

SIMATIC HMI

WinCC V7.5 SP2 WinCC:設定および通信

システムマニュアル

マルチユーザシステム

1

ファイルサーバー

2

WinCC ServiceMode

3

リダンダントシステム

4

プロセス通信

5

通信チャンネル

6

通信－診断

7

OPC - オープンな接続性
(Open Connectivity)

8

REST インターフェース

9

WinCC/Cloud Connector




10

オンラインヘルプの印刷

法律上の注意

警告事項

本書には、ユーザーの安全性を確保し製品の損傷を防止するうえ守るべき注意事項が記載されています。ユーザーの安全性に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。以下に表示された注意事項は、危険度によって等級分けされています。

 危険
回避しなければ、直接的な死または重傷に至る危険状態を示します。
 警告
回避しなければ、死または重傷に至るおそれのある危険な状況を示します。
 注意
回避しなければ、軽度または中度の人身傷害を引き起こすおそれのある危険な状況を示します。
通知
回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します。


複数の危険レベルに相当する場合は、通常、最も危険度の高い事項が表示されることになっています。安全警告サイン付きの人身傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

有資格者

本書が対象とする製品/システムは必ず有資格者が取り扱うものとし、各操作内容に関連するドキュメント、特に安全上の注意及び警告が遵守されなければなりません。有資格者とは、訓練内容及び経験に基づきながら当該製品/システムの取り扱いに伴う危険性を認識し、発生し得る危害を事前に回避できる者をいいます。

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

 警告
シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限りです。製品を正しく安全にご使用いただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

商標

®マークのついた称号はすべて Siemens AG の商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。しかしなお、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記載内容については定期的に検証し、訂正が必要な場合は次の版で更新いたします。

目次

1	マルチユーザシステム	15
1.1	WinCC のマルチユーザシステム	15
1.2	WinCC のクライアント/サーバーシステム	16
1.3	標準的な設定	19
1.4	数量構造および特性	21
1.5	クライアント/サーバーシナリオ	24
1.6	サーバーコンフィグレーション	28
1.6.1	サーバーコンフィグレーション	28
1.6.2	サーバー側でのプロジェクトの新規作成	30
1.6.3	コンピュータリストへのクライアント登録方法	32
1.6.4	オペレータオーソリゼーションのコンフィグレーション方法	33
1.6.5	パッケージエクスポートのコンフィグレーション方法	36
1.6.6	サーバープロジェクトでのクライアントのコンフィグレーション	40
1.7	クライアントコンフィグレーション	44
1.7.1	クライアントコンフィグレーション	44
1.7.2	クライアント側でのプロジェクトの新規作成	46
1.7.3	インポートパッケージのコンフィグレーション	49
1.7.4	標準サーバーの設定方法	52
1.7.5	優先サーバーのコンフィグレーション方法	55
1.7.6	クライアントのピクチャの開始のコンフィグレーション	58
1.7.7	異なるサーバーからのピクチャの表示	60
1.7.8	クライアント側でのピクチャ変更のコンフィグレーション	62
1.7.9	異なるサーバーからのデータの使用	64
1.7.10	異なるサーバーからのメッセージの表示	67
1.7.11	複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション	68
1.8	ランタイムでのシステム動作	70
1.8.1	ランタイムでのシステム動作	70
1.8.2	サーバーのスタートアップ	73
1.8.3	クライアントのスタートアップ	74
1.8.4	複数のネットワークカードを使ったサーバー通信の特殊機能	75
1.8.5	サーバーのシャットダウン	77
1.8.6	クライアントのシャットダウン	77
1.9	リモートコンフィグレーション	79
1.9.1	リモートコンフィグレーション	79
1.9.2	暗号化通信	81

1.9.3	サブネットに接続されていないコンピュータへのアクセス方法	83
1.9.4	複数のクライアントからのプロジェクトへのアクセス	87
1.9.5	編集用にプロジェクトを開く方法	89
1.9.6	サーバープロジェクトピクチャの編集方法	91
1.9.7	プロジェクトの実行方法	93
1.9.8	プロジェクトの終了方法	95
1.10	クライアント/サーバーシステムの OPC インターフェースの使用	97
2	ファイルサーバー	99
2.1	ファイルサーバーのセットアップ	99
3	WinCC ServiceMode	101
3.1	WinCC ServiceMode:標準プロジェクトとサービスプロジェクト	101
3.2	サービスプロジェクトの設定	103
3.3	サービスプロジェクトの使用と制約	104
3.4	WinCC ステータスおよびシステムトレイでのコントロール	106
3.5	ファンクションと前提条件	109
3.5.1	サービスプロジェクトの操作モード	109
3.5.2	サービスプロジェクトの実行の必要条件	110
3.6	WinCC ServiceMode の設定	112
3.6.1	プロジェクトをサービスプロジェクトとして定義する方法	112
3.6.2	サービスプロジェクト用の Autostart の設定方法	113
3.7	ランタイムのサービスプロジェクト	115
3.7.1	サービスプロジェクトを起動する方法	115
3.7.2	起動したサービスプロジェクトにログインおよびログオフする方法	116
3.7.3	サービス目的でのインタラクティブな操作を有効にする方法	117
4	リダンダントシステム	119
4.1	2 重化	119
4.2	WinCC の 2 重化	120
4.3	冗長システムの必要条件	123
4.4	2 重化の仕組み	125
4.5	冗長性システムの構成	130
4.5.1	冗長システムのセットアップ方法	130
4.5.2	同一ファンクションのコンフィグレーション	132
4.5.3	冗長サーバーを構成する方法	133
4.5.4	ユーザーアーカイブの同期の設定方法	136
4.5.5	冗長サーバーのプロジェクトの複製方法	138
4.5.6	ランタイムでの冗長プロジェクトの複製方法	140
4.6	WinCC 冗長性のシナリオ	142

4.6.1	プロセス接続エラー発生時のクライアントの切り替え.....	142
4.6.2	故障シナリオ.....	146
4.6.2.1	故障シナリオ.....	146
4.6.2.2	シナリオ 1：ランタイムにないサーバーコンピュータのプロジェクト.....	147
4.6.2.3	シナリオ 2 パートナサーバーへの接続エラー.....	149
4.6.2.4	シナリオ 3 クライアントへのネットワーク接続エラー.....	151
4.6.2.5	シナリオ 4 プロセス接続の不良.....	152
4.6.2.6	シナリオ 5 ソフトウェアエラー.....	153
4.6.3	WinCC リダンダントシステムタグ.....	154
4.6.4	WinCC リダンダントシステムメッセージ.....	160
5	プロセス通信.....	165
5.1	通信の基本.....	165
5.2	接続設定の基本規則.....	166
5.3	WinCC プロセス通信.....	167
5.3.1	WinCC プロセス通信.....	167
5.3.2	WinCC 通信の原則.....	168
5.3.3	ランタイム時の接続ステータスタグの設定.....	170
5.3.4	外部タグ.....	173
5.3.4.1	外部タグ.....	173
5.3.4.2	新規接続の作成方法.....	176
5.3.4.3	外部タグを以下のように設定します.....	177
5.3.4.4	WinCC データタイプでソートしたフォーマット調整.....	178
5.3.4.5	AS データタイプによってソートされたフォーマット調整.....	187
5.3.4.6	BinWrite メカニズムの原則.....	201
5.3.4.7	BinWrite 付きタグの設定方法.....	202
5.3.5	Ethernet 経由のカップリングのポートアドレス.....	204
5.3.6	SIMATIC S7-PLCSIM Advanced.....	205
6	通信チャンネル.....	207
6.1	Allen Bradley - Ethernet IP.....	207
6.1.1	WinCC チャンネル"Allen Bradley - Ethernet IP".....	207
6.1.2	チャンネルユニットの割り付け.....	207
6.1.3	サポートされるデータタイプ.....	208
6.1.4	チャンネルの設定.....	209
6.1.4.1	チャンネル"Allen Bradley - Ethernet IP"の設定.....	209
6.1.4.2	[Allen Bradley - Ethernet IP]チャンネル用に接続を設定する方法.....	210
6.1.4.3	例：通信パス.....	211
6.1.4.4	タグの設定.....	214
6.2	三菱製 Ethernet.....	226
6.2.1	WinCC チャンネル"三菱製 Ethernet".....	226
6.2.2	サポートされるデータタイプ.....	226
6.2.3	チャンネルの設定.....	227
6.2.3.1	"三菱製 Ethernet"チャンネルの設定.....	227

6.2.3.2	"三菱製 FX3U シリーズ"チャンネルユニット接続を設定する方法	228
6.2.3.3	"三菱製 Q シリーズ"チャンネルユニット接続を設定する方法	230
6.2.3.4	「三菱製 iQ シリーズ」チャンネルユニット接続を設定する方法	231
6.2.3.5	「三菱製 iQ-F シリーズ」チャンネルユニットの接続設定方法	233
6.2.3.6	タグの設定	234
6.3	Modbus TCP/IP	239
6.3.1	"Modbus TCP/IP"チャンネル	239
6.3.2	サポートされるデータタイプ	239
6.3.3	チャンネルの設定	240
6.3.3.1	"Modbus TCP/IP"チャンネルの設定	240
6.3.3.2	接続の設定方法	241
6.3.3.3	タグの設定	242
6.4	OPC チャンネル	248
6.4.1	WinCC OPC チャンネル	248
6.4.2	OPC 項目マネージャ	250
6.4.3	サポートされている WinCC データタイプの概要	254
6.4.4	WinCC OPC DA クライアント	255
6.4.4.1	WinCC OPC DA クライアントの機能	255
6.4.4.2	OPC 項目マネージャによる WinCC タグへのアクセス方法	257
6.4.4.3	OPC 項目マネージャによらない WinCC タグへのアクセス	263
6.4.4.4	WinCC OPC DA クライアント上での構造体の使用	265
6.4.4.5	OPC DA 通信に障害が発生した場合のエラー処理	269
6.4.5	WinCC OPC XML クライアント	275
6.4.5.1	WinCC OPC XML クライアントの機能	275
6.4.5.2	OPC 項目マネージャによる WinCC タグへのアクセス方法	276
6.4.5.3	OPC 項目マネージャを使用しない WinCC タグへのアクセス	281
6.5	OPC UA WinCC チャンネル	283
6.5.1	WinCC チャンネル「OPC UA WinCC チャンネル」	283
6.5.2	サポートされているデータタイプの概要	284
6.5.3	OPC UA チャンネルの設定	285
6.5.3.1	全般シーケンス	285
6.5.3.2	WinCC プロジェクトへの OPC UA WinCC チャンネルの挿入	286
6.5.3.3	OPC UA サーバーへの接続作成	286
6.5.3.4	NAT ルータ経由の OPC UA サーバーへの接続作成	291
6.5.3.5	証明書による認証の設定。	293
6.5.3.6	OPC UA サーバーへの接続の設定方法	300
6.5.3.7	OPC UA タグの設定	303
6.5.3.8	WinCC における OPC UA タイプの使用	310
6.5.3.9	WinCC における OPC UA アラームの使用	317
6.5.4	OPC UA WinCC チャンネルの OPC UA 配列	328
6.5.4.1	WinCC の配列	328
6.5.4.2	OPC UA 配列を手動で作成および設定する方法	332
6.5.5	エラー処理	334

6.5.5.1	OPC DA 通信に障害が発生した場合のエラー処理	334
6.6	PROFIBUS DP	335
6.6.1	WinCC チャンネル"PROFIBUS DP"	335
6.6.2	WinCC ドライバ Profibus DP のプロパティ	335
6.6.3	「Profibus DP」ドライバの統合	336
6.6.4	「PROFIBUS DP」ドライバの設定	337
6.7	S5 Ethernet 第 4 層	340
6.7.1	WinCC チャンネル"SIMATIC S5 Ethernet 第 4 層"	340
6.7.2	タグのデータタイプ	341
6.7.3	チャンネルの設定	342
6.7.3.1	チャンネル"SIMATIC S5 Ethernet 第 4 層"の設定	342
6.7.3.2	接続の設定方法	342
6.7.3.3	タグの設定	345
6.7.3.4	システムパラメータ	353
6.7.4	付録	356
6.7.4.1	付録	356
6.7.4.2	内部エラーコードと定数	356
6.8	S5 PROFIBUS FDL	365
6.8.1	WinCC チャンネル"SIMATIC S5 Profibus FDL"	365
6.8.2	サポートされるデータタイプおよびデータ範囲	366
6.8.3	WinCC チャンネル"SIMATIC S5 Profibus FDL"の機能	368
6.8.4	チャンネルの設定	370
6.8.4.1	チャンネル"SIMATIC S5 Profibus FDL"の設定方法	370
6.8.4.2	チャンネルユニット"FDL (CP5412/A2-1)"	370
6.8.4.3	接続の設定方法	371
6.8.4.4	タグの設定	373
6.8.4.5	システムパラメータ	377
6.8.5	特殊ファンクション	379
6.8.5.1	"SIMATIC S5 Profibus FDL"チャンネルの特殊ファンクション	379
6.8.5.2	[SIMATIC S5 Profibus FDL]チャンネルの未処理データタグ	379
6.8.5.3	通信タイプの設定	381
6.8.6	"SIMATIC S5 Profibus FDL"チャンネルの設定例	385
6.8.6.1	"SIMATIC S5 Profibus FDL"チャンネルの設定例	385
6.8.6.2	AS のデータ処理ブロックの設定方法	385
6.8.6.3	I/O フィールドの設定方法	388
6.9	S5 Programmiers Port AS511	390
6.9.1	WinCC チャンネル"SIMATIC S5 Programmiers Port AS511"	390
6.9.2	タグのデータタイプ	390
6.9.3	チャンネルの設定	391
6.9.3.1	"SIMATIC S5 Programmiers Port AS51"チャンネルの設定	391
6.9.3.2	接続の設定方法	391
6.9.3.3	タグの設定	393
6.10	S5 Serial 3964R	399

6.10.1	WinCC チャンネル"SIMATIC S5 Serial 3964R"	399
6.10.2	タグのデータタイプ	399
6.10.3	チャンネルの設定	400
6.10.3.1	"SIMATIC S5 Serial 3964R"チャンネルの設定	400
6.10.3.2	接続の設定方法	400
6.10.3.3	タグの設定	402
6.11	SIMATIC S7 Protocol Suite	408
6.11.1	WinCC チャンネル"SIMATIC S7 Protocol Suite"	408
6.11.2	チャンネルユニットの選択	410
6.11.3	サポートされているデータタイプの概要	413
6.11.4	チャンネルのコンフィグレーション	414
6.11.4.1	"SIMATIC S7 Protocol Suite"チャンネル - コンフィグレーション	414
6.11.4.2	"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法	415
6.11.4.3	AS シンボルをオフラインでダウンロードする方法	416
6.11.4.4	AS 構造の構成方法	421
6.11.4.5	AS プロジェクトデータのエクスポート方法	423
6.11.4.6	チャンネルユニット	425
6.11.4.7	タグのコンフィグレーション	445
6.11.4.8	システムパラメータ	451
6.11.5	特殊ファンクション	458
6.11.5.1	"SIMATIC S7 Protocol Suite"チャンネルの特殊ファンクション	458
6.11.5.2	S7 ファンクションブロック AR_SEND によるデータ交換	458
6.11.5.3	"SIMATIC S7 Protocol Suite"チャンネルの未処理データタグ	493
6.11.5.4	ソフトウェアの二重化	502
6.12	SIMATIC S7-1200, S7-1500 Channel	513
6.12.1	「SIMATIC S7-1200, S7-1500 Channel」チャンネル	513
6.12.2	サポートされているデータタイプの概要	514
6.12.3	チャンネルのコンフィグレーション	514
6.12.3.1	"SIMATIC S7-1200, S7-1500 Channel"チャンネルのコンフィグレーション	514
6.12.3.2	未処理データタグの設定	517
6.12.3.3	接続の設定方法	519
6.12.3.4	最適化されたブロックアクセスなしでのタグの構成方法	520
6.12.3.5	最適化されたブロックアクセスありでのタグの構成方法	522
6.12.3.6	AS シンボルをオフラインでダウンロードする方法	525
6.12.3.7	AS 構造の構成方法	530
6.12.3.8	AS プロジェクトデータのエクスポート方法	532
6.12.3.9	冗長システム S7-1500R/H	534
6.12.3.10	冗長化制御システムの設定	536
6.12.3.11	S7-1500R/H のソフトウェアの冗長化	540
6.13	SIMATIC TI Ethernet 第 4 層	546
6.13.1	WinCC チャンネル"SIMATIC TI Ethernet 第 4 層"	546
6.13.2	タグのデータタイプ	547
6.13.3	チャンネルの設定	547

6.13.3.1	チャンネル"SIMATIC TI Ethernet 第 4 層"の設定.....	547
6.13.3.2	接続の設定方法.....	547
6.13.3.3	タグの設定.....	550
6.13.3.4	システムパラメータ	555
6.14	SIMATIC TI Serial.....	559
6.14.1	WinCC チャンネル"SIMATIC TI シリアル"	559
6.14.2	タグのデータタイプ	559
6.14.3	チャンネルの設定	560
6.14.3.1	"SIMATIC TI シリアル"チャンネルの設定	560
6.14.3.2	接続の設定方法.....	560
6.14.3.3	タグの設定.....	562
6.15	SIMOTION	568
6.15.1	WinCC チャンネル"SIMOTION"	568
6.15.2	サポートされているデータタイプの概要.....	568
6.15.3	チャンネルの設定	569
6.15.3.1	"SIMOTION"チャンネルの設定	569
6.15.3.2	SIMOTION SCOUT プロジェクトのエクスポート方法	570
6.15.3.3	Simotion Mapper を使った WinCC プロジェクトの作成方法	571
6.15.3.4	Simotion Mapper を使った WinCC プロジェクトの変更方法	572
6.15.3.5	接続パラメータの変更方法	574
6.15.3.6	タグアドレスの変更方法.....	575
6.15.3.7	システムパラメータ設定	576
6.15.4	診断"SIMOTION"チャンネル.....	579
6.15.4.1	"SIMOTION"チャンネルの診断の可能性	579
6.15.4.2	ログファイルのエントリの説明	580
6.16	システム情報	584
6.16.1	"システム情報"チャンネル.....	584
6.16.2	WinCC システム情報チャンネル	584
6.16.3	サポートされるシステム情報の概要	586
6.16.4	他のソフトウェアの構成要素 との違い	591
6.16.5	チャンネルのコンフィグレーション	592
6.16.5.1	システム情報チャンネルをコンフィグレーションする方法.....	592
6.16.6	システム情報の評価と表示例	593
6.16.6.1	システム情報の呼び出しと評価方法	593
6.16.6.2	システム情報チャンネル内のタグをコンフィグレーションする方法	593
6.16.6.3	I/O フィールドで時間を表示する方法.....	595
6.16.6.4	空きディスクスペースを棒グラフで表示する方法.....	596
6.16.6.5	トレンドウィンドウで CPU 負荷を表示する方法	597
6.16.6.6	空きディスク容量に関するメッセージの設定方法.....	599
6.16.6.7	利用可能なディスク容量に関するメッセージを表示する方法	601
6.16.6.8	ステータス表示にプリンタステータスを表示する方法.....	602
6.16.6.9	WinCC スタートアップパラメータをチェックする方法	604
6.16.6.10	バーグラフの挿入方法	604

6.16.6.11	I/O フィールドの挿入方法	605
6.16.6.12	ランタイムの開始方法	605
6.16.7	特殊ファンクション	606
6.16.7.1	マルチユーザーおよびクライアントシステムにおける使用	606
6.16.7.2	複数のサーバーからのモニタシステム情報の例	606
7	通 信 — 診 断	615
7.1	チャンネルおよびタグの診断	615
7.2	エラー検出に関する一般情報	616
7.3	チャンネル診断	617
7.3.1	チャンネル診断	617
7.3.2	パフォーマンスタグによる接続の確認	617
7.3.3	[ステータス - 論理接続]ファンクションを使用してチャンネルを確認する方法	622
7.3.4	[チャンネル診断] に よ る チャンネルの診断	624
7.3.4.1	チャンネル診断の原理	624
7.3.4.2	ActiveX コントロールによるチャンネル診断	625
7.3.4.3	ActiveX コントロールとしてのチャンネル診断によるチャンネルのチェック方法	626
7.3.4.4	[チャンネル診断]を使用したチャンネルの診断	627
7.3.4.5	チャンネル診断によるチャンネルのチェック方法	628
7.3.4.6	チャンネルのトレースファンクションをコンフィグレーションする方法	629
7.3.4.7	ランタイムの開始方法	631
7.4	"システム情報"チャンネルの診断	632
7.4.1	"システム情報"チャンネル - 診断オプション	632
7.4.2	ログファイルのエントリの説明	632
7.4.3	無効なタグ値の原因の特定方法	633
7.4.3.1	無効なタグの原因の特定方法	633
7.4.3.2	チャンネルおよび接続のチェック方法	634
7.4.3.3	タグのチェック方法	636
7.5	診断チャンネル「SIMATIC S7-1200/S7-1500」	638
7.5.1	SysDiagControl によるシステム診断	638
7.5.2	システム診断の構成方法	642
7.6	"SIMATIC S7 Protocol Suite"チャンネルの診断	645
7.6.1	"SIMATIC S7 Protocol Suite"チャンネル - 診断オプション	645
7.6.2	ログファイルのエントリの説明	646
7.6.3	無効なタグ値の原因の特定方法	649
7.6.3.1	無効なタグの原因の特定方法	649
7.6.3.2	通信プロセッサの設定確認方法	649
7.6.3.3	SIMATIC NET での通信プロセッサのチェック	651
7.6.3.4	接続およびタグのコンフィグレーションのチェック方法	652

7.6.3.5	チャンネルおよび接続のチェック方法	654
7.6.3.6	タグのチェック方法	656
7.7	"SIMATIC S5 Profibus FDL"チャンネルの診断	658
7.7.1	"SIMATIC S5 PROFIBUS FDL"チャンネルの診断オプション	658
7.7.2	ログファイルのエントリの説明	659
7.7.3	無効なタグ値の原因の特定方法	660
7.7.3.1	無効なタグの原因の特定方法	660
7.7.3.2	通信プロセッサのコンフィグレーションのチェック	660
7.7.3.3	SIMATIC NET での通信プロセッサのチェック	662
7.7.3.4	接続およびタグのコンフィグレーションのチェック方法	663
7.7.3.5	チャンネルおよび接続のチェック方法	665
7.7.3.6	タグのチェック方法	667
7.8	"OPC"チャンネルの診断	669
7.8.1	"OPC"チャンネルの診断のための機能	669
7.8.2	ログファイルのエントリの説明	670
7.8.2.1	ログファイルのエントリの説明	670
7.8.2.2	"INFO"フラグのエントリ	671
7.8.2.3	"ERROR"フラグのエントリ	672
7.8.3	無効なタグ値の原因の特定方法	674
7.8.3.1	無効なタグの原因の特定方法	674
7.8.3.2	コンフィグレーションデータのチェック方法	674
7.8.3.3	チャンネルおよび接続のチェック方法	676
7.8.3.4	タグのチェック方法	678
7.9	タグの品質	680
7.9.1	タグの品質	680
7.9.2	タグの品質コード	683
7.9.3	タグステータス	690
7.9.4	モニタ接続ステータスへのタグステータスの使用	691
7.9.5	グローバルアクションによるタグステータスのモニタ	693
7.9.6	内部タグの作成方法	694
8	OPC - オープンな接続性(Open Connectivity)	697
8.1	OPC - オープンな接続性(Open Connectivity)	697
8.2	OPC の機能	698
8.3	OPC 仕様と互換性	699
8.4	WinCC における OPC の使用	700
8.5	WinCC OPC を使用するために Windows を構成する方法	702
8.6	WinCC OPC XML DA サーバー	703
8.6.1	WinCC OPC XML DA サーバーの機能	703
8.6.2	WinCC OPC XML DA サーバーのインストール	704
8.6.3	IIS を使用したセキュリティ設定の設定(WinCC OPC XML DA サーバー)	706

8.6.4	インストールのテスト(WinCC OPC XML DA サーバー)	707
8.7	WinCC OPC DA サーバー	709
8.7.1	WinCC OPC DA サーバーの機能.....	709
8.7.2	複数の OPC DA サーバーの使用	710
8.7.3	OPC DA サーバー名の問い合わせ	711
8.7.4	OPC DA 接続の例.....	713
8.7.4.1	WinCC 間の接続	713
8.7.4.2	WinCC と SIMATIC NET FMS OPC サーバー間の接続	717
8.7.4.3	WinCC と SIMATIC NET S7-OPC サーバー間の接続	719
8.7.4.4	WinCC と Microsoft Excel 間の接続	725
8.8	WinCC OPC HDA サーバー	730
8.8.1	WinCC OPC HDA サーバーの機能	730
8.8.2	WinCC OPC HDA サーバーのデータ構造.....	732
8.8.2.1	WinCC OPC HDA サーバーのデータ構造.....	732
8.8.2.2	サポートされている属性の概要	733
8.8.2.3	サポートされている集約の概要	733
8.8.2.4	サポートされているファンクションの概要	735
8.8.2.5	WinCC OPC HDA サーバーの時間形式	736
8.8.3	品質コード.....	738
8.8.4	サポートされている書き込みアクセス	739
8.8.5	OPC HDA 接続例.....	742
8.8.5.1	OPC HDA 接続例.....	742
8.8.5.2	HDA サーバーブラウザ	743
8.8.5.3	HDA サーバーブラウザを使用した WinCC アーカイブタグへのアクセスの設定方法...	744
8.8.5.4	WinCC アーカイブタグの値の読取り	746
8.8.6	WinCC の OPC HDA サーバーの非サイクリックロギング用特殊機能	748
8.9	WinCC OPC A&E サーバー	752
8.9.1	WinCC OPC A&E サーバーの機能.....	752
8.9.2	OPC A&E を使用した WinCC メッセージシステムのマッピング.....	754
8.9.2.1	OPC A&E を使用した WinCC メッセージシステムのマッピング.....	754
8.9.2.2	WinCC メッセージクラスおよびメッセージタイプのマッピング	755
8.9.2.3	WinCC メッセージの優先度のマッピング	756
8.9.2.4	WinCC メッセージシステムの属性.....	756
8.9.2.5	確認セオリー	759
8.9.3	OPC A&E の品質コード.....	761
8.9.4	OPC A&E 接続例	762
8.9.4.1	OPC A&E 接続例	762
8.9.4.2	WinCC メッセージシステムへのアクセスの設定方法.....	762
8.9.5	階層的アクセス機能付き OPC A&E サーバー.....	765
8.9.5.1	OPC A&E サーバーの機能	765
8.9.5.2	OPC A&E と階層的アクセス機能付き OPC A&E との違い.....	767
8.9.5.3	OPC A&E の WinCC メッセージシステムのマッピング	768
8.9.5.4	OPC A&E の品質コード.....	775

8.9.6	アーカイブメッセージの読取り	776
8.9.6.1	アーカイブイベントへのアクセス	776
8.9.6.2	OPC を使用したアーカイブメッセージへのアクセスの構文.....	777
8.9.6.3	アーカイブメッセージの読取りメソッド.....	779
8.9.6.4	アーカイブメッセージの識別	781
8.10	WinCC OPC UA サーバー	782
8.10.1	WinCC OPC UA サーバーの動作原理	782
8.10.2	OPC UA のセキュリティ概念	783
8.10.3	セキュリティメカニズムの構成	788
8.10.4	サポートされる OPC UA サービスとプロファイル	791
8.10.5	WinCC OPC UA サーバーの名前領域.....	793
8.10.6	OPC UA データアクセス	796
8.10.7	OPC UA Historical Access.....	797
8.10.8	OPC UA アラームと条件	798
8.10.9	WinCC メッセージシステムの属性.....	804
8.10.10	WinCC OPC UA サーバーの設定	806
8.10.10.1	WinCC OPC UA サーバーの設定ファイル.....	806
8.10.10.2	OPC UA サーバーのコンフィグレーション方法.....	811
8.11	診断	814
9	REST インターフェース	815
9.1	WinCC の REST インターフェース	815
9.2	WinCC REST サービスの設定方法.....	818
9.3	メソッドの概要.....	820
9.4	GET / POST メソッド	821
9.4.1	タグのランタイム値の読み取り	821
9.4.2	すべてのタグのランタイム値の読み取り	821
9.4.3	接続の設定データの読み取り	822
9.4.4	すべての接続の設定データの読み取り	823
9.4.5	タグの設定データの読み取り	824
9.4.6	すべてのタグの設定データの読み取り	825
9.4.7	タググループの設定データの読み取り	827
9.4.8	すべてのタググループの設定データの読み取り	827
9.4.9	構造タイプの設定データの読み取り	829
9.4.10	すべての構造タイプの設定データの読み取り	829
9.4.11	構造タイプのインスタンスの読み取り	832
9.4.12	複数の構造タイプのインスタンスの読み取り	833
9.5	PUT メソッド	835
9.5.1	値のタグへの書き込み	835
9.5.2	値の複数タグへの書き込み	835
9.6	設定データのフィルタ	837

10	WinCC/Cloud Connector	841
10.1	WinCC/Cloud Connector.....	841
10.2	ライセンシングクラウドコネクタ	844
10.3	MQTT を経由したクラウドへのデータ転送	845
10.4	REST を経由したクラウドへのデータ転送	849
10.5	WinCC タグ管理での設定	851
10.6	WinCC クラウドコネクタの MQTT 設定	853
10.7	WinCC クラウドコネクタの REST 設定	858
10.8	MQTT 経由のクラウド接続の設定方法	860
10.9	MindSphere 接続の設定方法	864
10.10	REST 経由のクラウド接続の設定方法	867
10.11	クラウド接続の診断.....	870
	索引.....	873

マルチユーザシステム

1.1 WinCC のマルチユーザシステム

内容

WinCC はクライアント/サーバーシステムのコンフィグレーションに使用でき、システム操作およびモニタ機能を複数のクライアントおよびサーバーに分散させることが可能です。したがって、大きなシステムの場合、個々のコンピュータにかかる負荷を減少させ、パフォーマンスを向上させることができます。

本章では以下を説明します。

- WinCC を使用して実行できるクライアント/サーバーシナリオ
- クライアント/サーバーシステムにおいてサーバーおよびクライアントをコンフィグレーションする方法
- 複数のサーバーに表示させるクライアントをコンフィグレーションする方法
- ランタイムでのクライアント/サーバーシステムの動作
- リモートクライアントからサーバープロジェクトをコンフィグレーションする方法

1.2 WinCC のクライアント/サーバーシステム

概要

WinCC は複数のクライアントおよびサーバーを備えたクライアント/サーバーシステムの設定に使用でき、大きなシステムをより効率的に操作およびモニタします。

サーバー間のオペレーションプロセスおよびモニタプロセスのタスクを複数のサーバーに分散することにより、個々のサーバーの利用率を減少させ、パフォーマンスを向上させます。さらに、WinCC を使用して、技術的にも位相的にも複雑な構造を有するシステム間のマッピングもできます。

クライアント/サーバーシステムは、たとえば以下のような場合に使用します。

- 複数のオペレータステーションおよびモニタステーション(クライアント)が、同じタスクを実行する必要がある大きなシステムの場合
- オペレータ制御タスクとモニタタスクを複数のオペレータステーション(システムの全メッセージを表示する中央クライアントなど)に分散させる場合。

クライアントは以下に使用できます。

- 1 台のサーバーを備えたマルチユーザシステムの設定 :
複数のクライアントがプロセスドライバ接続により、1 台のサーバーにあるプロジェクトにアクセスします。
マルチユーザシステムの場合、全データがサーバーから供給されるため、クライアントを設定する必要はありません。
- 複数のサーバーを備えた分散システムの設定 :
クライアントは、プロセスドライバ接続している複数のサーバーからのデータを表示できます。
分散システム内では、設定はクライアントごとに行われます。必要なサーバーデータはクライアントにインポートされ、修正された場合は自動的に更新されます。
プロセスデータはサーバーから提供されます。
- リモート設定 :
1 つのサーバープロジェクトは 1 台のクライアントから設定されます。
- リモートモニタリング :
1 つのサーバープロジェクトは 1 台のクライアントからモニタされます。

クライアント/サーバーシステムの設定条件

- WinCC を使用してクライアント/サーバーシステムを設定するには、[WinCC サーバー] オプションをすべての WinCC サーバーで使用可能にする必要があります。
- WinCC システムで使用されているすべての WinCC ステーションに同じ WinCC バージョンがインストールされている必要があります。
これらに含まれる内容例を以下に示します。
 - WinCC サーバー、冗長サーバー、アーカイブサーバー、ファイルサーバー
 - 独自のプロジェクトを持つ WinCC クライアント、独自のプロジェクトを持たない WinCC クライアント
 - WebNavigator サーバー、WebNavigator クライアント、WinCC/Audit、WinCC/Calendar オプションなどの WinCC オプションがインストールされた PCWinCC オプションがリリースされていない、異なる WinCC バージョンがサーバー、クライアント、または PC にインストールされた混合設定。
WinCC アップデートは、常に WinCC システム内のすべての PC にインストールしてください。クライアントまたはサーバーに異なるアップデートバージョンがインストールされると、WinCC プロジェクトで設定変更を行えないなどの状況が発生します。
- クライアント/サーバーシステム内の PC はすべてネットワーク (LAN) を介して相互接続する必要があります。
システム内のクライアントまたはサーバーとしてルータ経由で接続されている隣接するサブネットの PC にログオンすることもできます。
- デフォルトではリモートアクセスは無効です。
PC へのネットワークアクセスを有効にするには、Simatic Shell 設定でリモート通信を有効にします。

冗長システム

サーバーエラーなどの障害が発生した後などでもシステム作動を維持するには、冗長サーバーを設定します。冗長システムの設定の正確な手順については、WinCC 文書「冗長システム」を参照してください。

文書には冗長システムにおけるクライアントの設定に関する情報が含まれています。

冗長サーバーペアのクライアント

ローカルプロジェクトのないクライアントが含まれている WinCC プロジェクトは、サーバー(プライマリまたはスタンバイサーバー)上でのみ編集できます。

その間は、パートナーサーバーで WinCC プロジェクトを開くことはできません。WinCC エクスプローラとランタイムは閉じる必要があります。

1.2 WinCC のクライアント/サーバーシステム

プライマリサーバーまたはスタンバイサーバーをランタイムで負荷分散のための優先サーバーに設定することは依然として可能です。

ランタイム起動中はクライアント上で WinCC エクスプローラを終了した状態にする必要があります。ランタイムを起動するには、「AutoStart」を使用してください。これには、WinCC の「AutoStart の設定」ツールを使用します。

WinCC エクスプローラを開いたまま別のサーバーコンピュータに切り替えると、オペレータによる入力がロックされます(ユーザーインターフェースがグレースアウトします)。プロジェクトを開いたサーバーがクライアントの現在のサーバーでない場合、WinCC エディタを起動できません。

注記

WinCC プロジェクトを開くためのリモートアクセス

両方の冗長システムでプロジェクトを開いている場合、リモート設定はできません。

冗長システムを設定する場合、スタンバイコンピュータで WinCC プロジェクトを開かないようにします。

ランタイムで冗長化システムを設定するには、以下の手順で行います。

1. スタンバイコンピュータを終了し、プロジェクトを閉じます。
 2. プライマリサーバーをランタイムで、リモートまたはローカルに設定します。
 3. オンライン設定が完了したら、プロジェクトデュプリケータを使用してスタンバイコンピュータにオンラインでプロジェクトを複製します。
 4. スタンバイコンピュータで再びプロジェクトを開いて実行します。
-

下記も参照

クライアント/サーバーシステムの OPC インターフェースの使用 (ページ 97)

リモートコンフィグレーション (ページ 79)

ランタイムでのシステム動作 (ページ 70)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

標準的な設定 (ページ 19)

暗号化通信 (ページ 81)

1.3 標準的な設定

必要に応じて、異なるクライアント/サーバーソリューションを設定することができます。そのようにする場合、クライアント、ウェブクライアントおよびシンクライアントを使用するオプションがあります。

また、「数量構造および特性 (ページ 21)」の数量構造および特性に関する注記を順守します。

分散システムでの暗号化通信の使用

マルチユーザシステムのコンピュータ間に暗号化通信を設定します。

"Simatic Shell"を搭載したコンピュータに PSK キーを指定します。これは、共有の指定されたキーが通信の前に知られているコンピュータのみが、互いに通信することができるのを意味します。同じネットワークのための独自の PSK キーを使用して異なる環境を指定することもできます。

クライアント/サーバーシステムの異なる必要条件に対する設定

クライアント

設定に応じて、クライアント/サーバーシステムのクライアントは以下のことができます。

- 複数のクライアント上で 1 台のサーバーからの表示を行う(マルチユーザシステム)
- クライアント上で複数のサーバーの表示を行う(分散システム)
- クライアントから複数のサーバープロジェクト(リモート)を設定する。
- クライアントからサーバープロジェクト(リモート)を実行、終了させる。

複数のクライアントを設定するためには、各サーバーに[WinCC サーバー]オプションが必要です。

ウェブクライアント

たとえば以下の場合、ウェブクライアントはクライアントサーバーシステムにインストールされます。

- 狭帯域接続経由でのシステムへのアクセスが必要な場合
- データへの一時的アクセスのみが必要な場合
- インターネット経由などの長距離データアクセスが必要な場合

1.3 標準的な設定

ウェブクライアントには以下の利点があります。

- 異なるオペレーティングシステムを備えたクライアントコンピュータを設置できます。
- 1 台のサーバーに、複数のウェブクライアントから同時にアクセスできます。
- 大きな数量構造が実現できます。

ウェブクライアントを設定するには、WinCC の最小インストールおよび[WinCC WebNavigator]オプションが必要です。

シンククライアント

シンククライアントの主要な特徴は、基本的にウェブクライアントと同じです。ただし、他にも以下の特徴があります。

- また、堅牢なクライアントプラットフォームで使用することもできます。

シンククライアントを設定するには、WinCC の最小インストールおよび[WinCC WebNavigator]オプションが必要です。

下記も参照

クライアント/サーバーシナリオ (ページ 24)

異なるサーバーからのデータの使用 (ページ 64)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

異なるサーバーからのメッセージの表示 (ページ 67)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバー側でのプロジェクトの新規作成 (ページ 30)

コンピュータリストへのクライアント登録方法 (ページ 32)

パッケージエクスポートのコンフィグレーション方法 (ページ 36)

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

WinCC のクライアント/サーバーシステム (ページ 16)

数量構造および特性 (ページ 21)

暗号化通信 (ページ 81)

1.4 数量構造および特性

ランタイムの WinCC プロジェクトの特性

WinCC プロジェクトの特性は、システム設定、数量構造および設定によって異なります。これらの各要素は、画像の変更に要する時間およびアーカイブ速度などに影響することがあります。

お使いのシステムの設定に関する情報は、「標準的な設定 (ページ 19)」で参照できます。インストールに関する注意事項のハードウェア要件も順守してください。

設定に関する注記

次の注記は、マルチユーザーシステムだけでなく、すべてのタイプの WinCC プロジェクトにも適用されます。

ただし、これらの注記に留意する必要があります。マルチユーザーシステムの場合は、数量構造の増加が速く、設定の変更が大きくなる可能性があるため、特に注意が必要です。

設定時には、次の要素が WinCC プロジェクトの特性に影響することに注意してください。

- ウェブクライアント数
プロジェクトで処理されるタグ数は、使用されるそれぞれの追加ウェブクライアントと一緒に増加します。
各ウェブクライアントでは、WinCC プロジェクトの内部タグは、コンピュータに対してローカルなタグとして、ウェブサーバーで管理されます。設定に応じて、追加のウェブクライアントが管理されるタグの数を増加させます。
- プロセスウィンドウの画像ウィンドウの数
同時に表示される画像ウィンドウを最大 100 個にして使用することを推奨します。
- 画像ウィンドウのネスティング数(レベル)
画像オブジェクトの最大ネスティング深度として、20 レベルが承認されています。
- プロセス画面のスクリプト
同一サイクルを使用するすべてのスクリプトのプロセス時間を設定されたサイクル時間よりも長くならないようにする必要があります。

1.4 数量構造および特性

- タグ数
WinCC Runtime のサーバー負荷は、すべてのクライアントまたはウェブクライアントで同時に登録されているすべてのタグの合計によって発生します。
- モニタ数(マルチ VGA)
マルチ VGA が使用されている場合、WinCC クライアントの数が減ることがあります。この場合も、プロセス画像の数と複雑性により生じる潜在的なパフォーマンスの制限に注意してください。
モニタの数を 60 に制限することを推奨します。
例:すべての WinCC クライアントのそれぞれに 4 つのモニタを使用する場合、最大 15 の WinCC クライアントがサポートされます。

