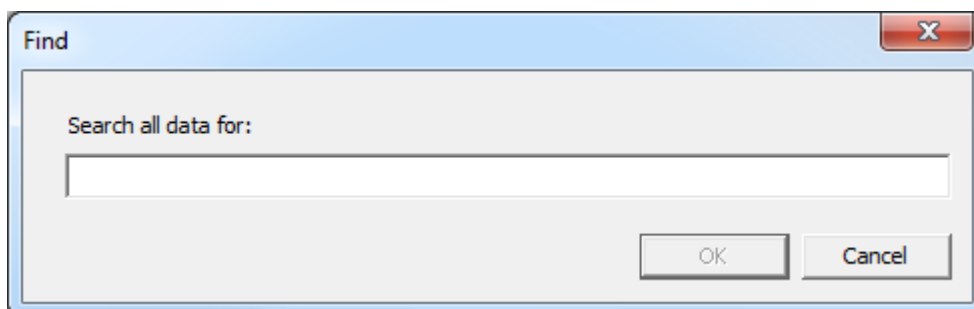


オプションの説明は次のとおりです。

- ・「**16 進数**」これが有効になっていると、TX/RX の詳細が 16 進表記を使用してフォーマットされます。
- ・「**ASCII**」これが有効になっていると、TX/RX の詳細が ASCII 表記を使用してフォーマットされます。
- ・「**検索**」このオプションは、イベントの詳細に適用する検索文字列を入力するためのダイアログを起動します。詳細については、[検索](#)を参照してください。
- ・「**コピー**」: このオプションは、プロトコル取り込みバッファのコンテンツを電子メールまたはファックスメッセージに簡単に「切り取り/貼り付け」できるようにテキストとしてフォーマットします。この情報は、テクニカルサポートがさまざまな通信の問題を解析および診断するために役立ちます。
- ・「**テキストファイルとして保存**」: このオプションは、ビュー内のすべてのイベントを特定のファイル名に(テキストとして)保存します。
- ・「**自動スクロール**」: このオプションは、新しいイベントを受信すると表示をスクロールします。これにより、最新のイベントが表示されます。これは、イベントがユーザーによって手動で選択されるとき(あるいは「検索」または「次を検索」によって選択されるとき)には無効になります。
- ・「**常に手前に表示**」: このオプションは、診断ウィンドウが常にほかのアプリケーションウィンドウの前に表示されるようにします。これはデフォルトの設定です。
- ・「**すべてのイベントを削除**」: このオプションは、イベントログによって維持されているログをクリアして、データを削除します。

検索

このダイアログでは、クライアントとサーバー間で転送された主要な情報を検索できます。



プロパティの説明は次のとおりです。

- ・「**すべてのデータを検索**」このフィールドは、検索基準を指定します。

● **注記**: 指定したテキストと一致するイベントまたは詳細が見つかったら、そのテキストが含まれている行がハイライトされます。「次を検索」操作を実行する(指定したテキストの次の一致を検索する)には、F3 キーを押します。最後の一致に到達すると、それを示すメッセージボックスが表示されます。検索基準は、いつでも Ctrl + F キーを押して変更できます。

イベント ログメッセージ

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

サーバーのサマリー情報

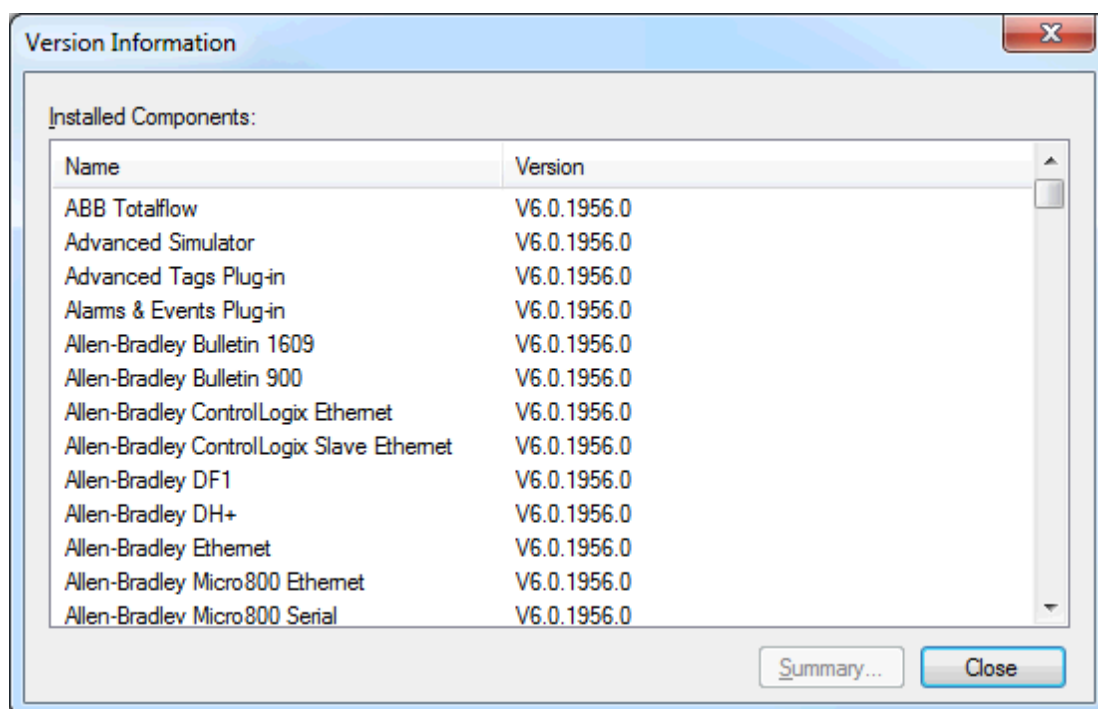
サーバーは、自身およびインストール済みのドライバーとプラグインに関して、基本的なサマリー情報を提供します。

サーバーについて

サーバーのバージョンはすぐに確認できるようになっており、ドライバー固有の情報をを見つけるための手立てになります。アクセスするには、サーバーの構成で「ヘルプ」|「サポート情報」の順にクリックします。インストール済みのコンポーネントすべてのバージョン情報を表示するには、「バージョン」をクリックします。

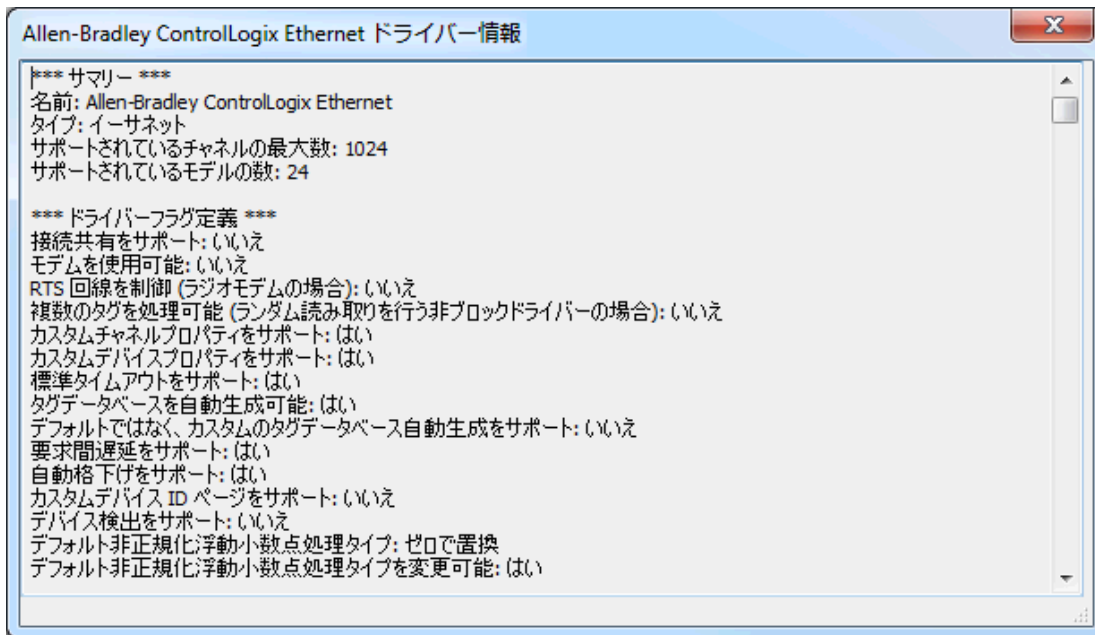
コンポーネントのバージョン情報

「バージョン情報」ウィンドウには、インストールされているすべてのドライバーおよびプラグインとともに、それらのバージョン情報が表示されます。個々のドライバーの情報を参照するには、コンポーネントを選択し、「サマリー」をクリックします。



ドライバー情報

「ドライバー情報」ウィンドウには、ドライバーのデフォルト設定のサマリーが表示されます。たとえば、各ドライバーでサポートされているチャンネルの最大数が表示されます。



参照できる情報に関するセクションは次のとおりです。

「**サマリー**」には、ドライバーの名前とタイプ、サポートされているチャンネルの最大数、ドライバーに含まれているモデルの数が表示されます。

「**通信のデフォルト**」には、ドライバーのデフォルト設定が表示されます。設定しようとしているデバイスの設定とは異なる場合があります。

「**ドライバーフラグ定義**」には、ドライバーのライブラリ関数とともに、それらの関数がドライバーで有効にされているかどうかが表示されます。

「**モデル情報**」には、デバイス固有のアドレス指定と機能が表示されます。サポートされている各モデルの名前に加え、アドレス指定値およびその他の機能が一覧表示されます。

<名前> デバイスドライバーは見つからなかったか、ロードできませんでした。

エラータイプ:

エラー

考えられる原因:

1. プロジェクトがある PC から別の PC に移動している場合、必要なドライバーがまだインストールされていない可能性があります。
2. 指定されたドライバーはインストール先サーバーから除去されている可能性があります。
3. 指定されたドライバーのバージョンはインストール先サーバーのバージョンに対応していない可能性があります。

解決策:

1. サーバーインストールを再実行して必要なドライバーを追加してください。
2. サーバーインストールを再実行して指定されたドライバーを再インストールしてください。
3. インストール先サーバーのディレクトリにドライバーが配置されていないことを確認してください (このドライバーはサーバーのバージョンに対応していません)。

'<名前>'ドライバの複数のコピー ('<名前>'と'<名前>') が存在するので、これをロードできません。競合するドライバを除去してからアプリケーションを再起動してください。

エラータイプ:

エラー

考えられる原因:

サーバー内のドライバのフォルダに複数のバージョンのドライバ DLL が存在します。

解決策:

1. サーバインストールを再実行して指定されたドライバを再インストールしてください。
2. テクニカルサポートに連絡して、正しいバージョンを確認してください。無効なドライバを除去してからサーバーを再起動し、プロジェクトをロードします。

プロジェクトファイルが無効です。

エラータイプ:

エラー

モデム回線 '<回線>'を開くことができませんでした [TAPI エラー = <コード>]。

エラータイプ:

エラー

考えられる原因:

TAPI はサーバー用にモデム回線を開こうとしましたがエラーが発生しました。

解決策:

示されたエラーの状況を修正した後、モデム回線のオープンを再試行してください。

ドライバレベルのエラーによってチャネルを追加できませんでした。

エラータイプ:

エラー

考えられる原因:

ドライバの問題によって失敗しました。

解決策:

ドライバエラーに関する追加のメッセージを参照し、関連する問題を修正してください。

ドライバレベルのエラーによってデバイスを追加できませんでした。

エラータイプ:

エラー

考えられる原因:

ドライバの問題によって失敗しました。

解決策:

ドライバエラーに関する追加のメッセージを参照し、関連する問題を修正してください。

バージョンが一致しません。

エラータイプ:

エラー

無効な XML ドキュメント:

エラータイプ:

エラー

考えられる原因:

サーバーは指定された XML ファイルを解析できません。

解決策:

サードパーティ製 XML エディタを使用してサーバープロジェクトが編集された場合、サーバーとドライバーのスキーマによってフォーマットが正しいことを確認してください。

プロジェクト <名前> をロードできません:

エラータイプ:

エラー

考えられる原因:

このプロジェクトはロードしようとしているバージョンとは互換性がないバージョンのサーバーで作成されました。

解決策:

通常、これは新しいバージョンのサーバーで作成されたプロジェクトを古いバージョンで開こうとした場合に起こります。

● 注記:

古いバージョンで作成されたプロジェクトを新しいバージョンでロードできるようにサーバーの後方互換性を確保するためのあらゆる試みがなされましたが、新しいバージョンのサーバーとドライバーのプロパティと構成は古いバージョンに存在しないことがあるので、古いプロジェクトを新しいバージョンで開いたりロードしたりできない可能性があります。

プロジェクトファイルを '<パス>' にバックアップできませんでした [<理由>]。保存操作は中止されました。保存先ファイルがロックされておらず、読み取り/書き込みのアクセス権があることを確認してください。バックアップしないでこのプロジェクトの保存を続行するには、「ツール」|「オプション」|「一般」でバックアップオプションを選択解除してからこのプロジェクトを保存し直してください。

エラータイプ:

エラー

考えられる原因:

1. 保存先ファイルが別のアプリケーションによってロックされていることはありません。
2. 保存先ファイルまたはそのファイルが存在するフォルダに対する読み取り/書き込みのアクセス権がありません。

解決策:

1. 保存先ファイルが別のアプリケーションによってロックされていないことを確認し、ファイルをロック解除するか、アプリケーションを終了してください。
2. 保存先ファイルとそのファイルが存在するフォルダに対する読み取りと書き込みのアクセス権があることを確認してください。

<機能名> が見つからなかったか、ロードできませんでした。

エラータイプ:

エラー

考えられる原因:

この機能はインストールされていないか、予期される場所にありません。

解決策:

サーバーのインストールを再実行して、指定された機能をインストールするように選択してください。

プロジェクトファイル<名前>を保存できません:

エラータイプ:

エラー

デバイスの検出でデバイスの上限 <数> を超えました。検出範囲を絞り込んでから再試行してください。

エラータイプ:

エラー

このプロジェクトをロードするために <機能名> は必須です。

エラータイプ:

エラー

現在の言語ではXMLプロジェクトをロードできません。XMLプロジェクトをロードするには、サーバー管理で製品言語の選択を英語に変更してください。

エラータイプ:

エラー

考えられる原因:

XMLプロジェクトをロードできるのは英語環境のみです。

解決策:

サーバー管理で製品言語の選択を英語に変更してから再試行してください。

オブジェクトが見つからないため、プロジェクトをロードできません。| オブジェクト = '<オブジェクト>'。

エラータイプ:

エラー

考えられる原因:

JSONプロジェクトファイルの編集によってファイルが無効な状態のままになっている可能性があります。

解決策:

JSONプロジェクトファイルに対して行ったすべての変更を元に戻してください。

プロジェクトのロード中に無効なモデルが見つかりました。| デバイス = '<デバイス>'。

エラータイプ:

エラー

考えられる原因:

指定されているデバイスには、このバージョンのサーバーではサポートされていないモデルが含まれています。

解決策:

新しいバージョンのサーバーでこのプロジェクトを開きます。

デバイスを追加できません。重複したデバイスがこのチャンネルにすでに存在している可能性があります。

エラータイプ:

エラー

自動生成されたタグ'<タグ>'はすでに存在し、上書きされません。

エラータイプ:

警告

考えられる原因:

サーバーはタグデータベース用にタグを再生成していますが、すでに存在するタグは上書きしないよう設定されています。

解決策:

これが必要な操作でない場合、デバイスの「重複タグ」プロパティの設定を変更してください。

デバイス'<デバイス>'のタグデータベースを生成できませんでした。デバイスが応答していません。

エラータイプ:

警告

考えられる原因:

1. デバイスは通信要求に応答しませんでした。
2. 指定されたデバイスは起動していないか、接続していないか、エラーになっています。

解決策:

1. (サーバーが接続できるように) デバイスが起動していて PC が起動していることを確認してください。
2. すべてのケーブル接続が正しいことを確認してください。
3. デバイス ID が正しいことを確認してください。
4. デバイスのエラーを修正してからタグ生成を再試行してください。

デバイス'<デバイス>'のタグデータベースを生成できませんでした:

エラータイプ:

警告

考えられる原因:

指定されたデバイスは起動していないか、接続していないか、エラーになっています。

解決策:

デバイスのエラーを修正してからタグ生成を再試行してください。

自動生成による上書きが多すぎるため、エラーメッセージの書き込みを停止しました。

エラータイプ:

警告

考えられる原因:

1. エラーログが一杯にならないようにするため、サーバーはタグ自動生成中に上書きできないタグに関するエラーメッセージの書き込みを停止しました。
2. タグ自動生成の範囲を狭めるか、問題のあるタグを除去してください。

アドレスが長すぎるのでタグ'<タグ>'を追加できません。アドレスの最大長は<数値>です。

エラータイプ:

警告

回線 '<回線>' はすでに使用されています。

エラータイプ:

警告

考えられる原因:

ターゲットのモデム回線はすでに開いており、これはおそらく別のアプリケーションによって使用されていることが原因です。

解決策:

モデムを開いているアプリケーションを見つけて、アプリケーションを終了するか、モデムを解放してください。

回線 '<回線>' でハードウェアエラーが発生しました。

エラータイプ:

警告

考えられる原因:

モデムに接続しているデバイス内のタグに対して要求を行った後でハードウェアエラーが返されました。

解決策:

このデバイスでのデータ収集を無効にしてください。モデムが宛先モデムに接続した後でこれを有効にします。

● **注記:**

1 回目のスキャンでエラーが発生しましたが再発していません。

回線 '<回線>' への接続で通信ハンドルが提供されませんでした。

エラータイプ:

警告

考えられる原因:

指定された通信ハンドルなしでモデム回線への接続が試みられました。

解決策:

モデムが正しくインストールされ、初期化されていることを確認してください。

回線 '<回線>' でダイヤルできません。

エラータイプ:

警告

考えられる原因:

モデムはダイヤル可能な状態にありません。

解決策:

番号をダイヤルするには、回線がアイドル状態であればなりません。モデムタグ _Mode をモニターし、アイドル状態を示しているときにダイヤルしてください。

チャネル '<名前>' でネットワークアダプタ '<アダプタ>' を使用できません。デフォルトのネットワークアダプタを使用します。

エラータイプ:

警告

考えられる原因:

プロジェクトで指定されているネットワークアダプタはこの PC に存在しません。サーバーはデフォルトのネットワークアダプタを使用します。

解決策:

この PC に使用するネットワークアダプタを選択し、プロジェクトを保存してください。

● 関連項目:

チャンネルプロパティ - ネットワークインタフェース

参照先 デバイス '<チャンネルデバイス>' のモデルタイプの変更を却下しています。

エラータイプ:

警告

TAPI 回線の初期化に失敗しました: <コード>。

エラータイプ:

警告

考えられる原因:

ランタイムを開始するためにテレフォニーサービスが実行中である必要はありません。サービスが無効な状態でシリアルドライバがプロジェクトに追加された場合、このエラーメッセージが報告されます。

解決策:

1. モデム通信が使用されていない場合、操作は必要ありません。
2. モデム通信が必要な場合、PC 上でテレフォニーサービスを開始する必要があります。

'<タグ>' での検証エラー: <エラー>。

エラータイプ:

警告

考えられる原因:

指定されたタグに無効なパラメータを設定しようとした。

ドライバー DLL '<名前>' をロードできません。

エラータイプ:

警告

考えられる原因:

プロジェクトの開始時に指定されたドライバーをロードできませんでした。

解決策:

1. インストールされているドライバーのバージョンを確認してください。ドライバーのバージョンがインストール先サーバーのバージョンに対応しているかどうかを Web サイトで確認してください。
2. ドライバーが破損している場合、ドライバーを削除してからサーバーインストールを再実行してください。

● 注記:

この問題は一般的に、ドライバーの DLL が破損しているか、ドライバーがサーバーのバージョンと互換性がない場合に起こります。

'<タグ>' での検証エラー: スケール変換パラメータが無効です。

エラータイプ:

警告

考えられる原因:

指定されたタグに無効なスケール変換パラメータを設定しようとした。

● 関連項目:

タグのプロパティ - スケール変換

回線 '<回線>' にモデム構成を適用できません。

エラータイプ:

警告

考えられる原因:

TAPI Manager は構成の変更をサーバーに適用できませんでした。

解決策:

1. モデムとのケーブル接続を確認してください。
2. モデムで構成の変更が適用されるように設定されていることを確認してください。
3. モデムが別のアプリケーションによって使用されていないことを確認してください。

デバイス '<デバイス>' は自動的に格下げされました。

エラータイプ:

警告

考えられる原因:

指定されたデバイスとの通信に失敗しました。このデバイスはポーリングサイクルから格下げされています。

解決策:

1. デバイスが再接続に失敗した場合、通信失敗の理由を調べて修正してください。
2. デバイスの格下げを中止するには、「自動格下げ」を無効にします。

● 関連項目:

自動格下げ

<ソース>: イーサネットカプセル化 IP '<address>' が無効です。

エラータイプ:

警告

考えられる原因:

イーサネットカプセル化チャネル上のデバイスに指定されている IP アドレスは有効な IP アドレスではありません。

解決策:

XML ファイル内の IP を修正してプロジェクトを再ロードしてください。

● 注記:

このエラーはサードパーティ製 XML ソフトウェアを使用して作成または編集された XML フォーマットのプロジェクトをロードする際に発生する可能性があります。

'<製品>' ドライバーは現在のところ XML 永続をサポートしていません。デフォルトのファイルフォーマットを使用して保存してください。

エラータイプ:

警告

考えられる原因:

指定されたドライバーは XML フォーマットをサポートしていません。

解決策:

プロジェクトを .opf フォーマットで保存してください。

プラグイン DLL '<名前>' をロードできません。

エラータイプ:

警告

考えられる原因:

プロジェクトの開始時に指定されたプラグインをロードできませんでした。

解決策:

1. インストールされているプラグインのバージョンを確認してください。プラグインのバージョンがインストール先サーバーと互換性があるかどうかを Web サイトで確認してください。
2. プラグインが破損している場合、プラグインを削除してからサーバーインストールを再実行してください。

● 注記:

この問題は一般的に、プラグインの DLL が破損しているか、プラグインがサーバーのバージョンと互換性がない場合に起こります。

'<デバイス>' に設定されているタイムゾーンは '<zone>' です。これはシステムで有効なタイムゾーンではありません。タイムゾーンをデフォルトの '<zone>' に戻します。

エラータイプ:

警告

ドライバー DLL '<名前>' をロードできません。理由:

エラータイプ:

警告

考えられる原因:

プロジェクトの開始時に指定されたプラグインをロードできませんでした。

解決策:

1. インストールされているプラグインのバージョンを確認してください。プラグインのバージョンがインストール先サーバーと互換性があるかどうかを Web サイトで確認してください。
2. プラグインが破損している場合、プラグインを削除してからサーバーインストールを再実行してください。

プラグイン DLL '<名前>' をロードできません。理由:

エラータイプ:

警告

考えられる原因:

プロジェクトの開始時に指定されたプラグインをロードできませんでした。

解決策:

1. インストールされているプラグインのバージョンを確認してください。プラグインのバージョンがインストール先サーバーと互換性があるかどうかを Web サイトで確認してください。
2. プラグインが破損している場合、プラグインを削除してからサーバーインストールを再実行してください。

自動ダイヤルを行うにはその電話帳に1つ以上の番号が含まれている必要があります。| チャンネル = '<チャンネル>'。

エラータイプ:

警告

考えられる原因:

「自動ダイヤル」プロパティが「有効」に設定されていますが、電話帳にエントリがありません。

解決策:

自動ダイヤルが必要な場合、電話帳に電話番号エントリを追加してください。自動ダイヤルが必要でない場合、「自動ダイヤル」を無効にしてください。

チャンネルが共有モデム接続を使用するためには、自動ダイヤルが有効になっていて、その電話帳に1つ以上の番号が含まれている必要があります。| チャンネル = '<チャンネル>'。

エラータイプ:

警告

考えられる原因:

チャンネルは1つ以上の既存のチャンネルとモデムを共有しており、自動ダイヤルが有効になっていないか、自動ダイヤル用の電話番号がありません。

解決策:

1. レポートされたチャンネルで自動ダイヤルを有効にします。
2. レポートされたチャンネルの電話帳に電話番号を追加します。

指定されたネットワークアダプタは、チャンネル'%1'| アダプタ = '%2' で無効です。

エラータイプ:

警告

考えられる原因:

プロジェクトで指定されているネットワークアダプタはこのPCに存在しません。

解決策:

このPCに使用するネットワークアダプタを選択し、プロジェクトを保存してください。

● 関連項目:

チャンネルプロパティ - ネットワークインタフェース

タグ生成要求によってタグは作成されませんでした。詳細はイベントログを参照してください。

エラータイプ:

警告

考えられる原因:

ドライバーではタグ情報が作成されませんでした。理由の提示が拒否されました。

解決策:

イベントログには、問題のトラブルシューティングに役立つ情報が含まれている可能性があります。

TAPI 構成が変更されました。再初期化しています...

エラータイプ:

情報

<製品> デバイスドライバが正 常にロードされました。

エラータイプ:

情報

<名 前> デバイスドライバを起 動しています。

エラータイプ:

情報

<名 前> デバイスドライバを停 止しています。

エラータイプ:

情報

回 線 '<modem>' で '<数 値>' をダイヤルしています。

エラータイプ:

情報

回 線 '<モデム>' は切 断されています。

エラータイプ:

情報

回 線 '<モデム>' でのダイヤルがユーザーによってキャンセルされました。

エラータイプ:

情報

回 線 '<モデム>' が <rate> ボーで接 続されました。

エラータイプ:

情報

'<モデム>' でリモート回 線がビジー状態です。

エラータイプ:

情報

'<モデム>' でリモート回 線が応 答していません。

エラータイプ:

情報

'<モデム>' で発 信 音がありません。

エラータイプ:

情報

電 話 番 号 が無 効 だ す (<数 値>)。

エラータイプ:

情報

'<モデム>' でダイヤルが中 止 されました。

エラータイプ:

情報

'<モデム>' 上のリモートサイトで回線がドロップされました。

エラータイプ:
情報

回線 '<モデム>' で着信呼び出しが検出されました。

エラータイプ:
情報

モデム回線が開きました: '<モデム>'。

エラータイプ:
情報

モデム回線が閉じました: '<モデム>'。

エラータイプ:
情報

<製品> デバイスドライバがメモリからアンロードされました。

エラータイプ:
情報

回線 '<モデム>' が接続されました。

エラータイプ:
情報

デバイス '<デバイス>' でシミュレーションモードが有効になっています。

エラータイプ:
情報

デバイス '<デバイス>' でシミュレーションモードが無効になっています。

エラータイプ:
情報

デバイス '<デバイス>' にタグを自動生成しようとしています。

エラータイプ:
情報

デバイス '<デバイス>' へのタグ自動生成が完了しました。

エラータイプ:
情報

モデム回線 '<モデム>' の切断を開始しています。

エラータイプ:
情報

クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが有効になりました。

エラータイプ:
情報

考えられる原因:

サーバーに接続しているクライアントアプリケーションによって、指定されているデバイスでの自動格下げが有効または無効になりました。

解決策:

クライアントアプリケーションによってこれが行われないようにするには、ユーザーマネージャから、システムレベルのタグに対するクライアントアプリケーションの書き込み権限を無効にします。

● 関連項目:

ユーザーマネージャ

デバイス '<デバイス>' でデータ収集が有効になっています。

エラータイプ:

情報

デバイス '<デバイス>' でデータ収集が無効になっています。

エラータイプ:

情報

オブジェクトタイプ '<名前>' はプロジェクトでは許可されません。

エラータイプ:

情報

プロジェクト '<名前>' のバックアップが '<パス>' に作成されました。

エラータイプ:

情報

通信を再確立可能かどうかを判定するためデバイス '<デバイス>' は自動昇格されました。

エラータイプ:

情報

ライブラリ <名前> のロードに失敗しました。

エラータイプ:

情報

マニフェスト作成リソース <名前> の読み取りに失敗しました。

エラータイプ:

情報

プロジェクトファイルはこのソフトウェアのより新しいバージョンで作成されました。

エラータイプ:

情報

クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが無効になりました。

エラータイプ:

情報

電話番号の優先順位が変更されました。| 電話番号名 = '<名前>'、更新後の優先順位 = '<優先順位>'。

エラータイプ:

情報

デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>。

エラータイプ:

情報

デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>、上書きされたタグ数 = <数>。

エラータイプ:

情報

デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>、上書きされなかったタグ数 = <数>。

エラータイプ:

情報

オブジェクトへのアクセスが拒否されました。| ユーザー = '<アカウント>'、オブジェクト = '<オブジェクトパス>'、アクセス許可 =

エラータイプ:

セキュリティ

User moved from user group. | User = '<name>', Old group = '<name>', New group = '<name>'.