データの読み込み、書き込み、アーカイブ時に WinCC プロジェクトのパフォーマンスを分析するには、「@PRF_...」システムタグを使用します。

サーバーでの複数のクライアントの同時起動

1 つのサーバーに接続されている複数のクライアントを同時に起動すると、過度の負荷がかかる可能性があります。この場合、クライアントはタイムアウトします。

クライアントは連続して起動することを推奨します。

クライアントおよびサーバーの可能数

使用されているクライアントタイプおよびタイプ数によって、異なる数量構造が実現できます。混合システムも可能です。つまり、1 つのクライアントサーバーシステム内でクライアントとウェブクライアントを並列して使用できます。

カスタムプロジェクトのある WinCC クライアントのみを使用する場合、WinCC ネットワーク内で最大 50 のクライアントが 1 台のサーバーに並行してアクセスできます。WinCC クライアントは、ランタイム時に最大 18 台のサーバーにアクセスできます。

最大 36 台のサーバーを、18 組のリダンダントサーバーペアの形で使用できます。

ウェブクライアントのみを使用している場合、最大 151 台のクライアント(クライアント 1 台とウェブクライアント 150 台)の数量構造を実現できます。

マルチ VGA がクライアントの数に及ぼす影響についても配慮してください。

複合システムの設定

複合システムを設定する場合は、以下の経験則を遵守して最大数量構造を実現します。

以下の値は、クライアントタイプで定義されます。

- ウェブクライアント/シンクライアント = 1
- クライアント = 2
- [リモートの設定]機能を備えたクライアント = 4

すべてのクライアントの値の合計が、次の値を超えないようにします。

- オペレータ機能なしの WinCC サーバー:160(サーバーごと)
- オペレータ機能ありの WinCC サーバー:16(サーバーごと)

例:

設定	有意性
[リモートの設定]機能を備えた 3 台のクライアント	$3 \times 4 = 12$
5 台のクライアント	$5 \times 2 = 10$
138 台のウェブクライアント	$138 \times 1 = 138$
合計	160

注記

WinCC サーバーとの混合設定なし

他の WinCC サーバーにアクセスする WinCC サーバーの複合設定は許可されていません。

下記も参照

WinCC のクライアント/サーバーシステム (ページ 16)

クライアント/サーバーシステムの OPC インターフェースの使用 (ページ 97)

リモートコンフィグレーション (ページ 79)

ランタイムでのシステム動作 (ページ 70)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

標準的な設定 (ページ 19)

1.5 クライアント/サーバーシナリオ

はじめに

WinCC は、アプリケーションによって異なるクライアント/サーバーシナリオを実装できます。

マルチユーザシステム

マルチユーザシステムは一般的に、複数のサーバーにデータを分散することが必要な小規模システム用に設定されています。

プロセスドライバ接続のサーバーは、すべての中央機能および複数のオペレーティングステーション(クライアント)の制御を行うように設定されます。

個々のオペレータステーションは、同一のタスクまたは異なるタスクを実行することができます。

クライアントはそのオペレータオーソリゼーションに応じて、以下の目的に使用できます。

- システムのモニタ
- システムのモニタおよび作動
- サービスコンピュータなどのサーバープロジェクトのリモート設定

クライアントはその設定に応じて、以下に使用できます。

- たとえばシステムのさまざまなポイントからプロセスを操作できるようにする場合、プロジェクトで共通の表示を行います。
- メッセージなどにかぎり、プロジェクトによって異なる表示を行います。

特定のオペレーティングコンソール上のオペレータが使用できる機能を定義するために、ユーザーオーソリゼーションが発行されます。

設定タイプ:

- 独自のプロジェクトを持たないクライアント。オペレータステーションが独自のプロジェクトデータを必要としない場合は、サーバープロジェクトでクライアントを設定します。
- 独自のプロジェクトを持つクライアント。オペレータステーションが独自のプロジェクトデータを必要とする場合(例えば、異なる開始画面)には、クライアントプロジェクトを作成します。

分散システム

複数のサーバーを設置した分散システムは、一般的に大規模システムにおいて特に大量のデータ処理が必要な場合に使用されます。複数のサーバーにタスクを分散する結果、個々のサーバーにかかる負荷が軽減されます。これにより、より大規模な典型的なアプリケーションを使用した場合のシステムパフォーマンスが向上します。

WinCC システムで分散システムを設定する場合、プロセスタスクはプロセス手順または機能性のいずれかに従った設定により、サーバーに分散されます。

- 技術的分散の場合、各サーバーは、特定のプレスユニットまたはドライユニットなど、システムが技術的に制限できる領域を担当します。
- 機能的な分散の場合、各サーバーは、視覚化、アーカイブ、アラーム発行などの特定のタスクを担当します。

ランタイムでは、分散システムのクライアントはそれぞれ最大で 18 台の異なるサーバー、または冗長サーバーペアからのデータを表示できます。分散システムの各クライアントは、基本画像とわずかなローカルデータを使用して個別に設定されます。プロセスデータの表示に必要なサーバーデータはサーバーからクライアントに転送され、必要であれば自動的に更新されます。

ファイルサーバー

クライアント/サーバーシステムのファイルサーバーを使用して、全プロジェクトをシステムで一元的に保存、管理します。したがって、たとえば全プロジェクトのバックアップコピーを定期的に作成することが容易になります。プロセスドライバ接続のサーバーはファイルサーバーにアクセスし、ファイルサーバー上でプロジェクトの設定ができます。ファイルサーバーは設定専用に使えます。

ファイルサーバーは必要に応じてハードウェアコンポーネントを追加することにより、特定の要望に対応できます。これにより、たとえば、バックアップのセキュリティのためディスクをミラー化することが可能になります。

中央アーカイブサーバー

接続しているすべての WinCC サーバーのプロセス値およびメッセージは、中央アーカイブサーバー(Process Historian など)に保存できます。保存されたプロセス値およびメッセージは、通常、WinCC オンライントレンドコントロールまたは WinCC アラームコントロールのランタイムのプロセス画面に表示されます。さらに、OLE-DB などの定義済みインターフェースを使用すれば、アーカイブされたプロセス値とメッセージに直接アクセスできます。このように、重要な生産データを、たとえば分析目的で全社全体で使えるようになります。

1.5 クライアント/サーバーシナリオ

サーバー間通信

2 台のサーバー間の通信中に、1 台のサーバーが別のサーバー上のデータにアクセスします。1 台のサーバーは最大で 18 台の他のサーバーまたは冗長サーバーペアのデータにアクセスできます。データにアクセスするサーバーは設定および操作においてクライアントとして動作しますが、標準サーバーは設定できません。

アクセスする各サーバーには WinCC サーバーライセンスが必要です。システムの設定中は、アクセスしているサーバーを数量構造に含める必要があります。

標準サーバー

分散システムでは、固有のサーバーに基づいたサーバー接頭語がデータに付けられるため、WinCC コントロールがメッセージおよびプロセスデータを表示できるようになります。

分散システムでは、標準サーバーはクライアント用に設定され、一意の接頭語が指定されていないデータも標準サーバーから要求できます。標準サーバーが定義されていない場合は、対応するローカルデータにアクセスが試みられます。ローカルデータ(メッセージおよびアーカイブなど)が管理されていない場合、アクセスは拒否されエラーメッセージが表示されます。

優先サーバー

冗長サーバーをマルチユーザーシステムまたは分散システムで使用する場合、クライアントの中から優先サーバーを設定します。

優先サーバーは冗長サーバーペアのサーバーで、マルチユーザーシステムのクライアントに対して優先度が高くなっています。優先サーバーは各クライアントで個別に選択でき、システムの操作性を確保します。

下記も参照

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

ランタイムでのシステム動作 (ページ 70)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.6 サーバーコンフィグレーション

1.6 サーバーコンフィグレーション

1.6.1 サーバーコンフィグレーション

概要

WinCC ネットワーク内にあるサーバーにより、以下のタスクが実行できます

- プロセスへの接続
- プロセス値の取得
- メッセージおよびプロセス値のアーカイブ
- プロセスからのデータのクライアントへの供給
- コンフィグレーションデータのクライアントへの供給

個々のサーバーで遂行されるタスクは技術面あるいは、機能面により分散できます。

- 技術面：各サーバーでプロセスおよびシステム固有のセクションを管理します。
- 機能面：各サーバーでプロセス全体に関わるランタイム固有のタスク(アラームロギングまたはアーカイブなど)を実行します。

コンフィグレーションのステップ

マルチユーザシステムのコンフィグレーション

マルチユーザシステムの場合、複数のクライアントがランタイムにおいて 1 台のサーバーからの表示を行うように設定されます。クライアントは 1 台のサーバーより独占的にデータを受け取り、個々の設定はされません。

マルチユーザシステムのサーバーは以下のように設定します。

1. サーバー上に"マルチユーザプロジェクト"タイプのプロジェクトを新規作成します。
2. 必要なプロジェクトデータ(ピクチャ、アーカイブ、タグなど)をサーバー上で設定します。
3. サーバー上で設定予定、あるいは、コンピュータリストにリモートでモニタされるクライアントを含みます。
4. リモート設定を有効化するように、操作権限をクライアントに割り振ります。
5. サーバーで自動パッケージインポートを実行します。
6. サーバープロジェクト(ピクチャの開始、ロックキーコンビネーションなど)上のクライアントプロパティを設定します。

分散システムのコンフィグレーション

分散システムでは、クライアントが複数のサーバーの表示を行うように設定します。クライアントはローカルデータを保存する自己プロジェクトを備えています。サーバーにより更新されたデータはパッケージエクスポート機能を経由してクライアントへ転送されます。

分散システムのサーバーは以下のように設定します。

1. 各サーバー上に"マルチユーザプロジェクト"タイプのプロジェクトを新規作成します。
2. 必要なプロジェクトデータ(ピクチャ、アーカイブ、タグなど)を各サーバー上で設定します。分散(技術面および機能面)に応じて、アーカイブのみといった固有プロジェクトデータに関連づけることもできます。
3. リモートコンフィグレーション可能なクライアントは、サーバーのコンピュータリストに登録する必要があります。
4. リモート設定を有効化するように、操作権限をクライアントに割り振ります。
5. パッケージエクスポートを(手動または自動で)設定します。
6. クライアント側でのクライアントプロジェクトのコンフィグレーション。
7. サーバーデータ(パッケージ)をクライアントで使用可能にします。

注記

常に、関連するクライアントを設定する前に、クライアント/サーバーシステムのサーバーを設定します。

下記も参照

優先サーバーのコンフィグレーション方法 (ページ 55)

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

パッケージエクスポートのコンフィグレーション方法 (ページ 36)

オペレータオーソリゼーションのコンフィグレーション方法 (ページ 33)

コンピュータリストへのクライアント登録方法 (ページ 32)

サーバー側でのプロジェクトの新規作成 (ページ 30)

クライアントコンフィグレーション (ページ 44)

クライアント/サーバーシナリオ (ページ 24)

WinCC のクライアント/サーバーシステム (ページ 16)

1.6.2 サーバー側でのプロジェクトの新規作成

概要

WinCC でプロジェクトを新規作成する場合は、以下のプロジェクトタイプから選択します。

- シングルユーザープロジェクト:
スタンドアローンオペレーティングステーション用のプロジェクト。すべてのタスクをランタイムで実行します(プロセスドライバ接続、オペレーティング、モニタ、アーカイブなど)。
クライアント/サーバーシステムには関係ありません。
- マルチユーザープロジェクト:
マルチユーザーシステムまたは分散システム用のサーバープロジェクト。複数のクライアントおよびサーバーを設定します。
- クライアントプロジェクト:
分散システム内の 1 台のクライアント用のプロジェクト。複数のサーバーに表示できます。

注記

プロジェクトタイプの変更

既存のプロジェクトは、プロジェクトタイプを修正([コンピュータプロパティ]ダイアログの[一般]タブ)することによりサーバープロジェクトに変換することもできます。

ポートの設定

オペレーティングシステムは、サーバーとクライアント間の通信に 1024～65535 の範囲でポートを動的に選択します。

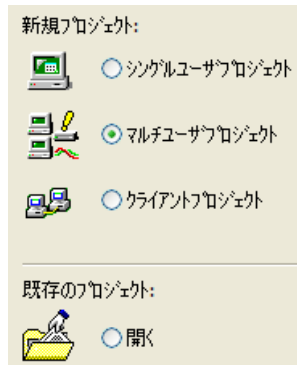
Simatic Shell の通信設定で、特定のポートを指定することもできます。

[暗号化された通信]を有効にして、希望するポートを入力します。

手順

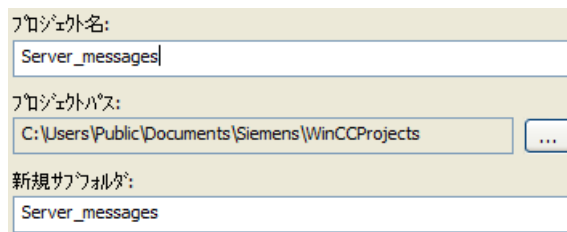
以下の手順はマルチユーザシステムまたは分散システムでのサーバープロジェクトの作成方法を説明しています。

1. WinCC エクスプローラでサーバーを開き、メニュー項目の[ファイル]>[新規]を選択します。
[WinCC エクスプローラ]ダイアログが開きます。



2. [マルチユーザプロジェクト]を選択し、[OK]をクリックします。
[プロジェクト新規作成]ダイアログが現れます。
3. プロジェクト名およびサブディレクトリ名(ディレクトリ名がプロジェクトと異なる場合)を入力します。
デフォルトでは、次のフォルダがプロジェクトパスとして使用されます。

– 「パブリック文書\Siemens\WinCCProjects」



4. [作成]ボタンをクリックします。
プロジェクトが作成され、WinCC エクスプローラで開きます。
現在のプロジェクトは自動的にサーバープロジェクトになります。

下記も参照

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

パッケージエクスポートのコンフィグレーション方法 (ページ 36)

オペレータオーソリゼーションのコンフィグレーション方法 (ページ 33)

コンピュータリストへのクライアント登録方法 (ページ 32)

サーバーコンフィグレーション (ページ 28)

クライアントコンフィグレーション (ページ 44)

1.6 サーバーコンフィグレーション

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.6.3 コンピュータリストへのクライアント登録方法

はじめに

クライアントがサーバーにリモートまたはランタイムでアクセスする場合、そのクライアントはサーバーのコンピュータリストに登録される必要があります。

前提条件

マルチユーザプロジェクト(マルチワークステーションシステムあるいは分散システム)を作成しました。これにより設定コンピュータは自動的にクライアント/サーバーシステムになります。

手順

1. WinCC エクスプローラで[コンピュータ]を選択し、ポップアップメニュー項目[新規コンピュータ...]を選択します。
[コンピュータプロパティ]ダイアログが表示されます。
2. クライアントコンピュータの名前を入力して、現在のサーバーへアクセスできるようにします。
3. [OK]をクリックし、プロジェクトのコンピュータリストにコンピュータを登録します。
4. クライアントサーバーシステム上の現在のサーバーへアクセスできるよう、すべてのコンピュータに対してこのプロセスを繰り返し実行します。

注記

クライアントコンピュータの名前を変更するには、コンピュータリストから選択したクライアントコンピュータを削除します。新しいクライアントコンピュータを、変更した名前でも新規コンピュータとしてコンピュータリストに含めます。

下記も参照

サーバーコンフィグレーション (ページ 28)

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

パッケージエクスポートのコンフィグレーション方法 (ページ 36)

オペレータオーソリゼーションのコンフィグレーション方法 (ページ 33)

サーバー側でのプロジェクトの新規作成 (ページ 30)

クライアントコンフィグレーション (ページ 44)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.6.4 オペレータオーソリゼーションのコンフィグレーション方法

WinCC におけるオペレータオーソリゼーション

クライアントがリモートあるいはランタイムでサーバープロジェクトを開いて処理をするには、サーバープロジェクト内で適切なクライアントオペレータオーソリゼーションを設定する必要があります。

この目的で、次のオペレータオーソリゼーションがサーバー上で使用できます。

- [リモート設定]:
クライアントはリモートステーションよりサーバープロジェクトを開き、プロジェクトにフルアクセスができます。
- "リモート有効化":
クライアントは実行時にサーバープロジェクトを配置することができます。
- "Web Access - モニタリングのみ":
Web クライアントはプラントの監視を許可されています。
したがって、オペレータオーソリゼーションは他のクライアントの設定には関係ありません。

クライアント側の設定

クライアントがサーバープロジェクトを設定するオーソリゼーションを有する場合、サーバープロジェクトのクライアントからオペレータオーソリゼーションを変更することも可能です。

オペレータ権限変更の際、ネットワーク内のコンピュータには通知されません。

新規クライアントがサーバーにログインすると、変更は有効になります。

1.6 サーバーコンフィグレーション

動作

クライアントが対応するサーバーのプロジェクトを開き、実行または終了するとオペレータ権限が要求されます。

対応するオペレータ権限がサーバーで利用できない場合、プロジェクトは処理できません。

サーバープロジェクトをクライアント側で閉じる場合、プロジェクトを再度開くとき、ログインが再度要求されます。

注記

オペレータ権限はユーザーに関連付けられています

設定されたオペレータオーソリゼーションはユーザーに対するものであり、コンピュータに対するものではありません。

つまり、割り付けられたオペレータオーソリゼーションは、ログインが同じすべてのオペレーティングステーションで有効になります。

オペレーティングシステムのオペレータオーソリゼーション

クライアントがサーバープロジェクトにアクセスするには、対応するプロジェクトフォルダでサーバー側のネットワークアクセスを有効にする必要があります。

1. [プロジェクトプロパティ]ダイアログで以下のオプションを無効にします。
[プロジェクトディレクトリが書き込み保護アクセス用にのみ共有されていること。]
2. プロジェクトにアクセスするユーザーに必要なすべての権限について、オペレーティングシステムのオーソリゼーションをセットアップします。

注記

Windows オペレータ権限

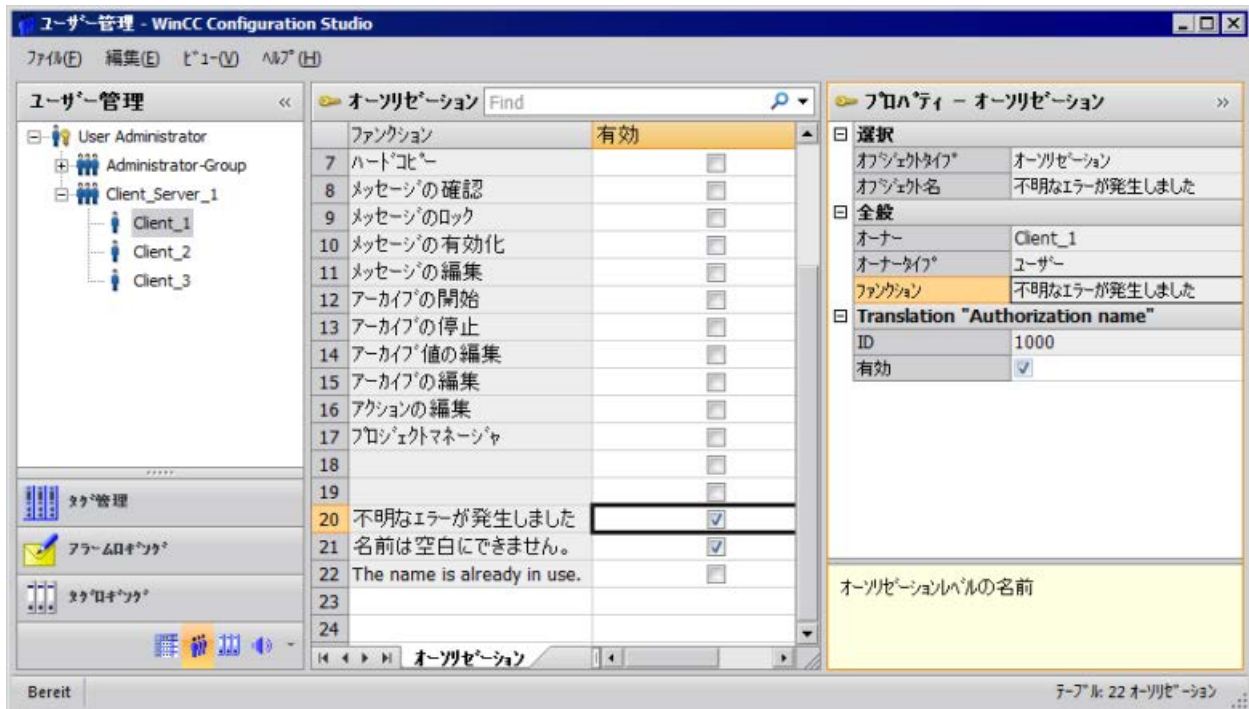
ネットワークセキュリティに関して、有効化したプロジェクトディレクトリに別のWindows オペレータオーソリゼーションを割り付けることができます。

オペレータオーソリゼーションの割り付けについての詳細な情報は、Windows 文書にて提供されます。

手順

1. WinCC エクスプローラのユーザー管理者を開きます。
2. ナビゲーシオンエリアでユーザーを選択します。

3. ユーザーにサーバープロジェクトへのフルアクセス権を付与するには、次の権限を有効にします。
- [リモート有効化]
 - [リモート設定]



4. [ユーザー管理者]を閉じます。

下記も参照

- サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)
- パッケージエクスポートのコンフィグレーション方法 (ページ 36)
- コンピュータリストへのクライアント登録方法 (ページ 32)
- サーバー側でのプロジェクトの新規作成 (ページ 30)
- サーバーコンフィグレーション (ページ 28)
- クライアントコンフィグレーション (ページ 44)
- クライアント/サーバーシナリオ (ページ 24)
- 数量構造および特性 (ページ 21)
- WinCC のクライアント/サーバーシステム (ページ 16)

1.6.5 パッケージエクスポートのコンフィグレーション方法

原理

パッケージとは現在のコンフィグレーションデータ(タグ、メッセージ、アーカイブなど)すべてを含むパケットデータで、分散システムあるいはマルチユーザーシステムに接続されたすべてのクライアントで利用できます。パッケージはサーバーからエクスポートされ、クライアントにインポートされます。

サーバーでエクスポートしクライアントでインポートする場合、はじめの1回は手動で行われます。その後のサーバー側およびクライアント側でのパッケージの更新は、すべて自動的に実行されます。パッケージを更新する場合、パラメータおよびその更新のトリガを調整できます。たとえば、手動コミッショニング中にパッケージをクライアントへ転送し、初回コンフィグレーションデータを配布します。クライアントのデータを最新に保つため、サーバーデータの各修正時に自動的にパッケージ更新するように、コンフィグレーションできます。

注記

すでに作成されたパッケージを含むプロジェクトを他のサーバーへコピーする場合、WinCC コンピュータプロパティのコンピュータ名をコピーしたプロジェクトの新規コンピュータに合わせます。コピーしたプロジェクトのパッケージを再度生成する場合は、[パッケージプロパティ]ダイアログボックスのコンピュータ名を更新する必要があります。

サーバー間の通信を選択した場合、1台のサーバーがもう1台のサーバーのデータにアクセスします。この時、アクセスするサーバーはインポートしたパッケージに関してクライアントのように動作します。そのため以下の記述では、クライアントに関する詳細が適用されます。

コンフィグレーションデータは通常作動中に手動あるいは自動で更新できます。

手動によるパッケージ作成

要求に応じて、新規パッケージがサーバーに手動で作成されます。クライアントはこれらをインポートできます。

パッケージの自動更新

Implicit Update ファンクションを用いて、サーバーでのパッケージのエクスポートおよびクライアントでのパッケージのインポートを自動化できます。

サーバー側の[コンフィグレーションパッケージ自動更新]ダイアログに表示されているオプションは、このサーバーからのパッケージのエクスポートおよび他のサーバーからのパッケージのインポート両方に影響を及ぼす可能性があります。このことは、以下のテーブルの[インポート]および[エクスポート]の列を見ればわかります。

ダイアログでは、WinCC に関して以下の可能性があります。

WinCC CS の設定	インポート	エクスポート	意味
プロジェクトが開かれるとサーバーデータを更新	可		プロジェクトが 開かれる と、クライアントは必ずインポートを実行します。
通知時の自動更新	可		以下の条件が満足されている場合、通知を受信しだい、クライアントは必ずインポートを実行します。 <ul style="list-style-type: none"> • [エクスポート後に通知]サーバー設定が有効になっている場合 • プロジェクトが実行されていない場合
モニタがコンフィグレーションデータに変わります。 <ul style="list-style-type: none"> • プロジェクトが開かれた場合はサーバーデータを生成します。 • プロジェクトが閉じられた場合はサーバーデータを生成します。 • 変更された場合は直ちにサーバーデータを生成します。 		可	サーバーがパッケージをエクスポートします。 <ul style="list-style-type: none"> • プロジェクトが開かれた場合 • プロジェクトが閉じられた場合 • プロジェクトデータに変更がある度
エクスポート後に通知します。		可	パッケージのエクスポート時、サーバーから通知が送信されます。 この設定を実行するには、[通知時の自動更新]クライアント設定が有効になっている必要があります。
自動インポート		可	サーバーは、以下の目的で、エクスポートしたその専用パッケージを再度インポートします。 <ul style="list-style-type: none"> • シンボルコンピュータ名が指定された特殊サーバーに関係なく、タグなどを設定する。 • その専用プロジェクトを使用しないクライアントの、特殊サーバーだけを表示する。

1.6 サーバーコンフィグレーション

WinCC RT の設定	インポート	エクスポート	意味
プロジェクトが開かれるとサーバーデータを更新	可		プロジェクトが 実行 されると、クライアントは必ずインポートを実行します。
通知時の自動更新	可		以下の条件が満足されている場合、通知を受信しだい、クライアントは必ずインポートを実行します。 <ul style="list-style-type: none"> • [通知時の自動更新]クライアント設定が有効になっている場合。 • プロジェクトが実行されている場合。

注記

たとえばコミショニング中やコンフィグレーションツールの使用中などに、プロジェクトデータを頻繁に変更しなければならない場合は、自動パッケージエクスポートは使用できません。

エクスポートパッケージを設定する場合は、WinCC エクスプローラのサーバーデータエディタを使用します。

必要条件

サーバープロジェクト開いておく必要があります。

手順

手動パッケージエクスポート

1. WinCC エクスプローラで[サーバーデータ]に進み、ショートカットメニューで[作成]を選択します。
2. [パッケージプロパティ]ダイアログボックスにおいて、記号および物理的サーバー名を指定します。この情報でクライアント側のパッケージの場所を識別します。
コンフィグレーション中できるだけ早期に、サーバーの物理的コンピュータ名およびコンピュータ名記号を定義します。コンピュータ名記号が変更された場合は、すべてのコンフィグレーションデータに適応させる必要があります。
コンピュータ名記号は通常、プロジェクト名と物理的コンピュータ名の組み合わせで構成されます。
3. [OK]をクリックします。サーバーデータが作成されます。設定のサイズによって異なりますが、これには時間がかかる場合があります。

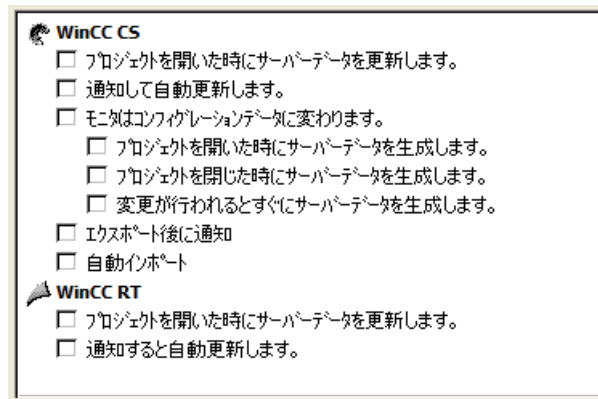
結果

サーバーデータの入ったパッケージは、WinCC エクスプローラ内[サーバーデータ]下のリストに置かれています。パッケージは、ファイルシステムのプロジェクトディレクトリに、<プロジェクト名>\<コンピュータ\パッケージ>*.pck の形式で保存されます。

これでクライアントはパッケージをインポートできます。

自動パッケージエクスポート

1. WinCC エクスプローラで[サーバーデータ]に進み、ショートカットメニューで[自動更新]を選択します。



2. 必要なオプションを選択します。複数選択も可能です。
3. [OK]をクリックし、選択を確定します。

結果

選択すると同時に自己のサーバーからのサーバーデータを含むパッケージが生成されます。また、他のサーバーからすでにインポートされたパッケージの場合は、たとえばプロジェクトを閉じた時に更新されます。選択すると同時に、自己のサーバーからのサーバーデータを含むパッケージが生成されます。また、他のサーバーからすでにインポートされたパッケージの場合は、たとえばプロジェクトを閉じた時に更新されます。

注記

SIMATIC Manager で作成した WinCC プロジェクトでは、[サーバーデータ]ショートカットメニューに、オプション[作成...]および[自動更新...]は含まれていません。これは WinCC で作成し、ファンクション[WinCC オブジェクトのインポート]を使用して SIMATIC Manager にインポートした、WinCC プロジェクトにも適用されます。このタイプのプロジェクトは TIA プロジェクトとも呼ばれます。


TIA プロジェクトを WinCC エクスプローラでコピーし、その後 WinCC エクスプローラでそのコピーを編集した場合、[サーバーデータ]ショートカットメニューにメニューアイテム[作成...]と[自動更新...]が含まれます。


生成されたパッケージの表示

パッケージが生成されている場合、WinCC エクスプローラデータウィンドウに以下のように表示されます。


キーボード、右：ロードされたパッケージ

キーボード、左：サーバーからエクスポートされたパッケージ

:ロードされたパッケージ、標準サーバーなし

:ロードされたパッケージ、標準サーバーあり

:サーバーエクスポートパッケージ(再インポートされない)

:自己プロジェクトへ再インポートされたローカルに作成されたパッケージ。

下記も参照

クライアントコンフィグレーション (ページ 44)

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

オペレータオーソリゼーションのコンフィグレーション方法 (ページ 33)

コンピュータリストへのクライアント登録方法 (ページ 32)

サーバー側でのプロジェクトの新規作成 (ページ 30)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.6.6 サーバープロジェクトでのクライアントのコンフィグレーション

原理

複数のクライアントがまさに 1 つのサーバーのビューを表示するマルチユーザシステムを設定する場合は、クライアント用に個別のプロジェクトを作成しないでください。

サーバープロジェクトで独自のプロジェクトを使用せずにクライアントの動作を設定します。

サーバークライアント設定

独自のプロジェクトを持たないクライアントは、クライアントが設定されているサーバーのビューしか持っていません。

このサーバーとサーバー間通信を介した別のサーバーまたは中央アーカイブサーバーへの接続は許可されません。

独自のプロジェクトを持たないクライアントの内部タグ

独自のプロジェクトを持たないクライアントの内部タグには、次のような特徴が適用されます。

- 「コンピュータローカル」設定が適用されます。
この設定では、プロジェクト全体またはコンピュータローカルベースでタグの変更を更新するかどうかを指定します。
内部タグは、常に WinCC サーバー上でプロジェクト全体で更新されます。独自のプロジェクトを持つクライアントでは、内部タグは常にコンピュータローカルベースで更新されます。
- 「コンピュータローカル」設定が有効な場合、「ランタイム永続化」設定は無効です。

WinCC エクスプローラがグレー表示されているときのプロジェクトの変更の適用

WinCC エクスプローラがグレー表示されている場合、クライアントのプロジェクト変更は適用されません。

開始条件

- 独自のプロジェクトを持たないクライアントのランタイムで、WinCC エディタが開いている。
- サーバーで WinCC Runtime が動作していない。

動作

エディタの変更(例えばプロジェクト関数のスクリプト変更)は適用されません。

ソリューション

独自のプロジェクトを持たないクライアント上で、WinCC エクスプローラがグレー表示されている場合には設定しないでください。

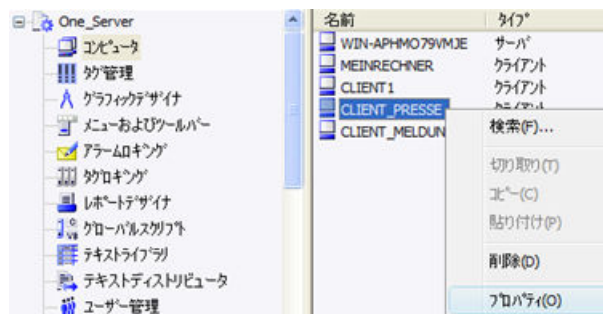
1.6 サーバーコンフィグレーション

必要条件

- サーバーデータを表示するクライアントは、サーバー側のコンピュータリストに登録されている必要があります。クライアントは、1 台の特定サーバーにだけアクセスしようとします。
- このサーバーは、他のサーバーからパッケージをインポートできません。
- サーバープロジェクトはサーバーで開きます。

手順

1. サーバにある WinCC エクスプローラのコンピュータリストを開きます。
2. 設定するクライアントを選択し、ポップアップメニューから[プロパティ]アイテムを選択します。



クライアントの[コンピュータプロパティ]ダイアログが開きます。

3. スクリプトを使用する場合、[スタートアップ]タブで、[グローバルスクリプトランタイム]などの、クライアントのランタイムで有効であることが必要なエディタを、選択します。
4. クライアントで起動するランタイムの言語の選択には、[パラメータ]タブを使用します。例えば、同じデータを異なる言語で表示する 2 台のクライアントを、設定することもできます。
5. [グラフィックランタイム]タブで、クライアントの開始画像を指定します。開始画像は、各クライアントに個別に選択できます。必要に応じて、ここでウィンドウ属性を定義します。
6. [OK]を選択して、設定を確定します。
7. 同じ方法で、プロジェクトの他のクライアントのプロパティを設定します。
8. サーバープロジェクトで、サーバーデータエディタのポップアップメニューで[暗黙の更新]エントリを開きます。[自動インポート]設定を有効にします。
9. サーバープロジェクトでパッケージを生成します。

下記も参照

パッケージエクスポートのコンフィグレーション方法 (ページ 36)

オペレータオーソリゼーションのコンフィグレーション方法 (ページ 33)

コンピュータリストへのクライアント登録方法 (ページ 32)

サーバー側でのプロジェクトの新規作成 (ページ 30)

サーバーコンフィグレーション (ページ 28)

クライアントコンフィグレーション (ページ 44)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7 クライアントコンフィグレーション

1.7.1 クライアントコンフィグレーション

はじめに

クライアントコンフィグレーションは、クライアントが複数のサーバーに表示できるように、分散システムをコンフィグレーションする場合にのみ必要です。クライアントが1台のサーバーからのデータのみを表示するように、マルチユーザシステムをコンフィグレーションする場合は、クライアントコンフィグレーションは必要ありません。クライアントは、サーバープロジェクトから全データおよび対応するランタイム環境を受信します。

クライアント/サーバーシステムに複数のサーバーが含まれ、クライアントが複数のサーバー(分散システム)に異なる表示をするようにコンフィグレーションする場合、各クライアントに個々のクライアントプロジェクトをコンフィグレーションします。ランタイムでは、各クライアントは18台までの異なるサーバーまたは冗長サーバーペアに表示できます。たとえば、サーバー1およびサーバー2からのメッセージを表示し、サーバー3からのプロセス値を表示して書き込み、サーバー4からの画像を表示する、などがあります。

分散システムのクライアントは、サーバー側での個々のオペレーティングオーソリゼーションに従って、以下を実施できます。

- プロセスのモニタ
- プロセスのモニタおよび作動
- サーバー側でのプロジェクトのリモートコンフィグレーション
- サーバー側でのプロジェクトのリモート実行および終了

注記

異なるサーバーからのデータを表示するため、サーバーの接頭語(つまり、サーバー名)は分散システム内で一意でなければなりません。

各クライアントには固有のコンフィグレーションがあり、クライアントのデータベースに、管理をあまり必要としない以下のようなクライアント固有のデータを、ローカルに保存します。

- ローカルタグ
- ユーザー管理者データ

- テキストライブラリからのデータ
- プロジェクトのプロパティ
- ユーザーサイクル

注記

サーバーコンフィグレーションのすべての外部データもクライアント側で使用でき、クライアントプロジェクトに正しく表示できなければなりません。外部データは、たとえば、WinCC に由来しない ActiveX コントロール、および OLE オブジェクトとして統合化されている外部グラフィックに、関連しています。

コンフィグレーションのステップ

1. サーバープロジェクトのコンフィグレーション
2. サーバーパッケージのエクスポートおよび作成
3. クライアント側でのパッケージインポートのコンフィグレーション
4. クライアント側でのクライアントプロジェクトのコンフィグレーション。

注記

サーバーでランタイムを無効にする場合は、クライアントでもランタイムを終了し、コンフィグレーションを続ける必要があります。

下記も参照

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

クライアントのピクチャの開始のコンフィグレーション (ページ 58)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

1.7 クライアントコンフィグレーション

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.2 クライアント側でのプロジェクトの新規作成

概要

WinCC でクライアントプロジェクトを新規作成する場合は、以下のプロジェクトタイプから選択します。

- シングルユーザープロジェクト:
スタンドアロンオペレーティングステーション用のプロジェクト。すべてのタスクをランタイムで実行します(プロセスドライバ接続、オペレーティング、モニタ、アーカイブなど)。
クライアント/サーバーシステムには関係ありません。
- マルチユーザープロジェクト:
マルチユーザーシステムまたは分散システム用のサーバープロジェクト。複数のクライアントおよびサーバーを設定します。
- クライアントプロジェクト:
分散システム内の 1 台のクライアント用のプロジェクト。複数のサーバーに表示できます。

注記

独自のプロジェクトを持たないクライアント

複数のクライアントが1台のみのサーバーの画面を表示するように、マルチユーザシステムが設定されている場合、クライアント用のローカルプロジェクトを作成せず、クライアントの動作をサーバープロジェクトで設定します。

プロジェクトタイプの変更

既存のプロジェクトは、[コンピュータプロパティ]ダイアログでプロジェクトタイプを修正することにより、クライアントプロジェクトに変換することもできます。

ポートの設定

オペレーティングシステムは、サーバーとクライアント間の通信に 1024～65535 の範囲でポートを動的に選択します。

Simatic Shell の通信設定で、特定のポートを指定することもできます。

[暗号化された通信]を有効にして、希望するポートを入力します。

手順

1. WinCC エクスプローラでクライアントを開き、メニュー項目の[ファイル]>[新規]を選択します。
[WinCC エクスプローラ]ダイアログが開きます。



2. [クライアントプロジェクト]を選択し、[OK]をクリックします。
[プロジェクト新規作成]ダイアログが現れます。

1.7 クライアントコンフィグレーション

3. プロジェクト名およびサブディレクトリ名(ディレクトリ名がプロジェクトと異なる場合)を入力します。
デフォルトでは、次のフォルダがプロジェクトパスとして使用されます。

– 「パブリック文書\Siemens\WinCCProjects」

プロジェクト名:	<input type="text" value="Client_2"/>
プロジェクトパス:	<input type="text" value="C:\Users\Public\Documents\Siemens\WinCCProjects"/> ...
新規サブフォルダ:	<input type="text" value="Client_2"/>

4. [作成]ボタンをクリックします。
プロジェクトが作成され、WinCC エクスプローラで開きます。

下記も参照

クライアント/サーバーシナリオ (ページ 24)

サーバープロジェクトでのクライアントのコンフィグレーション (ページ 40)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

クライアントのピクチャの開始のコンフィグレーション (ページ 58)

インポートパッケージのコンフィグレーション (ページ 49)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.3 インポートパッケージのコンフィグレーション

はじめに

分散システムの種々のサーバーのプロセスデータを表示するクライアントには、対応するデータの情報が必要です。この目的のため、設定データを含むパッケージが分散システムのサーバーで作成され、このパッケージがクライアントに提供されます。クライアントでは、使用するデータのあるサーバーからのパッケージが必要です。

概要

サーバーでパッケージをエクスポートし、クライアントでインポートする場合、はじめの 1 回は手動で行われます。その後のサーバー側およびクライアント側でのパッケージの更新は、すべて自動的に実行されます。更新をいつ行うか、および何によってトリガするかを設定できます。

注記

サーバー間の通信が選択されると、1 台のサーバーが別のサーバーのデータにアクセスします。この時、アクセスするサーバーはインポートされたパッケージに関してクライアントのように動作します。そのため以下の記述では、クライアントに関する詳細が適用されます。

シンボルコンピュータが指定された特殊サーバーに関係なく、タグなどを設定するために、サーバーはその専用パッケージを再度インポートできます。

パッケージをインポートするには、WinCC エクスプローラで[サーバーデータ]エディタを使用します。パッケージをインポートする方法には、以下の 3 種類があります。

手動ロード

サーバー上で生成されたパッケージがクライアント上にロードされます。インポートプロセスは、[ロード]コマンドを使用して手動でトリガされます。パッケージの最初のインポートは手動で実行する必要があります。

手動更新

サーバーによりクライアント上にすでにロードされているパッケージは、[更新]コマンドを使用して更新されます。

1.7 クライアントコンフィグレーション

自動更新

クライアント側でパッケージの自動更新を設定できるので、固有の条件に合致した場合、新規パッケージはクライアント側で自動的に更新されます。ただし、はじめのインポートは手動で実行しなくてはなりません。

設定	意味
WinCC CS 用 <ul style="list-style-type: none"> プロジェクトを開いてのサーバーデータの更新 通知時の自動更新 	<ul style="list-style-type: none"> プロジェクトが開かれると、クライアントは必ずインポートを実行します。 パッケージのエクスポート後、サーバーから通知が送信されます。その通知を受信するとクライアントは必ずインポートを実行します。 この設定は、サーバーでパッケージエクスポート用に[エクスポート後に通知]設定が実行されている場合にかぎり有効となります。
WinCC RT 用 <ul style="list-style-type: none"> プロジェクトを開いてのサーバーデータの更新 通知時の自動更新 	<ul style="list-style-type: none"> プロジェクトが実行されると、クライアントは必ずインポートを実行します。 パッケージのエクスポート後、サーバーから通知が送信されます。その通知を受信するとクライアントは必ずインポートを実行します。 この設定は、サーバーでパッケージエクスポート用に[エクスポート後に通知]設定が実行されている場合にかぎり有効となります。

必要条件

- サーバーでパッケージが作成されている。
- クライアントプロジェクトが開いている。

手順

手動ロード

- クライアントでクライアントプロジェクトを開きます。
- WinCC エクスプローラで[サーバーデータ]を選択し、ショートカットメニューで[ロード]を選択します。[ファイルを開く]ダイアログが現れます。

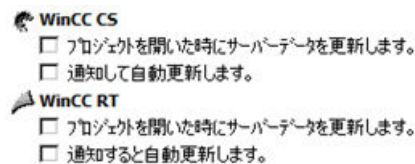
3. ロードするパッケージを選択し、[OK]をクリックします。
パッケージはデフォルトで、ディレクトリ「...\<サーバープロジェクト名>\<コンピュータ名>\<パッケージ>」に、名前「<プロジェクト名_コンピュータ名>*.pck」で保存されています。
ただし、どのようなデータ媒体に保存されているパッケージにもアクセスできます。
4. [開く]をクリックします。データがロードされます。対応するサーバーを使用できない場合、新規パッケージの要求時に、適切な障害エントリが表示されます。

手動更新

1. クライアントでクライアントプロジェクトを開きます。
2. WinCC エクスプローラで[サーバーデータ]を選択し、ショートカットメニューで[更新]コマンドを選択します。
3. データが更新されます。サーバー間通信の場合、他のサーバーからのパッケージがロードされないと、そのサーバー上に障害メッセージが表示されます。

自動更新

1. クライアントでクライアントプロジェクトを開きます。
2. WinCC エクスプローラで[サーバーデータ]を選択し、ショートカットメニューで[自動更新]を選択します。[設定自動パッケージ更新]ダイアログが表示されます。



3. 必要なオプションを選択します。複数選択も可能です。
4. [OK]をクリックして選択を確定します。サーバーデータはクライアント側で自動的に更新されます。たとえば、プロジェクトを開く、ネットワーク経由の通知に従う、などです。対応するサーバーを使用できなくても、クライアントに障害メッセージは表示されません。

注記

クライアント側でプロジェクト実行中に新規パッケージを追加する、またはパッケージを削除すると、表示が困難になる恐れがあります。この状況に対処するには、クライアントをいったん終了し、その後再起動します。

ロードされたパッケージの表示

パッケージがロードされている場合、以下のように WinCC エクスプローラのデータウィンドウに表示されます。

キーボード、右:ロードされたパッケージ

キーボード、左:エクスポートされたが、パッケージが未ロード



:ロードされたパッケージ、標準サーバーなし



:ロードされたパッケージ、標準サーバーあり

1.7 クライアントコンフィグレーション

下記も参照

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

クライアントのピクチャの開始のコンフィグレーション (ページ 58)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.4 標準サーバーの設定方法

はじめに

分散システムのクライアント用に標準サーバーを設定します。(標準サーバーは、一意のサーバー接頭語(例、タグ用)が指定されていない場合に、データを要求します。)

標準サーバーが構成要素用に設定されていない場合、ローカルクライアントデータ(例、内部タグ)にアクセスが試みられます。クライアントにローカルデータ管理(例、メッセージおよびアーカイブ)がない場合、アクセスは拒否されエラーメッセージが発行されます。

必要条件

標準サーバーは、対応するパッケージをインポートした後で、クライアントでのみ選択できます。

手順

1. WinCC エクスプローラで、クライアントの[サーバーデータ]エントリを選択します。
2. ポップアップメニューから[標準サーバー...]アイテムを選択します。
[標準サーバーの設定]ダイアログが表示されます。
3. シンボルコンピュータ名で、必要な構成要素のエントリをクリックします。ドロップダウンリストボックスからサーバーを選択します。リストには、クライアントにロードされたすべてのパッケージのシンボルコンピュータ名が含まれています。



4. ダイアログにリストされている構成要素は、WinCC インストールに依存します。オプションがインストールされている場合、表示されている構成要素に加えて、構成要素オプション (例、SSM - 分割画面マネージャ) をリストできます。
5. [OK] をクリックし、選択を確定します。

構成要素用の標準サーバーの選択タイミング

アラーム

オペレーティングメッセージをクライアントで生成する場合、アラーム用標準サーバーをあらかじめ指定しておかなくてはなりません。クライアント自体ではアラームロギングを設定できないので、メッセージはサーバーで発行する必要があります。

クライアントは、デフォルトサーバーから集中的にユーザーが指定したメッセージの選択を取得できます。

アーカイブ、画面、テキストライブラリ、ユーザーアーカイブ、タグ

標準サーバーがクライアントで設定されている場合、有効なサーバー接頭語が生成されていない構成要素からのデータは、定義された標準サーバーで検索されます。標準サーバーがクライアントで設定されていない場合、サーバー接頭語がないので、このデータのサーバーを特定できません。

アーカイブ、画面、テキストライブラリ、ユーザーアーカイブ、タグ用にデフォルトサーバーを設定することは、特殊なアプリケーションでのみ意味があります。SIMATIC のマニュアルやカスタマサポートから特定のサーバーを設定するように要求されない場合は、[標

1.7 クライアントコンフィグレーション

標準サーバーの設定]の"サーバーデータ"エディタの[標準サーバーなし]の設定を変更しないでください。

注記

WinCC クライアントの変数に標準サーバーが入力されている場合、タグ管理のランタイムでのヒントには、ステータス情報は表示されません。

基本プロセスコントロールを使用する場合は、標準サーバーを選択します。

アラーム

常にアラーム用の標準サーバーを指定する必要があります。

タグ

変数用に標準サーバーを指定することはできません。

SSM(分割画面マネージャ)

常に SSM 構成要素用の標準サーバーを指定します。

WinCC クライアントにトレンドグループが組み合わされてる場合、トレンドグループは標準サーバーおよびそのリダンダントパートナーサーバーに保存されます。他の WinCC クライアントも、このサーバーを SSM 構成要素用の標準サーバーとして、指定できます。そのため、設定されたトレンドグループも、これらの WinCC クライアントで使用できます。WinCC クライアントで SSM 構成要素用に設定された標準サーバーがない場合、コンパイルされたトレンドは、このコンピュータにローカルに保存されます。他の WinCC クライアントは、このようなトレンドグループを、WinCC オンライントレンドコントロールで表示できません。基本的に、これらのトレンドグループをサーバープロジェクトで表示することは、不可能です。

WinCC クライアントで画面構成が設定されている場合、標準サーバーが SSM 構成要素用に指定されている場合に限り、このサーバーに保存されます。標準サーバーが指定されていない場合は、WinCC クライアントの画面構成の設定はローカルに保存され、他のクライアントからはアクセスできません。基本的に、これらの画面構成をサーバープロジェクトに表示することは、不可能です。

サーバーに冗長化が設定されている場合、トレンドグループのデータと画面構成のデータも、冗長パートナーサーバーと同期しています。冗長化切り替え時には、コンパイルされたすべてのトレンドグループおよび画面構成を、WinCC クライアントから要求できます。

下記も参照

インポートパッケージのコンフィグレーション (ページ 49)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

クライアントのピクチャの開始のコンフィグレーション (ページ 58)

優先サーバーのコンフィグレーション方法 (ページ 55)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.5 優先サーバーのコンフィグレーション方法

はじめに

リダンダントサーバーを使用する場合、分散システムあるいはマルチユーザシステムのクライアント上の優先サーバーを設定します。

優先サーバーはリダンダントサーバーペアのサーバーで、分散システムのクライアントに対して優先度が高くなっています。優先サーバーが使用可能であるかぎり、クライアントはそのサーバーからデータを受信します。

優先サーバーは各クライアントに対して個々に定義できるので、クライアントをリダンダントサーバーに分散することが可能で、永続的な操作性が確保されます。構成されたサーバーへのネットワーク中断がある場合、クライアントはリダンダントパートナーサーバーに切り替わります。サーバーが再度使用可能になると、クライアントはその優先サーバーに再度切り替わります。

1.7 クライアントコンフィグレーション

クライアントをリダンダントサーバーに分散することにより負荷が分散され、システム全体のパフォーマンスが向上します。

注記

WinCC のリダンダントシステムのコンフィグレーションは、「リダンダントシステム」トピックを参照してください。

手順

分散システムとマルチユーザシステムのクライアント用優先サーバーは、別々に設定されます。

分散システムのクライアント用優先サーバーの構成

1. WinCC エクスプローラで、クライアント側の[サーバーデータ]エントリを選択します。
2. ショートカットメニューから[構成]を選択します。
[サーバーデータの構成]ダイアログが表示されます。
3. リストには、クライアントにパッケージを提供する全サーバーの、コンピュータ名記号および物理的コンピュータ名が含まれます。サーバーに対してリダンダントサーバーが利用可能である場合、物理的コンピュータ名を指定します。リダンダントサーバーペアから 1 台のサーバーを、優先サーバーとして選択します。
分散システムのリダンダントサーバーのペアには、唯一で共通のシンボル名があり、これによってサーバーをアドレス指定します。



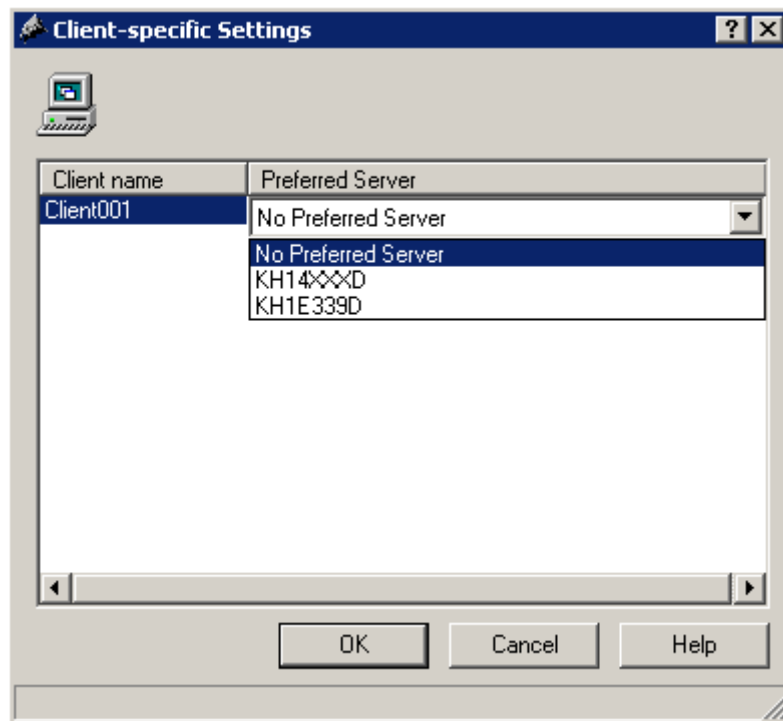
4. [OK]をクリックして、入力を終了します。

マルチユーザシステムのクライアント用優先サーバーの構成

クライアントはサーバーのコンピュータリストに入力される必要があります。

1. WinCC エクスプローラでサーバー側の[サーバーデータ]エントリを選択します。
2. ショートカットメニューから[クライアント固有の設定]を選択します。
[クライアント固有の設定]ダイアログが表示されます。

3. サーバーのコンピュータリストに、入力されたクライアントのリストが表示されます。必要なクライアントを選択し、[優先サーバー]列の2台のリダンダントサーバーから1台を、優先サーバーとして選択します。



4. [OK]をクリックして、入力を終了します。

クライアントのランタイム動作

優先サーバーとして指定されたリダンダントサーバーが使用可能であるかぎり、クライアントはこれに接続されたままです。

優先サーバーが故障した場合、クライアントはリダンダントパートナーサーバーに切り替わります。故障した優先サーバーが再度使用可能になると、クライアントはそのサーバーに戻ります。

下記も参照

クライアントのピクチャの開始のコンフィグレーション (ページ 58)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

1.7 クライアントコンフィグレーション

異なるサーバーからのピクチャの表示 (ページ 60)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.6 クライアントのピクチャの開始のコンフィグレーション

概要

分散システムのピクチャはすべてクライアントのピクチャの開始として使用できます。これはサーバーからのピクチャ、クライアント側のローカルピクチャなどのピクチャすべてにあてはまります。

以下の記述は、サーバーのピクチャをピクチャの開始(初期画面)として使用方法について、説明しています。

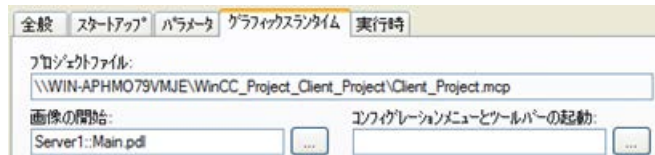
前提条件

ピクチャの開始として使用するピクチャのサーバーパッケージは、クライアントにインポートします。

手順

1. クライアントでクライアントプロジェクトを開きます。
2. WinCC エクスプローラでコンピュータを選択し、ポップアップメニューから[フ ° ロハ ° ティ]オプションを選択します。
3. グラフィックランタイムタブを実行します。

4. 初期画面名としてサーバーコンピュータ名を入力し、使用するピクチャを入力します。例：



5. [検索]ボタンを使用して、ピクチャを検索することもできます。選択ダイアログに、クライアントにロードされたすべてのサーバーパッケージのピクチャが表示されます。
6. [OK]をクリックして、入力を終了します。

下記も参照

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

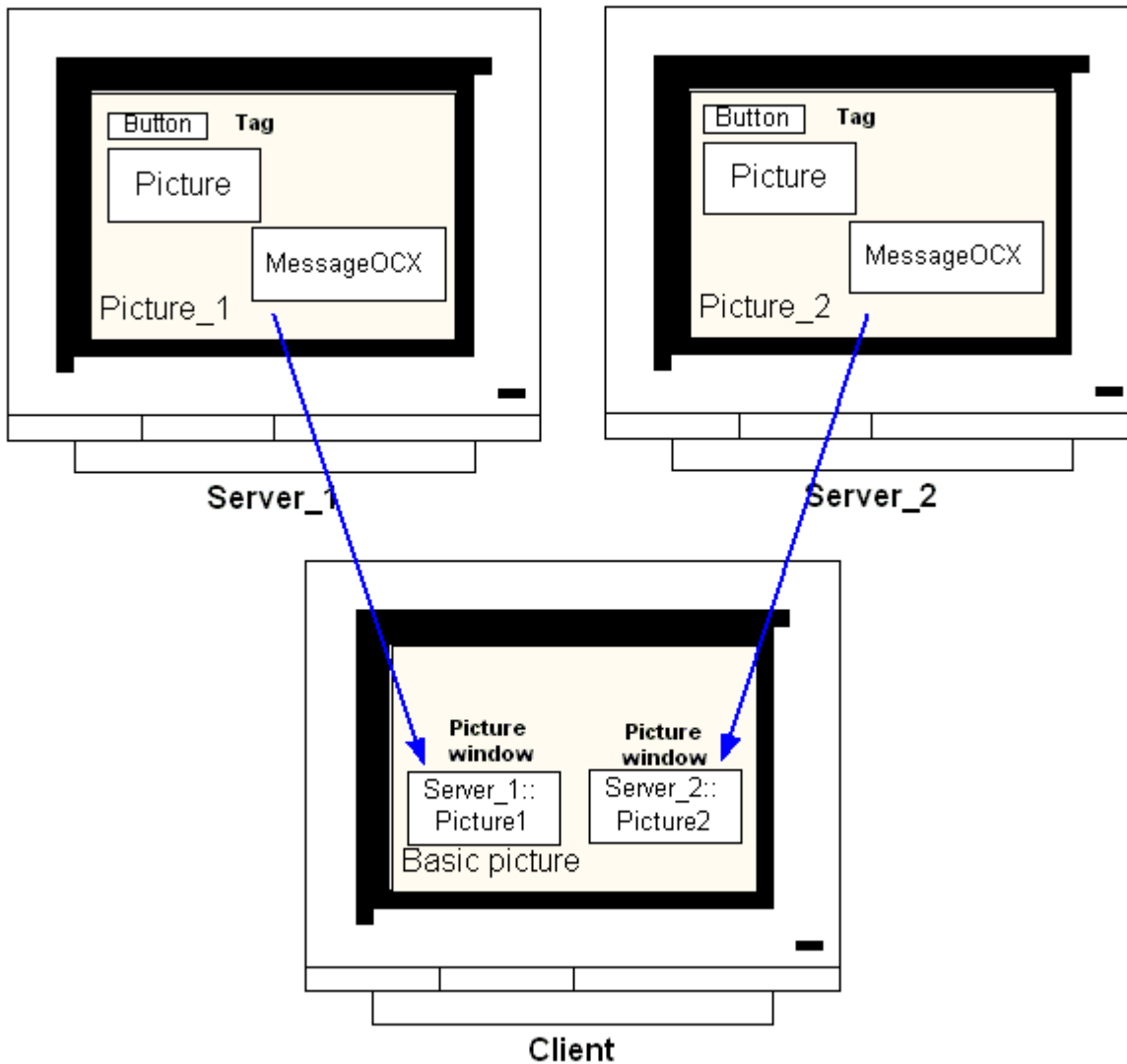
数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.7.7 異なるサーバーからのピクチャの表示

原理

異なるサーバーからのピクチャを、クライアント上にコンフィグレーションされた基本画面内のピクチャウィンドウに表示できます。



サーバーからのデータは各ピクチャウィンドウからアクセスできます。サーバーピクチャをピクチャウィンドウとしてクライアントピクチャに統合するため、サーバー接頭語をピクチャファイル名の前に付ける必要があります。

注記

サーバー接頭語は分散システム内で一意でなければなりません。

サーバーピクチャは、スクリプト(C または VBS)および直接リンク経由でピクチャウィンドウに挿入できます。

サーバーのピクチャはクライアントのピクチャウィンドウサイズに合わせる必要があります。

前提条件

対応するサーバーのパッケージをクライアントにインポートする必要があります。

手順

1. ピクチャウィンドウに挿入するクライアントでピクチャを開きます。
2. グラフィックデザイナーの標準パレットから、スマートオブジェクトグループの[ピクチャウィンドウ]を選択し、ピクチャに挿入します。
3. ピクチャウィンドウ上でダブルクリックし、[プロパティ]ダイアログを開きます。
4. [その他]グループから、[プロパティ]タブをダブルクリックし、[ピクチャ名]属性を選択して、ピクチャを検索します。

または、

[ピクチャ名]属性の[スタティック]列をダブルクリックし、ピクチャ名を直接<サーバー接頭語>::<ピクチャ名>という形式で入力します。

5. [プロパティ]ダイアログを閉じます。

注記

[ピクチャ名]属性でサーバー接頭語が自動的に指定されない場合、[サーバー接頭語]属性経由でもサーバー接頭語を入力できます。[サーバー接頭語]属性でダブルクリックすると、パッケージがクライアント側にある全サーバーを含んだ選択リストが現れます。

1.7 クライアントコンフィグレーション

下記も参照

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.8 クライアント側でのピクチャ変更のコンフィグレーション

概要

分散システムのクライアントを使用して、ターゲットピクチャにサーバー接頭語を付与することにより、サーバーピクチャ上のピクチャ変更をコンフィグレーションできます。WinCC によるコンフィグレーションでは、通常のピクチャ交換のコンフィグレーションでも、基本ピクチャの変更でも、違いはありません。

手順

以下の手順は、サーバー上でピクチャ変更を開始する、ボタンのコンフィグレーション方法例について説明しています。

1. グラフィックデザイナーのクライアントプロジェクトのピクチャを開きます。
2. Windows オブジェクトのグループからボタンをピクチャに挿入します。
[コンフィグレーション]ダイアログが現れます。
3. サーバー接頭語をターゲットとして、[ピクチャの変更]に入力し、ピクチャ名を<サーバー接頭語>::<ピクチャ名>の形式で入力します。例：



4. [OK]をクリックしてダイアログを閉じます。

その他の手順

ピクチャの変更は、ボタンの[プロパティ]ダイアログでもコンフィグレーションできます。

- [イベント]タブを使用して、マウスのクリックなどで直接接続をコンフィグレーションします。
- サーバー接頭語の付いたピクチャ名を、直接接続の定数として入力します。

下記も参照

サーバーコンフィグレーション (ページ 28)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

異なるサーバーからのデータの使用 (ページ 64)

1.7 クライアントコンフィグレーション

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

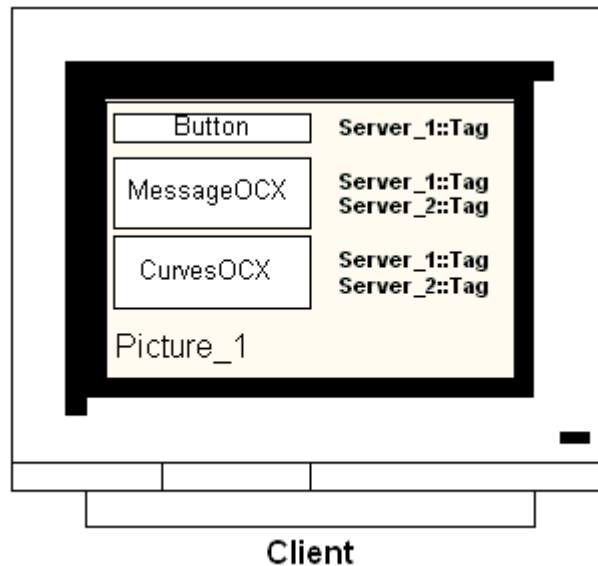
WinCC のクライアント/サーバーシステム (ページ 16)

1.7.9 異なるサーバーからのデータの使用

原理

分散システムのクライアントの基本ピクチャおよびそこに含まれるすべてのオブジェクトは、クライアント側で直接コンフィグレーションされます。複数のサーバーからのデータへのアクセスは、各基本ピクチャから実行できます。例：

- システムの A 部をモニタするサーバー_1 からのプロセス値出力フィールド、システムの残りの部分をモニタするサーバー_2 からのプロセス値出力フィールド。
- 異なるシステムパット/サーバーからのデータを、比較できる形式で表示するトレンド表示。
- 複数のサーバーからのメッセージを表示するメッセージウィンドウ。



クライアント上に出力される基本ピクチャは、他のクライアントにコピーできます。このためには、基本ピクチャでアドレスできるサーバーパッケージをターゲットクライアント側で使用可能にする必要があります。

注記

サーバー側でコンフィグレーションされ、パッケージとともにクライアントに転送されたタグはすべて、クライアント側では[タグの選択]ダイアログにより使用できます。

グローバルスクリプトからのCアクションとファンクション、またはVBSアクションと手順は、クライアント側で実行できなければなりません。グローバルCおよびVBSスクリプトは、パッケージのコンポーネントではありません。

手順

以下の手順は2台の異なるサーバーからのプロセスデータを、クライアント上のトレンド表示に表示させる方法例を説明したものです。

1. クライアントでクライアントプロジェクトを開きます。
2. グラフィックデザイナーを使用して、基本ピクチャとして使用するピクチャをコンフィグレーションします。
3. オブジェクトパレットのコントロールタブから、WinCC オンライントレンドコントロールを基本ピクチャに挿入します。[WinCC オンライントレンドコントロールのプロパティ]ダイアログが開きます。
4. 現在のプロセスをモニタする場合、データソースとして[オンラインタグ]を選択します。
5. [トレンド]タブを実行します。
6. [アーカイブ/タグの選択]を選択すると表示されるプロセス値を含んでいるはじめてのトレンド用のタグを選択し、[選択]ボタンを選択します。

1.7 クライアントコンフィグレーション

7. 以下の形式でタグ名を入力します。 [<サーバー接頭語 1>::<タグ名>]。内容を[OK]ボタンで確認します。
8. トレンドタブで[+]ボタンをクリックし、2 番目のトレンドを追加します。
9. 2 番目のトレンドを、2 番目のサーバーからのタグに以下の形式で関連付けます。 [<サーバー接頭語 2>::<タグ名>]。
- 10.[OK]を選択して、設定を確定します。

結果

ランタイムでは、2 個のトレンドがクライアント上のトレンド表示に現れます。トレンド 1 はサーバー 1 からのデータを表示し、トレンド 2 はサーバー 2 からデータを表示します。

下記も参照

インポートパッケージのコンフィグレーション (ページ 49)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.10 異なるサーバーからのメッセージの表示

一般的な手順

複数のサーバーからのメッセージは、分散システムのクライアント上に以下のように表示できます。

- メッセージが表示される各サーバーのメッセージ表示を設定します。
- メッセージ表示のソースとして、複数のメッセージサーバーを指定します。

注記

アラームコントロールがクライアントの基本画像に統合されている場合、アラームでのループファンクション実行時には、関連サーバーピクチャがクライアント上で基本画像として表示されます。元の基本画像へ戻ることはできません。

アラームコントロールがクライアントの画像ウィンドウに統合されている場合、[アラームでのループファンクション実行時には、関連サーバー画像が[アラームでのループ]画像ウィンドウに表示されます。関連ボタンをクリックして、基本クライアント画像に戻ります。

手順

1. クライアントでクライアントプロジェクトを開きます。
2. グラフィックデザイナを使用して、基本画像として使用する画像を設定します。
3. [オブジェクトパレット]、[アラームコントロール]タブから、WinCC オンライントレンドコントロールを基本画像に挿入します。[WinCC アラームコントロールのプロパティ]ダイアログが開きます。
4. アラームコントロールに全接続サーバーのメッセージを表示する場合は、[サーバーの選択]を選択して[すべてのサーバー]チェックボックスにチェックを入れます。
5. 固有のサーバーからのメッセージのみを表示する場合は、[すべてのサーバー]チェックボックスのチェックを外し、[選択]ボタンをクリックしてネットワークから WinCC サーバーを選択します。
6. [OK]をクリックしてダイアログを閉じます。

注記

マルチロケーションシステムでは、必ず、クライアント 1 名の選択ダイアログに表示される内容が、すべてのサーバーで同じ名前になるようにします。

1.7 クライアントコンフィグレーション

下記も参照

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアント側でのプロジェクトの新規作成 (ページ 46)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

1.7.11 複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション

原理

異なるサーバーからのメッセージをクライアント上の基本ピクチャに表示する場合、対応するメッセージシーケンスレポートを表示することもできます。全サーバーからのメッセージが収集され、正しいシーケンスで出力されます。

WinCC は、メッセージシーケンスレポート用の事前コンフィグレーションレイアウトおよび印刷ジョブを提供します。

手順

1. WinCC エクスプローラのレポートデザイナーで、[@CCAlgRtSequence.RPI] レイアウトを開きます。ラインレイアウトエディタが現れます。
2. [選択] ボタンをクリックします。[プロトコルテーブル選択] ダイアログが現れます。

3. [サーバーの追加]ボタンを使用して、メッセージシーケンスレポートにメッセージをロギングするサーバーを[選択されたサーバー]リストに挿入します。パッケージがクライアントにインポートされているサーバーのみが表示されます。
4. 矢印ボタンを使用して、ロギングするメッセージブロックを[レポートの列シーケンス]リストに転送します。
5. [OK]をクリックして入力を確認します。
6. WinCC エクスプローラで、印刷ジョブ[@レポートアラームロギング RT メッセージシーケンス]を開きます。
7. 個々のレイアウトには名前が付けられ、保存されています。[レイアウト]リストからレイアウトを選択します。[ラインプリンタ用ラインレイアウト]チェックボックスにチェックを入れます。
8. [プリンタセットアップ]タブの[プリンタ]チェックボックスにチェックを入れます。
9. 接続プリンタのリストから、レポートを印刷させるプリンタを選択します。
10. [OK]を選択して、設定を確認します。
11. WinCC エクスプローラでクライアントコンピュータを選択し、ポップアップメニューから[プロパティ]コマンドを選択します。[コンピュータプロパティ]ダイアログが現れます。
12. [スタートアップ]タブで[メッセージシーケンスレポート]を実行します。
13. [OK]を選択して、設定を確認します。

下記も参照

クライアント側でのプロジェクトの新規作成 (ページ 46)

複数のサーバーからのメッセージ用メッセージシーケンスレポートのコンフィグレーション (ページ 68)

異なるサーバーからのメッセージの表示 (ページ 67)

クライアント側でのピクチャ変更のコンフィグレーション (ページ 62)

異なるサーバーからのピクチャの表示 (ページ 60)

優先サーバーのコンフィグレーション方法 (ページ 55)

標準サーバーの設定方法 (ページ 52)

インポートパッケージのコンフィグレーション (ページ 49)

クライアントコンフィグレーション (ページ 44)

サーバーコンフィグレーション (ページ 28)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.8 ランタイムでのシステム動作

1.8.1 ランタイムでのシステム動作

はじめに

WinCC のクライアント/サーバーシステムを使用して、システム設定を複数のサーバーに配信し、個別のサーバーにかかる負荷を減らすことができます。サーバー上で設定したデータはクライアント側で表示させることができ、ランタイム時に最大 18 台の異なるサーバーまたは冗長サーバーペアからのデータをクライアント側で表示できます。

ランタイムでのエディタ動作

アーカイブ

アーカイブ[®] システムがオペレーティングステーションで実行されると、タグロギングランタイムはサーバー側ではアーカイブ[®] サーバーとして機能し、クライアント側ではアーカイブ[®] クライアントとして機能します。データベースへのアクセス、プロセスデータのコンパイル、ファイル格納は、アーカイブ[®] サーバーだけしかできません。クライアントは、アーカイブ[®] サーバーからのアーカイブ[®] データを受け取ります。

アーカイブ[®] データは、クライアント側でタグロギングランタイムが実行中であれば、どのクライアントからもテーブルまたはグラフィックとして表示できます。表示されるデータは常にアーカイブサーバーからのものです。クライアント側の操作はすべてサーバーに転送され、処理した結果がクライアントに戻されます。

グラフィック

ランタイム中にクライアントによって画像が呼び出されたとき、グラフィックランタイムはまずローカルに保存された画像を検索します。対応する名前の画像がローカルで見つからない場合は、そのサーバーのプロジェクトフォルダ内を検索します。使用可能な画像がない場合、対応するメッセージが表示されます。

画像の要求により別のエディタ(アラームロギング、グローバルスクリプト)との交換が必要となる場合、交換はローカルに行われます。画像は、ランタイムで複数のオペレーティングステーションで開き、処理できます。

注記

画像をクライアント側で作成する場合は、ローカルにクライアントにコピーする方が速くできます。ランタイムタブのクライアント側の[コンピュータフックロハティ]タブのアイコンで関係するディレクトリを、入力します。そのディレクトリを、「常に使用する」または「できれば使用する」のいずれかに指定することもできます。サーバープロジェクトで画像が修正されている場合は、修正済みの画像を再度クライアントのローカルディレクトリにコピーして、データを手動で更新する必要があります。

メッセージ

クライアント側にメッセージが表示された場合、クライアントはサーバーから表示されているデータを受け取ります。メッセージサーバーは、データベースから設定されたデータを受け取ります。

アーカイブデータおよびメッセージリストは、どのクライアント上にも表示できます。表示されるデータは、常にメッセージサーバーからのものです。新規作成メッセージを受信した場合、そのメッセージはメッセージサーバーにアーカイブされます。

オペレータステーションがメッセージを確認するとき、確認がメッセージサーバーに転送されます。サーバーはアーカイブのステータスの変更を入力し、関係するすべてのクライアントにその旨を伝送します。同じプロセスがメッセージのロックに適用されます。

ただし、メッセージサーバーがランタイムで使用できない場合は、メッセージウィンドウにはそのかわりに使用不可のメッセージが表示されます。サーバーが再び使用可能になれば、メッセージウィンドウに再度メッセージが表示されます。

レポート

真の意味では、WinCCのプロトコルシステムはランタイムを検出しません。プロトコルおよび印刷ジョブは、いつでも設定および実行できます。アーカイブまたはプロセスデータを表示する必要がある印刷ジョブだけが、ランタイムに応じて変わります。

プロトコルシステムは、どのクライアントでもスタートアップルーチン時に自動的に開始されます。サーバーはプロトコルサーバーとして動作し、クライアントはプロトコルクライアントとして動作します。スタートアップルーチン時にクライアントはサーバーにログインし、実行可能な印刷ジョブおよびそのステータスに関する現在の情報を受け取ります。印刷ジョブがクライアント側で開始されると、そのジョブはサーバーデータベースから関連するデータを取得します。印刷ジョブはローカルに開始されます。プロトコルサーバーは、クライアントから印刷ジョブのステータスに関する現在のデータを受信し、その他のクライアントにその情報を転送します。

1.8 ランタイムでのシステム動作

スクリプト

オペレーティングステーションがプロジェクトをローカルに実行した場合、サーバーのプロジェクトファンクションと標準ファンクションがローカルにロードされます。

ユーザー管理者

オペレーティング権限は、ユーザー管理者のランタイムコンポーネントにより確認されます。ユーザー管理者のランタイムコンポーネントは、すべてのコンピュータで WinCC の実行時に自動的に開始されます。ログインが変更された場合は、現在のオペレーティング権限リストがローカルデータベースからロードされます。

テキストライブラリ

サーバープロジェクトが実行される場合、テキストライブラリランタイムはサーバー側ではテキストサーバーとして、クライアント側ではテキストクライアントとして機能します。データは常に、サーバーのデータベースから読み込まれます。

システムエラー時の動作

サーバーが使用可能状態ではない場合、クライアントは再度スタートアップするまでサーバーに周期的にポーリングします。システムエラーでデータをサーバー上に表示できない場合、たとえば使用可能なグラフィックオブジェクトもすべて無効となります。

注記

実行中の WinCC に関してクライアントで問題が生じた場合、クライアントを再起動して、サーバーに影響を与えることなくサーバーに再接続できます。

下記も参照

サーバーのスタートアップ (ページ 73)

複数のネットワークカードを使ったサーバー通信の特殊機能 (ページ 75)

クライアントのシャットダウン (ページ 77)

サーバーのシャットダウン (ページ 77)

クライアントのスタートアップ (ページ 74)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

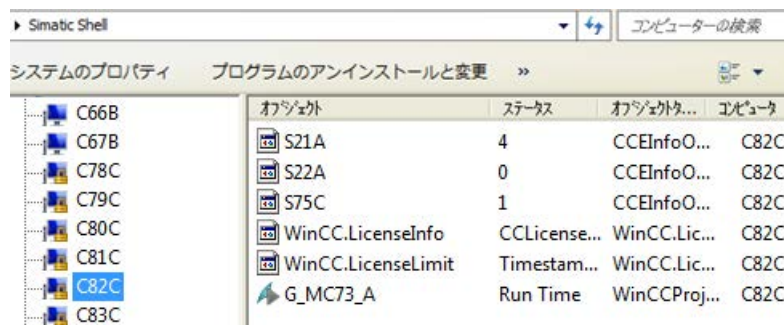
WinCC のクライアント/サーバーシステム (ページ 16)

1.8.2 サーバーのスタートアップ

原理

クライアント/サーバーシステムでは、サーバーはクライアントには関係なくスタートアップできます。サーバーがスタートアップすると、クライアントに対するサービスが使用可能となり、ネットワークに接続されているすべてのマシンに関する情報を取り出します。

[Simatic Shell]ダイアログのデータウィンドウにすべてのサーバーの現在の状態を表示できます。Windows エクスプローラから[Simatic Shell]を開きます。



通常のオペレーション時にサーバーにエラーが発生すると、クライアントのデータは更新されず、エラーが発生したサーバーに関する情報が転送されます。

注記

クライアント/サーバーシステムでファイルサーバーを使用する場合、ファイルサーバーとWinCC サーバーの両方がスタートアップされている場合にかぎり、システムは操作可能となります。

リモート実行

サーバーは、別のリモートコンピュータ(クライアントまたはサーバー)からスタートアップすることもできます。この手順は「プロジェクトの実行」を参照してください。

下記も参照

複数のネットワークカードを使ったサーバー通信の特殊機能 (ページ 75)

プロジェクトの実行方法 (ページ 93)

クライアントのシャットダウン (ページ 77)

サーバーのシャットダウン (ページ 77)

1.8 ランタイムでのシステム動作

クライアントのスタートアップ (ページ 74)

ランタイムでのシステム動作 (ページ 70)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.8.3 クライアントのスタートアップ

原理

クライアント/サーバーシステムでは、クライアントはサーバーに関係なくブートします。クライアント/サーバーシステムのクライアントがスタートアップすると、クライアントがネットワークに接続していると認識する WinCC サーバー経由で、以下に挙げた項目に関する現在の情報をすべて受信します。例：

- プロジェクト名
- サーバー名および IP アドレス
- サーバーのプロジェクトステータス(コンフィグレーションまたはルーチン)

ユーザーは、[Simatic Shell]ダイアログに対応する情報リストを見ることができます。サーバーの状態が変化すると、"Simatic shell"も更新されます。

サーバー使用不能

サーバーが使用不能になると、対応するエラーメッセージが出されます。さらに、そのサーバーからデータを受け取るグラフィックオブジェクトなどは無効になります。

スクリプトを使用して、クライアントに接続エラーを表示するようにコンフィグレーションできます。

下記も参照

複数のネットワークカードを使ったサーバー通信の特殊機能 (ページ 75)

プロジェクトの実行方法 (ページ 93)

クライアントのシャットダウン (ページ 77)

サーバーのシャットダウン (ページ 77)

サーバーのスタートアップ (ページ 73)

ランタイムでのシステム動作 (ページ 70)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.8.4 複数のネットワークカードを使ったサーバー通信の特殊機能

はじめに

複数のネットワークカードまたは SIMATIC NET SOFTNET ドライバを WinCC サーバーにインストールしてプロセスを接続し、サーバーの有効 TCP/IP プロトコルで通信する場合、サーバーと WinCC クライアントの通信が影響を受ける可能性があります。

考えられる原因は、サーバーの各ネットワークカードまたは SOFTNET ドライバが、固有の IP アドレスを持っていることです。したがって、特定の状況ではネットワーク内のサーバーにログオンすると、Windows は間違った IP アドレス(たとえば SOFTNET ドライバの IP アドレス)を経由して接続しようとします。接続に失敗した場合、Windows はこの IP アドレスによる通信を不良と認識し、コンピュータの別の IP アドレスを経由した接続を試みようとはしません。