エラータイプ:

セキュリティ

ユーザーグループが作成されました。| グループ = '<名前>'。

エラータイプ:

セキュリティ

ユーザーがユーザーグループに追加されました。| ユーザー = '<名前>'、グループ = '<名前>'。

エラータイプ:

セキュリティ

ユーザーグループの名前が変更されました。| 古い名前 = '<名前>'、新しい名前 = '<名前>'。

エラータイプ:

セキュリティ

ユーザーグループでアクセス許可定義が変更されました。| グループ = '<名前>'。

エラータイプ:

セキュリティ

ユーザーの名前が変更されました。| 古い名前 = '<名前>'、新しい名前 = '<名前>'。

エラータイプ:

セキュリティ

ユーザーが無効になりました。| ユーザー = '<名前>'。

エラータイプ:

セキュリティ

ユーザーグループが無効になりました。| グループ = '<名前>'。

エラータイプ:

セキュリティ

ユーザーが有効になりました。| ユーザー = '<名前>'。

エラータイプ:

セキュリティ

ユーザーグループが有効になりました。| グループ = '<名前>'。

エラータイプ:

セキュリティ

ユーザーのパスワードが変更されました。| ユーザー = '<名前>'。

エラータイプ:

セキュリティ

アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。

エラータイプ:

エラー

ランタイム動作モードを変更しています。

エラータイプ:

情報

ランタイム動作モードの変更が完了しました。

エラータイプ:

情報

インストールを実行するためにシャットダウンしています。

エラータイプ:

情報

OPC ProgID が ProgID リダイレクトリストに追加されました。| ProgID = '<ID>'。

エラータイプ:

情報

OPC ProgID が ProgID リダイレクトリストから除去されました。| ProgID = '<ID>'。

エラータイプ:

情報

無効な ProgID エントリが ProgID リダイレクトリストから削除されました。| ProgID = '<ID>'。

エラータイプ:

情報

管理者のパスワードが現在のユーザーによってリセットされました。| 管理者名 = '<名前>', 現在のユーザー = '<名前>'。

エラータイプ:
セキュリティ

ユーザーがユーザーグループから移動しました。| ユーザー = '<名前>', 古いグループ = '<名前>', 新しいグループ '<名前>'。

エラータイプ:
セキュリティ

ユーザーグループが作成されました。| グループ = '<名前>'。

エラータイプ:
セキュリティ

ユーザーがユーザーグループに追加されました。| ユーザー = '<名前>', グループ = '<名前>'。

エラータイプ:
セキュリティ

インポートによってユーザー情報が置き換えられました。| インポートされたファイル = '<absolute file path>'。

エラータイプ:
セキュリティ

ユーザーグループの名前が変更されました。| 古い名前 = '<名前>', 新しい名前 = '<名前>'。

エラータイプ:
セキュリティ

ユーザーグループでアクセス許可定義が変更されました。| グループ = '<名前>'。

エラータイプ:
セキュリティ

ユーザーの名前が変更されました。| 古い名前 = '<名前>', 新しい名前 = '<名前>'。

エラータイプ:
セキュリティ

ユーザーが無効になりました。| ユーザー = '<名前>'。

エラータイプ:
セキュリティ

ユーザーグループが無効になりました。| グループ = '<名前>'。

エラータイプ:
セキュリティ

ユーザーが有効になりました。| ユーザー = '<名前>'。

エラータイプ:
セキュリティ

ユーザーグループが有効になりました。| グループ = '<名前>'。

エラータイプ:

セキュリティ

管理者のパスワードのリセットに失敗しました。| 管理者名 = '<名前>'。

エラータイプ:

セキュリティ

管理者のパスワードのリセットに失敗しました。現在のユーザーは Windows 管理者ではありません。| 管理者名 = '<名前>'、現在のユーザー = '<名前>'。

エラータイプ:

セキュリティ

ユーザーのパスワードが変更されました。| ユーザー = '<名前>'。

エラータイプ:

セキュリティ

CSV タグインポート時の一般エラー。

エラータイプ:

エラー

ランタイムへの接続に失敗しました。| ランタイムのホストアドレス = '<ホストアドレス>'、ユーザー = '<名前>'、理由 = '<理由>'。

エラータイプ:

エラー

ユーザー情報が無効または見つかりません。

エラータイプ:

エラー

ランタイムプロジェクトを置き換えるにはユーザーアクセス許可が不十分です。

エラータイプ:

エラー

ランタイムプロジェクトの更新に失敗しました。

エラータイプ:

エラー

ランタイムプロジェクトの読み込みに失敗しました。

エラータイプ:

エラー

アクティブな参照カウントがあるので、チャンネル上のデバイスを置き換えられませんでした。| チャンネル = '<名前>'。

エラータイプ:

エラー

チャンネル上の既存の自動生成されたデバイスの置き換えに失敗し、削除に失敗しました。| チャンネル = '<名前>'。

エラータイプ:

エラー

チャンネルが有効でなくなりました。ユーザー入力を待機中に外部で除去された可能性があります。| チャンネル = '<名前>'。

エラータイプ:

エラー

デバイスドライバ DLL がロードされませんでした。

エラータイプ:

エラー

デバイスドライバは見つからなかったか、ロードできませんでした。| ドライバ = '<名前>'。

エラータイプ:

エラー

CSV データのインポート中にエラーが発生しました。\\n\\n識別レコードの読み取り中にフィールドバッファのオーバーフローが発生しました。

エラータイプ:

エラー

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が認識されません。| フィールド = '<名前>'。

エラータイプ:

エラー

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が重複しています。| フィールド = '<名前>'。

エラータイプ:

エラー

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド識別レコードが見つかりません。

エラータイプ:

エラー

CSV レコードのインポート中にエラーが発生しました。\\n\\nフィールドバッファのオーバーフロー。| レコードインデックス = '<数値>'。

エラータイプ:

エラー

CSV レコードのインポート中にエラーが発生しました。\\n\\n挿入に失敗しました。| レコードインデックス = '<数値>'、レコード名 = '<名前>'。

エラータイプ:

エラー

アプリケーションを起動できません。| アプリケーション = '<パス>', OS エラー = '<コード>'。

エラータイプ:

エラー

CSV レコードのインポート中にエラーが発生しました。\\n\\nマッピング先のタグアドレスがこのプロジェクトには有効ではありません。| レコードインデックス = '<数値>', タグアドレス = '<address>'。

エラータイプ:

エラー

CSV レコードのインポート中にエラーが発生しました。\\n\\nエイリアス名が無効です。名前に二重引用符を含めたり、名前をアンダースコアで始めることはできません。| レコードインデックス = '<数値>'。

エラータイプ:

エラー

無効な XML ドキュメント:

エラータイプ:

エラー

名前変更に失敗しました。その名前のオブジェクトがすでに存在します。| 提案された名前 = '<名前>'。

エラータイプ:

エラー

チャネル診断の開始に失敗しました

エラータイプ:

エラー

名前変更に失敗しました。名前にピリオドや二重引用符を含めたり、名前をアンダースコアで始めることはできません。| 提案された名前 = '<名前>'。

エラータイプ:

エラー

リモートランタイムとの同期化に失敗しました。

エラータイプ:

エラー

アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。

エラータイプ:

エラー

CSV レコードのインポート中にエラーが発生しました。タグ名が無効です。| レコードインデックス = '<数値>', タグ名 = '<名前>'。

エラータイプ:

警告

CSV レコードのインポート中にエラーが発生しました。タグまたはグループの名前が最大長を超えています。| レコードインデックス = '<数値>'、名前の最大長 (文字数) = '<数値>'。

エラータイプ:

警告

CSV レコードのインポート中にエラーが発生しました。アドレスが見つかりません。| レコードインデックス = '<数値>'。

エラータイプ:

警告

CSV レコードのインポート中にエラーが発生しました。タググループ名が無効です。| レコードインデックス = '<数値>'、グループ名 = '<名前>'。

エラータイプ:

警告

アクティブな接続があるので終了要求は無視されました。| アクティブな接続 = '<数>'。

エラータイプ:

警告

埋め込み依存ファイルの保存に失敗しました。| ファイル = '<パス>'。

エラータイプ:

警告

構成ユーティリティはサードパーティ製構成アプリケーションと同時に実行できません。両方のプログラムを閉じてから、使用するプログラムだけを開いてください。| 製品 = '<名前>'。

エラータイプ:

警告

プロジェクトを開いています。| プロジェクト = '<名前>'。

エラータイプ:

情報

プロジェクトを閉じています。| プロジェクト = '<名前>'。

エラータイプ:

情報

仮想ネットワークモードが変更されました。すべてのチャンネルと仮想ネットワークがこの影響を受けます。仮想ネットワークモードの詳細についてはヘルプを参照してください。| 新しいモード = '<モード>'。

エラータイプ:

情報

チャンネルでデバイス検出を開始しています。| チャンネル = '<名前>'。

エラータイプ:

情報

チャンネルでデバイス検出が完了しました。| チャンネル = '<名前>'、検出されたデバイス = '<数>'。

エラータイプ:

情報

チャンネルでデバイス検出がキャンセルされました。| チャンネル = '<名前>'。

エラータイプ:

情報

チャンネルでデバイス検出がキャンセルされました。| チャンネル = '<名前>'、検出されたデバイス = '<数>'。

エラータイプ:

情報

チャンネルでデバイス検出を開始できません。| チャンネル = '<名前>'。

エラータイプ:

情報

インストールを実行するためにシャットダウンしています。

エラータイプ:

情報

ランタイムプロジェクトがリセットされました。

エラータイプ:

情報

ランタイムプロジェクトが置き換えられました。| 新しいプロジェクト = '<パス>'。

エラータイプ:

情報

イベントロガーサービスに接続していません。

エラータイプ:

セキュリティ

アイテム '<名前>' の追加に失敗しました。

エラータイプ:

エラー

デバイスドライバ DLL がロードされませんでした。

エラータイプ:

エラー

無効なプロジェクトファイル: '<名前>'。

エラータイプ:

エラー

プロジェクトファイル: '<名前>' を開けませんでした。

エラータイプ:

エラー

使用中のプロジェクトと同じであるため、プロジェクト置換の要求を却下しています: '<名前>'。

エラータイプ:
エラー

ファイル名が既存のファイル '<名前>' を上書きしないようにしてください。

エラータイプ:
エラー

ファイル名は空にはできません。

エラータイプ:
エラー

ファイル名は <サブディレクトリ>/<名前>.{json,opf,sopf} の形式でなければなりません

エラータイプ:
エラー

ファイル名には 1 つ以上の無効な文字が含まれています。

エラータイプ:
エラー

「プロジェクトファイルの暗号化」が有効になっているプロジェクトファイルの .OPF ファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは .SOPF および .JSON です。

エラータイプ:
エラー

「プロジェクトファイルの暗号化」が無効になっているプロジェクトファイルの .SOPF ファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは .OPF および .JSON です。

エラータイプ:
エラー

アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。

エラータイプ:
エラー

暗号化されたプロジェクトファイル(.SOPF) にはパスワードが必要です。

エラータイプ:
エラー

パスワードを使用した .OPF および .JSON プロジェクトファイルの保存はサポートされていません。暗号化されたプロジェクトファイルを保存するには、.SOPF を使用してください。

エラータイプ:
エラー

'<名前>' へのオブジェクトの追加に失敗しました: <理由>。

エラータイプ:

警告

オブジェクト '<名前>' の移動に失敗しました: <理由>。

エラータイプ:

警告

オブジェクト '<名前>' の更新に失敗しました: <理由>。

エラータイプ:

警告

オブジェクト '<名前>' の削除に失敗しました: <理由>。

エラータイプ:

警告

スタートアッププロジェクト '<名前>' をロードできません: <理由>。

エラータイプ:

警告

スタートアッププロジェクト '<名前>' の更新に失敗しました: <理由>。

エラータイプ:

警告

定義済みのスタートアッププロジェクトにランタイムプロジェクトが置き換えられました。次の再起動時にランタイムプロジェクトは '<名前>' から復元されます。

エラータイプ:

警告

構成セッションがアクティブなのでユーザー定義のスタートアッププロジェクトは無視します。

エラータイプ:

警告

読み取り専用アイテム参照 '<名前>' に対する書き込み要求が却下されました。

エラータイプ:

警告

アイテム '<名前>' に書き込めません。

エラータイプ:

警告

アイテム '<名前>' への書き込み要求が失敗しました。書き込みデータ型 '<タイプ>' をタグデータ型 '<タイプ>' に変換できません。

エラータイプ:

警告

アイテム '<名前>' への書き込み要求が失敗しました。書き込みデータのスケール変換中にエラーが発生しました。

エラータイプ:

警告

属しているデバイスが無効になっているのでアイテム参照 '<名前>' への書き込み要求は却下されました。

エラータイプ:

警告

<名前> はシステムサービスとして実行するよう正常に設定されました。

エラータイプ:

情報

<名前> はサービスコントロールマネージャデータベースから正常に除去されました。

エラータイプ:

情報

ランタイムの再初期化が開始されました。

エラータイプ:

情報

ランタイムの再初期化が完了しました。

エラータイプ:

情報

スタートアッププロジェクト '<名前>' が更新されました。

エラータイプ:

情報

ランタイムサービスが開始されました。

エラータイプ:

情報

ランタイムプロセスが開始されました。

エラータイプ:

情報

ランタイムが終了処理を実行しています。

エラータイプ:

情報

ランタイムのシャットダウンが完了しました。

エラータイプ:

情報

インストールを実行するためにシャットダウンしています。

エラータイプ:

情報

'<名前>' から置き換えられたランタイムプロジェクトです。

エラータイプ:

情報

アプリケーションデータディレクトリが見つかりません。

エラータイプ:

情報

ランタイムプロジェクトは '<名前>' として保存されました。

エラータイプ:

情報

ランタイムプロジェクトが置換されました。

エラータイプ:

情報

構成セッションが <名前> (<名前>) によって開始されました。

エラータイプ:

セキュリティ

<名前> に割り当てられている構成セッションが終了しました。

エラータイプ:

セキュリティ

<名前> に割り当てられている構成セッションが書き込みアクセスに昇格されました。

エラータイプ:

セキュリティ

<名前> に割り当てられている構成セッションが読み取り専用に格下げされました。

エラータイプ:

セキュリティ

<名前> に割り当てられている構成セッションにアクセス許可の変更が適用されました。

エラータイプ:

セキュリティ

Failed to start Script Engine server. Socket error occurred binding to local port. |
Error = <error>, Details = '<information>'.