この場合、ネットワーク管理者が適切に修正しなければなりません。

ネットワークカードの順序の確認

コンピュータに複数のネットワークカードがインストールされている場合、端子接続用ネットワークカードが最初になっている必要があります。

Windows コントロールパネルの[ネットワーク接続]で、順序を確認します。

[詳細]メニューでメニューコマンド[詳細設定]を選択します。順序は、[接続]セクションにある[ネットワークカードと接続]タブの[詳細]ダイアログで使用できます。

診断

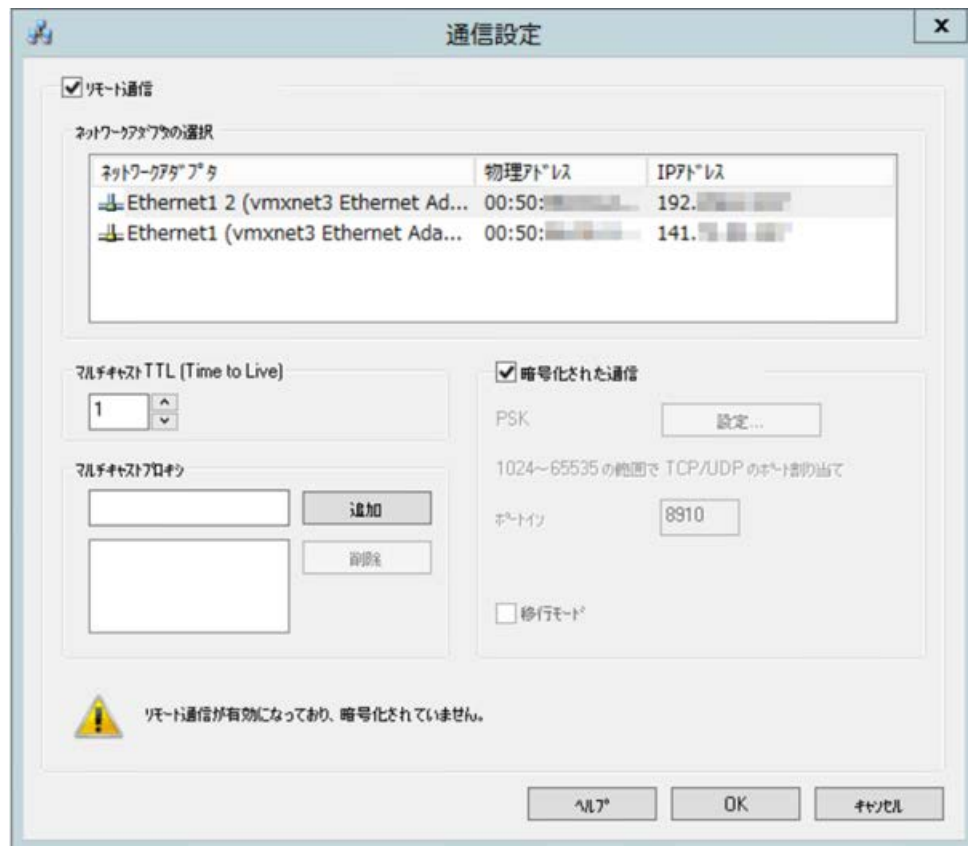
ディレクトリ「Simatic Shell」を使用して、ネットワークカードの設定をチェックできます。

コンピュータが間違った(アクセス不可能なネットワークエリアの)アドレスを示していることが判明した場合は、他のネットワークアダプタを選択します。

1.8 ランタイムでのシステム動作

手順

1. Windows エクスプローラのナビゲーションウィンドウで、[Simatic Shell]ディレクトリをクリックします。
2. ディレクトリのショートカットメニューから、[設定...]ダイアログを選択します。



3. ネットワークインターフェースを変更したい場合、"ネットワークアダプタ"エリアで希望するネットワークカードをクリックします。

さらに、プロセス接続に必要なウィンドウズのユーティリティが終了できることを、サーバーの SOFTNET ドライバのコンフィグレーションでチェックする必要があります。

上記チェックをおこなっても接続できない場合は、カスタマーサポートにお問合せください。

下記も参照

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

クライアントのスタートアップ (ページ 74)

サーバーのスタートアップ (ページ 73)

ランタイムでのシステム動作 (ページ 70)

WinCC のクライアント/サーバーシステム (ページ 16)

1.8.5 サーバーのシャットダウン

原理

クライアント/サーバーシステムのサーバーをシャットダウンした場合、接続しているクライアントにプロセスデータを提供できなくなります。システムから同時にログオフされ、[Simatic Shell]に無効としてマークされます。

リモート終了

サーバーは別のリモートコンピュータ(クライアントまたはサーバー)からも、シャットダウンできます。この手順については、「プロジェクトの終了」を参照してください。

下記も参照

クライアントのスタートアップ (ページ 74)

プロジェクトの終了方法 (ページ 95)

クライアントのシャットダウン (ページ 77)

サーバーのスタートアップ (ページ 73)

ランタイムでのシステム動作 (ページ 70)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.8.6 クライアントのシャットダウン

原理

クライアント/サーバーシステムのクライアントをシャットダウンするとシステムからログオフされます。

1.8 ランタイムでのシステム動作

下記も参照

プロジェクトの終了方法 (ページ 95)

サーバーのシャットダウン (ページ 77)

クライアントのスタートアップ (ページ 74)

サーバーのスタートアップ (ページ 73)

ランタイムでのシステム動作 (ページ 70)

クライアント/サーバーシナリオ (ページ 24)

数量構造および特性 (ページ 21)

WinCC のクライアント/サーバーシステム (ページ 16)

1.9 リモートコンフィグレーション

1.9.1 リモートコンフィグレーション

対応するオペレータ承認を得たクライアントは、サーバープロジェクトを遠隔操作することができます。例:

- サーバープロジェクトのリモート設定
- サーバープロジェクトの実行
- サーバープロジェクトの終了

リモートアクセスおよび RDP の詳細情報は、WinCC 情報システムのリリースノートの「WinCC の注意事項>リモートアクセスとリモートデスクトッププロトコル(RDP)」を参照してください。

Simatic Shell のファンクション

リモートアクセスの設定には、[Simatic Shell]ダイアログが使用できます。

[Simatic Shell]ダイアログに、ネットワークで利用できる WinCC プロジェクトを備えた有効なサーバーおよびコンピュータが表示されます。

これにはデモ版で実行できるプロジェクトもすべて含まれます。

[Simatic Shell]ダイアログ

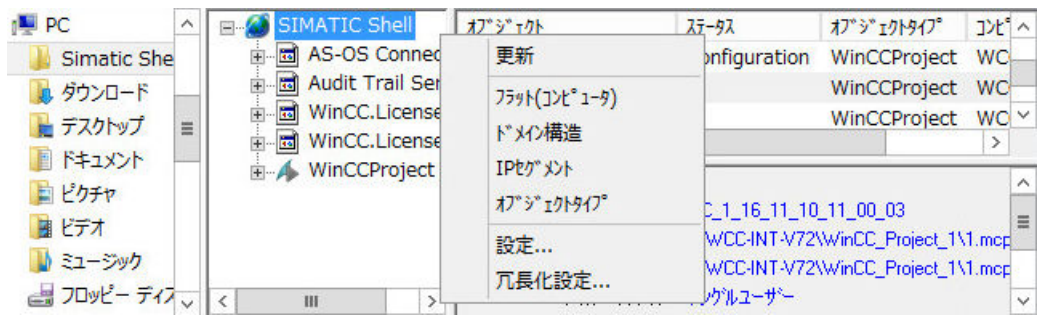
Windows エクスプローラから[Simatic Shell]ダイアログを開きます。

PC および WinCC プロジェクトは、以下の基準に従って構造化表示されます。

フラット(コンピュータ)	すべてのエントリは上から順番に表示されます。
ドメイン構造	サーバーとそれらの WinCC プロジェクトは、ドメインによりグループ化されます。
IP セグメント	サーバーとそれらの WinCC プロジェクトは、IP アドレスによりグループ化されます。
オブジェクトタイプ	表示はサーバータイプによりグループ化されます。

列見出しをダブルクリックしてエントリを並べ替えします。

1.9 リモートコンフィグレーション



複数のルーターに IGMP を設定

「インターネットグループ管理プロトコル(IGMP)」ネットワークプロトコルは、ターミナルバス上で使用されます。

複数のコンピュータを使用している場合、「クエリア」としてアクティブなルーターは 1 台だけです。次の設定に注意してください。

設定	設定
IGMP スヌーピング	「オン」
IGMP クエリア	「オン」を設定して 1 つのステーションのみを有効にする必要があります。 他のすべてのステーション設定を「オフ」にします。 複数のステーションがクエリアとして設定されている場合、スイッチの IP アドレスが最も小さいステーションだけがアクティブになります。
スヌーピングスイッチ IP	各ステーションには、別個の IGMP スイッチ IP アドレスを設定する必要があります。

下記も参照

- プロジェクトの終了方法 (ページ 95)
- プロジェクトの実行方法 (ページ 93)
- サーバープロジェクトピクチャの編集方法 (ページ 91)
- 編集用にプロジェクトを開く方法 (ページ 89)
- 複数のクライアントからのプロジェクトへのアクセス (ページ 87)
- サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 16)

暗号化通信 (ページ 81)

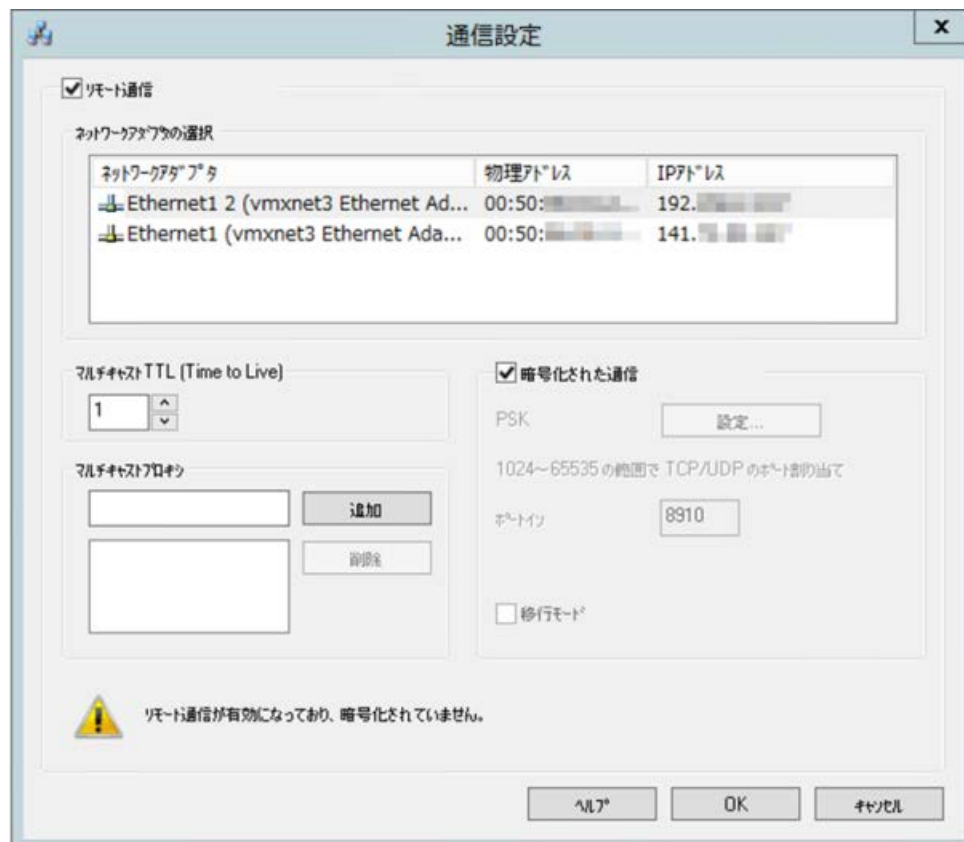
1.9.2 暗号化通信

コンピュータにアクセスするときは、常にコンピュータの暗号化通信が確立されていることを確認してください。

例えば移行など一時的な目的のためにのみ、非暗号化通信を使用します。

暗号化通信

Windows エクスプローラの[SIMATIC Shell]ショートカットメニューで通信設定を開きます。



暗号化された通信を使用する場合、接続は、同じ PSK のキーが指定されているコンピュータにのみ設定されます。これらのコンピュータとのみ通信することができます暗号化されていないコンピュータへの接続はできません。

1.9 リモートコンフィグレーション

同じネットワークのための独自の PSK キーを使用して異なる環境を指定することもできます。

暗号化された通信の設定に応じて、関連するコンピュータのみが **Simatic Shell** に表示されます。

設定に関する詳細情報は「サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)」を参照してください。

認証:自己署名証明書

自己署名証明書は WinCC ステーション間の通信ではサポートされていません。

サーバーでは自己署名証明書のみが検索可能である場合、通信に、設定済みの PSK キーが使用されています。

WinCC 用 Microsoft SQL Server

暗号化された通信についての詳細は、以下の Microsoft 記事を参照してください。

- データベースエンジンへの暗号化された接続の有効化 (<https://docs.microsoft.com/ja-jp/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-2017>)




移行モード

移行モードは、操作中にアップグレードにも利用できます。このモードでは、ネットワーク内で暗号化されている接続と暗号化されていない接続が並行して可能となります。

移行モードでは、ネットワーク内で接続が暗号化されたコンピュータと暗号化されていないコンピュータがすべて表示されます。

プラント全体の通信を暗号化する方法に関する一時的なソリューションとして、移行モードを使用してください。

コンピュータシンボル

 C83C	コンピュータは、暗号化された接続のみを許可します。
 C84C	コンピュータは暗号化された接続と暗号化されていない接続を許可します。 (移行モード)
 C85C	コンピュータは暗号化されていない接続を許可します。 (移行モード、または暗号化されていない接続による表示)

下記も参照

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

標準的な設定 (ページ 19)

データベースエンジンへの暗号化された接続の有効化 (<https://docs.microsoft.com/ja-jp/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-2017>)

1.9.3 サブネットに接続されていないコンピュータへのアクセス方法

例えば「Simatic Shell」を介して、ルーターの下流にあるネットワーク内のコンピュータをシステムに統合します。

[Simatic Shell]は WinCC の一部で、クライアント-サーバーシステムに統合されているすべてのコンピュータの集中保守および集中診断に使用されます。

原理

[Simatic Shell]の設定を使用して、サブネット内のコンピュータを、他のコンピュータからサブネット内のコンピュータに情報を伝える[エージェント]として利用できます。

暗号化通信を有効にすると、共有の指定されたキーが通信の前に知られているコンピュータのみが、互いに通信することができます。

ログオンした後、システム内にあるすべてのコンピュータはネットワークの範囲外にあるコンピュータとも通信できるようになります。既存のグループに追加される各コンピュータは、すべてのコンピュータの現在のステータスについて通知されます。

以下のようにコンピュータのステータスが変更された場合、システム内にある全コンピュータにメッセージが出されます。

- コンピュータがプロジェクトを実行した場合。
- コンピュータをシャットダウンした場合
- コンピュータをスタートアップしグループを入力した場合

ファイアウォールの設定

異なるネットワークからの WinCC コンピュータが相互に通信できるようにするには、ローカルの Windows ファイアウォールで次の設定を調整する必要があります。

すべての WinCC 固有のファイアウォールの規則では、他のネットワークからコンピュータの IP アドレス、または他のネットワークの完全な IP 範囲によって範囲を拡大する必要があります。

手順

1. Windows のコントロールパネルで[システムとセキュリティ]>[Windows ファイアウォール]カテゴリを開きます。
2. [詳細設定]をクリックします。
[セキュリティ強化機能搭載 Windows ファイアウォール]ダイアログが開きます。
3. [受信の規則]から、例えば CCAgent、OPC UA Discovery、WinCC ProjectManager など影響を受けるすべてのファイアウォールルールを 1 つずつ選択します。
各ルールには、[グループ]列に、「SIMATIC」で始まるグループ名があります。例えば、「SIMATIC Communication Services」、「SIMATIC WinCC OPC」。
4. ルールのショートカットメニューから[プロパティ]ダイアログを開きます。
5. [スコープ]タブで、通信パートナーの IP アドレスまたは IP スコープを[リモート IP アドレス]に追加します。

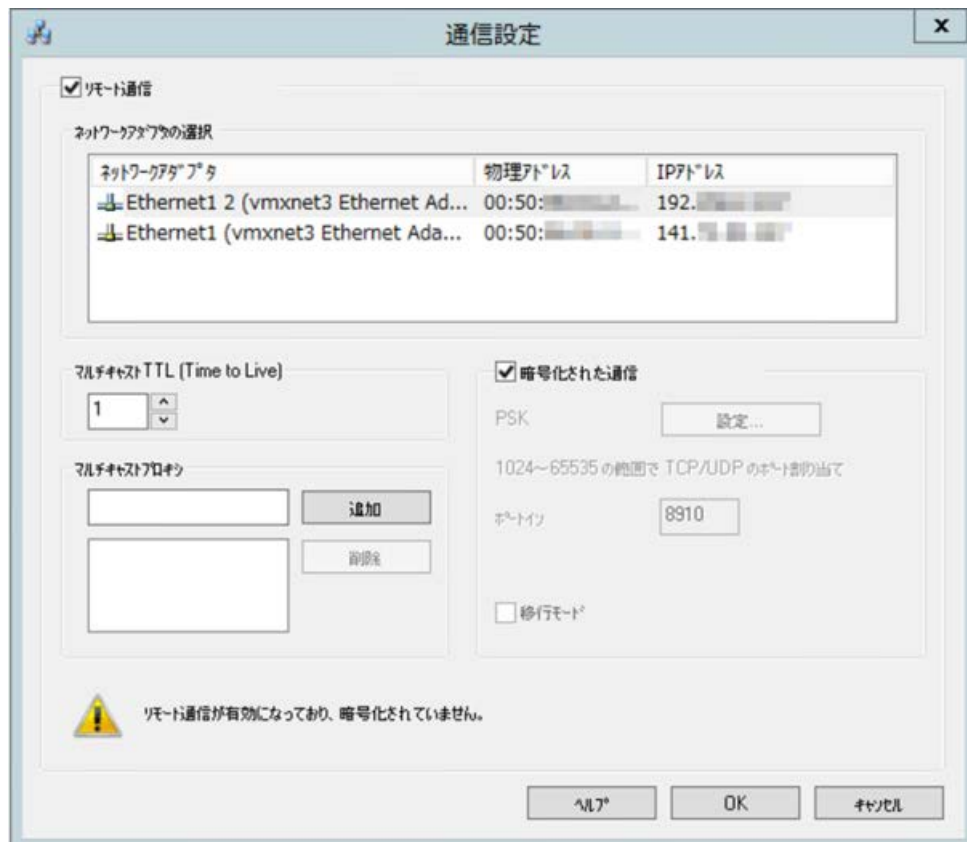
必要条件

- [リモート通信]オプションが有効になり、ネットワークアダプタが設定されます。

手順

1. クライアントとして WinCC コンピュータにアクセスするコンピュータで Windows エクスプローラを開きます。
2. エントリ [SIMATIC Shell]を選択します。
[Simatic Shell]ウィンドウが開きます。

3. [SIMATIC Shell]ショートカットメニューから[設定]コマンドを選択します。
[通信設定]ダイアログが開きます。



4. [マルチキャストライフタイム(TTL)]フィールド内の設定をチェックします。
値は、さまざまなサブネット(IP パラメータ TTL)間の遷移の最大数を指定します。
5. [マルチキャストプロキシ]入力フィールドに、サブネットの「エージェント」として指定されているコンピュータの IP アドレスを入力します。
サブネット内にあるどのコンピュータ(クライアントまたはサーバー)でも指定できます。
コンピュータをエージェントのリストに追加するには、[追加]をクリックします。

1.9 リモートコンフィグレーション

6. 暗号化通信を設定するには、「暗号化通信」オプションを選択します。
PSK キーを入力するには、[指定]ボタンをクリックします。



7. キーに対してキー強度の高い文字を入力してください。
キーは、最低 8 文字の長さで、小文字/大文字のほかには数字と記号を含める必要があります。
[OK]を選択して、設定を確定します。
8. デフォルト設定で割り付けられている使用可能なポートを使用しない場合は、受信ポートの割り付けを指定します。
9. 暗号化された接続と暗号化されていない接続を並行して許可するには、[移行モード]オプションを選択します。
このオプションは、操作中に更新する場合など一時的な目的にのみ使用してください。
- 10.[OK]を選択して、設定を確定します。

下記も参照

複数のクライアントからのプロジェクトへのアクセス (ページ 87)

プロジェクトの終了方法 (ページ 95)

プロジェクトの実行方法 (ページ 93)

サーバープロジェクトピクチャの編集方法 (ページ 91)

編集用にプロジェクトを開く方法 (ページ 89)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

暗号化通信 (ページ 81)

1.9.4 複数のクライアントからのプロジェクトへのアクセス

コンフィグレーションオプション

データタイプによっては、1台または複数のクライアントが、リモート操作中にサーバープロジェクトにアクセスできます。サーバーのデータベースに保存されたデータ(アラームロギング、タグロギング、タグ、ユーザ管理、テキストライブラリ)と、ファイルに基づいたデータ(ピクチャおよびグラフィック、レポート、スクリプト)は異なります。

注記

サーバーデータベースのデータは、複数のクライアントが同時に処理できます。ただし、複数のクライアントが同じデータにアクセスした場合、最後に修正したクライアントのデータが保存されることに注意します。サーバーデータベースのデータの場合、修正された値がほんの少ししかなくても、各エディタ内のデータがすべて保存されます。

ファイルに保存されたデータの場合、ファイルが開かれている間は他からアクセスできないようにデータがブロックされます。

アーカイブ (タグロギング)

アーカイブはサーバーデータベースに保存されています。タグロギングのデータはランタイムで修正できます。修正は、サーバーから接続されている全クライアントに伝送されます。

ピクチャ

ピクチャはサーバーまたはファイルサーバーにファイルとして保存されます。クライアントがサーバー側のピクチャにアクセスした場合、そのピクチャは別のクライアントからブロックされます。別々のプロジェクトのピクチャであれば別のクライアントから開くことができます。

ピクチャはランタイムで変更でき、保存すれば次にそのピクチャを選択したときに使用できます。ピクチャは編集用にローカルに保存することもできますが、サーバー側のピクチャと一致させる必要があります。

メッセージ

メッセージはサーバーデータベースに保存されています。アラームロギングシステムのデータは、ランタイムで修正できます。修正は、サーバーから接続されている全クライアントに伝送されます。

レポート

プロトコルはサーバーのプロジェクトフォルダに集中的に保存されます。プロトコルデータは、レイアウト(ファイル)と印刷ジョブ(プロジェクトデータベースへのエントリ)に、

1.9 リモートコンフィグレーション

分割されます。各サーバーに応じてプロトコルシステムを設定できるクライアントは1台だけです。

ピクチャは、編集用にローカルに保存することもできますが、各サーバー側のピクチャと手動で一致させる必要があります。プロトコルはランタイムに関わらず実行できるので、これはランタイム時にプロトコルシステムの修正を可能にすることを、目的としたものではありません。

スクリプト

スクリプトはサーバーのプロジェクトフォルダに集中的に保存されます。プロジェクト固有のスクリプトは、個々のローカルコンピュータで定義できます。スクリプトはファイルに保存されます。グラフィックデザイナアクションはピクチャに保存されます。ファイル(スクリプトまたはピクチャ)は編集、他のクライアントからブロックされます。サーバーに接続されていない場合、スクリプトはローカルに修正できますが、サーバー側のスクリプトと手動で一致させる必要があります。スクリプトはランタイムで修正できません。修正は、サーバーから接続されている全コンピュータに伝送されます。

注記

クライアントがサーバーにアクセスしないでスクリプトを設定した場合、スクリプトはローカルに保存されます。スクリプトをサーバーで使用可能にする場合は、スクリプトは適切なサーバー側のフォルダに手動でコピーする必要があります。

テキストライブラリのテキスト

テキストライブラリのテキストはサーバーデータベースに保存されます。テキストオブジェクトは個々に保存されます。テキストはランタイムで修正できます。修正は、サーバーから接続されている全コンピュータに伝送されます。ローカルコンピュータで定義されたコンフィグレーション言語が更新されます。

注記

アラームロギングやユーザ管理者など特定の WinCC エディタは、コンフィグレーション中テキストライブラリにある同一のデータベーステーブルにアクセスします。したがって、このエディタにかぎり、1台のオペレーティングステーションから同時に操作できます。

タグ

タグはサーバーデータベースに保存されています。

注記

クライアントプロジェクトを終了してタグを修正した場合、変更時に有効であったコンピュータをすべて再起動しないかぎり、その修正は有効となりません。

ユーザ管理者

ユーザ管理者のオペレーティング権限はサーバーデータベースに保存されます。ユーザ管理者データはランタイム時に修正できます。修正は接続されているコンピュータには通知されません。クライアントが再度ログインした時に、新規作成データは有効となります。

下記も参照

プロジェクトの実行方法 (ページ 93)

プロジェクトの終了方法 (ページ 95)

サーバープロジェクトピクチャの編集方法 (ページ 91)

編集用にプロジェクトを開く方法 (ページ 89)

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

1.9.5 編集用にプロジェクトを開く方法

原理

サーバープロジェクトは、プロジェクトの設定、またはランタイム操作時に、クライアントから編集できます。ランタイムでのデータ更新は、設定されているデータに依存します。

Windows エクスプローラの[Simatic Shell]ダイアログは、設定用に有効化されたネットワーク内のサーバープロジェクトすべてのリストを提示します。選択したサーバーの追加情報、例えば、現在有効なモード(設定/ランタイム)などを入手できます。

1.9 リモートコンフィグレーション

複数のクライアントから同じプロジェクトを同時に開いて編集できます。

注記

ランタイムのサーバープロジェクトの有効化

クライアント経由でサーバープロジェクトを処理するために開き、WinCCで[ランタイム有効化]コマンドを実行する場合、以下の点について注意する必要があります。

マルチユーザシステムのクライアントからランタイムを有効化した場合、サーバープロジェクトが開いていても、クライアントプロジェクトだけが有効になります。

サーバープロジェクトを有効化するには、[Simatic Shell]ダイアログの[リモート有効化]コマンドを使用します。

同様のことが[ランタイム無効化]コマンドに適用されます。

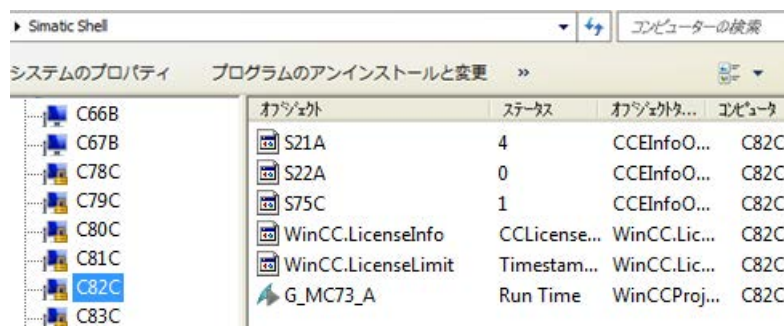
必要条件

クライアントからサーバープロジェクトを開いてリモート編集する場合、以下の条件を満たす必要があります。

- クライアントに登録されているユーザーは、サーバー WinCC プロジェクトの[リモート設定]用オペレータオーソリゼーションがあります。
- クライアントはサーバーのコンピュータリストにエントリされています。
- プロジェクトはネットワークからアクセス可能です。

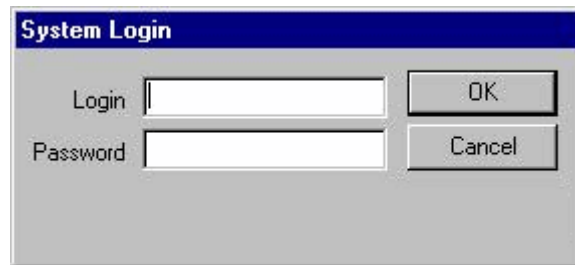
手順

1. クライアント側の Windows エクスプローラで、[Simatic Shell]エントリを選択します。
[Simatic Shell]ウィンドウが表示されます。
ネットワークで現在使用可能なすべてのサーバーとプロジェクトが、ナビゲーションウィンドウに表示されます。
または、別のウィンドウで[Simatic Shell]ダイアログを開きます。Windows エクスプローラの[Simatic Shell]ポップアップメニューで、[開く]コマンドを選択します。



2. 1 台のコンピュータを選択して、コンピュータ固有のプロジェクトを表示させます。

- プロジェクトリストから、開きたいプロジェクトを選択し、ショートカットメニューから[開く]コマンドを選択します。
[ログイン]ダイアログが表示されます。



- 現在のコンピュータのユーザー名とパスワードを入力します。
パスワードは、大文字と小文字を区別します。
[WinCC エクスプローラ - サーバーは利用できません]ダイアログで、[ローカルでサーバーを開始する]ボタンをクリックします。
クライアント側でプロジェクトが開き、設定が可能となります。

下記も参照

プロジェクトの終了方法 (ページ 95)

プロジェクトの実行方法 (ページ 93)

サーバープロジェクトピクチャの編集方法 (ページ 91)

複数のクライアントからのプロジェクトへのアクセス (ページ 87)

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

1.9.6 サーバープロジェクトピクチャの編集方法

原理

サーバーにある画像は、リモートクライアントから開いて編集、保存できます。ランタイム有効時に編集している場合、その修正は、次回に画像を実行したときに有効となります。

それぞれの画像は個々のファイルに保存されるため、一度にひとつの画像にアクセスできるのはクライアント 1 台に限られます。他のコンピュータからのその画像へのアクセスはブロックされます。

1.9 リモートコンフィグレーション

必要条件

- サーバーのプロジェクトフォルダでは、ネットワークからのアクセスを有効にしておく必要があります。
- クライアントに登録されているユーザーは、サーバー WinCC プロジェクトの[リモート設定]用オペレータオーソリゼーションがあります。

手順

1. クライアント側の Windows エクスプローラで、[Simatic Shell]エントリを選択します。
[Simatic Shell]ウィンドウが表示されます。
ネットワークで現在使用可能なすべてのサーバーとプロジェクトが、ナビゲーションウィンドウに表示されます。
2. プロジェクトリストから、開きたいプロジェクトを選択し、ショートカットメニューから[開く]コマンドを選択します。
[ログイン]ダイアログが表示されます。現在のコンピュータのユーザー名とパスワードを入力します。
パスワードは、大文字と小文字を区別します。
[WinCC エクスプローラ - サーバーは利用できません]ダイアログで、[ローカルでサーバーを開始する]ボタンをクリックします。クライアント側でプロジェクトが開き、設定が可能となります。
3. クライアント上の目的の画像をグラフィックデザイナーで開きます。
4. 画像を編集し、サーバーのプロジェクトフォルダに再度保存します。

下記も参照

プロジェクトの終了方法 (ページ 95)

プロジェクトの実行方法 (ページ 93)

編集用にプロジェクトを開く方法 (ページ 89)

複数のクライアントからのプロジェクトへのアクセス (ページ 87)

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

1.9.7 プロジェクトの実行方法

原理

クライアント/サーバーシステムは、リモートプロジェクトコンフィグレーションのオプションを提供するだけでなく、その実行や終了もリモートで行えます。

[SIMATIC Shell]ダイアログを使用してサーバープロジェクトをクライアントから実行する場合、そのサーバープロジェクトだけが実行されます。ただしツールバーにある[ランタイム開始]ボタンを使用して、WinCC 上でプロセスし、実行するためにサーバープロジェクトを開いた場合、サーバープロジェクトが開いていても、そのクライアントプロジェクトだけが実行されます。

注記

ランタイムを有効化できるのは、プロジェクトがローカルコンピュータにある場合だけです。

必要条件

クライアントからサーバープロジェクトを開いてリモート実行する場合、以下の条件を満たす必要があります。

- クライアントに登録されているユーザーは、サーバー WinCC プロジェクトの[リモートの実行]オペレータオーソリゼーションがあります。
- クライアントはサーバーのコンピュータリストにエントリされています。
- プロジェクトはネットワークからアクセス可能です。

手順

1. クライアント側の Windows エクスプローラで、[Simatic Shell]エントリを選択します。
[Simatic Shell]ウィンドウが表示されます。
ネットワークおよび現在の状態で使用可能である、すべてのサーバーやプロジェクトが表示されます。
2. 実行するプロジェクトを選択します。

1.9 リモートコンフィグレーション

3. ショートカットメニューで[リモートの実行]コマンドを選択します。
[ログイン]ダイアログが表示されます。
4. 現在のコンピュータのユーザー名とパスワードを入力します。サーバーでプロジェクトが実行されます。

注記

パスワードは、大文字と小文字を区別します。

エンジニアリングステーションから OS コンピュータをリモートで実行する方法

一般に OS プロジェクトやリモート実行では、以下の必要条件が適用されます:

- OS プロジェクトを対象とした WinCC プロジェクトで、OS サーバーとして ES コンピュータ名を入力します。
- OS プロジェクトをリモートで実行するために他のコンピュータを使用する場合、このコンピュータをクライアントコンピュータとして入力する必要があります。

サーバーおよびクライアント用のコンピュータ名は WinCC プロジェクトで同一にできないため、以下の手順に注意する必要があります。

1. コンピュータリストの[サーバー]で入力した ES コンピュータ名を、WinCC エクスプローラでダミーの名前に変更します。
2. プロジェクトを閉じます。
3. プロジェクトを開きます。
4. WinCC エクスプローラのコンピュータリストに新しいクライアントを追加します。
5. コンピュータリストの[クライアント]に ES コンピュータ名を入力します。
6. SIMATIC Manager を使用してターゲットシステムをロードします。
7. これで、エンジニアリングステーションから OS コンピュータ上のランタイムをリモートで実行できます。

下記も参照

プロジェクトの終了方法 (ページ 95)

サーバープロジェクトピクチャの編集方法 (ページ 91)

編集用にプロジェクトを開く方法 (ページ 89)

複数のクライアントからのプロジェクトへのアクセス (ページ 87)

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

1.9.8 プロジェクトの終了方法

原理

クライアント/サーバーシステムは、リモートプロジェクトコンフィグレーションのオプションを提供するだけでなく、その実行や終了もリモートで行えます。

[SIMATIC Shell]ダイアログを使用してサーバープロジェクトをクライアントから終了する場合、そのサーバープロジェクトだけが終了されます。ただしツールバーにある[ランタイム停止]ボタンを使用して、WinCC 上でプロセスし、終了するためにサーバープロジェクトを開いた場合、サーバープロジェクトが開いていても、そのクライアントプロジェクトだけが終了されます。

必要条件

クライアントからサーバープロジェクトを開いてリモート終了する場合、以下の条件を満たす必要があります。

- クライアントに登録されているユーザーは、サーバー WinCC プロジェクトの[リモートの実行]オペレータオーソリゼーションがあります。
- クライアントはサーバーのコンピュータリストにエントリされています。
- プロジェクトはネットワークからアクセス可能です。

手順

1. クライアント側の Windows エクスプローラで、[Simatic Shell]エントリを選択します。
[Simatic Shell]ウィンドウが表示されます。
[Simatic Shell]ダイアログに、ネットワークで使用可能なクライアント/サーバーシステムの、有効なサーバーおよびプロジェクトが表示されます。
2. 実行するプロジェクトを選択します。
3. ポップアップメニューから[リモートの終了]コマンドを選択します。[ログイン]ダイアログが表示されます。
4. 現在のコンピュータのユーザー名とパスワードを入力します。サーバーでプロジェクトが終了されます。

注記

パスワードは、大文字と小文字を区別します。

1.9 リモートコンフィグレーション

下記も参照

プロジェクトの実行方法 (ページ 93)

サーバープロジェクトピクチャの編集方法 (ページ 91)

編集用にプロジェクトを開く方法 (ページ 89)

複数のクライアントからのプロジェクトへのアクセス (ページ 87)

サブネットに接続されていないコンピュータへのアクセス方法 (ページ 83)

リモートコンフィグレーション (ページ 79)

WinCC のクライアント/サーバーシステム (ページ 16)

1.10 クライアント/サーバーシステムの OPC インターフェースの使用

原理

OPC(プロセスコントロール用 OLE)、はオートメーション産業部門の構成要素の世界的通信標準です。

OPC は Windows ベースの技術から開発された公開インターフェースで、これにより異なるメーカーの PLC 間、オペレーティング/モニタシステム間およびオフィスアプリケーション間で、エラーのない標準化されたデータ交換が可能となっています。

注記

オートメーション産業のトップ企業は、"OPC 協会"を設立して協力しあっています。

OPC 協会に関する詳細は、インターネットの「<http://www.opcfoundation.org>」で参照できます。

WinCC における OPC の使用

分散システムで使用する場合、各 WinCC サーバーはシステム全体をモニタできます。ただし、WinCC サーバーはメッセージの編集またはアーカイブなど、固有の範囲のタスクに対してのみ機能します。

WinCC OPC サーバーは、ソフトウェアインターフェースによって、WinCC Runtime へのアクセスを可能にします。WinCC OPC サーバーは対応する OPC 仕様にしたがって、全機能をサポートします。

いかなるソフトウェアも OPC 仕様を基本としている場合は、OPC クライアントとして条件を満たすことができます。この点で、OPC クライアントはソースの解析などに使用できます。独自に開発した OPC クライアントを作成すれば、固有の要求条件に最適に対応できます。

WinCC OPC サーバーモードで操作する場合、接続パックライセンスが、WinCC OPC サーバーとして使用するコンピュータにインストールされていなければなりません。OPC DA サーバーには接続パックライセンスは必要ありません。

OPC インターフェースは、WinCC がインストールされているクライアントとサーバーにインストールされます。

WinCC の OPC サーバーは以下の仕様をサポートします。

- OPC Data Access 2.05a und 3.00
- OPC XML Data Access 1.01

1.10 クライアント/サーバーシステムの OPC インターフェースの使用

- OPC Historical Data Access 1.20
- OPC Alarm & Events 1.10
- OPC UA 1.03

WinCC の OPC インターフェースの使用に関する詳細情報については、WinCC 情報システムの[通信]を参照してください。

下記も参照

WinCC のクライアント/サーバーシステム (ページ 16)

OPC の機能 (ページ 698)

ファイルサーバー

2.1 ファイルサーバーのセットアップ

はじめに

WinCC ファイルサーバーは、WinCC 構成要素の最小限の設定を備えたサーバーです。

プロジェクトをファイルサーバーに保存して集中管理できます。

これによって、全プロジェクトのバックアップコピーを定期的に作成することが容易になります。

注記

設定のみに使用

ファイルサーバーは、設定のためのみに使用します。

必要条件

インストールに関する注意事項で説明されている要件は、WinCC Fileserver V7 のインストールに適用されます。

その場合、以下の条件も適用されます。

- コンピュータがネットワーク(LAN)で使用可能
- ファイルサーバーを使用する場合は、管理者権限が必要です。

注記

WinCC 基本インストールのない PC

WinCC V7 と WinCC ファイルサーバー V7 を、1 台のコンピュータに同時にインストールすることはできません。

2.1 ファイルサーバーのセットアップ

インストール

コンピュータをファイルサーバーとして設定する場合は、コンピュータ上でファイルサーバー設定を実行します。

1. WinCC インストール DVD を起動します。
2. インストールタイプで「カスタムインストール」を選択します。
3. [プログラム]ダイアログで、[WinCC]グループの[WinCC Fileserver]エントリを選択します。

WinCC の最小インストールが、コンピュータで実行されます。

設定

ファイルサーバーにプロジェクトが保存されます。

アクセスの有効化

プロジェクトメンバー全員がプロジェクトにアクセスできるように、ファイルサーバーの対応するドライブとフォルダを共有する必要があります。

フォルダやドライブを共有するには、**Windows** 管理者権限が必要です。

設定コンピュータ上で共有するフォルダまたはドライブに、一意のドライブ文字を割り付けます。

これでプロジェクトメンバーは、ローカルプロジェクトと同様にファイルサーバーのプロジェクトを開くことができます。

WinCC ServiceMode

3.1 WinCC ServiceMode:標準プロジェクトとサービスプロジェクト

WinCC ServiceMode は、サービスとして WinCC Runtime を操作するオプションを提供します。

WinCC Runtime は、インタラクティブユーザーがコンピュータにログオンしていない場合にも、有効です。

このセクションでは、以下について説明します。

- WinCC ServiceMode を使用できる設定
- サービスプロジェクトとしてプロジェクトを設定する方法
- サービスプロジェクトを有効にする方法

概要

WinCC プロジェクトを、標準プロジェクトとしてあるいはサービスプロジェクトとして、設定できます。

WinCC プロジェクトを WinCC ServiceMode で動作させるには、サービスプロジェクトとして設定しなければなりません。

標準プロジェクト

WinCC Runtime を実行するには、ユーザーはコンピュータにログインしていなければなりません。

インタラクティブユーザーの入力が可能です。

サービスプロジェクト

WinCC Runtime は、インタラクティブユーザーがログインしていないコンピュータでも実行できます。

3.1 WinCC ServiceMode:標準プロジェクトとサービスプロジェクト

WinCC Runtime は、ログインしているユーザーによる操作も可能です。その場合、インタラクティブユーザーの入力も可能です。

注記

WinCC は、そのシステムにアクセス中のときは、実行できません

コントロールパネルおよび Windows タスクマネージャでの、WinCC のプロセスおよびサービスの変更は、許可されません。

以下の変更が影響を受けます。

- プロパティへの変更
- 手動でのアクセス:
 - 起動
 - 終了
 - 停止
 - 再開
 - 再起動
- 優先度の変更

個別のプロセスおよびサービスの間には、依存関係があります。

どのような変更もしてはいけません。

下記も参照

サービスプロジェクトの使用と制約 (ページ 104)

サービスプロジェクトの操作モード (ページ 109)

サービスプロジェクトを起動する方法 (ページ 115)

3.2 サービスプロジェクトの設定

概要

以下の設定で、WinCC Runtime をサーバーでサービスプロジェクトとして実行できます。

- Windows Server オペレーティングシステムを実装した WinCC サーバー
独自のプロジェクトを持つ WinCC クライアント
独自のプロジェクトを持たない WinCC クライアント
- WinCC WebNavigator サーバーまたは専用の Web サーバー
WinCC Web クライアント
- DataMonitor サーバーまたは専用の DataMonitor サーバー
DataMonitor クライアント

3.3 サービスプロジェクトの使用と制約

用途

サーバーで、WinCC サービスモードのプロジェクトは、サービスプロジェクトとして動作します。

WinCC Runtime はサービスとして起動します。

サービスプロジェクトは、自動的にまたは手動で、起動します。

ユーザーのログオンなしでの操作

サービスプロジェクトは、インタラクティブユーザーがコンピュータにログオンしていない状態で、実行できます。

インタラクティブユーザーがログオンしていない場合、インタラクティブな操作はできません。

ユーザーのログオンありでの操作

サービスプロジェクトでは、インタラクティブな操作は必ずしも必要ではありません。

インタラクティブユーザーは、例えばサービスの目的でログオンできます。この場合、ユーザーはサービスプロジェクトのインタラクティブな操作を有効にできます。

Autostart

サーバーの電源が入り、設定されたプロジェクトが有効になると、WinCC Runtime は、自動起動で自動的に開始します。

自動起動は、インタラクティブユーザーがログオンしていなくても実行できます。

手動起動

手動起動では、ユーザーがサーバーにログオンして、プロジェクトを起動しなければなりません。

ユーザーがサーバーからログオフしても、WinCC Runtime は有効なままです。

ユーザーのログオンとログオフ

サービスプロジェクトが有効の間、インタラクティブユーザーはいつでもサーバーにログオンおよびログオフできます。

制限

サービスプロジェクトは以下の制約を受けます。

スクリプト

インタラクティブユーザーは必ずしもサービスプロジェクト、C スクリプト、VB スクリプトにログオンしていないので、以下の場合に問題が発生します。

- 入力などのインタラクティブ操作がスクリプトで必要な場合。
- スクリプトは、メッセージボックスを開きます。

サービスモードの C スクリプトには、共通データ領域がありません。

このため、例えば"グローバルスクリプト"と"グラフィックデザイナー"の間でグローバル C 変数を交換することができません。

追加のプログラムまたはタスク

サービスプロジェクトでは、スタートアップリストに、追加のプログラムまたはタスクを追加できません。

非リリース構成要素

接続ステーション経由の OPC アクセスは、サービスプロジェクトにはリリースされません。

サービスプロジェクトの診断情報

一般的な規則として、ユーザーは、有効なサービスプロジェクトのサーバーには、ログオンしません。

WinCC は、サーバーで診断情報を表示できません。したがって、WinCC は診断情報をクライアントに転送します。

これについての詳細情報は、WinCC 情報システムの[WinCC での作業]>[プロジェクトでの作業]>[付録]>[WinCC 診断ウィンドウとライセンス情報]を参照してください。

注記

サービスプロジェクトの編集または移行

サービスプロジェクトを編集または移行するには、コンピュータで適切に ServiceMode ユーザーを管理する必要があります。

ServiceMode ユーザーが利用できない場合、ログオンしている Windows ユーザーは、プロジェクトの編集または移行のために適切に管理されている必要があります。

下記も参照

サービスプロジェクト用の Autostart の設定方法 (ページ 113)

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 101)

3.4 WinCC ステータスおよびシステムトレイでのコントロール

はじめに






WinCC は、トレイ領域と呼ばれるタスクバーの通知領域に"SIMATIC WinCC"シンボルを表示します。

このシンボルはプロジェクトステータスに関する情報を提供します。

WinCC プロジェクトは、このシンボルのショートカットメニューから起動および停止できます。

プロジェクトステータス

以下の表は、プロジェクトステータスとそれに対応する[SIMATIC WinCC]シンボルを示します。

アイコン	ステータス
	<ul style="list-style-type: none"> WinCC が起動していない。 プロジェクトが開いていない。
	<p>WinCC のステータスは以下の場合に変更されます。</p> <ul style="list-style-type: none"> WinCC がプロジェクトを開いた。 WinCC がプロジェクトを起動した。 WinCC がプロジェクトを停止した。 WinCC がプロジェクトを閉じた。
	プロジェクトが開いている。
	プロジェクトが起動されます。
	プロジェクトが有効になり、サーバーは「障害」ステータスです。

ポップアップメニューを使用したコントロールオプション

「SIMATIC WinCC」シンボルのショートカットメニューは、以下の機能を提供します。

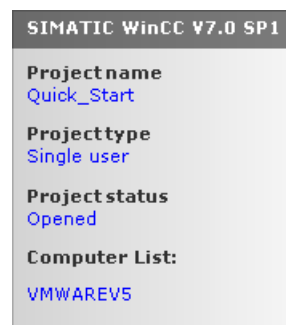
- ランタイムの有効化
- グラフィックランタイムの終了
- ランタイムを無効化する
- プロジェクトを閉じる
- ランタイム起動オプション(自動開始設定)

- [診断]ウィンドウを開く
- WinCC ライセンス解析

SIMATIC WinCC®ウィンドウ

[SIMATIC WinCC]ウィンドウを開くには、"SIMATIC WinCC"シンボルをクリックします。

例:ランタイムが有効なウィンドウ



ウィンドウには以下の情報が表示されます。



- プロジェクトの名前
- プロジェクトタイプ
- プロジェクトステータス
- コンピュータリスト
ローカルコンピュータは青色で表示されます。

コンピュータリスト



コンピュータリストには、ネットワーク上のすべてのコンピュータが表示されます。

プロジェクトが起動すると、すべての既存のコンピュータの接続ステータスが表示されます。

以下の表に、接続ステータスのアイコンとその意味を示します。

アイコン	ステータス
	<ul style="list-style-type: none"> • 接続なし • 接続が切断されている
	<ul style="list-style-type: none"> • ローカルコンピュータ • リダンダントパートナーサーバー

3.4 WinCC ステータスおよびシステムトレイでのコントロール

アイコン	ステータス
	接続されている <ul style="list-style-type: none">• スタンドバイサーバーと• マスタサーバーと(しかしスタンバイサーバーが優先サーバー)
	接続されている <ul style="list-style-type: none">• マスタサーバーと• スタンバイサーバー(優先サーバーとして)と

この表示は、ネットワーク内の PC のステータスに関する情報のみを提供します。

コントローラの接続ステータスを照会するには、WinCC エクスプローラの[ドライバ接続のステータス]機能、またはシステムタグ[@<接続名>@ConnectionStateEx]を使用します。

下記も参照

サービスプロジェクトを起動する方法 (ページ 115)

3.5 ファンクションと前提条件

3.5.1 サービスプロジェクトの操作モード

はじめに

この章では、WinCC サービスプロジェクトの操作モードを説明します。

標準プロジェクト

標準プロジェクトを以下のように起動します。

- ユーザーがシステムにログオンします。
- ユーザーが WinCC ランタイムを起動するか、あるいは WinCC ランタイムが自動的に起動します。

以下のケースのどれかが発生するまで、WinCC ランタイムは有効です。

- ユーザーが WinCC ランタイムを終了する。
- ユーザーがシステムからログオフする。
この場合、システムが WinCC ランタイムを停止させます。

サービスプロジェクト

サービスプロジェクトでは、WinCC ランタイムはサービスとして起動します。設定によって異なりますが、これらのサービスは以下の時に起動します。

- オペレーティングシステムが起動した後、自動的に。
- ユーザーがログインして、WinCC ランタイムを起動した後に。

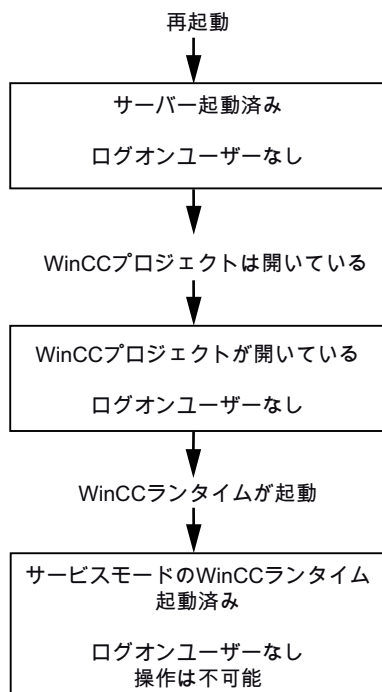
ユーザーが再びログオフしても、WinCC はそのままです。

WinCC ランタイムのデータには引き続きアクセスできます。

ログオンしているユーザーは、必要に応じてランタイム操作を起動できます。

以下の図は、サーバーの起動とランタイムによるサービスプロジェクトの自動起動との間の状態を、示しています。

3.5 ファンクションと前提条件



下記も参照

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 101)

サービスプロジェクトを起動する方法 (ページ 115)

3.5.2 サービスプロジェクトの実行の必要条件

サービスプロジェクトでは、インタラクティブな操作は必ずしも必要ではありません。

スクリプト

インタラクティブユーザーは必ずしもサービスプロジェクト、C スクリプト、VB スクリプトにログオンしていないので、以下の場合に問題が発生します。

- 入力などのインタラクティブ操作がスクリプトで必要な場合。
- スクリプトは、メッセージボックスを開きます。

分散 WinCC シナリオにおけるサービスプロジェクト

サービスプロジェクトに、専用の Windows ユーザーを設定しなければなりません。

サービスプロジェクトに設定された Windows ユーザーは、「SIMATIC HMI」ユーザーグループに属さなければなりません。

WinCC マルチユーザーシステムまたはサーバー-サーバー通信を使用する分散システムの場合、ローカル Windows ユーザーまたは Windows ドメインユーザーを使うことができます。

ローカル Windows ユーザー

- ユーザーは、ネットワークの全てのコンピュータで、ローカルの「SIMATIC HMI」ユーザーグループのメンバーでなければなりません。
- このユーザーのパスワードは、全てのコンピュータで同じでなければなりません。

Windows ドメインユーザー

- ユーザーは、全てのコンピュータで、ローカルの「SIMATIC HMI」ユーザーグループのメンバーである。
- ユーザーは、ローカルの「SIMATIC HMI」ユーザーグループのメンバーであるグループの、メンバーである。

パスワード

WinCC サービスプロジェクトの中断のないランタイム操作の要件:

- 設定されたユーザーのパスワードを変更できません。
- 設定されたユーザーのパスワードは期限切れになりません。

これを確実に実行するためには、ユーザーを設定する際に、以下のオプションを有効にします。

- [ユーザーがパスワードを変更できない]
- [パスワードが期限切れにならない]

下記も参照

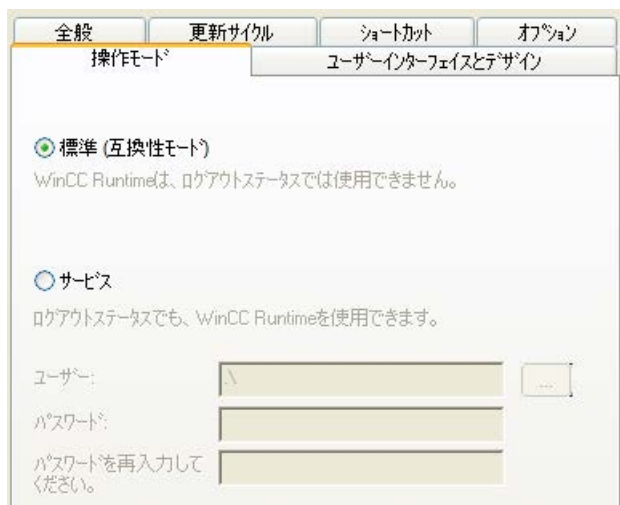
サービスプロジェクト用の Autostart の設定方法 (ページ 113)

3.6 WinCC ServiceMode の設定

3.6.1 プロジェクトをサービスプロジェクトとして定義する方法

はじめに

プロジェクトのプロパティで、プロジェクトを標準プロジェクトまたはサービスプロジェクトのどちらとして実行するかを、指定します。



手順 - サービスプロジェクトを定義

1. WinCC エクスプローラのナビゲーションウィンドウでプロジェクト名を選択します。
2. プロジェクトのショートカットメニューで、[プロパティ]エントリを選択します。
[プロジェクトプロパティ]ダイアログが開きます。
3. [動作モード]タブに切り替えます。
4. [サービス]オプションを有効にします。
WinCC は、プロジェクトを変換するためにプロジェクトを再ロードする必要がある旨のメッセージを、表示します。
5. WinCC サービスプロジェクトを実行するユーザーを、[ユーザー]フィールドに入力します。
このユーザーに必要なプロパティの詳細情報: 「サービスプロジェクトの実行の必要条件 (ページ 110)」
6. [パスワード]フィールドに、関連するパスワードを入力します。
7. [パスワード]フィールドのパスワードを、確認します。
8. [OK]をクリックして入力を確定します。
9. プロジェクトを再ロードします。

サービスプロジェクトの標準プロジェクトへの変換

サービスプロジェクトを標準プロジェクトに変換する場合、[動作モード]タブの[標準]オプションを有効にします。

標準プロジェクトのサービスプロジェクトへの変換

標準プロジェクトをサービスプロジェクトに変換する場合、[動作モード]タブの[サービス]オプションを有効にします。

注記

標準プロジェクトはサービスプロジェクトへ必ずしも変換できません

サービスプロジェクトは制約を受けます。変換を実行する前に、これらに注意してください。

詳細情報は、「サービスプロジェクトの使用と制約 (ページ 104)」を参照してください。

下記も参照

サービスプロジェクト用の Autostart の設定方法 (ページ 113)

3.6.2 サービスプロジェクト用の Autostart の設定方法

AutoStart の設定

コンピュータ起動されたとき、WinCC は選択された WinCC プロジェクトを有効にします。

[AutoStart 設定]ダイアログで、AutoStart プロジェクトとして希望するサービスプロジェクトを選択します。

注記

サービス再起動中のプロジェクトの有効化

[SIMATIC WinCC CCProjectMgr]サービスを再起動しているとき、AutoStart プロジェクトも再度有効化されます。

操作モードの変更:AutoStart の再設定

サービスプロジェクトから標準プロジェクトへ、あるいはその逆に変換した後、必ず「AutoStart」を再設定しなければなりません。

手順 - AutoStart の設定

1. Windows プログラムグループ[Siemens Automation]で、[AutoStart]エントリを選択します。
[AutoStart の設定]ダイアログが開きます。
ローカルコンピュータの設定が表示されます。
2. 希望するコンピュータを選択します。
以下のオプションがあります。
 - コンピュータ名を入力します。
 - [...]から、ネットワークパスのコンピュータを選択します。
 - ローカルコンピュータに対して自動起動を設置するには、[ローカルコンピュータ]をクリックします。選択されたコンピュータの現在の設定を表示するには、[設定の読み取り]をクリックします。
3. [プロジェクト]フィールドの [...] ボタンをクリックして、サービスプロジェクトを選択します。
このボックスに、プロジェクトファイルとそのフルパスが入力されています。
プロジェクトタイプは、パスの下に表示されます。
4. 自動起動動作の設定を構成します。
5. [Autostart の有効化]オプションを有効にします。
オプションが無効にされている場合、設定されたコンピュータに対して Autostart は実行されません。
6. [適用]を押して設定を確定し、[OK]を押して閉じます。

結果

次のコンピュータの起動時に、WinCC が自動的に起動し、選択されたプロジェクトが開かれます。

下記も参照


プロジェクトをサービスプロジェクトとして定義する方法 (ページ 112)

サービスプロジェクトの使用と制約 (ページ 104)

3.7 ランタイムのサービスプロジェクト

3.7.1 サービスプロジェクトを起動する方法

はじめに

WinCC は、タスクバーの通知領域、いわゆるシステムトレイに[SIMATIC WinCC]アイコンを作成します。

特に、次のファンクションを実行するため、このアイコンのショートカットメニューを使用できます。

- ランタイムの有効化。
- グラフィックランタイムの終了。
- ランタイムを無効化する。

詳細については、「WinCC ステータスおよびシステムトレイでのコントロール (ページ 106)」を参照してください。

必要条件

プロジェクトが、サービスプロジェクトとして保存されている必要があります。

これ以外の必要条件については、「サービスプロジェクトの実行の必要条件 (ページ 110)」を参照してください。

WinCC ServiceMode の自動起動

[AutoStart の設定]ツールでプロジェクトの自動起動を正しく設定すると、以下が実行されます。

- サーバーが起動されると、すぐにプロジェクトも自動的に起動されます。

ユーザーの入力は不要です。

結果


プロジェクトが有効化されていること。

ユーザーはシステムにログオンしていません。

3.7 ランタイムのサービスプロジェクト

WinCC ServiceMode の手動起動

以下の手順では、プロジェクトに対して自動起動が設定されていないと仮定しています。

1. サーバーを起動します。
2. サーバーにログオンします。
3. プロジェクトを開きます。
4. システムトレイのアイコン  のショートカットメニューで、[ランタイムを有効化]コマンドを選択します。
あるいは、WinCC エクスプローラでプロジェクトを有効化します。

結果

プロジェクトが起動されます。

WinCC はアイコン  を表示します。

サーバーからログオフした時に、WinCC Runtime が有効なままになるようにするには、WinCC エクスプローラだけを終了します。これを行なうには、[WinCC エクスプローラを終了]ダイアログの[WinCC エクスプローラを終了]エントリを選択します。

すると、WinCC Runtime は有効なままになります。

下記も参照

サービスプロジェクトの実行の必要条件 (ページ 110)

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 101)

サービスプロジェクトの操作モード (ページ 109)

WinCC ステータスおよびシステムトレイでのコントロール (ページ 106)

3.7.2 起動したサービスプロジェクトにログインおよびログオフする方法

はじめに

WinCC プロジェクトがランタイムにあるときに、サーバーで必要な作業を行なうために、サービスに再びログインおよびログオフすることができます。

注記

再起動を必要とする更新がインストールされると、WinCC ランタイムは終了します。

必要条件

サービスプロジェクトが起動されている。ユーザーがサーバーにログインしていない。

手順

1. サーバーにログインします。
2. 必要なアクションを実行します。
3. サーバーからログオフします。

結果

サーバーにログインおよびログオフしました。WinCC ランタイムは、影響を受けていません。

3.7.3 サービス目的でのインタラクティブな操作を有効にする方法


はじめに

WinCC サービスプロジェクトがランタイムにある間に、インタラクティブな操作を起動できます。

必要条件


サービスプロジェクトが有効である。インタラクティブな操作が起動されていない。
"SIMATIC HMI"グループのメンバーであるユーザーとして、ログインします。

手順 - インタラクティブな操作の起動

1. サーバーにログインします。
2. トレイ領域の  アイコンのポップアップメニューから、[グラフィックランタイムの開始]コマンドを選択します。
WinCC は、インタラクティブな操作をリリースします。WinCC プロジェクトを操作できます。

3.7 ランタイムのサービスプロジェクト

手順 - インタラクティブな操作の終了

1. トレイ領域の  アイコンのポップアップメニューから、[グラフィックランタイムの終了]コマンドを選択します。
WinCC はグラフィックランタイムを終了します。
2. 必要に応じて、ログオフします。

リダンダントシステム

4.1 2 重化

コンテンツ

WinCC オプションの[WinCC/冗長化]は、冗長システムを構成するために使用します。2 台の相互接続されたサーバーの並列操作と、誤作動が発生した場合のサーバーの自動切替えによって、WinCC およびシステムの使用可能性が向上します。

概要

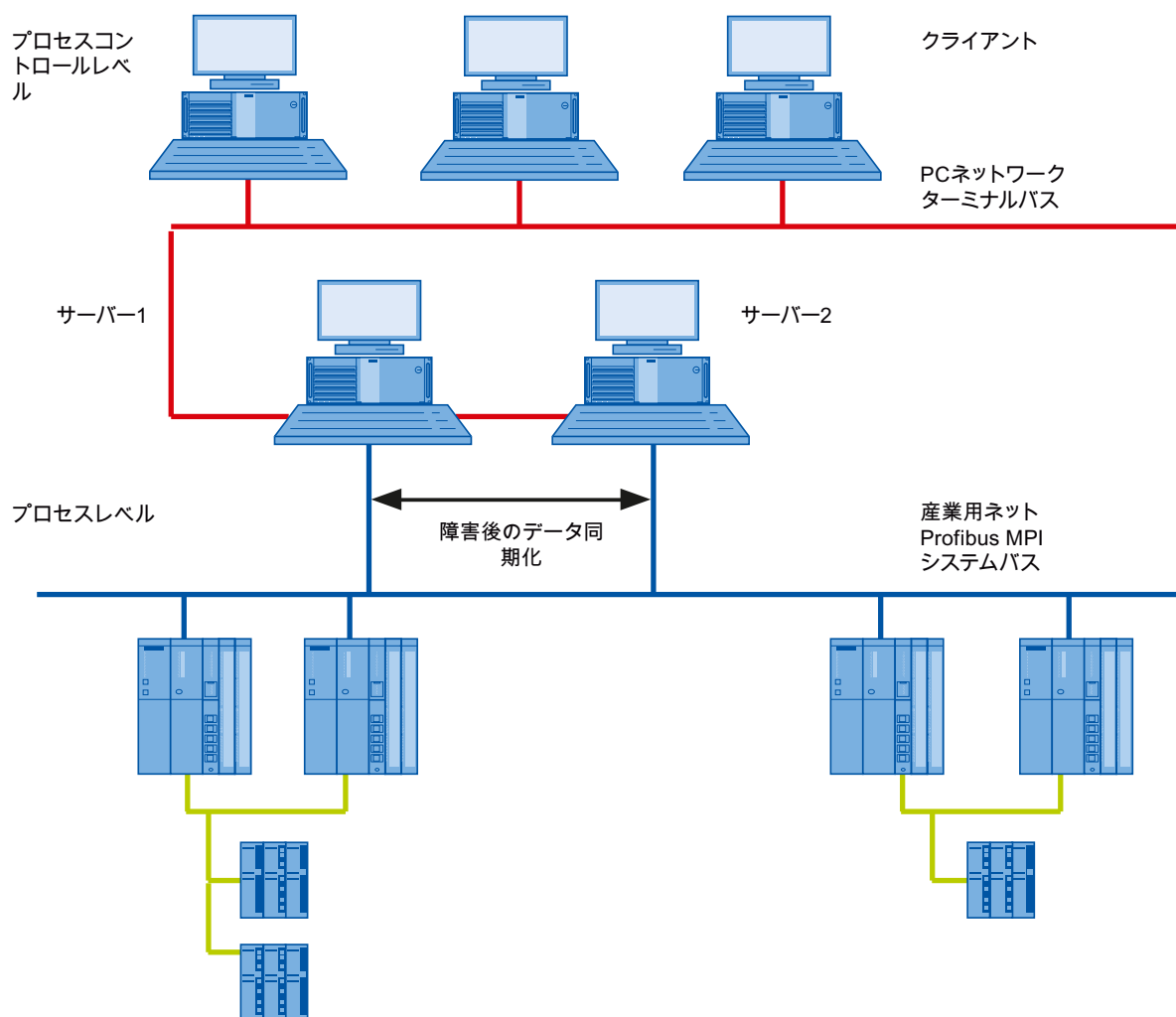
このドキュメントは以下について説明しています。

- 冗長システムの必要条件。
- WinCC における冗長システムの構築方法。
- 冗長化のためのサーバーの構成方法。
- 冗長アーカイブの同期の設定方法。

4.2 WinCC の 2 重化

はじめに

冗長性 WinCC プロジェクトは、同じ機能を実行し、並行して作動するように設定された 2 台の WinCC サーバーで構成されています。1 台はマスタサーバーで、もう 1 台はスタンバイサーバーです。2 台のサーバーはオートメーションシステム、クライアントそして互いに接続されています。



WinCC 冗長性の機能の概要

WinCC 冗長性は以下の機能を提供します。

- サーバーが故障するまたはプロセス接続失敗時のクライアントの自動切り替え。
- 故障したサーバを回復した後、またはプロセス接続エラーを消去した後の、メッセージアーカイブ、プロセス値およびユーザーアーカイブの自動同期化。
- 内部メッセージのオンライン同期化。
- タグ同期化をサポートする内部タグのオンライン同期化。
- ユーザーアーカイブのオンライン同期。
- 冗長サーバーにプロジェクトをコピーするための「プロジェクトデュプリケータ」。
- WinCC アプリケーションをモニタする「Application Health Check」ファンクション。
- ローカルシステムのハードウェアおよびソフトウェアの状態を監視する「SelfDiagnosis」関数。

「Application Health Check」ファンクション

「Application Health Check」ファンクションは、全ての重要な WinCC アプリケーションを、自動的にモニタします。

ライフビートのモニタリングは、ソフトウェアのエラーを検出すると、以下を実行します。

- 「@RedundantServerState」システムタグのサーバーステータスを、「故障」にセットします。
- 接続しているクライアントの冗長サーバーへの切り替えを、承認します。
- プロセスコントロールメッセージによるソフトウェアエラーのユーザーへの通知。エラーの原因がアラームサーバーの場合は、プロセスコントロールメッセージを生成できません。

注記

「Application Health Check」ファンクションによってソフトウェアのエラーが検出され、クライアントの切り替えが始まると、関連するサーバーが再起動されます。そうしないと、クライアントをサーバーに再接続できません。アーカイブはエラーが検知された時点まで遡及的に同期されます。

4.2 WinCC の 2 重化

「SelfDiagnosis」関数

[SelfDiagnosis]ファンクションは、冗長システムの可用性と安定性を確保する次のタスクで構成されます。

- ローカル HW と SW の問題を監視して報告する
- ローカルシステムのパフォーマンスを監視する
- データ量の状態を監視する
- 必要な場合、サーバーフェイルオーバー

故障が発生した場合には、次のタスクが実行されます。

- アプリケーションの再起動
- 必要に応じて、サーバーの状態が[障害]に設定され、サーバーが移行します。
- ログエントリが生成されます。
- システムアラームが発生されます。

4.3 冗長システムの必要条件

概要

WinCC の冗長性については、以下の前提条件に従う必要があります。

- マルチユーザー操作を使用する冗長 WinCC サーバーには、サーバーオペレーティングシステムを使用するコンピュータのみを使用できます。
- WinCC 冗長性オプションは両方のサーバーにインストールする必要があります。WinCC 冗長性ライセンスは冗長サーバーにインストールする必要があります。
- 2 つの冗長サーバーを機能的に等しく設定する必要があります。
- 2 つの冗長サーバーに追加して、さらに別の PC を冗長サーバーとして設定することはできません。
- サーバーは時間の同期が取れている必要があります。システム全体で時間の同期が取れていることを推奨します。時間の同期は、WinCC の[時間の同期]オプションで設定できます。
- オートメーションシステムおよびクライアントからのメッセージと確認には必ずフレームにタイムスタンプ(時系列順メッセージ)が必要です。これによりエントリの重複が回避されます。例えば、アラームクロックをオートメーションシステムで使用するします。
- 下位のオートメーションシステムからプロセス値、メッセージおよびアクティブメッセージブロックを同時に両方のサーバーに送信する必要があります。
- 冗長サーバー間には、下記の追加接続の 1 つが存在する必要があります。
 - ネットワークアダプタ
 - シリアル接続

この追加接続によって、「マスタ」または「スタンバイ」ステータスの正確な定義が確保されます。

[冗長性]エディタを使用して、WinCC エクスプローラのネットワークカード接続経路で、追加接続を設定します。TCP/IP プロトコルを対応する IP アドレスと共に使用します。この IP アドレスは、ターミナルバスと同じサブネットに存在してはいけません。

4.3 冗長システムの必要条件

注記

WinCC および有効な WinCC 冗長性コミッショニング中のランタイム動作

サーバーコンピュータ上でコミッショニング中に、WinCC Runtime は頻繁に有効と無効に切替わります。このアクティブな WinCC 冗長性での繰り返し実行により、アーカイブが毎回同期されます。これにより、WinCC Runtime 動作の著しい劣化が発生する場合があります。従って、コミッショニング中は WinCC 冗長性を無効にしておくことをお勧めします。

無停電電源装置

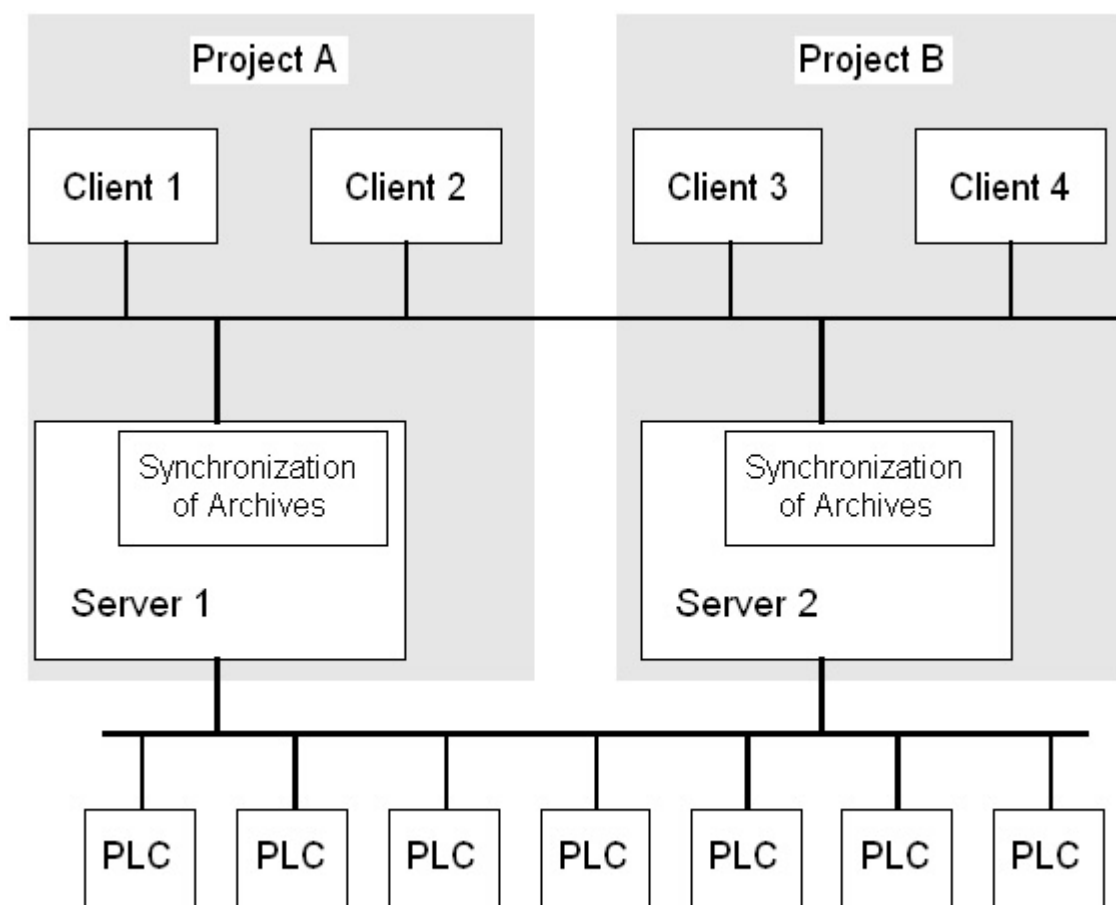
不意の停電にも WinCC を安全に終了させるために、無停電電源装置(UPS)の使用を推奨します。

4.4 2 重化の仕組み

はじめに

両方のサーバーは同一の権限を持ち、互いに独立して作動します。ユーザーは、両方のサーバーにアクセスできます。どちらかのサーバーが故障すると、同等の冗長サーバーをいつでも利用できます。

以下の図は、並行冗長サーバーのアーカイブとアーカイブの同期化を示しています。



冗長サーバーの識別

2 つのサーバーのうち、1 つをデフォルトマスタとして構成できます。

このサーバーのランタイムでシステム変数「@RM_MASTER」を「1」に設定します。コンピュータの故障などによってタグのステータスが変わると、クライアントの接続先は元の

4.4 2 重化の仕組み

「スタンバイ」コンピュータに切り替わります。以前の「スタンバイ」コンピュータは現在の「マスタ」です。

サーバーはランタイム時に互いにモニタし合い、パートナーサーバーの不具合をすばやく検出します。

ネットワークアダプタ経由の追加接続かサーバー間のシリアル接続を使用して、ステータスをモニタします。この接続によって、冗長パートナーサーバー間の通信を改善できます。これにより冗長化を利用しやすくなります。アーカイブの同期に追加の接続は使用されてません。

正常作動時の WinCC のアーカイブ

サーバーは通常ランタイムで完全に並列に実行されます。各サーバーコンピュータは個々のプロセスドライバで接続されており、固有のデータアーカイブを持っています。

プロセスデータおよびメッセージは、オートメーションシステムによって両方の冗長サーバーに送信され、両方の冗長サーバーによって適切に処理されます。

ユーザーアーカイブ、内部メッセージ、内部タグは、連続してオンラインで同期できます。

2 つのサーバーは TCP/IP プロトコルのある LAN で通信し、アーカイブを同期します。

サーバーの故障

一方のサーバーが故障すると、クライアントは故障したサーバーから冗長パートナーサーバーへ自動的に切り替えられます。これにより、すべてのクライアントが常にプロセスをモニタし、操作できます。

故障中、正常なサーバーは、WinCC プロジェクトのすべてのメッセージおよびプロセスデータを、アーカイブし続けます。

故障したサーバーがオンラインに復帰すると、すべてのメッセージアーカイブ、プロセス値アーカイブ、およびユーザーアーカイブが、復帰したサーバーへ自動的にコピーされます。このとき、正常サーバーと故障サーバーとのアーカイブデータのギャップが埋められます。

注記

冗長性障害:最低 69 秒

技術的な理由で、ダウンタイムは、2 つのシステムの自動同期が最低 69 秒持続する必要があります。

クライアントの切り替えをトリガするファクタ

サーバーに障害が発生している間のデフォルト(マスタ)サーバーからパートナーサーバーへのクライアントの切り替えは、システムによって自動的に実行されます。

サーバーの切り替えは、以下のファクタによって起こります。

- サーバーへのネットワーク接続の故障
- サーバーの障害
- プロセス接続の誤動作
- 「Application Health Check」ファンクションが WinCC アプリケーションの不具合を検出し、切り替えをトリガした。
- プロジェクトが停止した。

プロセス接続でのエラー発生時にクライアント切り替えの冗長性オプションが有効な場合、「マスタ」サーバーおよびリダンダントパートナーサーバーへの論理接続の不具合数が、周期的に計測されます。

「マスタ」サーバーの論理接続の不具合数がリダンダントパートナーサーバーの不具合数よりも多い場合、マスタサーバーにログオンしているクライアントは、リダンダントパートナーサーバーに切り替えられます。

プロセスリンクのエラーが是正されると、クライアントは、元来接続されていた優先サーバーに切り替えられます。

プロセスリンクのモニタは、冗長サーバーが両方ともランタイムになるまで開始されません。

注記

サーバー上のソフトウェアエラー

サーバー上でソフトウェアエラーが発生した場合は、クライアントの接続をリダンダントパートナーに切り替えるのではなく、システムをブロックできます。

サーバー復帰後にアーカイブの同期をトリガするファクタ

サーバー間のアーカイブの同期は以下のエラーが修正された後に開始されます。

- プロセス接続エラー。ただし、プロセス接続モニタをオフにすることができます。
- パートナーサーバーへのネットワーク接続の障害。
- サーバーの故障。
- プロジェクトが作動していない。

4.4 2 重化の仕組み

サーバー復帰後の同期

故障したサーバーが復帰したら、WinCC 冗長性は不足しているデータを故障したサーバーに転送します。

これは、メッセージアーカイブ、プロセス値アーカイブ、ユーザーアーカイブおよび内部タグに適用されます。

メッセージアーカイブとプロセス値アーカイブの場合、ダウンタイム中にアクティブだったすべてのセグメントが転送されます。転送後、2 台の同等のサーバーが再び利用可能になります。

アーカイブの同期はバックグラウンドで、WinCC のプロセス管理およびアーカイブ処理と並行して実行されます。したがって、システムの作動および監視は常に保証されています。

内部タグの比較

内部タグには、プロパティ「タグ同期」が必要です。

冗長サーバーの 1 台でタグの 1 つが修正されると、直ちにパートナコンピュータで内部タグが比較されます。

内部タグには、たとえば「@RM_Master」などの、文字「@」で始まる名前を持つシステムタグも含まれます。システムタグには、オンライン同期を構成できません。

プロセス接続エラー後の同期

プロセス接続モニタリングを有効にすると、サーバーとオートメーションシステム間の障害が解消されると自動的にすべてのアーカイブの同期が開始されます。

プロセス接続モニタが作動すると、該当するサーバーが、構成されたすべての接続に対してライフビートモニタを実行します。関連するオートメーションシステムがサーバーに確認メッセージを送信できない場合、オートメーションシステムへのプロセス接続に障害が発生したことをサーバーが検知します。

1 つまたは複数のオートメーションシステムに対するネットワークエラーが検出されると、そのプロジェクトに属するオートメーションシステムのすべてのアーカイブの同期が実行されます。障害の発生していないオートメーションシステムのアーカイブも同期されます。このオプションを無効にすると、ランタイム中のサーバー負荷が防止されます。

ネットワーク接続のモニタリングが無効になっているとき、オートメーションシステムのネットワークのエラーは認識されないため、アーカイブの同期は実行されません。

オンライン同期

サーバー間での直接的な同期をサポート:

- 以下のアラームロギングを実施:
 - 内部メッセージタグ
 - タグ接続のないメッセージ
 - システムオペレーションメッセージ
 - 「バッチ」メッセージ
- ユーザーアーカイブを対象
- タグ同期付き内部タグを対象

ブロックされているメッセージの比較

故障したサーバが復帰すると、現在ブロックされているメッセージが検索され、オートメーションシステムの全般クエリで同期されます。

メッセージが1台のサーバーでのみパッシブにブロックされている場合、ブロック情報が同期されます。

4.5 冗長性システムの構成

4.5.1 冗長システムのセットアップ方法

はじめに

ここで、WinCC 冗長性システムの設定方法の概要について理解します。クライアント-サーバシステムの構造に関する一般的な情報は、WinCC 情報システムの「分散システム」の章に記載されています。

Windows でのサーバーの入力

2 つの冗長サーバーはネットワーク上で互いに認識していなければなりません。さらに、ユーザーおよびパスワードが、冗長サーバー同士で一致する必要があります。ユーザーに管理者またはユーザー権限を設定する必要があります。ユーザーは「SIMATIC HMI」ユーザーグループのメンバーでなければなりません。

サーバーでのプロジェクトの構成

WinCC 冗長性の設定時に、以下の事項を決定します。

- 標準マスタ。
- パートナーサーバー。
- クライアントの切り替え動作。
- アーカイブの同期のタイプ。

プロジェクトを複製する前に、WinCC エクスプローラで「サーバーデータ」エディタを使ってサーバーパッケージを作成します。なるべく標準サーバーにサーバーパッケージを作成します。

注記

同期を行う場合は、必要なユーザーアーカイブのみを構成します。同期するユーザーアーカイブの数が増えれば、それだけ同期処理にも時間がかかり、システムへの負荷が大きくなります。

WinCC プロジェクトの複製

冗長パートナーサーバーに機能的に同等の WinCC プロジェクトを配備するには、「プロジェクトデублиケータ」を使ってデフォルトのサーバーからプロジェクトを複製します。マスタサーバーおよびスタンバイサーバーに同じプロジェクト設定が作成されます。

注記

複製を行う前に、プロジェクトが複製されるコンピュータのメモリが十分であることを確認します。既存のプロジェクトを複製する場合、このプロジェクトが開いておくことはできません。

スタンバイサーバーの構成

冗長性のステータスをモニタリングするには、「冗長性」エディタでスタンバイサーバーにマスタサーバーへの追加接続を設定する必要があります。

クライアントの構成

クライアントで WinCC 冗長性を使用するには、「サーバーデータ」エディタで以下の手順を実行します。

- デフォルトサーバーのパッケージを作成します。
- 希望のサーバーを設定し、パッケージの自動更新を有効にします。

リダンダントサーバーの起動

1. 最初に構成済みのマスタサーバーを起動します。
2. 次に接続したクライアントを起動します。
3. クライアントが起動したら、2 番目のサーバーと接続されたクライアントを起動します。

最初の同期が実行されます。この同期化のためのダウン時間には、最初のサーバーの起動と 2 番目のサーバーの起動の合間も含まれます。

注記

冗長サーバーの起動中は、最初のサーバーを完全に起動してから冗長パートナーを起動しなければならないことに注意が必要です。サーバーの最初の起動中は、有効なクライアントがあってははいけません。

冗長サーバーのペアを完全に無効にした後、再起動の間特定の順番を守る必要があります。最後に無効にすべきサーバーだったサーバーを、最初に起動します。このサーバーが完全に起動した後、冗長パートナーを起動できます。

4.5 冗長性システムの構成

冗長サーバーの停止

冗長サーバーを停止にする前に、2 番目のサーバーが機能していて、エラーなしで動作している必要があります。

アーカイブの同期化は、対応するプロセスコントロールメッセージが示すように、停止前に完了している必要があります。

注記

最初のサーバーのアーカイブの同期化が完了する前に 2 番目のサーバーを停止すると、データが失われることがあります。これは、コミッショニング中にサーバーの有効/無効を頻繁に切り替える場合に特に重要です。

下記も参照

冗長サーバーを構成する方法 (ページ 133)

ユーザーアーカイブの同期の設定方法 (ページ 136)

4.5.2 同一ファンクションのコンフィグレーション

プロセスデータアーカイブとメッセージアーカイブ

タグロギングおよびアラームロギングは、冗長サーバー間で機能的に等しくなるように設定する必要があります。追加の測定ポイントまたはアーカイブという形態で追加できるようにするため、同一のアーカイブが必要です。同期に拡張は含まれません。拡張はパートナーサーバーでユーザー自身が調整する必要があります。

WinCC はハードディスクベースの以下のアーカイブを同期します。

- プロセス値アーカイブ
- 圧縮アーカイブ
- メッセージアーカイブ

メインメモリアーカイブの同期は実行されません。

ユーザーアーカイブ

ユーザーアーカイブは両方のサーバーで同一構造で構築されている必要があります：

これから同期を行うユーザーアーカイブの設定は、プロパティおよびフィールドとレコード構造において等しくなければなりません。

注記

オンラインでの変更のロードを使用して実行できない設定データの変更の同期

アーカイブで削除したフィールドなど、ユーザーアーカイブ設定データの変更は、オンラインでの変更のロードでは冗長サーバーペアには転送できません。

ユーザー管理(ユーザー管理者)

ユーザー管理の変更は、自動的に同期されません。これは、WinCC UserAdminControl 経由のランタイムでの設定にも適用されます。

ユーザー管理の設定を変更する場合、以下のオプションがあります。

- エンジニアリングステーションで変更を設定する。変更を冗長サーバーに転送する。
- 両方の冗長サーバーで同じ変更を設定する。

下記も参照

WinCC の 2 重化 (ページ 120)

4.5.3 冗長サーバーを構成する方法

はじめに

WinCC エクスプローラで「冗長性」エディタを使って冗長サーバとアーカイブの同期を設定します。

必要条件

- 2 つの冗長サーバーを機能的に等しく設定する必要があります。

4.5 冗長性システムの構成

手順

1. WinCC エクスプローラで「冗長性」エディタを開きます。
[全般]タブに進みます。
[サーバー]フィールドは、WinCC 冗長性を設定するコンピュータの名前を含みます。

The screenshot shows the 'General' tab of the WinCC Redundancy Editor. It includes fields for 'Server' (WCC-DEV), 'Default Master' (unchecked), 'Redundant Partner Server' (SV_REDUND_07), and a 'Reference' button. The 'Local Computer Settings' section shows 'Network Adapter for Redundant Partner Connection' set to 'None'. The 'Redundant Partner Network Address' section has 'Dynamic' selected, with an IP address field and a 'Port' of 1765. The 'Serial Interface for Redundant Partner Connection' is set to 'COM1'. The 'Options' section has several checkboxes: 'Synchronize after partner server goes online' (checked), 'Synchronize after partner server goes online (Alarm Reset)' (checked), 'Synchronize Alarm Reset online' (checked), 'Synchronize after disconnection of connection' (unchecked), and 'Switch WinCC client when connection fails' (checked). At the bottom, there are checkboxes for 'Apply all specified options and synchronize with user-defined' (checked) and 'Execute redundancy' (checked).