エラータイプ:

エラー

考えられる原因:

The port conflicts with another application.

解決策:

Use the server administration settings to update the Script Engine port.

スクリプトから未処理の例外が発生しました。| 関数 = '<関数>', エラー = '<エラー>'.

エラータイプ:

エラー

考えられる原因:

スクリプトから例外が発生しました。

解決策:

例外を引き起こした条件を修正するか、スクリプト論理を更新してください。

Script Engine service stopping.

エラータイプ:

情報

Script Engine service starting.

エラータイプ:

情報

Config API SSL 証明書に不正な署名が含まれています。

エラータイプ:

エラー

Config API が SSL 証明書をロードできません。

エラータイプ:

エラー

Config API サービスを開始できません。ポートへのバインド中に問題が発生した可能性があります。

エラータイプ:

エラー

考えられる原因:

Config API 設定で指定されている HTTP または HTTPS ポートは、別のアプリケーションによってすでにバインドされています。

解決策:

Config API またはブロックしているアプリケーションの構成を変更して異なるポートを使用するか、ポートをブロックしているアプリケーションを停止します。

Config API SSL 証明書が期限切れになっています。

エラータイプ:

警告

Config API SSL 証明書は自己署名されています。

エラータイプ:

警告

ポート <ポート番号> で、構成 API が SSL なしで開始しました。

エラータイプ:

情報

ポート <ポート番号> で、構成 API が SSL ありで開始しました。

エラータイプ:

情報

OPC .NET サーバーの起動に失敗しました。詳細については、Windows アプリケーションのイベントログを参照してください。.NET 3.5 Framework がインストールされていることも確認してください。| OS エラー = '<エラーの理由>'。

エラータイプ:

エラー

OPC .NET サーバーはインストールされていないため、起動できませんでした。インストールを再実行してください。

エラータイプ:

エラー

OPC .NET サーバーの起動がタイムアウトしました。OPC .NET Configuration Manager を使用することによってサーバーが動作していることを確認してください。

エラータイプ:

警告

サーバーインスタンスの証明書 '<証明書の場所>' が見つかりません。OPC UA Configuration Manager を使用して証明書を再発行してください。

エラータイプ:

エラー

サーバーインスタンスの証明書 '<証明書の場所>' をインポートできませんでした。OPC UA Configuration Manager を使用して証明書を再発行してください。

エラータイプ:

エラー

考えられる原因:

1. サーバーインスタンスの証明書を含むファイルが存在しないか、このファイルにアクセスできません。
2. 証明書の解読に失敗しました。

解決策:

1. ファイルが、ユーザーがアクセス許可を持つ有効なインスタンス証明書を参照していることを確認してください。
2. 証明書を再発行して、暗号化を再表示してください。
3. 新しい証明書をインポートしてください。

UA Server の証明書が失効しました。OPC UA Configuration Manager を使用して証明書を再発行してください。

エラータイプ:

エラー

考えられる原因:

証明書の有効期間が現在のシステムの日付より前になっています。

解決策:

1. 有効期限の切れていない証明書をインポートしてください。
2. 証明書を再発行して、有効期限の切れていない新しい証明書を生成してください。

クライアント接続の待機中にソケットエラーが発生しました。| エンドポイント URL = '<エンドポイント URL>', エラー = <エラーコード>、詳細 = '<description>'。

エラータイプ:

エラー

考えられる原因:

クライアント接続の待機中にエンドポイントソケットがエラーを返しました。

解決策:

エラーメッセージの詳細を確認して、問題を診断してください。

UA Server は UA Discovery Server を介した登録に失敗しました。| エンドポイント URL: '<エンドポイント URL>'。

エラータイプ:

エラー

考えられる原因:

1. UA Server エンドポイント URL およびセキュリティポリシーは、UA Discovery Server でサポートされていません。
2. UA Server の UA Discovery Server での登録は、期待どおりに完了できませんでした。

解決策:

UA Discovery Server エンドポイントでの UA Server エンドポイント URL およびセキュリティポリシーを確認してください。

証明書のロードに失敗したため、UA Server を開始できません。

エラータイプ:

エラー

考えられる原因:

1. UA Server アプリケーションインスタンス証明書の有効期間が現在のシステムの日付より前になっています。
2. サーバーインスタンスの証明書を含むファイルが存在しないか、このファイルにアクセスできません。
3. 証明書の解読に失敗しました。

解決策:

1. 有効期限の切れていない証明書をインポートしてください。
2. 証明書を再発行して、有効期限の切れていない新しい証明書を生成してください。
3. ファイルが、ユーザーがアクセス許可を持つ有効なインスタンス証明書を参照していることを確認してください。
4. 証明書を再発行して、暗号化を再表示してください。

UA Server は UA Discovery Server からの登録解除に失敗しました。| エンドポイント URL: '<エンドポイント URL>'。

エラータイプ:

警告

考えられる原因:

1. UA Server エンドポイント URL およびセキュリティポリシーは、UA Discovery Server でサポートされていません。
2. UA Server の UA Discovery Server からの登録解除は、期待どおりに完了できませんでした。

解決策:

UA Discovery Server エンドポイントでの UA Server エンドポイント URL およびセキュリティポリシーを確認してください。

UA Server は UA Discovery Server を介した登録に成功しました。| エンドポイント URL: '<エンドポイント URL>'。

エラータイプ:

情報

UA Server は UA Discovery Server からの登録解除に成功しました。| エンドポイント URL: '<エンドポイント URL>'。

エラータイプ:

情報

ReadProcessed 要求がタイムアウトになりました。| 経過時間 = <秒> (秒)。

エラータイプ:

エラー

ReadAtTime 要求がタイムアウトになりました。| 経過時間 = <秒> (秒)。

エラータイプ:

エラー

DDE アイテムの追加に失敗しました。| アイテム = '<アイテム名>'。

エラータイプ:

エラー

DDE クライアントがトピックの追加に失敗しました。| トピック = '<トピック>'。

エラータイプ:

エラー

考えられる原因:

トピック名が無効です。

解決策:

エイリアスマップを表示して、参照を有効なトピックに修正します。

● 関連項目:

エイリアスマップ

アイテムに書き込めません。| アイテム = '<アイテム名>'。

エラータイプ:

警告

指定された領域は有効ではありません。サブスクリプションフィルタを設定できませんでした。| 領域 = '<領域名>'。

エラータイプ:

エラー

指定されたソースは有効ではありません。サブスクリプションフィルタを設定できませんでした。| ソース = '<ソース名>'。

エラータイプ:

エラー

Config API SSL 証明書に不正な署名が含まれています。

エラータイプ:

エラー

Config API が SSL 証明書をロードできません。

エラータイプ:

エラー

Config API サービスを開始できません。ポートへのバインド中に問題が発生した可能性があります。

エラータイプ:

エラー

考えられる原因:

Config API 設定で指定されている HTTP または HTTPS ポートは、別のアプリケーションによってすでにバインドされています。

解決策:

Config API またはブロックしているアプリケーションの構成を変更して異なるポートを使用するか、ポートをブロックしているアプリケーションを停止します。

Config API SSL 証明書が期限切れになっています。

エラータイプ:

警告

Config API SSL 証明書は自己署名されています。

エラータイプ:

警告

ポート <ポート番号> で、構成 API が SSL なしで開始しました。

エラータイプ:

情報

ポート <ポート番号> で、構成 API が SSL ありで開始しました。

エラータイプ:

情報

ThingWorx への接続に失敗しました。| プラットフォーム = <ホスト:ポート|リソース>、エラー = <理由>。

エラータイプ:

エラー

考えられる原因:

ThingWorx Platform への接続を確立できませんでした。

解決策:

1. ホスト、ポート、リソース、およびアプリケーションキーがすべて有効で正しいことを確認してください。
2. ホストマシンが ThingWorx Platform の Composer にアクセスできることを確認してください。
3. 自己署名証明書を使用する場合、または暗号化を使用しない場合は、適切な証明書の設定が有効になっていることを確認してください。

アイテムの追加中にエラーが発生しました。| アイテム名 = '<アイテム名>'。

エラータイプ:

エラー

考えられる原因:

アイテム <TagName> をサーバーに追加してスキャンすることができませんでした。

解決策:

1. 有効なチャネルおよびデバイスにタグが存在することを確認してください。
2. QuickClient などの別のクライアントを使用してタグを読み込めることを確認してください。

プラットフォーム上での自動バインド完了イベントのトリガーに失敗しました。

エラータイプ:

エラー

考えられる原因:

自動バインドプロセスが完了する前に、ThingWorx 接続が終了しました。

解決策:

すべての自動バインドが完了してから、ThingWorx プロジェクトプロパティの再初期化または変更を実行してください。

ThingWorx への接続が失敗しました (理由は不明)。| プラットフォーム <ホスト:ポートリソース>、エラー = <error>。

エラータイプ:

エラー

考えられる原因:

ThingWorx Platform への接続に失敗しました。

解決策:

1. ホスト、ポート、リソース、およびアプリケーションキーがすべて有効で正しいことを確認してください。
2. ホストマシンが ThingWorx Platform の Composer にアクセスできることを確認してください。
3. 自己署名証明書を使用する場合、または暗号化を使用しない場合は、適切な証明書の設定が有効になっていることを確認してください。
4. エラーコードとアプリケーションレポートをご用意の上、テクニカルサポートまでご連絡ください。

接続バッファ内の容量不足が原因で、1 つまたは複数の値変更の更新が失われました。| 失われた更新の数 = <数>。

エラータイプ:

エラー

考えられる原因:

ThingWorx Platform を利用できないため、またはインスタンスによって収集されるデータが多すぎるため、データはドロップされます。

解決策:

1. 一部のデータが ThingWorx Platform で更新されていること、およびプラットフォームに到達可能であることを確認してください。
2. タグのスキャン速度を落として、ThingWorx Platform へ移動するデータ量を減らしてください。

アイテムのパブリッシングに失敗しました。多次元配列はサポートされていません。| アイテム名 = '%s'。

エラータイプ:

エラー

考えられる原因:

アイテム <アイテム名> は、そのデータが多次元配列であるタグを参照しています。

解決策:

サポートされているデータ型を持つタグを参照するアイテムを修正します。

ディスクが満杯であるため、「ストアアンドフォワード」データストアでデータを保存できません。

エラータイプ:

エラー

考えられる原因:

更新の保存に使用されているディスクは、500 MiB の容量が満杯になっています。

解決策:

1. 更新の保存に使用されているディスクの空き容量を増やしてください。
2. _DeleteStoredData システムタグを使用して、データストア内に保存されているデータを削除してください。
3. データの保存に使用されているディスクをより大きなディスクに置き換えてください。

「ストアアンドフォワード」データストアのサイズの制限値に達しました。

エラータイプ:

エラー

考えられる原因:

ThingWorx のインタフェースは、更新が生成される速度で更新をプラットフォームに送信できません。

解決策:

1. ThingWorx のインタフェースが ThingWorx Platform に接続できることを確認してください。
2. ThingWorx のインタフェースによって収集されている更新速度を低下させてください。

ThingWorx への接続が切断されました。| プラットフォーム = <ホスト:ポートリソース>。

エラータイプ:

警告

考えられる原因:

接続が切断されました。サービスが停止されたか、インタフェースがプラットフォームに到達できなくなりました。

解決策:

1. プロジェクトプロパティでネイティブインタフェースが有効になっていることを確認してください。
2. ホストマシンが ThingWorx Platform の Composer にアクセスできることを確認してください。

プロパティの自動バインドに失敗しました。| 名前 = '<プロパティ名>'。

エラータイプ:

警告

考えられる原因:

この名前のプロパティは、すでにこの Thing の下に存在します。

解決策:

1. データが現在のものであるかどうか、プロパティを確認してください。
2. データが現在のものではない場合、その Thing のプロパティを削除して、AddItem サービスをもう一度実行してください。

Thing を再起動できませんでした。| 名前 = '<Thing 名>'。

エラータイプ:

警告

考えられる原因:

AddItem サービスが完了すると、再起動サービスが Thing で呼び出されます。これにより、Composer で変更を視覚化できます。このエラーが表示された場合でも、データの変更はプラットフォームに送信されます。

解決策:

Composer を再起動して、Thing を再起動します。

プロパティへの書き込みに失敗しました。| プロパティ名 = '<名前>'、理由 = <理由>。

エラータイプ:

警告

考えられる原因:

変換の問題により、タグに書き込めません。

解決策:

1. KEPSEServerEX および ThingWorx Platform のタグのデータ型が正しく、一貫性があることを確認してください。
2. 書き込まれる値がデータ型の適切な範囲内にあることを確認します。

アイテムを追加する ThingWorx 要求が失敗しました。このアイテムはすでに追加されています。| アイテム名 = '<名前>'。

エラータイプ:

警告

考えられる原因:

このタグがすでに Thing に追加されています。

解決策:

1. データが現在のものであるかどうか、プロパティを確認してください。
2. データが現在のものではない場合、その Thing のプロパティを削除して、AddItem サービスをもう一度実行してください。

アイテムを除去する ThingWorx 要求が失敗しました。アイテムは存在しません。| アイテム名 = '<名前>'。

エラータイプ:

警告

考えられる原因:

このタグがすでに Thing から除去されています。または、そのようなタグが存在しません。

解決策:

その後もこのタグが Thing のプロパティの下に表示される場合は、ThingWorx Composer でそのプロパティを削除します。

すべてのスキャンで更新を送信するようにサーバーが設定されていますが、1 つまたは複数のプロパティのプッシュタイプが、値が変更された場合にのみプッシュするように設定されています。| 数 = <数>。

エラータイプ:

警告

考えられる原因:

ThingWorx Platform のプッシュタイプが、一部のアイテムで「Change Only」に設定されています。このプッシュタイプでは、データ値が変更された場合にプラットフォームでデータが更新されます。

解決策:

「スキャンごとに送信」オプションを使用するには、この値を「Always」に設定します。

1 つまたは複数のプロパティのプッシュタイプが、更新をプラットフォームにプッシュしないように設定されています。| 数 = <数>。

エラータイプ:

警告

考えられる原因:

ThingWorx Platform のプッシュタイプが一部のアイテムで「Never」に設定されているため、すべてのデータ変更がプラットフォームで自動的に更新されません。

解決策:

これが目的の動作でない場合は、ThingWorx Platform でプッシュタイプを変更します。

アイテムを除去するための ThingWorx 要求が失敗しました。このアイテムはバインドされており、force フラグが false に設定されています。| アイテム名 = '<名前>'。

エラータイプ:

警告

考えられる原因:

アイテムがプロパティにバインドされており、Force フラグが True に設定されていないため、RemoveItems サービスはこのアイテムを除去できませんでした。

解決策:

ForceRemove フラグを True として明示的に呼び出して、サービスを再度実行してください。

プロパティへの書き込みに失敗しました。| Thing 名 = '<名前>'、プロパティ名 = '<名前>'、理由 = <理由>。

エラータイプ:

警告

考えられる原因:

変換の問題により、タグに書き込めません。

解決策:

1. KEPServerEX および ThingWorx Platform のタグのデータ型が正しく、一貫性があることを確認してください。
2. 書き込まれる値がデータ型の適切な範囲内にあることを確認します。

Thing へのプロパティ更新のプッシュでエラーが発生しました。| Thing 名 = '<名前>'。

エラータイプ:

警告

考えられる原因:

名前付き Thing のプロパティ更新は、プラットフォームに正常にパブリッシングされませんでした。

解決策:

プラットフォームのログで、プロパティ更新が失敗している理由 (アクセス許可の問題など) を確認します。

「ストアアンドフォワード」データストアに接続または追加できません。インメモリ保存を使用します。| インメモリ保存サイズ (更新) = <数>。

エラータイプ:

警告

考えられる原因:

1. 「ストアアンドフォワード」サービスが動作していません。
2. サービスには指定したストレージディレクトリへのアクセス権がありません。
3. ポートが競合しているため、「ストアアンドフォワード」サービスが接続を受け入れられません。

解決策:

1. サーバーランタイムを再起動します。
2. 指定したストレージ場所が「ストアアンドフォワード」サービスによってアクセス可能であることを確認してください。
3. サーバー管理でストアアンドフォワードの新しいポートを設定することにより、ポートの競合を解決します。

ファイルIO エラーまたはデータストア破損のため、「ストアアンドフォワード」データストアがリセットされました。

エラータイプ:

警告

考えられる原因:

1. データストアはユーザーまたは別のプログラムによって破損しました。
2. データストアはハードウェアエラーによって破損しました。
3. データのディスクからの読み取り中にエラーが発生しました。ハードウェアに問題がある可能性があります。

解決策:

1. ユーザーアクセス制御を使用して、データストアの場所にアクセスできるユーザーを制限してください。
2. データストアを別のディスクに移動してください。

プラットフォームによって開始された設定の変更を適用できません。アクセス許可が拒否されました。| ユーザー = '<ユーザー名>'。

エラータイプ:

警告

考えられる原因:

ユーザーグループ "ThingWorx Interface Users" は、アクセス許可 "Project Modification:Servermain.Project" が "Deny" に設定されています。

解決策:

ユーザーグループ "ThingWorx Interface Users" のアクセス許可 "Project Modification:Servermain.Project" を "Allow" に設定してください。

ThingWorx Platform への構成の転送に失敗しました。

エラータイプ:

警告

ThingWorx Platform への構成の転送に失敗しました。| 理由 = '<理由>'

エラータイプ:

警告

考えられる原因:

1. 詳細は、理由を参照してください。
2. ユーザーが編集しているため、ランタイムプロジェクトはロックされています。
3. ThingWorx インタフェースのユーザーアカウントには、操作を実行するために必要な権限がありません。

「ストアアンドフォワード」データストアで保存された更新の削除に失敗しました。

エラータイプ:

警告

考えられる原因:

ハードウェアまたはオペレーティングシステムのエラーにより、操作が完了しませんでした。

解決策:

マシンを再起動してから、もう一度試してください。

ThingWorx Platform からの構成の転送に失敗しました。

エラータイプ:

警告

ThingWorx Platform からの構成の転送に失敗しました。| 理由 = '<理由>'

エラータイプ:

警告

考えられる原因:

1. 詳細は、理由を参照してください。
2. ユーザーが編集しているため、ランタイムプロジェクトはロックされています。
3. ThingWorx インタフェースのユーザーアカウントには、操作を実行するために必要な権限がありません。

使用しているアプリケーションキーが適切にフォーマットされ、有効であることを確認してください。

エラータイプ:

警告

考えられる原因:

不適切な認証のため、ThingWorx Platform への接続に失敗しました。

解決策:

1. アプリケーションキーの有効期限が切れていないことを確認してください。
2. アプリケーションキーが適切にフォーマットされていることを確認してください。
3. アプリケーションキーが正しく入力されたことを確認してください。

ThingWorx に接続しました。| プラットフォーム = <ホスト:ポート/リソース>、Thing 名 = '<名前>'。

エラータイプ:

情報

考えられる原因:

ThingWorx Platform に接続しました。

プロジェクト設定の変更による ThingWorx 接続の再初期化が、プラットフォームから開始されました。

エラータイプ:

情報

考えられる原因:

このメッセージは、SetConfiguration サービスを使用している場合に、KEPServerEX イベントログを監視しているオペレーターに対して変更が行われたことを伝えます。

インタフェースのシャットダウンまたは再初期化のために、保留中の自動バインドがドロップします。| 数 = <数>。

エラータイプ:

情報

考えられる原因:

AddItems サービス呼び出しによる自動バインドの実行中に、サーバーのシャットダウンまたは初期化が呼び出されました。

解決策:

自動バインドが実行されないすべてのアイテムは、手動で作成して ThingWorx Composer でバインドする必要があります。

1 つまたは複数の自動バインド要求が処理されました。| 数 = <数>。

エラータイプ:

情報

考えられる原因:

AddItems サービスの一部に自動バインド操作が含まれています。この操作は実際のアイテムの追加操作よりも時間がかかる可能性があります。このメッセージは、自動バインドを実行するアイテムの数に注意するように呼びかけるものです。

プロジェクト設定の変更が構成 API から開始されたため、ThingWorx 接続を再初期化しています。

エラータイプ:

情報

考えられる原因:

このメッセージは、構成 API を使用している場合に、KEPServerEX イベントログを監視しているオペレーターに対して変更が行われたことを伝えます。

Thing へのプロパティ更新のプッシュが再開しました: エラー状態は解決しました。| Thing 名 = '<名前>'。

エラータイプ:

情報

ThingWorx からの構成転送が開始しました。

エラータイプ:

情報

ThingWorx からの構成の転送が中断しました。

エラータイプ:

情報

「ストアアンドフォワード」データストアを初期化しました。| データストアの場所: '<場所>'。

エラータイプ:

情報

考えられる原因:

ThingWorx ネイティブインターフェースが、ストアアンドフォワードを使用するように設定されています。

保存されたデータが「ストアアンドフォワード」データストアから削除されました。

エラータイプ:

情報

考えられる原因:

クライアントが `_DeleteStoredData` システムタグを使用して、「ストアアンドフォワード」データストアの ThingWorx インタフェースに対してキャッシュされたデータを削除しました。

ストアアンドフォワードモードが変更されました。| フォワードモード = '<モード>'。

エラータイプ:

情報

考えられる原因:

`_ForwardMode` システムタグは接続されたクライアントによって書き込まれ、書き込みの値によって設定が変更されました。

「ストアアンドフォワード」データストアを初期化しました。| フォワードモード = '<モード>' | データストアの場所 = '<場所>'。

エラータイプ:
情報

考えられる原因:

ThingWorx ネイティブインターフェースが、ストアアンドフォワードを使用するように設定されています。

無効なデータストアパスのため、データストアへの追加でエラーが発生しました。| パス = '<パス>'

エラータイプ:
エラー

考えられる原因:

「ストアアンドフォワード」を使用してコンポーネントによって指定されたパスが無効です。詳細は、コンポーネントのドキュメントと、メッセージの本文に含まれている検証エラーを参照してください。

解決策:

メッセージに記載されているエラーを修正します。

「ストアアンドフォワード」サーバーの開始に失敗しました。ローカルポートへのバインド中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:
エラー

考えられる原因:

The port conflicts with another application.

解決策:

サーバー管理設定を使用して、「ストアアンドフォワード」ポートを更新します。

「ストアアンドフォワード」サービスを中止しています。

エラータイプ:
情報

「ストアアンドフォワード」サービスを開始しています。

エラータイプ:
情報

データストアへの接続時にファイルが破損しました。データストアは再作成されました。| データストアのパス = '<パス>'。

エラータイプ:
情報

考えられる原因:

データストアによって使用されているファイルは、システム、別のアプリケーション、またはユーザーが破損しました。

解決策:

1. 古いデータストアは自動的に置き換えられました。ユーザーの操作は必要ありません。
2. この問題が繰り返し発生する場合、その他のアプリケーションやユーザーがアクセスできない場所にデータストアのディレクトリを変更することを検討してください。

構成が変更されたため、データストアは上書きされました。| データストアのパス = '<パス>'。

エラータイプ:
情報

考えられる原因:
データストアのサイズパラメータが変更されました。

● **注記:**
データストアのサイズを変更すると、データストアのすべてのファイルが再作成されます。ThingWorx Platform からの切断のためにデータストアでデータがアクティブに保存されたのではない限り、データが失われた可能性は低いと考えられます。

既存のデータストアは古いバージョンのサーバーで作成されたため、このデータストアに追加できません。データストアは再作成されました。| データストアのパス = '<パス>'。

エラータイプ:
情報

考えられる原因:
サーバーは、新しいデータストアフォーマットを使用するバージョンにアップグレードされました。

解決策:
古いデータストアは新しいバージョンのデータストアで置き換えられました。ユーザーの操作は必要ありません。

COM ポートが別のアプリケーションによって使用されています。| ポート = '<ポート>'。

エラータイプ:
エラー

考えられる原因:
デバイスに割り当てられているシリアルポートが別のアプリケーションによって使用されています。

解決策:

1. チャンネルに正しいポートが割り当てられていることを確認してください。
2. 現在のプロジェクトのコピーが1つだけ動作していることを確認してください。

指定されたパラメータでは COM ポートを設定できません。| ポート = COM<数値>、OS エラー = <エラー>。

エラータイプ:
エラー

考えられる原因:
指定された COM ポートのシリアルパラメータが有効ではありません。

解決策:
シリアルパラメータを確認し、必要に応じて変更してください。

ドライバーの初期化に失敗しました。

エラータイプ:
エラー

シリアル I/O スレッドを作成できません。

エラータイプ:

エラー

考えられる原因:

サーバープロセスには新しいスレッドの作成に使用可能なリソースがありません。

解決策:

各タググループが1つのスレッドを消費します。1つのプロセスでの一般的な上限は約2000スレッドです。プロジェクト内のタググループの数を減らしてください。

COMポートが存在しません。| ポート = '<ポート>'。

エラータイプ:

エラー

考えられる原因:

指定されたCOMポートはターゲットコンピュータ上に存在しません。

解決策:

適切なCOMポートが選択されていることを確認してください。

COMポートを開く際にエラーが発生しました。| ポート = '<ポート>', OSエラー = <エラー>。

エラータイプ:

エラー

考えられる原因:

ターゲットコンピュータ上の内部のハードウェアまたはソフトウェアの問題によって、指定されたCOMポートを開くことができませんでした。

解決策:

COMポートが機能していてほかのアプリケーションからアクセス可能であることを確認してください。

接続に失敗しました。アダプタにバインドできません。| アダプタ = '<名前>'。

エラータイプ:

エラー

考えられる原因:

指定されたネットワークアダプタは、システムデバイスリストで見つからないので、通信用にバインドできません。これはあるPCから別のPCにプロジェクトを移動した場合（かつプロジェクトでデフォルトを使用する代わりにネットワークアダプタが指定されている場合）に起こる可能性があります。サーバーはデフォルトアダプタに戻ります。