2. WinCC 冗長性を設定および使用する場合、[冗長化の起動] チェックボックスを有効にします。
3. [デフォルトマスタ] オプションを有効にし、両方のサーバーが同時に起動されているときに、上記で入力したサーバーがマスタサーバーとして常に起動されるように指定します。このオプションを有効にしないと、選択したサーバーはスタンバイサーバーになります。

通知

「デフォルトマスタ」に指定できるのは1台の冗長サーバのみ

[デフォルトマスタ] オプションが2台の冗長サーバーのうち1台でのみ有効になっていることを確認してください。

さもないと、クライアントの冗長切り替え時に、不具合が発生する場合があります。

4. パートナーサーバーのコンピュータ名を入力し、[検索]をクリックします。

5. ネットワークアダプタによる冗長パートナーへの接続があるかどうかをモニタリングするステータスを指定します。
必要に応じて、固定ネットワークアドレスと冗長パートナーのポートを入力するには、[静的]オプションを選択します。
ネットワークアダプタによる接続は、シリアル接続よりも優先されます。
シリアル接続を使用する場合は、インターフェースを選択します。
これらの設定は[Simatic Shell]で設定することもできます。Windows エクスプローラから、[Simatic Shell]フォルダのショートカットメニューで[冗長設定]を選択します。
6. 対応するオプション設定を選択して、冗長性が復元された後、またはプロセス接続が中断された後に実行する同期を指定します。
 - パートナサーバーのオンライン復帰後のタグロギングの同期
 - パートナサーバーのオンライン復帰後のアラームロギングの同期
 - アラームロギングのオンライン同期。オペレータメッセージとタグ接続のないメッセージまたは内部メッセージタグありのメッセージは同期されます。
 - プロセスリンクが切断した後の同期(タグロギング+アラームロギング):サーバーと自動化システムとの間の切断が解消された後、プロセス接続の監視は自動アーカイブの同期を開始します。
 - プロセス接続エラーの場合の WinCC クライアント切り替え。サーバーに接続しているクライアントは、冗長パートナーサーバーに切り替わります。
シナリオについては、「プロセス接続エラー発生時のクライアントの切り替え (ページ 142)」で説明します。
7. [OK]をクリックして設定を保存します。
8. 設定を適用するには、必要に応じてランタイムを再起動します。
ランタイムを再起動しない場合、以下のオプションの変更のみが即座に有効になります。
 - パートナサーバーのオンライン復帰後のタグロギングの同期
 - パートナサーバーのオンライン復帰後のアラームロギングの同期
 - アラームロギングのオンライン同期。
 - プロセスリンクが切断した後の同期(タグロギング+アラームロギング)他のオプションの変更は、ランタイムを再起動した後にのみ有効です。

下記も参照

プロセス接続エラー発生時のクライアントの切り替え (ページ 142)

ユーザーアーカイブの同期の設定方法 (ページ 136)

WinCC の 2 重化 (ページ 120)

WinCC リダンダントシステムメッセージ (ページ 160)

冗長システムのセットアップ方法 (ページ 130)

故障シナリオ (ページ 146)

4.5 冗長性システムの構成

4.5.4 ユーザーアーカイブの同期の設定方法

概要

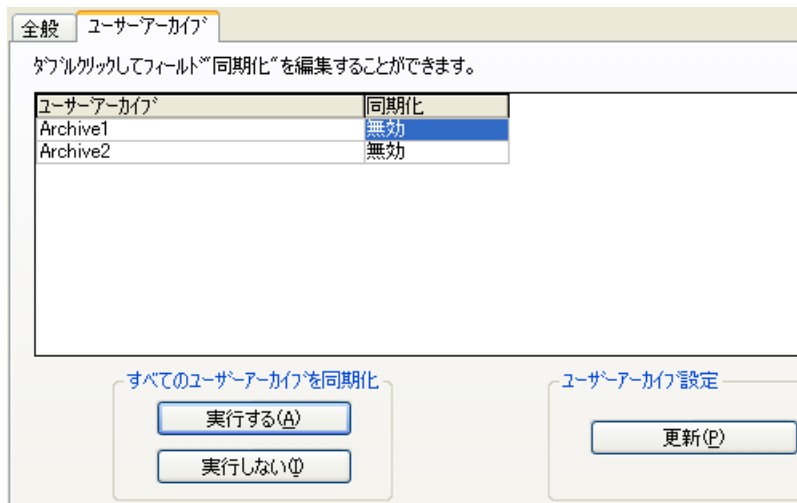
ユーザーアーカイブはオペレーション、独立したプログラムまたはオートメーションシステムで処理されます。冗長システムに対して、ユーザーアーカイブの自動同期を設定します。

必要条件

ユーザーアーカイブの設定は2台の冗長サーバーで同一である必要があります。これにはプロジェクトデублиケータを使用します。

手順

1. WinCC エクスプローラで「冗長性」エディタを開きます。「ユーザーアーカイブ」タブを開きます。



2. 「ユーザーアーカイブ」列には、設定されているすべてのユーザーアーカイブが行単位で表示されます。[同期] コラムをダブルクリックして、個々のユーザーアーカイブの同期を有効または無効にします。この設定は、両方のパートナーサーバーで同一でなければなりません。
3. [すべてのユーザーアーカイブの同期] フィールドの2つのボタンにより、表示されたすべてのユーザーアーカイブの同期を有効化または無効化できます。
4. 「冗長化」エディタを呼び出した後で「ユーザーアーカイブ」エディタでユーザーアーカイブの設定を変更しましたか?その場合、[更新] ボタンを押してユーザーアーカイブの現在の設定を適用します。
5. [OK]をクリックして設定を保存します。

注記

オンラインでの変更のロードを使用して実行できない設定データの変更の同期

アーカイブで削除したフィールドなど、ユーザーアーカイブ設定データの変更は、オンラインでの変更のロードでは冗長サーバーペアには転送できません。

注記

ユーザーアーカイブのアーカイブ同期の変更後にランタイムを再起動する

ユーザーアーカイブのアーカイブ同期の変更は、ランタイムを再起動した後にのみ有効になります。

ユーザーアーカイブの並行編集

レコードを冗長ユーザーアーカイブに同時に追加するときは、以下の項目に注意してください。

- 復帰後に同期が行われた場合、レコードは以前に故障したサーバーにのみ追加できます。そうしないと、スクリプトまたはユーザーアーカイブコントロールに、エラーメッセージが発生します。
- オンライン同期中であっても、レコードが冗長アーカイブで同期されるまで、多少時間がかかることがあります。

注記

サーバーコンピュータがどちらも機能しない場合、またはどちらもシャットダウンした場合、最後に使用したサーバーコンピュータを最初に起動する必要があります。さもないと、保存されていない変更が失われます。

下記も参照

冗長サーバーを構成する方法 (ページ 133)

WinCC の 2 重化 (ページ 120)

冗長システムのセットアップ方法 (ページ 130)

WinCC リダンダントシステムメッセージ (ページ 160)

故障シナリオ (ページ 146)

4.5.5 冗長サーバーのプロジェクトの複製方法

概要

2 台の冗長サーバーはハードウェアおよびソフトウェアに関して機能的に同一に設定する必要があります。

WinCC の設定の完了後および WinCC プロジェクトの各変更後、WinCC プロジェクトデュープリケータを使用して冗長パートナプロジェクトを生成します。

プロジェクトデュープリケータは以下を実行します。

- 画像、スクリプトおよびアーカイブなど、すべての関連するプロジェクトデータを冗長パートナにコピーする。
- コンピュータが WinCC 冗長性を使用するように設定されていない場合、対象コンピュータで必要なすべての設定を行う。

コンピュータ固有の設定は、後で手動で変更しなければなりません。

注記

プロジェクトを冗長サーバーに転送するために、Windows エクスプローラは使用できません。

SIMATIC Manager の[オンラインでの変更のロード]機能を使って小規模の変更を保存して、ランタイムでサーバーに転送することができます。

原理

プロジェクトデュープリケータで複製するプロジェクトを選択します。

プロジェクトの複製先であるコンピュータとフォルダを指定します。プロジェクトフォルダはターゲットフォルダ内に作成されます。

ローカルコンピュータでプロジェクトを複製することはできません。常にネットワーク上でアクセス権がある別のコンピュータのプロジェクトを複製します。

プロジェクトのステータスにより、設定データおよびランタイムデータを、選択したフォルダに複製することができます:

プロジェクトステータス	設定データ	ランタイムデータ
閉じたプロジェクト	+	+
開いている無効のプロジェクト	+	-
ランタイムのプロジェクト	+	-

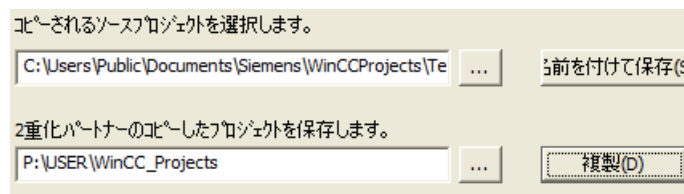
複製できるのはプロジェクト全体およびフォルダ構造全体のみです。どのデータおよびフォルダも複製操作から除外できません。

必要条件

- WinCC 冗長性オプションが両方のコンピュータにインストールされていること。
- 複製のターゲットフォルダがターゲットコンピュータに作成されており、アクセスが可能なこと。
- ターゲットフォルダへのアクセス権があること。
- ターゲットコンピュータのハードディスクに十分な空きスペースがあること。
- ターゲットコンピュータには正しい WinCC バージョンをインストールする必要があります。コンピュータを起動する必要があります。
- ターゲットコンピュータでランタイムが終了していること。
- ターゲットコンピュータでプロジェクトが閉じられていること。

手順

1. 「Siemens Automation」 Windows プログラムグループで、[プロジェクトデュプリケータ]エントリを選択します。
WinCC プロジェクトデュプリケータが開きます。



2. [コピーされるソースプロジェクトを選択]ボックスに複製するプロジェクトを入力します。パスおよびプロジェクトファイル<PROJECT>.MCP を直接入力するか、... ボタンをクリックして検索します。
3. [冗長化パートナー用の複製プロジェクトの保存場所]ボックスに、複製したプロジェクトを保存するパスを入力します。フォルダパスおよびプロジェクトファイル<PROJECT>.MCP を直接入力するか、... ボタンをクリックして検索します。

4.5 冗長性システムの構成

4. [複製]ボタンをクリックします。
[コピー]ウィンドウが開きます。複製時に、プロジェクトデュプリケータは、ファイルとフォルダを進捗バーとともに表示します。複製を中止するには、[キャンセル]ボタンを押します。
複製後、[プロジェクトデュプリケータに関する注意]ウィンドウが開きます。
WinCC に、確認する必要がある設定が表示されます。

注記

複製元のコンピュータで開いていた WinCC プロジェクトを複製すると、進捗バーは表示されません。

5. [閉じる]ボタンでプロジェクトデュプリケータを閉じます。
6. 複製したプロジェクトの設定を確認し、必要に応じて変更します。
7. 以下の事項を確認します。
 - コンピュータ名。
 - 冗長化エディタの設定。
 - 必要な場合、エディタの設定。

プロジェクトベースのアクセス保護があるプロジェクトの複製

プロジェクトベースのアクセス保護がある WinCC プロジェクトを冗長サーバーに転送するためには、SIMATIC STEP 7 がインストールされていなければなりません。

[WinCC プロジェクトデュプリケータ]ダイアログの[複製]ボタンをクリックすると、STEP 7 プロジェクトのパスワードを入力しなければなりません。

SIMATIC STEP 7 がインストールされていないか、またはパスワードが間違っていると、プロジェクトデュプリケータはエラーメッセージを出して中止します。

4.5.6 ランタイムでの冗長プロジェクトの複製方法

概要

冗長プロジェクトを編集すると、冗長化サーバーの動作中のプロジェクトを更新することもできます。

[オンラインでの変更のロード]機能で小規模な変更を保存して、それらをランタイムでサーバーに転送することができます。「オンラインでの変更のロード」の章にある文書も参照してください。

プロジェクトデuplicータを使用した複製

一部の設定は、[オンラインでの変更のロード]機能によって保存できません。この場合は、プロジェクトデuplicータを使用して、プロジェクトの複製を冗長化サーバーに生成する必要があります。

注記

冗長化システムではない場合

通常の操作中の変更では、パートナーサーバーの 1 つを停止する必要があります。この間、冗長性を使用できません。

必要条件

- ターゲットフォルダが作成されていること。
- ターゲットフォルダへのアクセス権があること。
- コピーしたプロジェクトを保存する冗長化サーバーに、十分なハードディスク空きスペースがあること。

手順

このセクションでは、サーバー 1 とサーバー 2 の 2 台のサーバーを備えた冗長システムを例として、この機能を使用する方法を説明します。

1. 冗長サーバー 1 で、ランタイムを終了してプロジェクトを閉じます。
2. ランタイムでサーバー 2 に対する設定変更を行い、変更を保存します。
3. サーバー 2 でプロジェクトデuplicータを起動します。
4. [複製]ボタンを使って、「1」で無効化されたプロジェクトのターゲットフォルダにサーバー 1 のプロジェクトを複製して、上書きします。
5. サーバー 1 でプロジェクトを開きます。
6. 設定をチェックします。
7. ランタイムを起動し、冗長化の同期化を待ちます。

4.6 WinCC 冗長性のシナリオ

4.6.1 プロセス接続エラー発生時のクライアントの切り替え

概要

冗長システムは、2 台の機能的に等しいサーバーで構成されます。一方のサーバーが「マスタ」サーバーで、もう一方のサーバーが冗長パートナーサーバーです。

影響を受けない状態のサーバーには、以下のステータスがあります。

- マスターサーバーには、「マスタ」ステータスがあります。
- 冗長サーバーには、「スタンバイ」ステータスがあります。

クライアントはそれぞれの優先サーバます。優先サーバーが指定されていない場合、マスタサーバーに接続されます。

ランタイムで両方のサーバーが稼働すると、プロセス接続モニタが起動します。「マスタ」サーバーおよび冗長パートナーサーバーの論理接続の不具合数を、WinCC 冗長化が周期的に測定します。

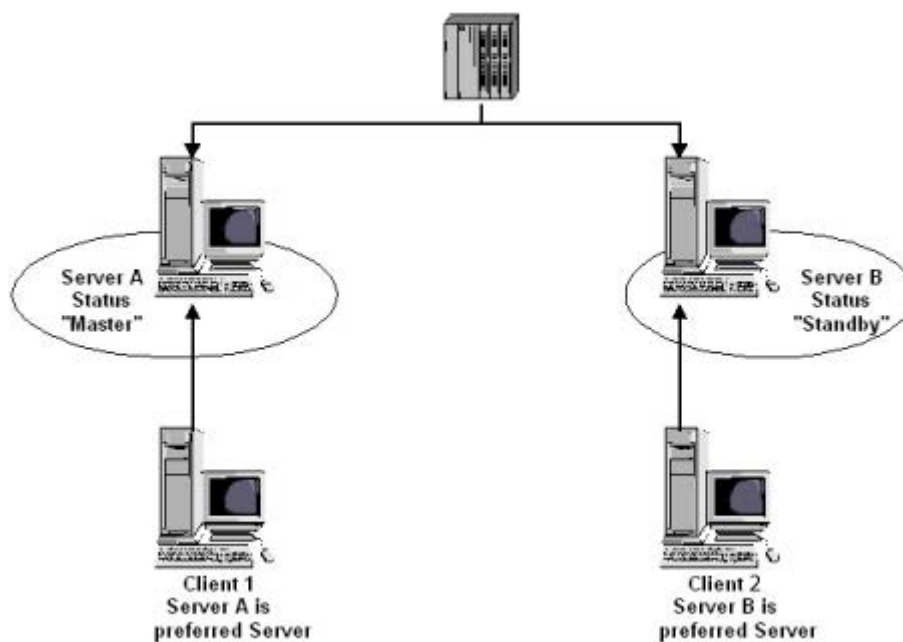
「マスタ」サーバーの論理接続不具合数が冗長パートナーサーバーよりも多い場合、サーバーのステータスは「@RedundantServerState」システムタグで「エラー」になります。クライアントの接続は、「マスタ」ステータスを持つ冗長パートナーサーバーに切り替えられます。

正常な作動状態

システムは、以下のコンピュータで構成されます。

- 冗長サーバー A
- 冗長サーバー B

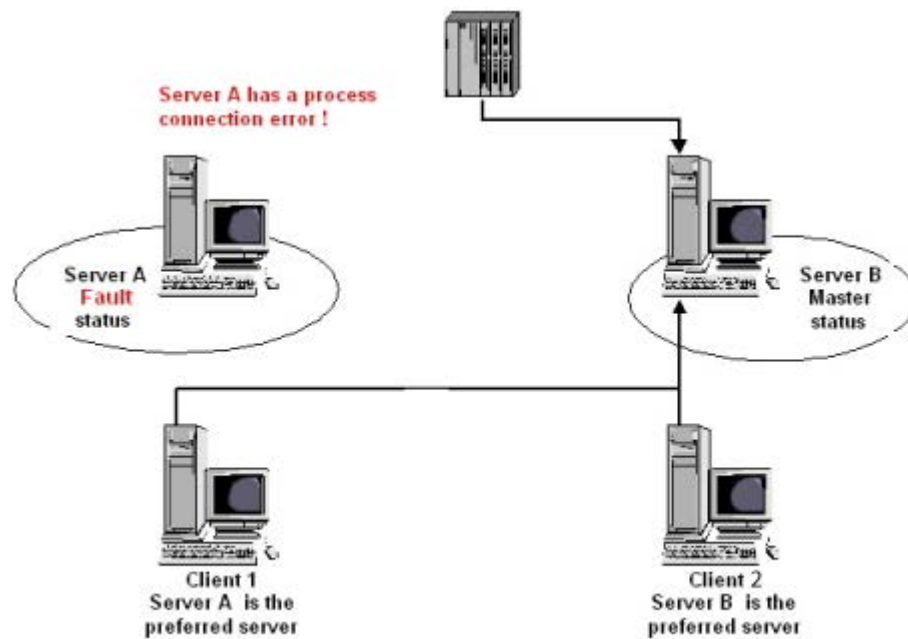
- 優先サーバー A があるクライアント 1
- 優先サーバー B があるクライアント 2



4.6 WinCC 冗長性のシナリオ

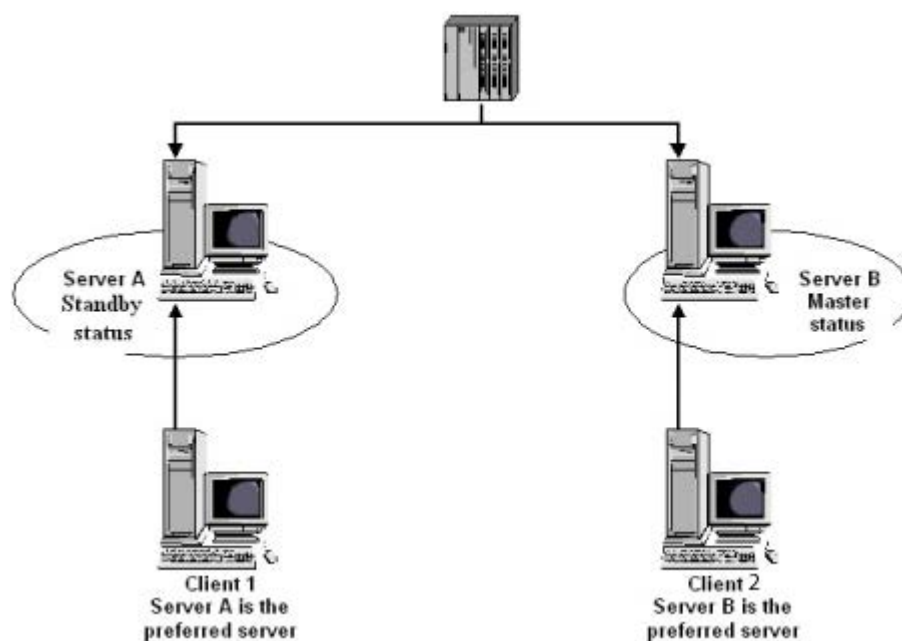
サーバー A のプロセス接続エラー

サーバー A にプロセスリンクエラーがあります。サーバー B にはエラーがありません。サーバー A の論理接続の不具合の数は、サーバー B より大です。従って、サーバー A は「エラー」ステータスを受け取ります。この結果、クライアント 1 は、冗長サーバー B に切り替えられます。



プロセスリンクエラーの終了

サーバー A のプロセスリンクエラーが解消されると、サーバー A は「スタンバイ」ステータスになります。サーバーが優先サーバーとして入力されたため、クライアント 1 はサーバー A に切り替わります。クライアント 2 はその優先サーバーであるサーバー B に接続されたままです。



注記

OPC カプラはモニタされません。従って、OPC カプラのエラーの場合には、クライアントの切り替えは起こりません。

下記も参照

故障シナリオ (ページ 146)

冗長システムのセットアップ方法 (ページ 130)

WinCC の 2 重化 (ページ 120)

冗長サーバーを構成する方法 (ページ 133)

ユーザーアーカイブの同期の設定方法 (ページ 136)

4.6 WinCC 冗長性のシナリオ

4.6.2 故障シナリオ

4.6.2.1 故障シナリオ

はじめに

実際に発生する一部のエラーを例に挙げ、WinCC 冗長性がどのように機能するかについて説明します。

1. シナリオ 1：ランタイムにないサーバーコンピュータのプロジェクト (ページ 147)
2. シナリオ 2 パートナサーバーへの接続エラー (ページ 149)
3. シナリオ 3 クライアントへのネットワーク接続エラー (ページ 151)
4. シナリオ 4 プロセス接続の不良 (ページ 152)
5. シナリオ 5 ソフトウェアエラー (ページ 153)

WinCC 冗長化が、現在のエラーそのものを認識するか、またはエラーメッセージに以下のアクションで反応します。

- イベントの回数を保存。
- アーカイブの同期。
- 「マスタ」および「スタンバイ」識別子の変更。
- クライアントの切り替え。
- メッセージのトリガ。

サーバー PC の起動

サーバー PC が起動しているとき、冗長構成要素は、パートナーサーバーがすでに作動しているかどうかを確認します。

- パートナーサーバーが既に起動されている場合、サーバーコンピュータに「スタンバイ」ステータスが設定されます。
- 起動中にパートナーサーバーが有効でない場合、サーバーコンピュータに「マスタ」ステータスが設定されます。

WinCC 冗長化システムタグ

サーバーコンピュータのステータスが、「@RM_MASTER」システムタグに保存されます。

サーバーコンピュータのステータス	「@RM_MASTER」ステータス
マスタ	1
スタンバイ	0

「@RM_MASTER_NAME」タグには、「マスタ」ステータスを持つサーバーシステムの名前(例、「サーバー 1」)が含まれます。

「@RedundantServerState」タグには各冗長サーバの冗長ステータス(例、「スタンバイ」)が表示されます。

冗長化では上記のタグのみを設定します。両方のサーバーは常に完全に同等です。

スクリプトまたは他のアプリケーションが、これらのタグを評価できます。

「@RM_MASTER」タグだけを変更できます。

システムタグの概要は WinCC リダンダントシステムタグ(ページ 154)に記載されています。

ステータス情報の交換

冗長性ステータスは、個別の接続でコントロールされます。接続は、以下のように確立できます。

- ネットワークアダプターを使用
- シリアルインターフェースを使用

ネットワークアダプタによる接続は、シリアル接続よりも優先されます。

注記

アーカイブ同期は端末バス経由で実行されます。アーカイブ同期はステータス接続では実行されません。

4.6.2.2 シナリオ 1：ランタイムにないサーバーコンピュータのプロジェクト

はじめに

プロジェクトがサーバー 2 で無効の場合、このシナリオは WinCC 冗長性の動作を示します。

4.6 WinCC 冗長性のシナリオ

以下のアクションがトリガされます。

- サーバー 1 にサーバー 2 のダウンタイム(日付と時刻)が保存されます。
- サーバー 1 は、サーバー 2 の故障をシステムメッセージによってレポートします。
- サーバー 1 がスタンバイサーバーの場合、サーバー 1 はマスタサーバーの役割を引き継ぎます。「@RM_MASTER」タグが設定され、「@RM_MASTER_NAME」および「@RedundantServerState」タグは変更されます。
- サーバー 2 に接続されているクライアントが、サーバー 1 に切り替わります。

サーバー 2 のオンラインへの復帰

ダウン時間は、サーバー 2 のアーカイブにギャップがあることを意味します。このギャップは、以下の手段によって埋めることができます。

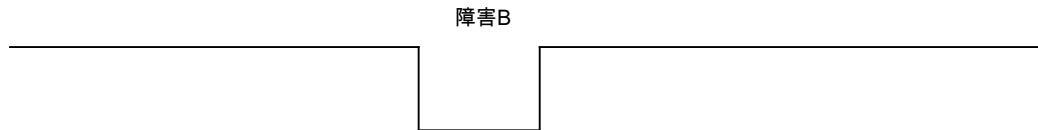
- サーバー 1 が、サーバー 2 の復帰時間(日付と時刻)を保存します。
- サーバー 1 は、サーバー 2 の復帰をシステムメッセージによってレポートします。
- サーバー 1 からの以下のアーカイブの冗長同期が、サーバー 2 で実行されます。
 - メッセージアーカイブ
 - プロセスデータアーカイブ
 - ユーザーアーカイブ
- サーバ 1「@RM_MASTER」が設定されたまま、サーバ 2「@RM_MASTER」がリセットされます。
「@RM_MASTER_NAME」および「@RedundantServerState」はどちらのサーバーでも変更されません。
- 優先サーバーとしてサーバー 2 を構成されているクライアントは、サーバー 2 に切り替えられます。

オンライン同期に比べて、サーバー故障発生後のアーカイブ同期には時間がかかる場合があります。同期の時間は、同期すべきレコードの数や、コンピュータおよびネットワークの負荷によります。

交互に発生するサーバー故障

2 台のサーバー間で故障が交互に発生する場合、サーバーの同期は交互に行われます。同期後、すべてのデータは両方のアーカイブで使用可能になります。

Server1:



Server2:



同期が設定されると、常に同期が実行されます。

障害 A

サーバー 1 からサーバー 2 へすべての値が転送されます。

障害 B

サーバー 2 からサーバー 1 へすべての値が転送されます。

障害 C

サーバー 1 からサーバー 2 へすべての値が転送されます。

これらすべてのプロセスは、同時に行われている下位のオートメーションシステムから取得したプロセス値のアーカイブ処理およびメッセージのアーカイブ処理に関わりなく、バックグラウンドで自動的に行われます。

4.6.2.3 シナリオ 2 パートナサーバーへの接続エラー

はじめに

このシナリオでは、パートナーサーバーへの接続障害の場合の冗長性の動作を示します。

このイベントの発生に先立って、両方のサーバーは障害無くランタイムで動作しているとします。

たとえば、サーバー 1 でネットワーク接続が引っ張られると、既述の接続障害が発生します。

初期条件 1

接続エラー時、サーバー 1 がマスタサーバー、サーバー 2 がスタンバイサーバーです。

接続エラーが発生する

接続に失敗すると、以下の動作がトリガされます。

- サーバー 2 がマスタサーバーになり、障害の時刻(日付と時刻)を保存します。
- サーバー 2 がは、パートナーサーバーに故障してサーバー 2 が「マスタ」サーバーに切り替わったというシステムメッセージを表示します。
- タグ「@RM_MASTER」、「@RM_MASTER_NAME」および「@RedundantServerState」が両方のサーバーに適用されます。

接続が回復する

接続障害中は、アラームロギングのメッセージおよびユーザーアーカイブを同期化できません。

以下の測定が実施されます。

- マスタサーバー 2 が復帰時刻を保存。
- サーバー 2 が、パートナーサーバーの復帰を、システムメッセージで表示します。
- マスタサーバーからスタンバイサーバーへの冗長の同期。
- アラームロギングのオンライン同期で、以下の事項がサーバー 1 からサーバー 2 にレポートされ、システムメッセージとしてサーバー 1 に表示されます。
 - 冗長操作でエラーが発生した。
 - サーバー 1 が「スタンバイ」ステータスに切り替わった。
 - サーバー 1 の復帰。
- 両方のサーバーの「@RM_MASTER」、「@RM_MASTER_NAME」および「@RedundantServerState」タグは変更されません。

初期条件 2

接続エラー時、サーバー 1 がスタンバイサーバー、サーバー 2 がマスタサーバーです。

接続エラーが発生する

接続に失敗すると、以下の動作がトリガされます。

- サーバー 2 はマスタサーバーのままで、障害の時刻(日付と時刻)を保存します。
- サーバー 2 が、パートナーサーバーの故障を、システムメッセージで表示します。

- サーバー 1 が「エラー」内部ステータスに変わります。
クライアントのサーバー 1 が優先サーバーとして、サーバー 2 に切り替わります。
- 両方のサーバーの「@RM_MASTER」、「@RM_MASTER_NAME」および「@RedundantServerState」タグは変更されません。

接続が回復する

接続障害中は、アラームロギングのメッセージおよびユーザーアーカイブを同期化できません。

以下の測定が実施されます。

- サーバー 2 が復帰時刻を保存します。
- サーバー 2 が、パートナーサーバーの復帰を、システムメッセージで表示します。
- マスタサーバーからスタンバイサーバーへの冗長の同期。
- アラームロギングのオンライン同期で、以下の事項がサーバー 1 からサーバー 2 にレポートされ、システムメッセージとしてサーバー 1 に表示されます。
 - 冗長操作でエラーが発生した。
 - サーバー 1 の復帰。
- 両方のサーバーの「@RM_MASTER」、「@RM_MASTER_NAME」および「@RedundantServerState」タグは変更されません。

4.6.2.4 シナリオ 3 クライアントへのネットワーク接続エラー

はじめに

このシナリオでは、サーバー 2 とサーバー 2 に属する「CL5」クライアントの間のネットワーク接続に、障害があります。サーバー 1 がマスタサーバーであること。

以下のアクションがトリガされます。

- クライアント「CL5」は、障害が発生したサーバー 2 から作動しているサーバー 1 に、自動的に切り替えられます。

4.6 WinCC 冗長性のシナリオ

クライアントへのネットワーク障害の終了

ネットワーク障害の終了時に以下のアクションがトリガされます。

- 故障する前にサーバー 1 がすでにマスタサーバーであった場合は、両方のサーバーの「@RM_MASTER」、「@RM_MASTER_NAME」および「@RedundantServerState」タグは変更されません。
- クライアント「CL5」は再び優先サーバーのサーバー 2 に切り替わります。

下記も参照

WinCC リダンダントシステムタグ (ページ 154)

シナリオ 2 パートナサーバーへの接続エラー (ページ 149)

シナリオ 1: ランタイムにないサーバーコンピュータのプロジェクト (ページ 147)

故障シナリオ (ページ 146)

シナリオ 4 プロセス接続の不良 (ページ 152)

4.6.2.5 シナリオ 4 プロセス接続の不良

はじめに

このシナリオでは、オートメーションシステムへのネットワーク接続が遮断されたため、サーバー 2 でプロセスリンクに障害が発生します。

オートメーションシステムへの接続の障害

オートメーションシステムへの接続の障害は、サーバーへの接続が失敗した場合に、WinCC 冗長性でのみ認識されます。

2 台のサーバーへのオートメーションシステムの接続の中断は冗長性という意味ではエラーではなく、例えばオートメーションシステムの故障です。

エラーへの反応

WinCC が障害を認識すると、以下のアクションがトリガされます。

- プロセスリンクの障害がサーバー 2 でレポートされます。
- サーバー 1 は、パートナーサーバー 2 が故障したというメッセージを受け取ります。
- サーバー 1 は、サーバー 2 のエラーの時刻(日付と時刻)を保存します。

- 「冗長性」エディタで「プロセス接続内の障害によるクライアントの変更」オプションを設定すると、このサーバーに接続しているクライアントがパートナーサーバーに切り替わります。
- 「@RM_MASTER」タグは、サーバー 1 では「マスタ」に、サーバー 2 では「スタンバイ」に設定されます。「@RM_MASTER_NAME」タグおよび「RedundantServerState」タグが適用されます。サーバー 2 で、「@RedundantServerState」タグが「エラー」に設定されます。

サーバー 2 でのプロセスリンクエラーの終了

プロセス接続モニタが作動すると、サーバー 2 のアーカイブのギャップが、以下の手段によって埋められます。

- サーバー 1 はサーバー 2 の復帰時刻を保存します。
- サーバー 1 のプロセス接続には障害がないので、サーバー 1 からサーバー 2 へ冗長の同期が実行されます。障害の発生していないオートメーションシステムのデータを含め、すべてのオートメーションシステムのデータを同期します。
- サーバー 2 では、「@RedundantServerState」タグが「エラー」から「スタンバイ」に変更されます。
- サーバー 2 で行うプロセスリンクエラーの修正が、システムメッセージによって通知されます。

4.6.2.6 シナリオ 5 ソフトウェアエラー

はじめに

このシナリオでは、サーバー 2 でモニタされているソフトウェアにエラーが発生します。この障害の時点で、サーバー 2 が「マスタ」ステータスで、サーバー 1 が「スタンバイ」ステータスです。複数のクライアントが、両方のサーバーに接続されています。

「Application Health Check」ファンクションが、WinCC ソフトウェアのエラーを検出した場合、以下のアクションが起動されます。

- 「Application Health Check」は WinCC 冗長性のエラーをレポートします。サーバー 2 のステータスが、「@RedundantServerState」タグで「エラー」に設定されます。「@RM_Master」タグが「スタンバイ」に設定されます。
- サーバー 1 で「@RM_Master」が「マスタ」に設定されます。「@RM_MASTER_NAME」タグおよび「RedundantServerState」タグが適用されます。

4.6 WinCC 冗長性のシナリオ

- サーバー 2 に接続されているクライアントが、サーバー 1 に切り替わります。
- アラームサーバー自身がエラーを起こしていない場合、プロセスコントロールメッセージが、ソフトウェアのエラーをユーザーに通知します。

サーバー 2 でのソフトウェアエラーの終了時の測定

影響を受けるサーバー 2 プロジェクトを無効にします。サーバー 2 を再起動します。プロジェクトがサーバー 2 で実行されると、アーカイブが自動的に同期されます。

- サーバー 2 で「@RedundantServerState」が「スタンバイ」に設定されます。サーバー 1 は「マスタ」のままです。
- サーバー 1 が、サーバー 2 の復帰時間(日付と時刻)を保存します。
- このサーバーへの再接続が可能になります。アーカイブの同期化は、サーバー 2 のソフトウェアエラーが検出された時まで遡った時点までのみ実行されます。

下記も参照

WinCC リダンダントシステムタグ (ページ 154)

シナリオ 3 クライアントへのネットワーク接続エラー (ページ 151)

シナリオ 2 パートナサーバーへの接続エラー (ページ 149)

シナリオ 1: ランタイムにないサーバーコンピュータのプロジェクト (ページ 147)

故障シナリオ (ページ 146)

シナリオ 4 プロセス接続の不良 (ページ 152)

4.6.3 WinCC リダンダントシステムタグ

WinCC 冗長性システムタグ

システムタグは、冗長性ステータスの診断のために、内部の[Performance]および[Redundancy]タググループで作成されます。

冗長性タグ

WinCC エクスプローラで[冗長性]エディタを開いてから[OK]を使用してこのエディタを再度閉じた場合、WinCC 冗長性によりシステムタグが作成されます。

「@RM_MASTER」および「@RM_MASTER_NAME」システムタグは、2 台の冗長化サーバーのマスタ/スタンバイ制御時およびクライアント変更のために使用されます。

他のアプリケーションまたはスクリプトを介してシステムタグを読むことができます。

「@RM_MASTER」タグのみを変更できます。

パフォーマンスタグ

パフォーマンスタグ「@PRF_REDUNDANCY_...」は、冗長化システムの状態を表しています。

WinCC プロジェクトが作成されると、システムタグが[Performance]タググループに作成されます。

追加情報:

- [WinCC での作業] > [プロジェクトでの作業] > [ランタイムの設定] > [パフォーマンスタグでのシステム診断]

冗長性タグの概要

システムタグ	意味
@LocalMachineName	ローカルのコンピュータ名が含まれます。
@RedundantServerState	サーバーの冗長性ステータス: 0:未定義のステータスまたは初期値 1:サーバーはプライマリサーバー（マスタ）です 2:サーバーはスタンバイです 3:サーバーは「障害」ステータスです 4:サーバーはスタンドアローンまたは冗長化操作なし
@RM_MASTER	タグ値=1:プライマリサーバーであることを識別します。 サーバーがスタンバイサーバーになると、「@RM_MASTER」が[0]にリセットされます。 スクリプトなどを介して、タグの値を変更できます。
@RM_MASTER_NAME	プライマリサーバーの名前
@RM_SERVER_NAME	クライアントが接続されているサーバーの名前

4.6 WinCC 冗長性のシナリオ

システムタグ	意味
@RM_UA_ONL_"Archiv name"	診断に使用されます: <ul style="list-style-type: none">• 1:ユーザーアーカイブが変更されました。• 0:ユーザーアーカイブのオンライン同期が完了しています。 対応するアーカイブ名を持つ個別のタグが、各ユーザーアーカイブに挿入されます。
@RM_Offline_UA_Name	診断に使用。 このタグには、同期したばかりのユーザーアーカイブの名前が含まれます。

パフォーマンスタグの概要

パフォーマンスタグ「@PRF_REDUNDANCY_...」は、[RedundancyControl]診断ツールでも評価される冗長化サーバーの状態を示しています。

システムタグ	説明
@PRF_REDUNDANCY_IS_SYNCHRONIZED	<p>同期ステータス:</p> <ul style="list-style-type: none">0:冗長化アプリケーションは同期されません。1:すべてのアプリケーションの冗長性同期が完了しています。 <p>ステータスは、WinCC および冗長性のためにログオンされている他のアプリケーション(例、SIMATIC BATCH)の影響を受けることがあります。</p>

4.6 WinCC 冗長性のシナリオ

システムタグ	説明
@PRF_REDUNDANCY_VALIDATION	<p>サーバーの評価ポイント。検証値はプライマリサーバーとなるサーバーを決定します。</p> <p>検証値は、例えば、接続およびランタイムステータスに依存しています。</p> <p>正しく設定されている冗長性を使用していると、この検証値は両方の冗長化サーバーで同じになります。</p> <p>検証値が異なる場合、より高い値を持つサーバーがプライマリサーバーとなります。</p> <p>標準値:</p> <ul style="list-style-type: none"> 37:サーバーステータスは良好です。 <ul style="list-style-type: none"> ランタイムがアクティブです。 シリアルインターフェース経由の冗長化接続 35:サーバーステータスは良好です。 <ul style="list-style-type: none"> ランタイムがアクティブです。 LAN 経由の冗長化接続 <35:サーバーには内部「障害」ステータスがあります。サーバーの接続ステータスまたはステータスを確認します。「障害」ステータスは、重大な操作状態の場合に設定されます。例えば、サーバーアプリケーションが応答しなくなったときなどです。 <p>1 台のサーバーが「障害」ステータスになっている場合、パートナーサーバーがプライマリサーバーとなります。</p> <p>サンプル計算:</p> <ul style="list-style-type: none"> ランタイムがサーバー上で無効になっている場合、検証値は 4 ポイント低下します。 端子バスに到達できない場合、検証値は 20 ポイント低下します。
@PRF_REDUNDANCY_PARTNER_VALIDATION	<p>冗長化パートナーサーバーの評価ポイント</p> <p>正しく設定されている冗長性を使用していると、この値は両方の冗長化サーバーで同じになります。</p>

システムタグ	説明
@PRF_REDUNDANCY_AS_COUNT	<p>サーバーの AS 接続の数</p> <p>正しく設定されている冗長性を使用していると、この値は両方の冗長化サーバーで同じになります。</p> <p>次の条件で冗長性の切り換えが引き起こされます:</p> <ul style="list-style-type: none"> 冗長化サーバーの検証値が同じである。 AS 接続の数が異なる。 <p>この場合、より多くの AS 接続数のあるサーバーがプライマリサーバーとなります。</p>
@PRF_REDUNDANCY_PARTNER_AS_COUNT	<p>冗長化パートナーサーバーの AS 接続の数</p> <p>正しく設定されている冗長性を使用していると、この値は両方の冗長化サーバーで同じになります。</p>
@PRF_REDUNDANCY_CURRENT_STATE	<p>サーバーの冗長性ステータス:</p> <ul style="list-style-type: none"> 0:未定義のステータス 1:サーバーはプライマリサーバーです 2:サーバーはスタンバイです 3:サーバーは「障害」ステータスです 4:サーバーはスタンドアローンまたは冗長化操作なし
@PRF_REDUNDANCY_PARTNER_CURRENT_STATE	冗長化パートナーサーバーの冗長性ステータス
@PRF_REDUNDANCY_FAULT_POSTPONED	<p>タグ値=1:サーバーには「FAULT_POSTPONED」ステータスがあります。</p> <p>ローカルサーバーの内部状態が「障害」ですが、パートナーサーバーを「マスタ」状態と見なすことができません。冗長性のフェイルオーバーはできません。原因は、例えば、冗長性同期の実行である可能性があります。</p> <p>冗長性切り換えの条件に適合するとすぐに、サーバーは「障害」ステータスに変わります。</p> <p>「@PRF_REDUNDANCY_CURRENT_STATE」タグは、値「3」と仮定します。</p>
@PRF_REDUNDANCY_PARTNER_FAULT_POSTPONED	タグ値=1:冗長化パートナーサーバーには、「FAULT_POSTPONED」ステータスがあります。
@PRF_REDUNDANCY_SWITCHOVER_COUNT	<p>ランタイムの起動後または「@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET」経由で最後にリセットされてからの冗長性切り換えの数。</p>

4.6 WinCC 冗長性のシナリオ

システムタグ	説明
@PRF_REDUNDANCY_SWITCHOVER_COUNT_PERIOD	定義された期間の冗長性切り換えの数 初期設定: <ul style="list-style-type: none"> 時間期間:1 暦日 値は、毎日午前 0:00 にリセットされます。
@PRF_REDUNDANCY_SWITCHOVER_COUNT_RESET	リセットタグが、次のパフォーマンスタグの値をリセットします。 <ul style="list-style-type: none"> @PRF_REDUNDANCY_SWITCHOVER_COUNT

[RedundancyControl]診断ツール

包括的な冗長性診断のため、[RedundancyControl]診断ツールを使用できます。

追加情報については、アプリケーション例「WinCC V7.x および WinCC Professional での冗長性」で参照できます:

- <https://support.industry.siemens.com/cs/ww/en/view/109772627> (<https://support.industry.siemens.com/cs/ww/en/view/109772627>)

下記も参照

WinCC リダンダントシステムメッセージ (ページ 160)

<https://support.industry.siemens.com/cs/ww/en/view/109772627> (<https://support.industry.siemens.com/cs/ww/en/view/109772627>)

4.6.4 WinCC リダンダントシステムメッセージ**概要**

WinCC 冗長性は一連のシステムメッセージを提供します。システムメッセージを使用するには、アラームロギングエディタの [オプション] メニューで [WinCC システムメッセージ...] コマンドを選択する必要があります。

以下のシステムメッセージは WinCC 冗長性から出力させることができます。

メッセージ No.	WinCC メッセージテキスト
1012200	REDRT: パートナーステーション故障。 パートナーサーバーで WinCC が終了しました。
1012201	REDRT: パートナーステーション再起動。 パートナーサーバーで WinCC が再起動しました。
1012202	REDRT: プロジェクトが同一機能を有していない。
1012203	REDRT: アーカイブ同期化誤動作しました。
1012204	REDRT: 冗長性で内部エラーが発生しました。
1012205	REDRT: パートナーへの接続エラー。 パートナーサーバーへの接続が切断されました。
1012206	REDRT: パートナーへの接続確立。 パートナーサーバーへの接続が確立されました。
1012207	REDRT: パートナーサーバーの WinCC が実行されていません。 実行中に WinCC が起動していないことが判明しました。
1012208	REDRT: アーカイブの同期開始。 このメッセージはアーカイブの同期開始時に表示されます。
1012209	REDRT: アーカイブの同期終了。 このメッセージはアーカイブの同期終了時に表示されます。
1012216	REDRT: 同期が中断された。 追加の障害により同期が中断された。
1012217	REDRT: パートナーサーバープロジェクトが実行されていません。 実行時に WinCC がパートナーサーバーまたはランタイムで実行されていないことが判明しました。
1012218	SWITCH: クライアントが自動で切り替えられました。 クライアントがパートナーサーバーに自動で切り替えられました。
1012219	SWITCH: クライアントが手動で切り替えられました。 クライアントがパートナーサーバーに手動で切り替えられました。
1012220	REDRT: 同期の準備ができていないユーザーアーカイブがあります。 パートナーサーバーのアーカイブ構造が 1 つ以上のアーカイブと異なるか、パートナーサーバー上で同期が実行されていないため、ローカルで構成されたユーザーアーカイブ同期の準備ができていないものがあります。

4.6 WinCC 冗長性のシナリオ

メッセージ No.	WinCC メッセージテキスト
1012221	REDRT:すべてのユーザーアーカイブに同期の準備ができています。 ローカルで構成されたユーザーアーカイブに同期の準備ができており、 パートナーサーバーのアーカイブ構造が一致しています。
1012226	REDRT:パートナーサーバープロジェクトが有効です。 起動中に WinCC がパートナーサーバー上で実行されていることが認識 されました。
1012227	REDRT:エラー - パートナーコンピュータはサーバーではありません 起動中に、構成されているパートナーコンピュータがサーバーではな いことが認識されました。
1012240	REDRT:<アプリケーション名>中の<エラー説明>エラーは切り替えをト リガします。 記名されたアプリケーションでのエラーにより、アプリケーションの ヘルスチェックによってスイッチがトリガされました。
1012241	REDRT:<ステータスの説明>ステータスに切り替えます メッセージはステータスの変更を示します。
1012244	REDRT:アラームロギングのオンライン同期中に過負荷が発生しました 同期するメッセージが数が多すぎます。
1012245	REDRT:シリアル接続障害
1012246	REDRT:シリアル接続が再確立しました。
1012247	REDRT:<メッセージが生成されたコンピュータ名>OS サーバー(マス タ)<コンピュータ名>OS サーバー(スタンバイ)<コンピュータ名>冗長 エラー 障害のシナリオにより、マスタサーバーおよびスタンバイサーバー、 またはこの 2 台のいずれかが、冗長エラーを送信します。冗長化が損 なわれる可能性があります。
1012248	REDRT:OS サーバー(予備)の冗長性が再確立されました
1012349	REDRT:ネットワークカード(MAC)アドレスの接続が失われました 冗長 LAN を使用したパートナーサーバーへの接続が中断または切断さ れました。
1012350	REDRT:ネットワークカード(MAC)アドレスの接続が再確立しました 冗長 LAN を使用したパートナーサーバーとの接続が再構築されました。
1012351	REDRT:冗長制御:システムブロックが検出されました。障害ステータス に切り替えます。
1012352	REDRT:冗長制御:システムブロックが検出されました。できるだけ早く コンピュータを再起動します。

メッセージ No.	WinCC メッセージテキスト
1012354	冗長性のコントロール:故障状況が変化しましたが、サーバーの隔離が実行されていません
1012355	冗長性のコントロール:故障状況が変化しましたが、サーバーの隔離が @1%s@によりロックされています。原因: @2%s@
1012356	冗長性のコントロール:故障状況が変化しました => サーバーが隔離されました
1012357	冗長性のコントロール:故障状況が変化しましたが、自動再起動が実行されていません
1012358	冗長性のコントロール:故障状況が変化しましたが、ネットワークアダプタが外され、DHCP が有効なため、自動再起動はロックされています
1012359	冗長性のコントロール:コンピュータの再起動が @1%s@によりロックされています。原因: @2%s@
1012360	冗長性のコントロール:@1%s@ 秒未満が期限切れになった状態で最後の再起動が実行されたため、コンピュータの再起動がキャンセルされました
1012361	冗長性のコントロール:@1%s@が再起動してから@2%s@秒間再起動が許可されていないので、コンピュータの再起動がキャンセルされました
1012362	冗長性のコントロール:@1%s@秒でコンピュータを再起動
1012700	自己診断:ノード@10%s@の値@7%s@が無効です。
1012701	自己診断:ノード@10%s@の値@7%s@が高域エラー限界を超えました。
1012702	自己診断:ノード@10%s@の値@7%s@が低域エラー限界未満になりました。
1012703	自己診断: @100%s@:ステーション@10%s@の値@7%s@が警告制限高域に違反しました。
1012704	自己診断: @100%s@:ステーション@10%s@の値@7%s@が低域エラー限界未満になりました。
1012705	自己診断: @100%s@:ノード@10%s@の値@7%s@はエラー限界を超えていません。
1012706	自己診断: @100%s@:ノード@10%s@の値@7%s@が OK です。
1012707	自己診断: @100%s@:ノード@10%s@が@2%s@を発生させます。

4.6 WinCC 冗長性のシナリオ

下記も参照

WinCC リダンダントシステムタグ (ページ 154)

プロセス通信

5.1 通信の基本

はじめに

通信は、2つの通信パートナー間のデータの交換として定義されます。

通信

通信パートナーは、他の通信パートナーと通信できる位置にあり、データを交換できるネットワークの任意の構成要素である可能性があります。WinCCでは、これらは、オートメーションシステム(AS)の中央モジュールと通信モジュールであり、PCの通信プロセッサであることがあります。

通信パートナー間で転送されるデータは、さまざまな目的に使用できます。WinCCの場合、以下があります。

- プロセスのコントロール
- プロセスからのデータの呼び出し
- プロセスにおける予期されていない状態の通知
- プロセスデータのアーカイブ

5.2 接続設定の基本規則

取得サイクルと更新時間

設定ソフトウェアで定義されるタグの取得サイクルは、達成できる更新時間の主要な要因です。

更新時間は、取得サイクル、転送時間および処理時間の合計です。

更新時間を最適にするには、設定時に以下の点を考慮に入れます。

- データ領域の最大および最小サイズを最適化します。
- 同じ類のデータ領域を、1つのグループとして定義します。複数の小さな領域の代わりに1つの大きな領域を設定すると、更新時間が改善されます。
- 取得サイクルが短すぎると、パフォーマンスが低下します。取得サイクルは、プロセス値の変更速度に合わせて設定します。たとえば、オーブンの温度は電気ドライブの速度よりはるかにゆっくりと変化します。
- アラームや画面のタグは、ギャップのない1つのデータ領域に配置します。
- 最低1つの取得サイクルでこれらを使用できる場合のみ、コントローラの変化が確実に検出されます。
- 伝送レートを、エラーのない転送が可能な最高の値に設定します。

画像

画面の更新速度は、表示されるデータの種類と量によって変わります。

短い更新時間にするために、素早く更新する必要のあるオブジェクトのみに短い取得時間を設定するようにしてください。

カーブ

ビットトリガされたカーブを使用するとき、グループビットが[カーブ転送領域]で設定されている場合、この領域でそのビットが設定されている全てのカーブは WinCC ステーションで更新されます。次のサイクルでビットがリセットされます。

全てのビットが WinCC ステーションでリセットされた後だけ、グループビットを PLC プログラムで再設定できます。

5.3 WinCC プロセス通信

5.3.1 WinCC プロセス通信

はじめに

WinCC から、オートメーションシステムのプロセスタグ(外部タグ)に、アクセスできます。ただし、WinCC でプロセス接続を設定する前に、チェックリストを使って、以下の前提条件が満たされているかを確認する必要があります。

- オートメーションシステムに、WinCC の通信ドライバがサポートしている通信インターフェースが、装備されている必要があります。
- このインターフェースは、制御プログラムが通話呼び出しによってインターフェースにアクセスできるように、オートメーションシステムで設定されていなければなりません。通信ハードウェアの設定パラメータは、既知でなければなりません。
- WinCC がアクセスすべきタグのアドレスが、既知でなければなりません。アドレスは使用するオートメーションシステムによって異なることに注意してください。
- それぞれの通信ハードウェア(通信プロセッサ、標準 I/O ポート COMx など)が、WinCC システムにインストールされている必要があります。このハードウェアをインストールするために、付属しているオペレーティングシステムのドライバ(ハードウェアドライバ)もまた、事前にインストールされていなければなりません。通信プロセッサのハードウェアおよびソフトウェアの設定が、既知でなければなりません。
- WinCC システムで使用される通信プロセッサによっては、さらに多くの設定が必要になることがあります。例えば、Industrial Ethernet あるいは PROFIBUS を使用する場合、ローカルのデータベースを作成する必要があります。これらの接続パラメータも既知でなければなりません。

ランタイム中の操作では、外部タグにアクセスできるように、WinCC と AS の間に物理的な接続も存在しなくてはなりません。

S7DOS 設定

S7DOS を使用している場合は、バージョン「S7DOS V9」の IPv4 プロトコルが必要です。

そのため、ネットワークアダプタまたは SIMATIC Ethernet CP の Ethernet プロパティで IPv4 プロトコルを有効のままにしておきます。

このようにして、S7DOS のモジュール検出が TCP、RFC1006、ISO プロトコルで機能することを保証します。

5.3.2 WinCC 通信の原則

はじめに

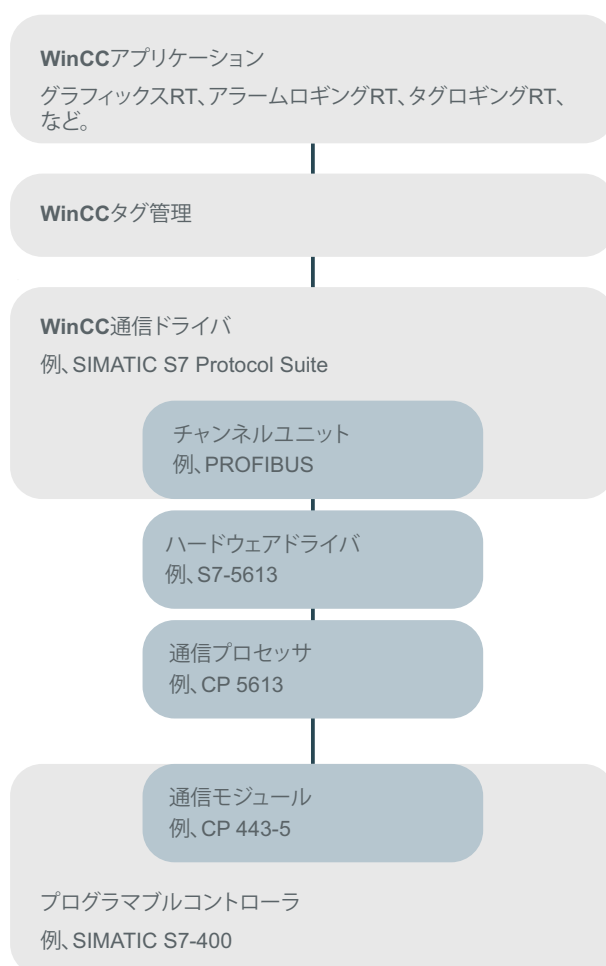
WinCC は、いわゆるタグ管理を使ってタグを一元的に管理します。プロジェクトで作成され、プロジェクトデータベースに保存されたすべてのデータとタグは、WinCC Runtime で取得および管理されます。

グラフィックランタイム、アラームロギングランタイム、またはタグロギングランタイム (グローバルスクリプト)などのすべてのアプリケーションは、タグ管理から WinCC タグの形式でデータを要求する必要があります。

WinCC とオートメーションシステム(AS)間の通信

WinCC との産業用通信における通信は、タグとプロセス値を使用して情報が交換されることを意味します。

プロセス値を取得するために、WinCC 通信ドライバは要求フレームを AS に送信します。すると、要求されたプロセス値に対応する応答フレームで WinCC に返します。



WinCC と AS の間の物理的な接続は、最初に存在する必要があります。

転送媒体や通信ネットワークなどのこの接続のプロパティは、通信の条件を定義し、WinCC で通信を設定するために必要です。

通信ドライバ

通信ドライバは、AS と WinCC のタグ管理の間に接続を確立するソフトウェア構成要素で、プロセス値を含む WinCC タグの提供を可能にします。WinCC には、さまざまなバスシステムを使用して異なる AS を接続するための多数の通信ドライバがあります。

各通信ドライバは、WinCC プロジェクトで一度だけ統合することができます。

WinCC では、通信ドライバは「チャンネル」とも呼ばれ、ファイル拡張子は「*.chn」です。コンピュータにインストールされたすべての通信ドライバは、WinCC インストールディレクトリの「\bin」サブディレクトリにあります。

通信ドライバには、異なる通信ネットワークに対して異なるチャンネルユニットがあります。

チャンネルユニット

各チャンネルユニットは、1つの基礎ハードウェアドライバが装備されたインターフェース、つまり PC の 1つの通信プロセッサとのインターフェースとして動作します。したがって、使用される各チャンネルユニットは、関連する通信プロセッサに割り付ける必要があります。

一部のチャンネルユニットでは、いわゆるシステムパラメータで、追加の設定が行なわれます。

OSI モデルのトランスポートレイヤ(第 4 層)で動作するチャンネルユニットでは、トランスポートパラメータも定義されます。

接続(論理的)

WinCC と AS が正しく物理的に接続されると、AS との(論理的な)接続を確立または設定するために、WinCC に通信ドライバと関連するチャンネルユニットが必要です。

ランタイムには、この接続を通してデータ交換が行なわれます。システムタグを使用して、この接続を確立または終了し、接続ステータスをクエリすることができます。

WinCC では、接続は特定の通信サービスを実行するための 2 つの通信パートナで設定された論理的な割り付けです。

すべての接続には、通信パートナのアドレス指定に必要な情報と接続を確立するためのその他の属性を含む 2 つのエンドポイントがあります

特定の接続パラメータを使用して、チャンネル単位で接続を設定します。通信ドライバに応じて、チャンネル単位で複数の接続を作成することもできます。

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

5.3.3 ランタイム時の接続ステータスタグの設定

WinCC Runtime が有効になると、設定されたコントローラへの接続が確立されます。

ランタイム時に個々の接続を選択的に無効化または有効化するには、「ConnectionStates」タググループのシステムタグを使用します。

別のシステムタグを使用すると、現在の接続ステータスをクエリできます。

サポートされるチャンネル

接続ステータスのシステムタグは、WinCC V7.5 以降でサポートされている通信ドライバで使用できます。

システム情報の評価にのみ使用される「システム情報」チャンネルは例外です。

診断:接続特性

接続の動作を評価するには、WinCC パフォーマンスタグを使用します。追加情報:

- 「パフォーマンスタグによる接続の確認 (ページ 617)」

タググループ「ConnectionStates」

チャンネルの接続状態を指定または決定するために、接続ごとに次のシステムタグが作成されます。

- @<Connectionname>@ForceConnectionStateEx
このタグを使用して、ランタイム時に接続を確立または終了します。
- @<Connectionname>@ConnectionStateEx
このタグを使用して、ランタイム時に接続のステータスを判断します。

タグのタグタイプは「符号なし 32 ビット値(DWORD)」です。

接続の名前を変更すると、2 つのシステムタグの名前も変更されます。

タグ値

タグ	使用	値	説明
@<...>@ForceConnectionStateEx	接続状態の決定	1	接続の確立 開始値 = 1: ランタイムが有効化されると、接続が確立されます。
		0	接続の終了 開始値 = 0: ランタイムが有効化されると、接続は無効のままです。 接続のタグはアーカイブされません。

タグ	使用	値	説明
@<...>@ConnectionStateEx	現在の接続ステータスの決定	1	接続を使用する準備が完了しています。
		0	断線、または終了されました。

必要条件

- 必要な接続は、タグ管理で作成されます。

手順

- タグ管理のナビゲーションエリアで目的の接続を選択します。
- 接続のコンテキストメニューで、エントリ[有効化/無効化タグの作成]を選択します。
新しいタググループ「ConnectionStates」が「内部タグ」のナビゲーションツリーに作成されます。
このグループには、作成された2つのタグが含まれます。
- コントロールシステムの各タグに対して別個のアドレスを設定します。
この目的で、使用されていないまたは架空のアドレスを使用します。このアドレスは、タグ転送にのみ必要です。

タグの使用

接続ステータスの問い合わせ

接続のステータスを特定するには、タグ「@<...>@ConnectionStateEx」の値を読み取ります。

接続の終了

接続を無効にするには、「@<...>@ForceConnectionStateEx」タグで値「0」を設定します。
関連するプロセスタグのアーカイブが停止されます。

接続の確立

中断された接続を再度有効にするには、「@<...>@ForceConnectionStateEx」タグで値「1」を設定します。

対応する通信チャンネルのプロセスタグが再度アーカイブされます。

接続ステータスの視覚化

例えば、SVG オブジェクトを使用するプロセス画像で接続ステータスを視覚化できます。

V2.0 以降では、SVG ライブラリ [IndustryGraphicLibrary] には、[SIMATIC > SystemDiagnostic] フォルダに次のオブジェクトが含まれます:

- SysDiag_DiagnosticsIndicator
- SysDiag_SignalLamp
- SysDiag_SignalTower

下記も参照

[ステータス - 論理接続] ファンクションを使用してチャンネルを確認する方法 (ページ 622)

モニタ接続ステータスへのタグステータスの使用 (ページ 691)

パフォーマンスタグによる接続の確認 (ページ 617)

5.3.4 外部タグ

5.3.4.1 外部タグ

はじめに

AS の特定のデータへのアクセスを取得するために、WinCC でタグが必要です。AS への接続に依存するこれらのタグは、外部タグと呼ばれます。対照的に、プロセス接続を持たないタグは、内部タグと呼ばれます。

データタイプとタイプ変換

外部タグを設定するとき、タグ名およびデータタイプを指定する必要があります。一部のデータタイプでは、タイプ変換も指定する必要があります。

データタイプは、WinCC におけるデータフォーマットを決定します。タイプ変換は、AS データフォーマットから WinCC データフォーマットへの変換を指定するために使用されます。タイプ変換は、両方の転送方法に適用されます。

- AS では:例えば、特定のファンクション(タイマー値/BCD 表示など)のため、あるいはアドレス指定する情報(データブロックまたは I/O 領域におけるバイトアドレス、ワードアドレスなど)のため。
- WinCC では:例えば、アナログ値の処理または計算用に。

実際には、AS データフォーマットが通常存在します。次の可能性は、WinCC フォーマットの選択肢に存在します。

- WinCC データフォーマットを AS フォーマットに適合させることができます。これは、両側で同じフォーマットを使用するタイプ変換を選択し、WinCC データタイプに応じて、符号を考慮する場合に可能です(例、"WordToSignedWord")。選択したデータタイプでこれができない場合、WinCC で変更しなくてはなりません。
- WinCC フォーマットとは、WinCC で処理される値をベースにしています。

データタイプおよび(必要に応じて)タイプ変換を選択するとき、次を順守する必要があります。

- 符号：変換でこれを考慮に入れておく必要があるか？操作中に負のタグ値も発生することがあるか？(例えば、閉ループ制御エラー(%単位))
- 値の範囲：操作中に発生するタグ値が両方のフォーマットの値の範囲内か？あるいは、WinCC または AS で値のオーバーフローが予想されるか？オーバーフローが発生すると、値が他方で表示されないか、以降の処理中に問題を引き起こすことがあります。
- 同じ値の範囲での異なるタイプ変換：データタイプの複数のタイプ変換が同じ値範囲となることもあります。例えば、値範囲[0～127]での"ByteToUnsignedDword"および"ByteToUnsignedWord"。この場合、AS データが存在するフォーマットを必ず確認し、このフォーマットが、範囲が過大であるために不必要にリソースを浪費していないかを確認します(例、Word ではなく DWord)。

AS で必要とされる値範囲が、選択されたタイプ変換でカバーされていない場合、WinCC でデータタイプを変更しなければなりません。

注記

例えばアドレスエラーなどのために、プロセスタグが正しく設定されていない場合、オートメーションシステムとの通信が中断されることがあります。

WinCC データタイプとタイプ変換

以下の表は、どの WinCC データタイプがタイプ変換をサポートするかを示します。

データタイプ	タイプ変換
2 進タグ	いいえ
符号なし 8 ビット値	はい
符号付き 8 ビット値	はい
符号なし 16 ビット値	はい

データタイプ	タイプ変換
符号付き 16 ビット値	はい
符号なし 32 ビット値	はい
符号付き 32 ビット値	はい
浮動小数点数 32 ビット IEEE 754	はい
浮動小数点数 64 ビット IEEE 754	はい
テキストタグ、8 ビット文字セット	いいえ
テキストタグ、16 ビット文字セット	いいえ
未処理データタイプ	いいえ

注記

タイプ変換の場合、AS によって送信されるデータが、選択されたタイプ変換内で WinCC によって解釈されることがあります。データが WinCC によって解釈できない場合、`"..\Siemens\WinCC\Diagnose"`ディレクトリの`"WinCC_sys_0x.log"`ファイルにエラーが入力されます。

数値タグタイプのリニアスケールリング

数値データタイプに対して、リニアスケールリングを実行できます。プロセスの変数の値範囲を、WinCC タグの定義された値範囲に直線的にマッピングすることができます。

例えば、この値がユーザーによって WinCC に入力されるとき(mbar 単位)、プロセスはバーで圧力セットポイントの仕様を呼び出すことがあります。リニアスケールリングを使用して、プロセスの値範囲[0～1]を WinCC タグの値範囲[0～1000]に変換できます。

テキストタグの長さ情報

[テキストタグ 8 ビット文字セット]データタイプおよび[テキストタグ 16 ビット文字セット]データタイプのタグに対して、長さ情報が必要です。後で 10 文字に適合されるテキストタグは、[テキストタグ 8 ビット文字セット]データタイプの場合は 10 の長さ、および[テキストタグ 16 ビット文字セット]データタイプの場合は 20 の長さがある必要があります。

オートメーションシステムのアドレス指定

WinCC タグは、AS のデータタイプに割り当てられます。これらは、特定の 방법으로 AS でアドレス指定されなくてはなりません。アドレス指定方法は、通信パートナーのタイプに依存します。

タグ名の接頭語および接尾語

AS からタグをダウンロードすると、接続のインスタンスのタグ名に接頭語と接尾語を定義できます。[タグ選択]ダイアログを使用してプロセスタグをインポートすると、接続のすべてのタグに接頭語や接尾語が自動的に追加されます。接頭語や接尾語を変更しても、既にインポートされたタグには影響しません。

5.3.4.2 新規接続の作成方法

はじめに

外部タグは、AS への接続に基いてのみ作成できます。必要な接続が存在しない場合、まず作成する必要があります。

必要条件

- 必要な通信プロセッサとそれぞれのハードウェアドライバが、インストールされている。
- 例えば、"SIMATIC S7 Protocol Suite"などの必要な通信ドライバも、インストールされている。

手順

1. コンフィグレーションスタジオで、ナビゲーションバーの[タグ管理]を選択します。
2. ナビゲーションエリアで、例えば"PROFIBUS"などの必要なチャンネルユニットを選択します。
3. チャンネルユニットのポップアップメニューで、[新規接続...]エントリを選択します。
4. データ領域で接続に一意の名前を付けます。
5. [プロパティ]ウィンドウでこの接続に必要なパラメータを定義します。詳細な情報は、関連するチャンネルのヘルプ/文書を参照してください。

5.3.4.3 外部タグを以下のように設定します

はじめに

タグを作成する手順は、ほとんど全てのデータタイプに対して同様です。
ただし一部のデータタイプでは、特殊な設定が必要です（ステップ 5～7）。

必要条件

- 必要な通信プロセッサとハードウェアドライバがインストールされている。
- 例えば、"SIMATIC S7 Protocol Suite"などの必要な通信ドライバがインストールされている。
- チャンネルユニット(例えば"PROFIBUS")に基いて、接続が既に作成されている。

手順

1. ナビゲーションエリアのツリー表示で、タグを作成する接続を選択します。
2. [名前]列の先頭の空きセルに、たとえば"WinCCTag_01"などの WinCC プロジェクト内で一意のタグ名を入力します。
3. [データタイプ]フィールドで、たとえば"浮動小数点数 64 ビット IEEE 754"などのタグのデータタイプを定義します。
4. AS の[AS 長さ]フィールドで、タグのアドレスエリアを指定します。
2 進または 8 ビットタグを使用したビット/バイトアクセスをサポートしないチャンネルでは、まず[ビット/バイトタグ]ダイアログ、次に[タグプロパティ]ダイアログも表示されます。
詳細情報は「BinWrite メカニズムの原則」を参照してください。
[OK]ボタンで、[ビットタグ/バイトタグ]ダイアログまたは[タグプロパティ]ダイアログを閉じます。
5. 数値タグで、WinCC は[フォーマット調整]フィールドでのフォーマット調整を提案します。
必要な場合、他のフォーマット調整を選択します。表示は"X から Y"の順です。ここで例えば"DoubleToDouble"のように、X = WinCC フォーマット、Y = AS フォーマットです。
6. [線形スケーリング]チェックボックスを有効にして、数値タグを直線的にスケールします。
[プロセス値範囲] (AS)および[タグ値範囲] (WinCC)の上下限を入力します。
7. テキストタグの[長さ]フィールドが有効になります。ここで、テキストタグの長さを文字数で入力します。
8. [OK]ボタンで、全てのダイアログを閉じます。

5.3.4.4 WinCC データタイプでソートしたフォーマット調整

概要

外部タグの設定時に、すべての数値データタイプ用に他のフォーマットを調整する必要があります。

データタイプによって WinCC 側のデータフォーマットが決定されます。フォーマットの調整で、WinCC フォーマットから AS フォーマットへの変換も定義します。定義を転送の両方向に適用します。

以下の選択ボックスで、必要な WinCC データタイプを選択します。それぞれの可能なフォーマット調整のリストと値範囲が、以下の表で提供されます。

WinCC データタイプ

表 5-1 符号付き 8 ビット値

フォーマット調整"符号付き 8 ビット値"	値の範囲
CharToUnsignedByte	0...127
CharToUnsignedWord	0...127
CharToUnsignedDword	0...127
CharToSignedByte	-128...+127 (変換なし)
CharToSignedWord	-128...+127
CharToSignedDword	-128...+127
CharToMSBByte	-127...+127
CharToMSBWord	-128...+127
CharToMSBDword	-128...+127
CharToBCDByte	0...99
CharToBCDWord	0...127
CharToBCDDword	0...127
CharToSignedBCDByte	-9...+9
CharToSignedBCDWord	-128...+127
CharToSignedBCDDword	-128...+127
CharToExtSignedBCDByte	-79...+79
CharToExtSignedBCDWord	-128...+127

フォーマット調整"符号付き 8 ビット値"	値の範囲
CharToExtSignedBCDDword	-128...+127
CharToAikenByte	0...99
CharToAikenWord	0...127
CharToAikenDword	0...127
CharToSignedAikenByte	-9...+9
CharToSignedAikenWord	-128...+127
CharToSignedAikenDword	-128...+127
CharToExcessByte	0...99
CharToExcessWord	0...127
CharToExcessDword	0...127
CharToSignedExcessByte	-9...+9
CharToSignedExcessWord	-128...+127
CharToSignedExcessDword	-128...+127

表 5-2 符号なし 8 ビット値

フォーマット調整"符号なし 8 ビット値"	値の範囲
ByteToUnsignedByte	0...255 (変換なし)
ByteToUnsignedWord	0...255
ByteToUnsignedDword	0...255
ByteToSignedByte	0...127
ByteToSignedWord	0...255
ByteToSignedDword	0...255
ByteToBCDByte	0...99
ByteToBCDWord	0...255
ByteToBCDDword	0...255
ByteToAikenByte	0...99
ByteToAikenWord	0...255
ByteToAikenDword	0...255
ByteToExcessByte	0...99

5.3 WinCC プロセス通信

フォーマット調整"符号なし 8 ビット値"	値の範囲
ByteToExcessWord	0...255
ByteToExcessDword	0...255

表 5-3 符号付き 16 ビット値

フォーマット調整"符号付き 16 ビット値"	値の範囲
ShortToUnsignedByte	0...255
ShortToUnsignedWord	0...32767
ShortToUnsignedDword	0...32767
ShortToSignedByte	-128...+127
ShortToSignedWord	-32768...+32767 (変換なし)
ShortToSignedDword	-32768...+32767
ShortToMSBByte	-127...+127
ShortToMSBWord	-32767...+32767
ShortToMSBDword	-32768...+32767
ShortToBCDByte	0...99
ShortToBCDWord	0...9999
ShortToBCDDword	0...32767
ShortToSignedBCDByte	-9...+9
ShortToSignedBCDWord	-999...+999
ShortToSignedBCDDword	-32768...+32767
ShortToExtSignedBCDByte	-79...+79
ShortToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDDword	-32768...+32767
ShortToAikenByte	0...99
ShortToAikenWord	0...9999
ShortToAikenDword	0...32767
ShortToSignedAikenByte	-9...+9
ShortToSignedAikenWord	-999...+999
ShortToSignedAikenDword	-32768...+32767
ShortToExcessByte	0...99

フォーマット調整"符号付き 16 ビット値"	値の範囲
ShortToExcessWord	0...9999
ShortToExcessDword	0...32767
ShortToSignedExcessByte	-9...+9
ShortToSignedExcessWord	-999...+999
ShortToSignedExcessDword	-32768...+32767

表 5-4 符号なし 16 ビット値

フォーマット調整"符号なし 16 ビット値"	値の範囲
WordToUnsignedWord	0...65535 (変換なし)
WordToUnsignedByte	0...255
WordToUnsignedDword	0...65535
WordToSignedByte	0...127
WordToSignedWord	0...32767
WordToSignedDword	0...65535
WordToBCDByte	0...99
WordToBCDWord	0...9999
WordToBCDDword	0...65535
WordToAikenByte	0...99
WordToAikenWord	0...9999
WordToAikenDword	0...65535
WordToExcessByte	0...99
WordToExcessWord	0...9999
WordToExcessDword	0...65535
WordToSimaticBCDCounter	0...999
WordToSimaticCounter	0...999

5.3 WinCC プロセス通信

表 5-5 符号付き 32 ビット値

フォーマット調整"符号付き 32 ビット値"	値の範囲
LongToSignedDword	-2147483648...+2147483647 (変換なし)
LongToUnsignedByte	0...255
LongToUnsignedWord	0...65535
LongToUnsignedDword	0...2147483647
LongToSignedByte	-128...+127
LongToSignedWord	-32768...+32767
LongToSignedQword	-2147483648...+2147483647
LongToMSBByte	-127...+127
LongToMSBWord	-32767...+32767
LongToMSBDword	-2147483647...+2147483647
LongToMSBQword	-2147483648...+2147483647
LongToBCDByte	0...99
LongToBCDWord	0...9999
LongToBCDDword	0...99999999
LongToSignedBCDByte	-9...+9
LongToSignedBCDWord	-999...+999
LongToSignedBCDDword	-9999999...+9999999
LongToSignedBCDQword	-2147483648...+2147483647
LongToExtSignedBCDByte	-79...+79
LongToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDDword	-79999999...+79999999
LongToExtSignedBCDQword	-79999999...+79999999
LongToAikenByte	0...99
LongToAikenWord	0...9999
LongToAikenDword	0...99999999
LongToSignedAikenByte	-9...+9
LongToSignedAikenWord	-999...+999
LongToSignedAikenDword	-9999999...+9999999
LongToSignedAikenQword	-999999999...+999999999

フォーマット調整"符号付き 32 ビット値"	値の範囲
LongToExcessByte	0...99
LongToExcessWord	0...9999
LongToExcessDword	0...99999999
LongToSignedExcessByte	-9...+9
LongToSignedExcessWord	-999...+999
LongToSignedExcessDword	-9999999...+9999999
LongToSignedExcessQword	-99999999...+99999999
LongToSimaticBCDTimer	10...9990000
LongToSimaticTimer	10...9990000
LongToSimaticLTime	00:00:00.000...596.31.23.647