解決策:

ネットワークアダプタのプロパティをデフォルトに変更（または新しいアダプタを選択）してから、プロジェクトを保存し、再試行してください。

Winsockのシャットダウンに失敗しました。| OSエラー = <エラー>。

エラータイプ:

エラー

Winsockの初期化に失敗しました。| OSエラー = <エラー>。

エラータイプ:

エラー

解決策:

1. 基礎となるネットワークサブシステムでネットワーク通信の準備が整っていません。数秒待ってからドライバーを再起動してください。
2. Windows ソケットの実装でサポートされているタスク数の上限に達しました。Winsock を使用している可能性がある 1 つ以上のアプリケーションを終了してからドライバーを再起動してください。

このドライバーを使用するには Winsock V1.1 以上がインストールされている必要があります。

エラータイプ:

エラー

考えられる原因:

システムで見つかった Winsock DLL のバージョン番号は 1.1 より古いバージョンです。

解決策:

Winsock をバージョン 1.1 以上にアップグレードしてください。

ローカルポートへのバインド中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

エラー

デバイスが応答していません。

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 間の接続が切断しています。
2. 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デバイス ID が割り当てられている可能性があります。
4. デバイスからの応答を受信するまでに、「要求のタイムアウト」デバイス設定で許可されているより長い時間がかかりました。

解決策:

1. PC と PLC デバイス間のケーブル接続を確認してください。
2. 指定した通信パラメータがデバイスのパラメータと一致することを確認してください。
3. この名前のデバイスのデバイス ID が実際のデバイスの ID と一致することを確認してください。
4. 応答全体が処理されるように「要求のタイムアウト」設定を大きくしてください。

デバイスが応答していません。| ID = '<デバイス>'。

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 間のネットワーク接続が切断しています。
2. デバイスとドライバーに設定されている通信パラメータが一致しません。

3. デバイスからの応答を受信するまでに、「要求のタイムアウト」デバイス設定で許可されているより長い時間がかかりました。

解決策:

1. PC と PLC デバイス間のケーブル接続を確認してください。
2. 指定した通信パラメータがデバイスのパラメータと一致することを確認してください。
3. 応答全体が処理されるように「要求のタイムアウト」設定を大きくしてください。

チャンネルでのシリアル通信エラー。| エラーマスク = <マスク>。

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 間のシリアル接続が切断しています。
2. シリアル接続の通信パラメータが不正です。

解決策:

1. エラーマスクコードと関連情報を調査してください。
2. PC と PLC デバイス間のケーブル接続を確認してください。
3. 指定した通信パラメータがデバイスのパラメータと一致することを確認してください。

● 関連項目:

エラーマスクコード

タグの書き込みで無効な配列サイズが検出されました。<デバイス名>.<アドレス>。

エラータイプ:

警告

考えられる原因:

クライアントが、更新前に書き込もうとしています。

解決策:

書き込む前に配列で読み取りを実行してください。

デバイスのアドレスに書き込めません。| 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 間の接続が切断しています。
2. 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デバイス ID が割り当てられている可能性があります。

解決策:

1. PC と PLC デバイス間のケーブル接続を確認してください。
2. 指定した通信パラメータがデバイスのパラメータと一致す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デバイス ID が実際のデバイスの ID と一致することを確認してください。

ドライバーがタグを処理している間はこのページ上のアイテムを変更できません。

エラータイプ:

警告

考えられる原因:

データクライアントがサーバーに接続してチャネル/デバイスからデータを受信している間に、チャネル構成またはデバイス構成を変更しようとした。

解決策:

変更を行う前にすべてのデータクライアントをサーバーから切断してください。

指定されたアドレスはデバイス上で有効ではありません。| 無効なアドレス = '<アドレス>'。

エラータイプ:

警告

考えられる原因:

タグアドレスに無効なアドレスが割り当てられています。

解決策:

要求されたアドレスをクライアントアプリケーションで修正してください。

アドレス '<アドレス>' はデバイス '<名前>' 上で有効ではありません。

エラータイプ:

警告

ドライバーがタグを処理している間にこのプロパティを変更することはできません。

エラータイプ:

警告

デバイス '<名前>' 上のアドレス '<アドレス>' に書き込めません。

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 間の接続が切断しています。
2. 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デバイス ID が割り当てられている可能性があります。

解決策:

1. PC と PLC デバイス間のケーブル接続を確認してください。
2. 指定した通信パラメータがデバイスのパラメータと一致す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デバイス ID が実際のデバイスの ID と一致することを確認してください。

接続中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

警告

考えられる原因:

指定されたソケットの操作中にデバイスとの通信に失敗しました。

解決策:

エラーの発生原因についての説明、および該当する場合の修復方法が記載されている、エラーと詳細のガイダンスに従ってください。

データの受信中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

警告

考えられる原因:

指定されたソケットの操作中にデバイスとの通信に失敗しました。

解決策:

エラーの発生原因についての説明、および該当する場合の修復方法が記載されている、エラーと詳細のガイダンスに従ってください。

データの送信中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

警告

考えられる原因:

指定されたソケットの操作中にデバイスとの通信に失敗しました。

解決策:

エラーの発生原因についての説明、および該当する場合の修復方法が記載されている、エラーと詳細のガイダンスに従ってください。

読み取り可否のチェック中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

警告

考えられる原因:

指定されたソケットの操作中にデバイスとの通信に失敗しました。

解決策:

エラーの発生原因についての説明、および該当する場合の修復方法が記載されている、エラーと詳細のガイダンスに従ってください。

書き込み可否のチェック中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。

エラータイプ:

警告

考えられる原因:

指定されたソケットの操作中にデバイスとの通信に失敗しました。

解決策:

エラーの発生原因についての説明、および該当する場合の修復方法が記載されている、エラーと詳細のガイダンスに従ってください。

%s |

エラータイプ:
情報

<名前> デバイスドライバー '<名前>'

エラータイプ:
情報

リソース

このユーザーマニュアルのほかに、顧客のサポート、質問への回答、特定の実装に関する詳細情報の提供、または特定の問題のトラブルシューティングに役立つさまざまなリソースが用意されています。

[知識ベース](#)

[技術白書](#)

[接続ガイド](#)

[テクニカルノート](#)

[トレーニングプログラム](#)

[トレーニングのビデオ](#)

[Kepware テクニカルサポート](#)

[PTC テクニカル サポート](#)

索引

,

'<タグ>'での検証エラー

<エラー>。 204

スケール変換パラメータが無効です。 204

'<デバイス>'に設定されているタイムゾーンは '<zone>' です。これはシステムで有効なタイムゾーンではありません。タイムゾーンをデフォルトの '<zone>' に戻します。 206

'<モデム>'でダイヤルが中止されました。 208

'<モデム>'でリモート回線がビジー状態です。 208

'<モデム>'でリモート回線が応答していません。 208

'<モデム>'で発信音がありません。 208

'<モデム>'上のリモートサイトで回線がドロップされました。 209

'<製品>'ドライバーは現在のところXML永続をサポートしていません。デフォルトのファイルフォーマットを使用して保存してください。 205

'<名前>'から置き換えられたランタイムプロジェクトです。 221

'<名前>'ドライバーの複数のコピー ('<名前>'と '<名前>')が存在するので、これをロードできません。競合するドライバーを除去してからアプリケーションを再起動してください。 199

'<名前>'へのオブジェクトの追加に失敗しました

<理由>。 220

%

%s | 243

「

「ストアアンドフォワード」サーバーの開始に失敗しました。ローカルポートへのバインド中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 236

「ストアアンドフォワード」サービスを開始しています。 236

「ストアアンドフォワード」サービスを中止しています。 236

「ストアアンドフォワード」データストアで保存された更新の削除に失敗しました。 233

「ストアアンドフォワード」データストアに接続または追加できません。インメモリ保存を使用します。| インメモリ保存サイズ (更新) = <数>。 232

「ストアアンドフォワード」データストアのサイズの制限値に達しました。 229

「ストアアンドフォワード」データストアを初期化しました。| データストアの場所 '<場所>'。 235

「ストアアンドフォワード」データストアを初期化しました。| フォワードモード = '<モード>' | データストアの場所 = '<場所>'。 236

「プロジェクトファイルの暗号化」が無効になっているプロジェクトファイルの .SOPF ファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは .OPF および .JSON です。 219

「プロジェクトファイルの暗号化」が有効になっているプロジェクトファイルの .OPF ファイルタイプとしての保存はサポートされていません。サポートされているファイルタイプは .SOPF および .JSON です。 219

<

<ソース>

イーサネットカプセル化 IP '<address>' が無効です。 205

<機能名> が見つからなかったか、ロードできませんでした。 200

<製品> デバイスドライバがメモリからアンロードされました。 209

<製品> デバイスドライバが正常にロードされました。 208

<名前> デバイスドライバ '<名前>' 243

<名前> デバイスドライバは見つからなかったか、ロードできませんでした。 198

<名前> デバイスドライバを起動しています。 208

<名前> デバイスドライバを停止しています。 208

<名前> に割り当てられている構成セッションが終了しました。 222

<名前> に割り当てられている構成セッションが書き込みアクセスに昇格されました。 222

<名前> に割り当てられている構成セッションが読み取り専用に格下げされました。 222

<名前> に割り当てられている構成セッションにアクセス許可の変更が適用されました。 222

<名前> はサービスコントロールマネージャデータベースから正常に除去されました。 221

<名前> はシステムサービスとして実行するよう正常に設定されました。 221

1

1 つまたは複数のプロパティのプッシュタイプが、更新をプラットフォームにプッシュしないように設定されています。| 数 = <数>。 231

1 つまたは複数の自動バインド要求が処理されました。| 数 = <数>。 234

A

AttributeServiceSet 193

C

COM ポートが存在しません。| ポート = '<ポート>'。 238

COM ポートが別のアプリケーションによって使用されています。| ポート = '<ポート>'。 237

COM ポートを開く際にエラーが発生しました。| ポート = '<ポート>'、OS エラー = <エラー>。 238

Config API SSL 証明書が期限切れになっています。 223, 227

Config API SSL 証明書に不正な署名が含まれています。 223, 227

Config API SSL 証明書は自己署名されています。 223, 227

Config API が SSL 証明書をロードできません。 223, 227

Config API サービスを開始できません。ポートへのバインド中に問題が発生した可能性があります。 223, 227

CORS 39

CSV 96

CSV タグインポート時の一般エラー。 214

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド識別レコードが見つかりません。 215

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が重複しています。| フィールド = '<名前>'。
215

CSV データのインポート中にエラーが発生しました。\\n\\nフィールド名が認識されません。| フィールド = '<名前>'。
215

CSV データのインポート中にエラーが発生しました。\\n\\n識別レコードの読み取り中にフィールドバッファのオーバーフローが発生しました。 215

CSV レコードのインポート中にエラーが発生しました。\\n\\nエイリアス名が無効です。名前に二重引用符を含めたり、名前をアンダースコアで始めることはできません。| レコードインデックス = '<数値>'。 216

CSV レコードのインポート中にエラーが発生しました。\\n\\nフィールドバッファのオーバーフロー。| レコードインデックス = '<数値>'。 215

CSV レコードのインポート中にエラーが発生しました。\\n\\nマッピング先のタグアドレスがこのプロジェクトには有効ではありません。| レコードインデックス = '<数値>'、タグアドレス = '<address>'。 216

CSV レコードのインポート中にエラーが発生しました。\\n\\n挿入に失敗しました。| レコードインデックス = '<数値>'、レコード名 = '<名前>'。 215

CSV レコードのインポート中にエラーが発生しました。アドレスが見つかりません。| レコードインデックス = '<数値>'。
217

CSV レコードのインポート中にエラーが発生しました。タググループ名が無効です。| レコードインデックス = '<数値>'、グループ名 = '<名前>'。 217

CSV レコードのインポート中にエラーが発生しました。タグまたはグループの名前が最大長を超えています。| レコードインデックス = '<数値>'、名前の最大長 (文字数) = '<数値>'。 217

CSV レコードのインポート中にエラーが発生しました。タグ名が無効です。| レコードインデックス = '<数値>'、タグ名 = '<名前>'。 216

D

DCOM 30

DDE 24

DDE アイテムの追加に失敗しました。| アイテム = '<アイテム名>'。 226

DDE クライアントがトピックの追加に失敗しました。| トピック = '<トピック>'。 226

DiscoveryServiceSet 193

F

Failed to start Script Engine server. Socket error occurred binding to local port. | Error = <error>, Details = '<information>'。 222

FastDDE/SuiteLink 24

G

GET 要求 URI 165

H

HTTP ポート 39

HTTPS ポート 39

I

ID 77

iFIX アプリケーションからの構成 153

iFIX アプリケーションのプロジェクトの起動 161

iFIX データベースマネージャでの I/O アドレスの指定 153

iFIX データベースマネージャでの I/O ドライバーの指定 153

iFIX データベースマネージャでの信号条件の指定 154

iFIX ネイティブインタフェース 25

iFIX 信号条件のオプション 155

IP アドレス 79-80

J

JSON 133

JSON 応答の構造体 165

M

MonitoredItemServiceSet 193

O

OPC .NET 23

OPC .NET サーバーの起動がタイムアウトしました。OPC .NET Configuration Manager を使用することによってサーバーが動作していることを確認してください。 224

OPC .NET サーバーの起動に失敗しました。詳細については、Windows アプリケーションのイベントログを参照してください。 .NET 3.5 Framework がインストールされていることも確認してください。 | OS エラー = '<エラーの理由>'。 224

OPC .NET サーバーはインストールされていないため、起動できませんでした。インストールを再実行してください。 224

OPC AE 21

OPC DA 21

OPC DA サービス 192

OPC ProgID が ProgID リダイレクトリストから除去されました。 | ProgID = '<ID>'。 212

OPC ProgID が ProgID リダイレクトリストに追加されました。 | ProgID = '<ID>'。 212

OPC UA 23

OPC 診断のイベント 186

OPC 診断ビューア 183

OPF 133

OtherServices 193

P

Password 168

R

ReadAtTime 要求がタイムアウトになりました。| 経過時間 = <秒> (秒)。 226

ReadProcessed 要求がタイムアウトになりました。| 経過時間 = <秒> (秒)。 226

S

Script Engine service starting. 223

Script Engine service stopping. 223

SecureChannelServiceSet 193

SessionServiceSet 193

SOPF 133

SSL 40

SubscriptionServiceSet 194

T

TAPI 回線の初期化に失敗しました

<コード>。 204

TAPI 構成が変更されました。再初期化しています... 207

Thing へのプロパティ更新のプッシュが再開しました

エラー状態は解決しました。| Thing 名 = '<名前>'。 235

Thing へのプロパティ更新のプッシュでエラーが発生しました。| Thing 名 = '<名前>'。 232

Thing を再起動できませんでした。| 名前 = '<Thing 名>'。 230

ThingWorx 58

ThingWorx Platform からの構成の転送に失敗しました。 233

ThingWorx Platform からの構成の転送に失敗しました。| 理由 = '<理由>' 233

ThingWorx Platform への構成の転送に失敗しました。 233

ThingWorx Platform への構成の転送に失敗しました。| 理由 = '<理由>' 233

ThingWorx からの構成の転送が中断しました。 235

ThingWorx からの構成転送が開始しました。 235

ThingWorx に接続しました。| プラットフォーム = <ホスト

ポートリソース>、Thing 名 = '<名前>'。 234

ThingWorx ネイティブインタフェース 25

ThingWorx への接続が失敗しました(理由は不明)。| プラットフォーム <ホスト

ポートリソース>、エラー = <error>。 228

ThingWorx への接続が切断されました。| プラットフォーム = <ホスト

ポートリソース>。 229

ThingWorx への接続に失敗しました。| プラットフォーム = <ホスト

ポートリソース>、エラー = <理由>。 227

U

UA Server の証明書が失効しました。OPC UA Configuration Manager を使用して証明書を再発行してください。
224

UA Server は UA Discovery Server からの登録解除に失敗しました。| エンドポイント URL
'<エンドポイント URL>'。 225

UA Server は UA Discovery Server からの登録解除に成功しました。| エンドポイント URL
'<エンドポイント URL>'。 226

UA Server は UA Discovery Server を介した登録に失敗しました。| エンドポイント URL
'<エンドポイント URL>'。 225

UA Server は UA Discovery Server を介した登録に成功しました。| エンドポイント URL
'<エンドポイント URL>'。 226

User moved from user group. | User = '<name>', Old group = '<name>', New group = '<name>'. 211

V

ViewServiceSet 194

W

Winsock のシャットダウンに失敗しました。| OS エラー = <エラー>。 238

Winsock の初期化に失敗しました。| OS エラー = <エラー>。 238

あ

アーキテクチャ 163

アイテム '<名前>' に書き込めません。 220

アイテム '<名前>' の追加に失敗しました。 218

アイテム '<名前>' への書き込み要求が失敗しました。書き込みデータのスケール変換中にエラーが発生しました。
220

アイテム '<名前>' への書き込み要求が失敗しました。書き込みデータ型 '<タイプ>' をタグデータ型 '<タイプ>' に変換
できません。 220

アイテムに書き込めません。| アイテム = '<アイテム名>'。 226

アイテムのパブリッシングに失敗しました。多次元配列はサポートされていません。| アイテム名 = '%s'。 229

アイテムの追加中にエラーが発生しました。| アイテム名 = '<アイテム名>'。 228

アイテムを除去する ThingWorx 要求が失敗しました。アイテムは存在しません。| アイテム名 = '<名前>'。 231

アイテムを除去するための ThingWorx 要求が失敗しました。このアイテムはバインドされており、force フラグが false
に設定されています。| アイテム名 = '<名前>'。 231

アイテムを追加する ThingWorx 要求が失敗しました。このアイテムはすでに追加されています。| アイテム名 = '<名
前>'。 230

アカウント '<名前>' には、このアプリケーションを実行するためのアクセス許可がありません。システム管理者にお問い合わせください。 212, 216, 219

アクセス許可 18, 36

アクティブな参照カウントがあるので、チャンネル上のデバイスを置き換えられませんでした。| チャンネル = '<名前>'。 214

アクティブな接続があるので終了要求は無視されました。| アクティブな接続 = '<数>'。 217

アドレス '<アドレス>' はデバイス '<名前>' 上で有効ではありません。 241
アドレスが長すぎるのでタグ '<タグ>' を追加できません。アドレスの最大長は <数値> です。 202
アプリケーションデータ 17
アプリケーションデータディレクトリが見つかりません。 222
アプリケーションを起動できません。| アプリケーション = '<パス>'、OS エラー = '<コード>'。 216

い

イーサネットカプセル化 80
イーサネットカプセル化の使用 148
イベント 47
イベントロガーサービスに接続していません。 218
イベントログ 30
イベントログとは 94
イベントログの表示 94
イベントログメッセージ 197
インストールを実行するためにシャットダウンしています。 212, 218, 221
インタフェースと接続性 21
インタフェースのシャットダウンまたは再初期化のために、保留中の自動バインドがドロップします。| 数 = <数>。 234
インポート 96
インポートによってユーザー情報が置き換えられました。| インポートされたファイル = '<absolute file path>'。 213

え

エイリアスのプロパティ 93
エイリアスの作成および使用 142
エイリアスマップとは 92
エイリアスを使用したプロジェクトの最適化 146
エイリアス名 93
エクスポート 96
エラー時に格下げ 79

お

オブジェクト '<名前>' の移動に失敗しました
<理由>。 220
オブジェクト '<名前>' の更新に失敗しました
<理由>。 220
オブジェクト '<名前>' の削除に失敗しました
<理由>。 220
オブジェクトが見つからないため、プロジェクトをロードできません。| オブジェクト = '<オブジェクト>'。 201
オブジェクトタイプ '<名前>' はプロジェクトでは許可されません。 210
オブジェクトへのアクセスが拒否されました。| ユーザー = '<アカウント>'、オブジェクト = '<オブジェクトパス>'、アクセス許可 = 211

オプション - ランタイム接続 67

オプション - 一般 66

き

キャッシュからの初回更新 78

く

クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが無効になりました。 210

クライアントアプリケーションによってデバイス '<デバイス>' での自動格下げが有効になりました。 209

クライアント接続の待機中にソケットエラーが発生しました。| エンドポイント URL = '<エンドポイント URL>', エラー =
<エラーコード>, 詳細 = '<description>'。 225

クランプ 90

こ

このドライバーを使用するには Winsock V1.1 以上がインストールされている必要があります。 239

このプロジェクトをロードするために <機能名> は必須です。 201

コンテンツの取得 164

コンポーネント 19

コンポーネントと概念 68

コマ区切り変数 96

さ

サーバーインスタンスの証明書 '<証明書の場所>' が見つかりません。 OPC UA Configuration Manager を使用して
証明書を再発行してください。 224

サーバーインスタンスの証明書 '<証明書の場所>' をインポートできませんでした。 OPC UA Configuration Manager
を使用して証明書を再発行してください。 224

サーバーでの DDE の使用 147

サーバーのオプション 66

サーバーのサマリー情報 197

サーバーの実行 122

サーバープロジェクトでのモデムの使用 118

サーバープロジェクトの最適化 144

サービスポート 42

サブグループを許可 81

サマータイム 82

し

システムサービス 178

システムタグ 98
システム要件 17
シミュレーション 78
シリアル I/O スレッドを作成できません。 237

す

スキャンしない、要求ポールのみ 78
スキャンモード 78
スキャン速度オーバーライド 94
スクリプトから未処理の例外が発生しました。| 関数 = '<関数>'、エラー = '<エラー>'。 222
スケール変換後 89
スタートアッププロジェクト '<名前>' が更新されました。 221
スタートアッププロジェクト '<名前>' の更新に失敗しました
 <理由>。 220
スタートアッププロジェクト '<名前>' をロードできません
 <理由>。 220
ステータスバー 47
ストアアンドフォワード - システムタグ 64
ストアアンドフォワード - 補充率の例 63
ストアアンドフォワードサービス 43, 162
ストアアンドフォワードモードが変更されました。| フォワードモード = '<モード>'。 235
すべてのスキャンで更新を送信するようにサーバーが設定されていますが、1 つまたは複数のプロパティのプロシユタイプ
 が、値が変更された場合にのみプッシュするように設定されています。| 数 = <数>。 231

せ

セキュリティ 17, 24, 29, 33, 38, 40, 48, 51, 55, 58, 67, 122, 133, 164, 178

そ

ソフトウェア要件 18

た

タイトルバー 44
タイプ定義 166
タイムアウト前の試行回数 83
タイムゾーン 82
タググループとは 91
タググループのプロパティ 91
タグスケール変換の追加 132
タグとは 83

タグに指定のスキャン速度を適用 78
タグのブラウズ 129
タグのプロパティ - スケール変換 89
タグのプロパティ - 一般 84
タグの管理 96
タグの書き込みで無効な配列サイズが検出されました。<デバイス名>.<アドレス>。 240
タグ生成 80
タグ生成要求によってタグは作成されませんでした。詳細はイベントログを参照してください。 207

ち

チャンネル '<名前>' でネットワークアダプタ '<アダプタ>' を使用できません。デフォルトのネットワークアダプタを使用します。 203
チャンネル、デバイス、タグ、およびタググループの適切な名前の指定 145
チャンネルが共有モデム接続を使用するためには、自動ダイヤルが有効になっていて、その電話帳に1つ以上の番号が含まれている必要があります。| チャンネル = '<チャンネル>'。 207
チャンネルが有効でなくなりました。ユーザー入力を待機中に外部で除去された可能性があります。| チャンネル = '<名前>'。 215
チャンネルでデバイス検出がキャンセルされました。| チャンネル = '<名前>'、検出されたデバイス = '<数>'。 218
チャンネルでデバイス検出がキャンセルされました。| チャンネル = '<名前>'。 218
チャンネルでデバイス検出が完了しました。| チャンネル = '<名前>'、検出されたデバイス = '<数>'。 218
チャンネルでデバイス検出を開始しています。| チャンネル = '<名前>'。 217
チャンネルでデバイス検出を開始できません。| チャンネル = '<名前>'。 218
チャンネルでのシリアル通信エラー。| エラーマスク = <マスク>。 240
チャンネルとは 68
チャンネルの追加と構成 123
チャンネル割り当て 77
チャンネル作成ウィザード 123
チャンネル上の既存の自動生成されたデバイスの置き換えに失敗し、削除に失敗しました。| チャンネル = '<名前>'。 215
チャンネル診断の開始に失敗しました 216

て

ディスクが満杯であるため、「ストアアンドフォワード」データストアでデータを保存できません。 229
データ 170
データコレクション 78
データストアへの接続時にファイルが破損しました。データストアは再作成されました。| データストアのパス = '<パス>'。 236
データの受信中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 242
データの送信中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 242
テキストシーケンスを追加 87
デスクトップ対話の許可 142
デバイス '<デバイス>' でシミュレーションモードが無効になっています。 209
デバイス '<デバイス>' でシミュレーションモードが有効になっています。 209

デバイス '<デバイス>' でデータ収集が無効になっています。 210
デバイス '<デバイス>' でデータ収集が有効になっています。 210
デバイス '<デバイス>' にタグを自動生成しようとしています。 209
デバイス '<デバイス>' のタグデータベースを生成できませんでした 202
デバイス '<デバイス>' のタグデータベースを生成できませんでした。デバイスが応答していません。 202
デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>、上書きされたタグ数 = <数>。 211
デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>、上書きされなかったタグ数 = <数>。 211
デバイス '<デバイス>' のタグ生成結果。| 作成されたタグ数 = <数>。 211
デバイス '<デバイス>' は自動的に格下げされました。 205
デバイス '<デバイス>' へのタグ自動生成が完了しました。 209
デバイス '<名前>' 上のアドレス '<アドレス>' に書き込めません。 241
デバイスが応答していません。 239
デバイスが応答していません。| ID = '<デバイス>'。 239
デバイスとは 76
デバイスドライバ DLL がロードされませんでした。 215, 218
デバイスドライバは見つからなかったか、ロードできませんでした。| ドライバ = '<名前>'。 215
デバイスのアドレスに書き込めません。| アドレス = '<アドレス>'。 240
デバイスのプロパティ - タグ生成 80
デバイスの検出でデバイスの上限 <数> を超えました。検出範囲を絞り込んでから再試行してください。 201
デバイスの追加と構成 125
デバイスを追加できません。重複したデバイスがこのチャンネルにすでに存在している可能性があります。 201
デバイス起動時 81
デバイス作成ウィザード 126
デバイス要求ポール 151
デフォルト 47
テンプレート 96

と

ドライバ 77
ドライバ DLL '<名前>' をロードできません。 204
ドライバ DLL '<名前>' をロードできません。理由 206
ドライバがタグを処理している間にこのプロパティを変更することはできません。 241
ドライバがタグを処理している間はこのページ上のアイテムを変更できません。 241
ドライバの初期化に失敗しました。 237
ドライバレベルのエラーによってチャンネルを追加できませんでした。 199
ドライバレベルのエラーによってデバイスを追加できませんでした。 199

は

バージョンが一致しません。 199
はじめに 16
パスワード 34, 36, 51, 67, 122, 134-135
パスワードを使用した .OPF および .JSON プロジェクトファイルの保存はサポートされていません。暗号化されたプロ

ジェクトファイルを保存するには、.SOPF を使用してください。 219
バックアップ 30

ふ

ファイル IO エラーまたはデータストア破損のため、「ストアアンドフォワード」データストアがリセットされました。 232
ファイル名が既存のファイル
 '<名前>' を上書きしないようにしてください。 219
ファイル名には 1 つ以上の無効な文字が含まれています。 219
ファイル名は <サブディレクトリ>/<名前>.{json,opf,sopf} の形式でなければなりません 219
ファイル名は空にはできません。 219
プラグイン DLL '<名前>' をロードできません。 206
プラグイン DLL '<名前>' をロードできません。理由 206
プラットフォームによって開始された設定の変更を適用できません。アクセス許可が拒否されました。| ユーザー = '<ユーザー名>'。 233
プラットフォーム上での自動バインド完了イベントのトリガーに失敗しました。 228
プレビュー 88
プロジェクト '<名前>' のバックアップが '<パス>' に作成されました。 210
プロジェクト <名前> をロードできません 200
プロジェクトツリービュー 46
プロジェクトのテスト 135
プロジェクトのプロパティ 48
プロジェクトのプロパティ- 識別 48
プロジェクトのプロパティ- DDE 52
プロジェクトのプロパティ- FastDDE/SuiteLink 55
プロジェクトのプロパティ- iFIX PDB 設定 56
プロジェクトのプロパティ- OPC .NET 53
プロジェクトのプロパティ- OPC AE 54
プロジェクトのプロパティ- OPC DA コンプライアンス 48
プロジェクトのプロパティ- OPC HDA 57
プロジェクトのプロパティ- OPC UA 50
プロジェクトのプロパティ- ThingWorx ネイティブインタフェース 58
プロジェクトのロード中に無効なモデルが見つかりました。| デバイス = '<デバイス>'。 201
プロジェクトの設計 122
プロジェクトの保存 133
プロジェクトファイル
 '<名前>' を開けませんでした。 218
プロジェクトファイル <名前> を保存できません 201
プロジェクトファイルが無効です。 199
プロジェクトファイルはこのソフトウェアのより新しいバージョンで作成されました。 210
プロジェクトファイルを '<パス>' にバックアップできませんでした [<理由>]。保存操作は中止されました。保存先ファイルがロックされておらず、読み取り/書き込みのアクセス権があることを確認してください。バックアップしないでこのプロジェクトの保存を続行するには、「ツール」|「オプション」|「一般」でバックアップオプションを選択解除してからこのプロジェクトを保存し直してください。 200
プロジェクトを開いています。| プロジェクト = '<名前>'。 217
プロジェクトを閉じています。| プロジェクト = '<名前>'。 217

プロジェクト設定の変更が構成 API から開始されたため、ThingWorx 接続を再初期化しています。 235
プロジェクト設定の変更による ThingWorx 接続の再初期化が、プラットフォームから開始されました。 234
プロセスモード 19
プロセス配列データ 144
プロトコル 79-80
プロパティエディタ 47
プロパティタグ 110
プロパティのタイプ 168
プロパティの自動バインドに失敗しました。| 名前 = '<プロパティ名>'。 230
プロパティの定義 166
プロパティへの書き込みに失敗しました。| Thing 名 = '<名前>'、プロパティ名 = '<名前>'、理由 = <理由>。 232
プロパティへの書き込みに失敗しました。| プロパティ名 = '<名前>'、理由 = <理由>。 230
プロパティ変更時 81

ほ

ポート 30, 79-80
ポート <ポート番号> で、構成 API が SSL ありで開始しました。 223, 227
ポート <ポート番号> で、構成 API が SSL なしで開始しました。 223, 227
ポーリング時 82
ボタンバー 44

ま

マッピング先 94
マニフェスト作成リソース <名前> の読み取りに失敗しました。 210

め

メソッド 82
メニューバー 44
メモリ 31
メンバー 166

も

モデムタグ 112
モデム回線 '<モデム>' の切断を開始しています。 209
モデム回線 '<回線>' を開くことができませんでした [TAPI エラー = <コード>]。 199
モデム回線が開きました
 '<モデム>'。 209
モデム回線が閉じました
 '<モデム>'。 209

モデル 77

ゆ

ユーザーインターフェースのナビゲーション 44

ユーザーがユーザーグループから移動しました。| ユーザー = '<名前>', 古いグループ = '<名前>', 新しいグループ '<名前>'。 213

ユーザーがユーザーグループに追加されました。| ユーザー = '<名前>', グループ = '<名前>'。 211, 213

ユーザーが無効になりました。| ユーザー = '<名前>'。 212-213

ユーザーが有効になりました。| ユーザー = '<名前>'。 212-213

ユーザーグループが作成されました。| グループ = '<名前>'。 211, 213

ユーザーグループが無効になりました。| グループ = '<名前>'。 212-213

ユーザーグループが有効になりました。| グループ = '<名前>'。 212, 214

ユーザーグループでアクセス許可定義が変更されました。| グループ = '<名前>'。 211, 213

ユーザーグループの名前が変更されました。| 古い名前 = '<名前>', 新しい名前 = '<名前>'。 211, 213

ユーザーのパスワードが変更されました。| ユーザー = '<名前>'。 212, 214

ユーザーの名前が変更されました。| 古い名前 = '<名前>', 新しい名前 = '<名前>'。 211, 213

ユーザー情報が無効または見つかりません。 214

ユーザー定義のタグの追加 127

ら

ライブラリ<名前>のロードに失敗しました。 210

ランタイムが終了処理を実行しています。 221

ランタイムサービスが開始されました。 221

ランタイムのシャットダウンが完了しました。 221

ランタイムの再初期化が開始されました。 221

ランタイムの再初期化が完了しました。 221

ランタイムプロジェクトがリセットされました。 218

ランタイムプロジェクトが置き換えられました。| 新しいプロジェクト = '<パス>'。 218

ランタイムプロジェクトが置換されました。 222

ランタイムプロジェクトの更新に失敗しました。 214

ランタイムプロジェクトの読み込みに失敗しました。 214

ランタイムプロジェクトは '<名前>' として保存されました。 222

ランタイムプロジェクトを置き換えるにはユーザーアクセス許可が不十分です。 214

ランタイムプロセスが開始されました。 221

ランタイムへの接続に失敗しました。| ランタイムのホストアドレス = '<ホストアドレス>', ユーザー = '<名前>', 理由 = '<理由>'。 214

ランタイム動作モードの変更が完了しました。 212

ランタイム動作モードを変更しています。 212

り

リソース 244

リモートランタイムとの同期化に失敗しました。 216

ろ

ローカルポートへのバインド中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 239

ログの設定 31

ログファイルのパス 31

呂

暗号 135

暗号化 38, 58, 122, 133, 135

暗号化されたプロジェクトファイル (.SOPF) にはパスワードが必要です。 219

暗号化されたプロジェクトを開く 135

盛

一般 77

毘

永続データストア 31

永続モード 31

永続性なし 31

彼

応答コード 182

于

仮想ネットワークモードが変更されました。すべてのチャンネルと仮想ネットワークがこの影響を受けます。仮想ネットワークモードの詳細についてはヘルプを参照してください。| 新しいモード = '<モード>'。 217

嗜

回線 '<modem>' で '<数値>' をダイヤルしています。 208

回線 '<モデム>' が <rate> ボーで接続されました。 208

回線 '<モデム>' が接続されました。 209

回線 '<モデム>' でのダイヤルがユーザーによってキャンセルされました。 208

回線 '<モデム>' で着信呼び出しが検出されました。 209

回線 '<モデム>' は切断されています。 208
回線 '<回線>' でダイヤルできません。 203
回線 '<回線>' でハードウェアエラーが発生しました。 203
回線 '<回線>' にモデム構成を適用できません。 205
回線 '<回線>' はすでに使用されています。 203
回線 '<回線>' への接続で通信ハンドルが提供されませんでした。 203

梱

概要

iFIX アプリケーション内でのデータブロックの作成 153

抗

拡張データストア 31

柜

格下げまでのタイムアウト回数 79

格下げ期間 79

格下げ時に要求を破棄 79

笛

管理メニューへのアクセス 26

管理者のパスワードが現在のユーザーによってリセットされました。| 管理者名 = '<名前>'、現在のユーザー = '<名前>'。 213

管理者のパスワードのリセットに失敗しました。| 管理者名 = '<名前>'。 214

管理者のパスワードのリセットに失敗しました。現在のユーザーは Windows 管理者ではありません。| 管理者名 = '<名前>'、現在のユーザー = '<名前>'。 214

镁

間隔 82

斂

既存のデータストアは古いバージョンのサーバーで作成されたため、このデータストアに追加できません。データストアは再作成されました。| データストアのパス = '<パス>'。 237

磨

強固なセキュリティ 18, 133

珞

現在の言語ではXML プロジェクトをロードできません。XML プロジェクトをロードするには、サーバー管理で製品言語の選択を英語に変更してください。 201

楫

構成 API サービス 163

構成が変更されたため、データストアは上書きされました。| データストアのパス = '<パス>'。 237

構成セッションが<名前> (<名前>) によって開始されました。 222

構成セッションがアクティブなのでユーザー定義のスタートアッププロジェクトは無視します。 220

構成ユーティリティはサードパーティ製構成アプリケーションと同時に実行できません。両方のプログラムを閉じてから、使用するプログラムだけを開いてください。| 製品 = '<名前>'。 217

瀝

災害復旧 30

任

作成 82

凧

削除 81

卢

参照先デバイス '<チャンネルデバイス>' でのモデルタイプの変更を却下しています。 204

伟

使用しているアプリケーションキーが適切にフォーマットされ、有効であることを確認してください。 234

使用中のプロジェクトと同じであるため、プロジェクト置換の要求を却下しています
'<名前>'。 219

扱

指定されたアドレスはデバイス上で有効ではありません。| 無効なアドレス = '<アドレス>'。 241

指定されたソースは有効ではありません。サブスクリプションフィルタを設定できませんでした。| ソース = '<ソース名>'。
227

指定されたネットワークアダプタは、チャンネル '%1' | アダプタ = '%2' で無効です。 207

指定されたパラメータでは COM ポートを設定できません。| ポート = COM<数値>、OS エラー = <エラー>。 237

指定された領域は有効ではありません。サブスクリプションフィルタを設定できませんでした。| 領域 = '<領域名>'。
226

狛

資格証明 51, 134

既

時刻の同期化 82

時刻同期化しきい値 82

膊

自動ダイヤル 120

自動ダイヤルを行うにはその電話帳に 1 つ以上の番号が含まれている必要があります。| チャンネル = '<チャンネル>'。
207

自動格下げ 79

自動生成されたタグ '<タグ>' はすでに存在し、上書きされません。 202

自動生成による上書きが多すぎるため、エラーメッセージの書き込みを停止しました。 202

諸

識別 77

酤

重複タグ 81

暘

書き込み可否のチェック中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 242

背

証明書 41

証明書のロードに失敗したため、UA Server を開始できません。 225

訓

詳細ビュー 46

鐵

上書き 81

儷

冗長 83

啟

新しいプロジェクトの開始 122

撮

親グループ 81

攘

数値範囲を追加 87

璿

生データ値 89

生成 81

雹

静的タグ(ユーザー定義) 91

静的テキストを追加 86

捅

接続 30

接続されたデバイスで電源サイクルを行ったときの通信の問題の解決 145

接続に失敗しました。アダプタにバインドできません。| アダプタ = '<名前>'。 238

接続のタイムアウト 83

接続バッファ内の容量不足が原因で、1 つまたは複数の値変更の更新が失われました。| 失われた更新の数 = <数>。 228

接続中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 242

覬

設定 26

設定 - Config API サービスの設定 38

設定 - ProgID リダイレクト 32

設定 - ユーザーマネージャ 33, 37

設定 - ランタイムオプション 29

設定 - ランタイムプロセス 28

設定 - 管理 27

設定 - 構成 27

設定 - 証明書ストア 40

綱

絶対 82

絺

線形 89

橄

組み込み診断 183

攄

操作方法 142

对

属しているデバイスが無効になっているのでアイテム参照 '<名前>' への書き込み要求は却下されました。 221

勸

単一ファイル 31

辺

通信シリアル化タグ 115

通信タイムアウト 82-83

通信パラメータ 79

通信を再確立可能かどうかを判定するためデバイス '<デバイス>' は自動昇格されました。 210

通信管理 118

通信診断 194

嬭

定義済みのスタートアッププロジェクトにランタイムプロジェクトが置き換えられました。次回の再起動時にランタイムプロジェクトは '<名前>' から復元されます。 220

选

適切なネットワークケーブルの選択 146

適用 47

際

電話帳 119

電話番号が無効です (<数値>)。 208

電話番号の優先順位が変更されました。| 電話番号名 = '<名前>'、更新後の優先順位 = '<優先順位>'。 210

紾

統計タグ 111

創

動作モード 77

動的タグ 90

動的タグアドレス指定の使用 148

唇

同時クライアント 164

募

匿名 33, 51

誣

読み取り可否のチェック中にソケットエラーが発生しました。| エラー = <エラー>、詳細 = '<情報>'。 242

読み取り専用アイテム参照 '<名前>' に対する書き込み要求が却下されました。 220

霧

非正規化浮動小数点値の使用 150

饅

負数化 90

骯

復号 135

枸

複数タグの生成 86

複数のタグの生成 130

𣎵

平方根 89

攸

保存されたデータが「ストアアンドフォワード」データストアから削除されました。 235

站

埋め込み依存ファイルの保存に失敗しました。| ファイル = '<パス>'。 217

烁

無効な ProgID エントリが ProgID リダイレクトリストから削除されました。| ProgID = '<ID>'。 212

無効な XML ドキュメント 200, 216

無効なデータストアパスのため、データストアへの追加でエラーが発生しました。| パス = '<パス>' 236

無効なプロジェクトファイル

'<名前>'。 218

厭

名前 77

名前変更に失敗しました。その名前のオブジェクトがすでに存在します。| 提案された名前 = '<名前>'。 216

名前変更に失敗しました。名前にピリオドや二重引用符を含めたり、名前をアンダースコアで始めることはできません。| 提案された名前 = '<名前>'。 216

裕

要求のタイムアウト 83

要求間遅延 83