表 5-6 符号なし 32 ビット値

フォーマット調整"符号なし 32 ビット値"	値の範囲
DwordToUnsignedDword	0...4294967295 (変換なし)
DwordToUnsignedByte	0...255
DwordToUnsignedWord	0...65535
DwordToUnsignedQword	0...4294967295
DwordToSignedByte	0...127
DwordToSignedWord	0...32767
DwordToSignedDword	0...2147483647
DwordToBCDByte	0...99
DwordToBCDWord	0...9999
DwordToBCDDword	0...99999999
DwordToBCDQword	0...999999999
DwordToAikenByte	0...99
DwordToAikenWord	0...9999
DwordToAikenDword	0...99999999
DwordToAikenQword	0...999999999
DwordToExcessByte	0...99
DwordToExcessWord	0...9999

5.3 WinCC プロセス通信

フォーマット調整"符号なし 32 ビット値"	値の範囲
DwordToExcessDword	0...999999999
DwordToExcessQword	0...9999999999
DwordToSimaticBCDTimer	10...9990000
DwordToSimaticTimer	10...9990000
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

表 5-7 浮動小数点数 32 ビット IEEE 754

フォーマット調整"浮動小数点数 32 ビット IEEE 754"	値の範囲
FloatToFloat	+/-3.402823e+38 (変換なし)
FloatToUnsignedByte	0...255
FloatToUnsignedWord	0...65535
FloatToUnsignedDword	0...4.294967e+09
FloatToSignedByte	-128...+127
FloatToSignedWord	-32768...+32767
FloatToSignedDword	-2.147483e+09...+2.147483e+09
FloatToDouble	+/-3.402823e+38
FloatToMSBByte	-127...+127
FloatToMSBWord	-32767...+32767
FloatToMSBDword	-2.147483e+09...+2.147483e+09
FloatToBCDByte	0...99
FloatToBCDWord	0...9999
FloatToBCDDword	0...9.999999e+07
FloatToSignedBCDByte	-9...+9
FloatToSignedBCDWord	-999...+999
FloatToSignedBCDDword	-9999999...+9999999
FloatToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
FloatToAikenByte	0...99
FloatToAikenWord	0...9999

フォーマット調整"浮動小数点数 32 ビット IEEE 754"	値の範囲
FloatToAikenDword	0...9,999999e+07
FloatToSignedAikenByte	-9...+9
FloatToSignedAikenWord	-999...+999
FloatToSignedAikenDword	-9999999...+9999999
FloatToExcessByte	0...99
FloatToExcessWord	0...9999
FloatToExcessDword	0...9.999999e+07
FloatToSignedExcessByte	-9...+9
FloatToSignedExcessWord	-999...+999
FloatToSignedExcessDword	-9999999...+9999999
FloatToSimaticBCDTimer	10...9990000
FloatToS5Float	+1.701411e+38
FloatToSimaticTimer	10...9990000

表 5-8 浮動小数点数 64 ビット IEEE 754

フォーマット調整"浮動小数点数 64 ビット IEEE 754"	値の範囲
DoubleToDouble	+1.79769313486231e+308 (変換なし)
DoubleToUnsignedByte	0...255
DoubleToUnsignedWord	0...65535
DoubleToUnsignedDword	0...4294967295
DoubleToUnsignedQword	0...18446744073709551616
DoubleToSignedByte	-128...+127
DoubleToSignedWord	-32768...+32767
DoubleToSignedDword	-2147483648...+2147483647
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
DoubleToFloat	+3.402823e+38
DoubleToMSBByte	-127...+127
DoubleToMSBWord	-32767...+32767
DoubleToMSBDword	-2147483647...+2147483647

5.3 WinCC プロセス通信

フォーマット調整"浮動小数点数 64 ビット IEEE 754"	値の範囲
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
DoubleToBCDByte	0...99
DoubleToBCDWord	0...9999
DoubleToBCDDword	0...99999999
DoubleToBCDQword	0...99999999999999
DoubleToSignedBCDByte	-9...+9
DoubleToSignedBCDWord	-999...+999
DoubleToSignedBCDDword	-9999999...+9999999
DoubleToSignedBCDQword	-99999999999999...+99999999999999
DoubleToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDWord	-7999...+7999
DoubleToExtSignedBCDDword	-79999999...+79999999
DoubleToExtSignedBCDQword	-79999999999999...+79999999999999
DoubleToAikenByte	0...99
DoubleToAikenWord	0...9999
DoubleToAikenDword	0...99999999
DoubleToAikenQword	0...99999999999999
DoubleToSignedAikenByte	-9...+9
DoubleToSignedAikenWord	-999...+999
DoubleToSignedAikenDword	-9999999...+9999999
DoubleToSignedAikenQword	-99999999999999...+99999999999999
DoubleToExcessByte	0...99
DoubleToExcessWord	0...9999
DoubleToExcessDword	0...99999999
DoubleToExcessQword	0...99999999999999
DoubleToSignedExcessByte	-9...+9
DoubleToSignedExcessWord	-999...+999
DoubleToSignedExcessDword	-9999999...+9999999
DoubleToSignedExcessQword	-99999999999999...+99999999999999
DoubleToSimaticBCDTimer	10...9990000

フォーマット調整"浮動小数点数 64 ビット IEEE 754"	値の範囲
DoubleToS5Float	+/-1.701411e+38
DoubleToSimaticTimer	10...9990000

5.3.4.5 AS データタイプによってソートされたフォーマット調整

概要

外部タグの設定時に、すべての数値データタイプ用に他のフォーマットを調整する必要があります。

データタイプによって WinCC 側のデータフォーマットが決定されます。フォーマットの調整で、WinCC フォーマットから AS フォーマットへの変換も定義します。定義を転送の両方向に適用します。

以下の選択ボックスで、必要な AS データタイプを選択します。それぞれの可能なフォーマット調整のリストと値範囲が、以下の表で提供されます。

AS データタイプ

タイプ変換と値の範囲:

表 5-9 AikenByte

タイプ変換"AikenByte"	値の範囲
ByteToAikenByte	0...99
AikenByte	0...99
DoubleToAikenByte	0...99
DwordToAikenByte	0...99
FloatToAikenByte	0...99
LongToAikenByte	0...99
ShortToAikenByte	0...99
WordToAikenByte	0...99

5.3 WinCC プロセス通信

表 5-10 AikenWord

タイプ変換"AikenWord"	値の範囲
ByteToAikenWord	0...255
CharToAikenWord	0...127
DoubleToAikenWord	0...9999
DwordToAikenWord	0...9999
FloatToAikenWord	0...9999
LongToAikenWord	0...9999
ShortToAikenWord	0...9999
WordToAikenWord	0...9999

表 5-11 AikenDWord

タイプ変換"AikenDWord"	値の範囲
ByteToAikenDword	0...255
CharToAikenDword	0...127
DoubleToAikenDword	0...999999999
DwordToAikenDword	0...999999999
FloatToAikenDword	0...9,999999e+07
LongToAikenDword	0...999999999
ShortToAikenDword	0...32767
WordToAikenDword	0...65535

表 5-12 AikenQWord

タイプ変換"AikenQWord"	値の範囲
DoubleToAikenQword	0...9999999999999999
DwordToAikenQword	0...9999999999

表 5-13 BCDByte

タイプ変換"BCDByte"	値の範囲
ByteToBCDByte	0...99
CharToBCDByte	0...99
DoubleToBCDByte	0...99
DwordToBCDByte	0...99
FloatToBCDByte	0...99
LongToBCDByte	0...99
ShortToBCDByte	0...99
WordToBCDByte	0...99

表 5-14 BCDWord

タイプ変換"BCDWord"	値の範囲
ByteToBCDWord	0...255
CharToBCDWord	0...127
DoubleToBCDWord	0...9999
DwordToBCDWord	0...9999
FloatToBCDWord	0...9999
LongToBCDWord	0...9999
ShortToBCDWord	0...9999
WordToBCDWord	0...9999

表 5-15 BCDDWord

タイプ変換"BCDDWord"	値の範囲
ByteToBCDDword	0...255
CharToBCDDword	0...127
DoubleToBCDDword	0...99999999
DwordToBCDDword	0...99999999
FloatToBCDDword	0...9.999999e+07
LongToBCDDword	0...99999999

5.3 WinCC プロセス通信

タイプ変換"BCDDWord"	値の範囲
ShortToBCDDword	0...32767
WordToBCDDword	0...65535

表 5-16 BCDQWord

タイプ変換"BCDQWord"	値の範囲
DoubleToBCDQword	0...9999999999999999
DwordToBCDQword	0...999999999

表 5-17 Double

タイプ変換"Double"	値の範囲
DoubleToDouble	+1.79769313486231e+308 (変換なし)
FloatToDouble	+3.402823e+38

表 5-18 ExcessByte

タイプ変換"ExcessByte"	値の範囲
ByteToExcessByte	0...99
CharToExcessByte	0...99
DoubleToExcessByte	0...99
DwordToExcessByte	0...99
FloatToExcessByte	0...99
LongToExcessByte	0...99
ShortToExcessByte	0...99
WordToExcessByte	0...99

表 5-19 ExcessWord

タイプ変換"ExcessWord"	値の範囲
ByteToExcessWord	0...255
CharToExcessWord	0...127

タイプ変換"ExcessWord"	値の範囲
DoubleToExcessWord	0...9999
DwordToExcessWord	0...9999
FloatToExcessWord	0...9999
LongToExcessWord	0...9999
ShortToExcessWord	0...9999
WordToExcessWord	0...9999

表 5-20 ExcessDWord

タイプ変換"ExcessDWord"	値の範囲
ByteToExcessDword	0...255
CharToExcessDword	0...127
DoubleToExcessDword	0...999999999
DwordToExcessDword	0...999999999
FloatToExcessDword	0...9.9999999e+07
LongToExcessDword	0...999999999
ShortToExcessDword	0...32767
WordToExcessDword	0...65535

表 5-21 ExcessQWord

タイプ変換"ExcessQWord"	値の範囲
DoubleToExcessQword	0...9999999999999999
DwordToExcessQword	0...999999999

表 5-22 ExtSignedBCDByte

タイプ変換"ExtSignedBCDByte"	値の範囲
CharToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDByte	-79...+79

5.3 WinCC プロセス通信

タイプ変換"ExtSignedBCDByte"	値の範囲
LongToExtSignedBCDByte	-79...+79
ShortToExtSignedBCDByte	-79...+79

表 5-23 ExtSignedBCDWord

タイプ変換"ExtSignedBCDWord"	値の範囲
CharToExtSignedBCDWord	-128...+127
DoubleToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDWord	-7999...+7999
LongToExtSignedBCDWord	-7999...+7999
ShortToExtSignedBCDWord	-7999...+7999

表 5-24 ExtSignedBCDDWord

タイプ変換"ExtSignedBCDDWord"	値の範囲
CharToExtSignedBCDDword	-128...+127
DoubleToExtSignedBCDDword	-79999999...+79999999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
LongToExtSignedBCDDword	-79999999...+79999999
ShortToExtSignedBCDDword	-32768...+32767

表 5-25 ExtSignedBCDQWord

タイプ変換"ExtSignedBCDQWord"	値の範囲
DoubleToExtSignedBCDQword	-7999999999999999...+7999999999999999
LongToExtSignedBCDQword	-79999999...+79999999

表 5-26 Float

タイプ変換"Float"	値の範囲
DoubleToFloat	+/-3.402823e+38
FloatToFloat	+/-3.402823e+38 (変換なし)

表 5-27 MSBByte

タイプ変換"MSBByte"	値の範囲
CharToMSBByte	-127...+127
DoubleToMSBByte	-127...+127
FloatToMSBByte	-127...+127
LongToMSBByte	-127...+127
ShortToMSBByte	-127...+127

表 5-28 MSBWord

タイプ変換"MSBWord"	値の範囲
CharToMSBWord	-128...+127
DoubleToMSBWord	-32767...+32767
FloatToMSBWord	-32767...+32767
LongToMSBWord	-32767...+32767
ShortToMSBWord	-32767...+32767

表 5-29 MSBDWord

タイプ変換"MSBDWord"	値の範囲
CharToMSBDword	-128...+127
DoubleToMSBDword	-2147483647...+2147483647
FloatToMSBDword	-2.147483e+09...+2.147483e+09
LongToMSBDword	-2147483647...+2147483647
ShortToMSBDword	-32768...+32767

表 5-30 MSBQWord

タイプ変換"MSBQWord"	値の範囲
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
LongToMSBQword	-2147483648...+2147483647

5.3 WinCC プロセス通信

表 5-31 S5Float

タイプ変換"S5Float"	値の範囲
DoubleToS5Float	+/-1.701411e+38
FloatToS5Float	+/-1.701411e+38

表 5-32 SignedByte

タイプ変換"SignedByte"	値の範囲
ByteToSignedByte	0...127
CharToSignedByte	-128...+127 (変換なし)
DoubleToSignedByte	-128...+127
DwordToSignedByte	0...127
FloatToSignedByte	-128...+127
LongToSignedByte	-128...+127
ShortToSignedByte	-128...+127
WordToSignedByte	0...127

表 5-33 SignedWord

タイプ変換"SignedWord"	値の範囲
ByteToSignedWord	0...255
CharToSignedWord	-128...+127
DoubleToSignedWord	-32768...+32767
DwordToSignedWord	0...32767
FloatToSignedWord	-32768...+32767
LongToSignedWord	-32768...+32767
ShortToSignedWord	-32768...+32767 (変換なし)
WordToSignedWord	0...32767

表 5-34 SignedDWord

タイプ変換"SignedDWord"	値の範囲
ByteToSignedDword	0...255
CharToSignedDword	-128...+127
DoubleToSignedDword	-2147483648...+2147483647
DwordToSignedDword	0...2147483647
FloatToSignedDword	-2.147483e+09...+2.147483e+09
LongToSignedDword	-2147483648...+2147483647 (変換なし)
ShortToSignedDword	-32768...+32767
WordToSignedDword	0...65535

表 5-35 SignedQWord

タイプ変換"SignedQWord"	値の範囲
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
LongToSignedQword	-2147483648...+2147483647

表 5-36 SignedAikenByte

タイプ変換"SignedAikenByte"	値の範囲
CharToSignedAikenByte	-9...+9
DoubleToSignedAikenByte	-9...+9
FloatToSignedAikenByte	-9...+9
LongToSignedAikenByte	-9...+9
ShortToSignedAikenByte	-9...+9

表 5-37 SignedAikenWord

タイプ変換"SignedAikenWord"	値の範囲
CharToSignedAikenWord	-128...+127
DoubleToSignedAikenWord	-999...+999

5.3 WinCC プロセス通信

タイプ変換" SignedAikenWord "	値の範囲
FloatToSignedAikenWord	-999...+999
LongToSignedAikenWord	-999...+999
ShortToSignedAikenWord	-999...+999

表 5-38 SignedAikenDWord

タイプ変換" SignedAikenDWord "	値の範囲
CharToSignedAikenDword	-128...+127
DoubleToSignedAikenDword	-9999999...+9999999
FloatToSignedAikenDword	-9999999...+9999999
LongToSignedAikenDword	-9999999...+9999999
ShortToSignedAikenDword	-32768...+32767

表 5-39 SignedAikenQWord

タイプ変換" SignedAikenQWord "	値の範囲
DoubleToSignedAikenQword	-999999999999999...+999999999999999
LongToSignedAikenQword	-999999999...+999999999

表 5-40 SignedBCDByte

タイプ変換" SignedBCDByte "	値の範囲
CharToSignedBCDByte	-9...+9
DoubleToSignedBCDByte	-9...+9
FloatToSignedBCDByte	-9...+9
LongToSignedBCDByte	-9...+9
ShortToSignedBCDByte	-9...+9

表 5-41 SignedBCDWord

タイプ変換" SignedBCDWord "	値の範囲
CharToSignedBCDWord	-128...+127
DoubleToSignedBCDWord	-999...+999

タイプ変換" SignedBCDWord "	値の範囲
FloatToSignedBCDWord	-999...+999
LongToSignedBCDWord	-999...+999
ShortToSignedBCDWord	-999...+999

表 5-42 SignedBCDDWord

タイプ変換" SignedBCDDWord "	値の範囲
CharToSignedBCDDword	-128...+127
DoubleToSignedBCDDword	-9999999...+9999999
FloatToSignedBCDDword	-9999999...+9999999
LongToSignedBCDDword	-9999999...+9999999
ShortToSignedBCDDword	-32768...+32767

表 5-43 SignedBCDQWord

タイプ変換" SignedBCDQWord "	値の範囲
DoubleToSignedBCDQword	-999999999999999...+999999999999999
LongToSignedBCDQword	-2147483648...+2147483647

表 5-44 SignedExcessByte

タイプ変換" SignedExcessByte "	値の範囲
CharToSignedExcessByte	-9...+9
DoubleToSignedExcessByte	-9...+9
FloatToSignedExcessByte	-9...+9
LongToSignedExcessByte	-9...+9
ShortToSignedExcessByte	-9...+9

表 5-45 SignedExcessWord

タイプ変換" SignedExcessWord "	値の範囲
CharToSignedExcessWord	-128...+127
DoubleToSignedExcessWord	-999...+999

5.3 WinCC プロセス通信

タイプ変換" SignedExcessWord "	値の範囲
FloatToSignedExcessWord	-999...+999
LongToSignedExcessWord	-999...+999
ShortToSignedExcessWord	-999...+999

表 5-46 SignedExcessDWord

タイプ変換" SignedExcessDWord "	値の範囲
CharToSignedExcessDword	-128...+127
DoubleToSignedExcessDword	-9999999...+9999999
FloatToSignedExcessDword	-9999999...+9999999
LongToSignedExcessDword	-9999999...+9999999
ShortToSignedExcessDword	-32768...+32767

表 5-47 SignedExcessQWord

タイプ変換" SignedExcessQWord "	値の範囲
DoubleToSignedExcessQword	-999999999999999...+999999999999999
LongToSignedExcessQword	-999999999...+999999999

表 5-48 SimaticCounter

タイプ変換" SimaticCounter "	値の範囲
WordToSimaticCounter	0...999

表 5-49 SimaticBCDCounter

タイプ変換" SimaticBCDCounter "	値の範囲
WordToSimaticBCDCounter	0...999

表 5-50 SimaticTimer

タイプ変換"SimaticTimer"	値の範囲
DoubleToSimaticTimer	10...9990000
DwordToSimaticTimer	10...9990000
FloatToSimaticTimer	10...9990000
LongToSimaticTimer	10...9990000

表 5-51 SimaticBCDTimer

タイプ変換"SimaticBCDTimer"	値の範囲
DoubleToSimaticBCDTimer	10...9990000
DwordToSimaticBCDTimer	10...9990000
FloatToSimaticBCDTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

表 5-52 SimaticLTime

タイプ変換"SimaticLTime"	値の範囲
LongToSimaticLTime	00:00:00.000...596.31.23.647

表 5-53 SimaticLTimeOfDay

タイプ変換"SimaticLTimeOfDay"	値の範囲
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

表 5-54 UnsignedByte

タイプ変換"UnsignedByte"	値の範囲
ByteToUnsignedByte	0...255 (変換なし)
CharToUnsignedByte	0...127
DoubleToUnsignedByte	0...255
DwordToUnsignedByte	0...255
FloatToUnsignedByte	0...255

5.3 WinCC プロセス通信

タイプ変換"UnsignedByte"	値の範囲
LongToUnsignedByte	0...255
ShortToUnsignedByte	0...255
WordToUnsignedByte	0...255

表 5-55 UnsignedWord

タイプ変換"UnsignedWord"	値の範囲
ByteToUnsignedWord	0...255
CharToUnsignedWord	0...127
DoubleToUnsignedWord	0...65535
DwordToUnsignedWord	0...65535
FloatToUnsignedWord	0...65535
LongToUnsignedWord	0...65535
ShortToUnsignedWord	0...32767
WordToUnsignedWord	0...65535 (変換なし)

表 5-56 UnsignedDWord

タイプ変換"UnsignedDWord"	値の範囲
ByteToUnsignedDword	0...255
CharToUnsignedDword	0...127
DoubleToUnsignedDword	0...4294967295
DwordToUnsignedDword	0...4294967295 (変換なし)
FloatToUnsignedDword	0...4.294967e+09
LongToUnsignedDword	0...2147483647
ShortToUnsignedDword	0...32767
WordToUnsignedDword	0...65535

表 5-57 UnsignedQWord

タイプ変換"UnsignedQWord"	値の範囲
DoubleToUnsignedQword	0...18446744073709551616
DwordToUnsignedQword	0...4294967295

5.3.4.6 BinWrite メカニズムの原則

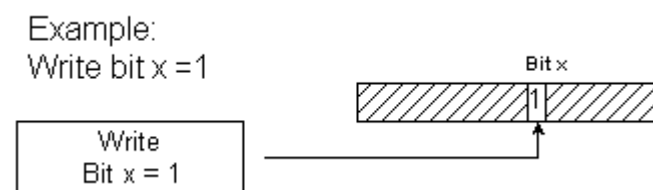
はじめに

WinCC では、全ての通信ドライバおよびそのチャンネルユニットが、接続されたオートメーションシステムのアドレス範囲への 直接のビット単位またはバイト単位のアクセス(ショート: ビットアクセス/バイトアクセス)をサポートしているわけではありません。代わりに、BinWrite メカニズムを使用します。

ビットアクセス/バイトアクセス

ビットアクセス/バイトアクセスの通信ドライバのチャンネルユニットで、必要なビットまたはバイトを直接読み書きできます。

以下の図では、ビット x に、直接のビットアクセス/バイトアクセスを介して、値=1 が割り当てられます。



BinWrite メカニズム

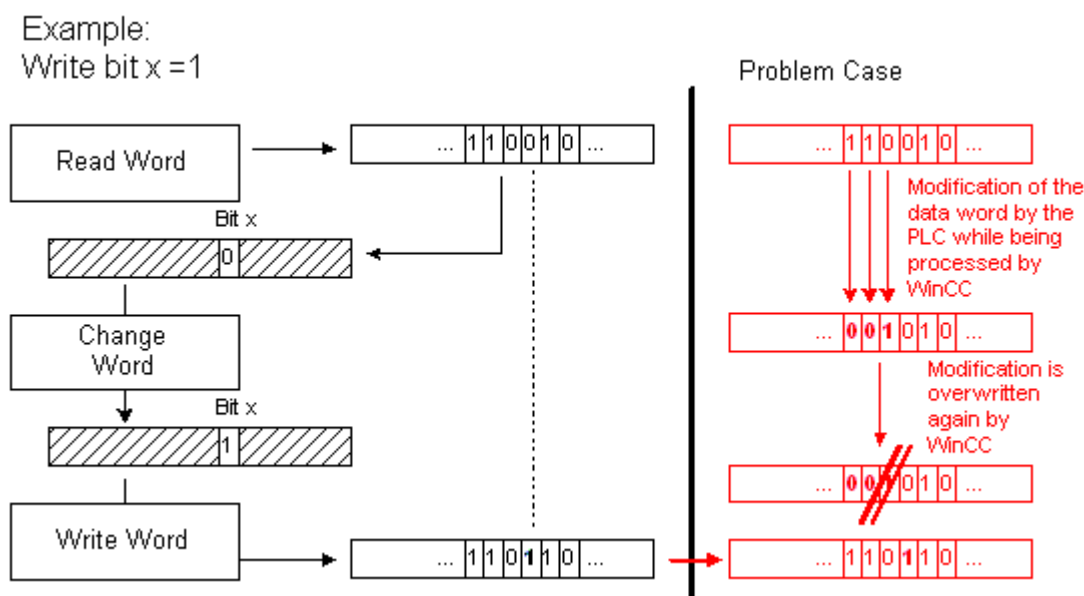
以下の通信ドライバはビットアクセス/バイトアクセスをサポートしておらず、代わりにそれぞれのチャンネルユニットに対して BinWrite メカニズムを使用します。

- Modbus Serial
- SIMATIC S5 Ethernet 第 4 層
- SIMATIC S5 Programmers Port AS511
- SIMATIC S5 Serial 3964R

- SIMATIC TI Ethernet 第4層
- SIMATIC TI Serial

ビットまたはバイトを書き込むには、チャンネルユニットは、まず **BinWrite** メカニズムでデータワード全体を読み取ります。アドレス指定対象のデータが、読み込まれたワードで変更されます。次に、変更されたビットまたはバイトではなく、ワード全体(!)が書き戻されます。

以下の図では、ビット x に、BinWrite メカニズムで値 = 1 が割り当てられます。



注記

データのワードが、WinCC で BinWrite メカニズムによってこのデータのワードが読み込まれたのと同時に AS で変わる場合("問題の場合"の図を参照)、WinCC がデータのワードを書き戻すとすぐに、AS でこの変更が失われます。

5.3.4.7 BinWrite 付きタグの設定方法

はじめに

ビットアクセス/バイトアクセスをサポートしない通信ドライバのチャンネルユニットに"2進タグ"を設定する場合、ダイアログを使って **BinWrite** メカニズムを有効にして、設定しなければなりません。そうしないと、このメカニズムは存在しません。


必要条件

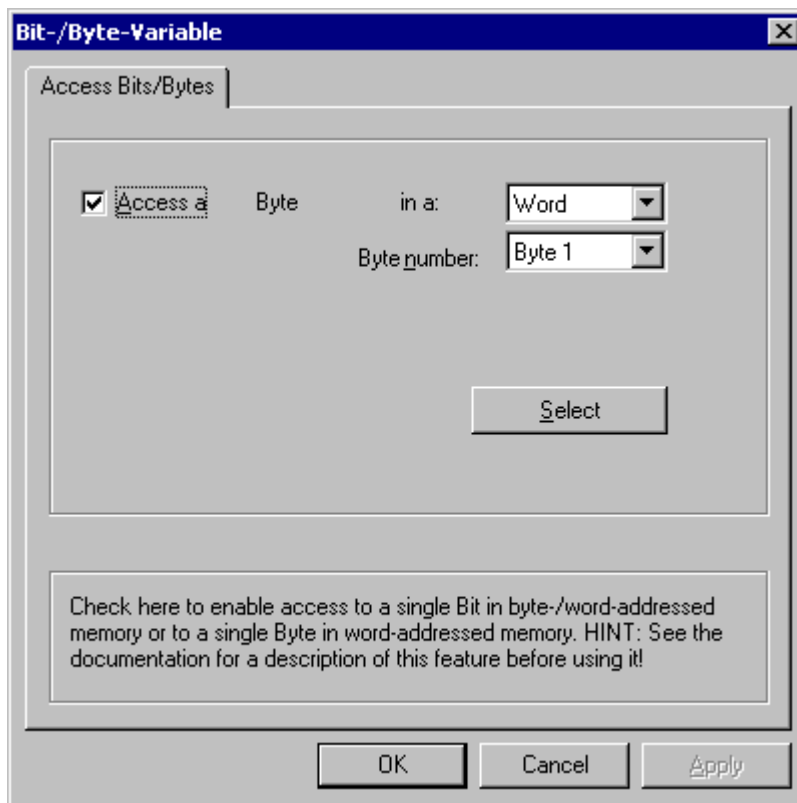
- 必要な通信プロセッサとハードウェアドライバがインストールされている。
- 例えば「SIMATIC S5 Ethernet Layer 4」のような、必要な通信ドライバがインストールされている。ただし、これはビットアクセス/バイトアクセスをサポートしません。
- そのチャンネルユニットに基いて、接続が既に作成されている。

手順

1. タグを設定する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. 次のいずれかのデータタイプを選択します。
 - 「2 進タグ」
 - 「符号なし 8 ビット値」
 - 「符号付き 8 ビット値」

5.3 WinCC プロセス通信

5. [ビットタグ/バイトタグ]ダイアログを開きます。
この目的で、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。
[ビットタグ/バイトタグ]ダイアログが開きます。AS のアドレス範囲は、ビットアクセス/バイトアクセス付きチャンネルのボタンによって設定されます。



6. [1 ビットにアクセス]または[1 バイトにアクセス]チェックボックスを有効にして、通常の設定を行ないます。
このダイアログの表示は、ステップ 2 で選択したデータタイプによって異なります。
7. [OK]をクリックしてダイアログを閉じます。

5.3.5 Ethernet 経由のカップリングのポートアドレス

ポートアドレス



注意

Ethernet 通信

Ethernet ベースの通信を使用する場合、自分のデータネットワークのセキュリティはエンドユーザーの責任です。例えば目標とされる攻撃によってデバイスが過負荷になる場合、機能は保証されません。

Ethernet によって接続される場合、ポートアドレスに関する情報が必要になることがあります。この情報は、ファイアウォールまたはルーターの設定に必要です。WinCC アプリケーションがデフォルトで使用するポートアドレスが、表に示されています。

	ポートアドレス TCP/IP	ポートアドレ ス UDP
S7 通信	102	
HTTP (通信 ; 転送)	80	
HTTPS (通信 ; 転送)	443	
WebServices (SOAP)	80 HTTP 443 HTTPS	
OPC-XML (OPC マスタとしての CE)	80 HTTP 443 HTTPS	
SendEmail	25	
転送(Ethernet 経由 ; CE-Stub ; PC ローダー ; PC)	2308 代替 50523	
ロギング(Ethernet 経由) CSV ファイル	139, 445	137, 138
Modbus Ethernet	502	
Allen-Bradley Ethernet CIP	44818	
Allen-Bradley Ethernet CSP2	2222	

5.3.6 SIMATIC S7-PLCSIM Advanced

WinCC は SIMATIC S7-PLCSIM Advanced シミュレーションソフトウェアを使用して仮想コントローラのシミュレーションをサポートします。

詳細な情報は、PLCSIM 製品ドキュメントに記載されています。

- Industry Online Support: SIMATIC S7-PLCSIM Advanced (<https://support.industry.siemens.com/cs/jp/ja/ps/24466/man>)

下記も参照

Industry Online Support: SIMATIC S7-PLCSIM Advanced (<https://support.industry.siemens.com/cs/jp/ja/ps/24466/man>)

通信チャンネル

6.1 Allen Bradley - Ethernet IP

6.1.1 WinCC チャンネル"Allen Bradley - Ethernet IP"

はじめに

チャンネル"Allen Bradley - Ethernet IP"は、Allen-Bradley オートメーションシステムをリンクするために使用します。通信は、Ethernet IP プロトコルで処理されます。

使用する通信ハードウェアに応じて、システムでは以下のチャンネルユニット経由の通信がサポートされます。

- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x
- Allen Bradley E/IP ControlLogix

6.1.2 チャンネルユニットの割り付け

はじめに

WinCC から既存のまたは予定のネットワークへの接続を作成するために、チャンネル用のチャンネルユニットを選択しなくてはなりません。

6.1 Allen Bradley - Ethernet IP

チャンネルユニットの割り付け

以下の表に、チャンネル"Allen Bradley - Ethernet IP"のチャンネルユニットのネットワークおよびオートメーションシステム(AS)への割り付けを示します。

チャンネルのチャンネルユニット	通信ネットワーク	AS
Allen Bradley E/IP PLC5	Ethernet/IP	Ethernet ポート付き PLC-5
Allen Bradley E/IP SLC50x	Ethernet/IP	SLC5/05 などの Ethernet ポート付き SLC-500
Allen Bradley E/IP ControlLogix	Ethernet/IP	ControlLogix 5500

6.1.3 サポートされるデータタイプ

概要

論理接続に必要なタグを定義します。以下のデータタイプが、「Allen Bradley - Ethernet IP」チャンネルによってサポートされます。

- 2 進タグ
- 符号付き 8 ビット値
- 符号なし 8 ビット値
- 符号付き 16 ビット値
- 符号なし 16 ビット値
- 符号付き 32 ビット値
- 符号なし 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- テキストタグ、8 ビットフォント(最大長:82 文字)
- テキストタグ、16 ビットフォント(最大長:82 文字)

6.1.4 チャンネルの設定

6.1.4.1 チャンネル"Allen Bradley - Ethernet IP"の設定

はじめに

WinCC では、オートメーションシステム(AS)との WinCC の通信に、論理接続が必要です。このセクションでは、"Allen Bradley - Ethernet IP"チャンネルをどのように設定するかを示します。

TCP/IP プロトコルの実装時に、論理接続用に AS の IP アドレスを定義します。IP アドレスは、ドットで区切られた 4 つの数値で構成されます。数値は 0～255 の範囲でなければなりません。

注記

タイムアウト動作

TCP/IP プロトコルの使用時には、中断された接続を即座には検出できません。メッセージのチェックバックに 1 分間かかります。

接続可能なコントローラ

以下の Allen-Bradley PLC に対する接続を、実装できます。

- Allen-Bradley ControlLogix 5500
- Allen-Bradley CompactLogix 5300
- Ethernet ポート付き PLC-5
- SLC5/05 などの Ethernet ポート付き SLC-500
- MicroLogix

リリースされている通信タイプ

以下の通信タイプはシステムテスト済みであり、"Allen Bradley - Ethernet IP"チャンネル用にリリースされています。

- ポイントツーポイント接続：
- オプションの量のコントローラとの、WinCC ステーションからの複数ポイント接続。

6.1 Allen Bradley - Ethernet IP

オンライン設定

"Allen Bradley - Ethernet IP"チャンネルのオンライン設定は、サポートされていません。

6.1.4.2 [Allen Bradley - Ethernet IP]チャンネル用に接続を設定する方法

はじめに

[Allen Bradley - Ethernet IP]チャンネルは、3 つのチャンネルユニット用に設定できます。

- Allen Bradley E/IP ControlLogix
- Allen Bradley E/IP PLC5
- Allen Bradley E/IP SLC50x

設定は、3 つすべてのチャンネルユニットについて同一で、次のタスクから構成されています。

1. 接続の設定
2. タグの設定

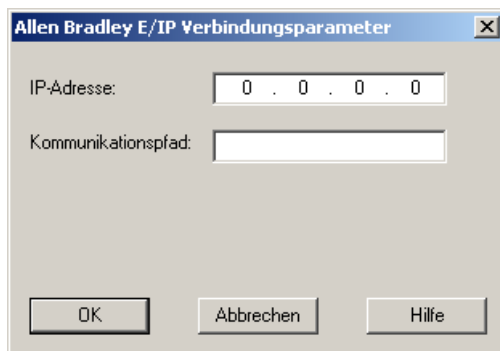
必要条件

- チャンネル"Allen Bradley - Ethernet IP"の通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理で、必要なチャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
新しい接続が作成されます。
3. 新しい接続を選択します。
[一般]の下にある[プロパティ - 接続]エリアで、接続名を入力します。

4. ショートカットメニューで[接続パラメータ]接続を選択します。
[Allen Bradley E/IP 接続パラメータ]ダイアログが開きます。



5. コントローラの Ethernet/IP モジュールの IP アドレスを入力します。
デフォルトでは Ethernet IP デバイス用に、ポート 44818 が永久的に設定されています。
6. [通信パス]フィールドで CIP パスを Ethernet モジュールからコントローラに定義します。
直接接続およびルーティング経由の接続を設定できます。
異なる CIP ネットワークに位置していたとしても、PLC と Ethernet モジュールとの間に論理的な接続を作成します。
7. [OK]ボタンをクリックして、ダイアログを閉じます。

6.1.4.3 例：通信パス

例 1:直接接続

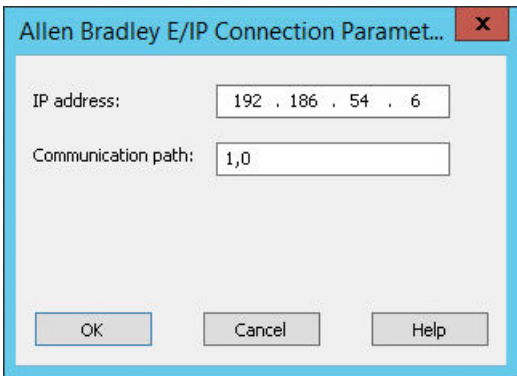
同じ Allen-Bradley ラック上の PLC との接続。

通信パス:

- 1,0

番号	意味
1	バックプレーン接続を意味します。
0	CPU スロット番号を意味します。

6.1 Allen Bradley - Ethernet IP



例 2:ルーティング経由の接続

PLC を他の Allen-Bradley ラックと接続します。

2 つの Allen-Bradley ラックが Ethernet を使用してネットワーク接続されています。

2 つの同一のチャンネルユニットは、2 つの異なるチャンネルユニットと同じ方法で設定されます。例:

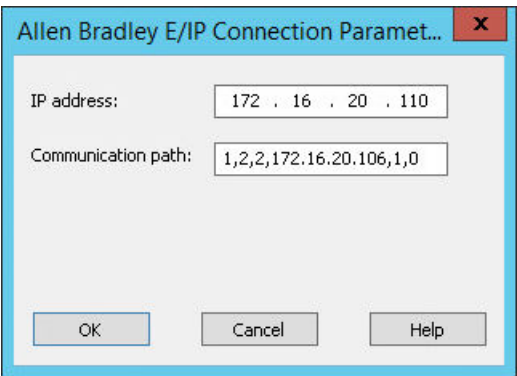
- WinCC - Control Logix (1) - Control Logix (2)
- WinCC - Control Logix (1) - SLC50x (1)

例えば SLC50x または Control Logix へのルーティングには異なるプロトコルが使用されますが、アドレス構造は引き続き同じです。

通信パス:

- 1,2,2,172.16.20.106,1,0

番号	意味
1	最初のモジュール「Control Logix (1)」のバックプレーン接続。
2	第 2 の Ethernet モジュールの CPU スロット番号を意味します。
2	Ethernet 接続を意味します。
172.16.20.106	例えば「Control Logix (2)」モジュールの 3 番目の Ethernet モジュールなど、ネットワークのリモート AB ラックの IP アドレス。
1	例えば「Control Logix (2)」または「SLC50x (1)」など、2 番目のモジュールのバックプレーン接続。
0	CPU のスロット番号



例 3:ルーティングを介した複数モジュールの接続

PLC を他の Allen-Bradley ラックと接続します。

2 つ以上の Allen-Bradley ラックが Ethernet に接続されています。

シナリオ:

- WinCC - Control Logix (1) - Control Logix (2) - Control Logix (3)
- WinCC - Control Logix (1) - Control Logix (2) - SLC50x (1)

例えば SLC50x または Control Logix へのルーティングには異なるプロトコルが使用されますが、アドレス構造は引き続き同じです。

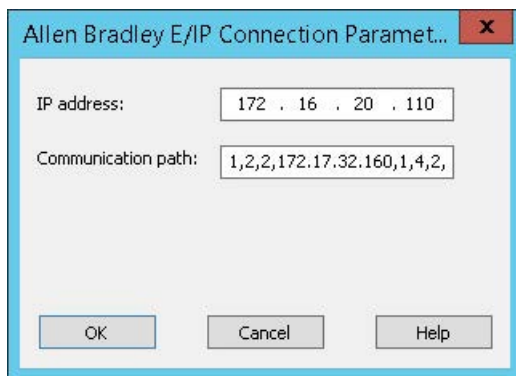
通信パス:

- 1,2,2,172.17.32.160,1,4,2,172.17.32.156,1,0

番号	意味
1	最初のモジュール「Control Logix (1)」のバックプレーン接続。
2	第 2 の Ethernet モジュールの CPU スロット番号を意味します。
2	Ethernet 接続を意味します。
172.17.32.160	例えば「Control Logix (2)」モジュールの 1 番目の Ethernet モジュールなど、ネットワークのリモート AB ラックの IP アドレス。
1	2 番目のモジュール「Control Logix (2)」のバックプレーン接続。
4	3 番目の Ethernet モジュールのスロット番号を意味します。
2	Ethernet 接続を意味します。
172.17.32.156	例えば「Control Logix (3)」や「SLC50x (1)」モジュールなど、ネットワークの別のリモート AB ラックの IP アドレス。

6.1 Allen Bradley - Ethernet IP

番号	意味
1	例えば「Control Logix (3)」または「SLC50x (1)」など、3 番目のモジュールのバックプレーン接続。
0	CPU のスロット番号



6.1.4.4 タグの設定

タグの設定

はじめに

チャンネル[Allen Bradley - Ethernet IP]経由での WinCC とオートメーションシステム(AS)の間の接続用に、異なるデータタイプのタグを WinCC で作成できます。許可されるデータタイプのリストを、このセクションに示します。

タグの更新

複数のタグが同時に PLC からの画像で取得される場合、[Allen Bradley - Ethernet IP]チャンネルは、更新を最適化しようと試みます。ただし、これは以下の条件を満たす場合にのみ達成されます。

- タグが同じアドレス範囲にある。
- アドレス範囲内で、複数のタグができるだけ互いの近くにある。

これらの推奨事項を遵守しない場合は、多量のタグが付いている画像の更新で、顕著な相違が発生します。特定の環境下で、取得サイクルを維持できなくなる可能性があります。

タグ設定時に以下の規則を守ると、接続のパフォーマンスが最高になります。

- 同時に最大 2000 タグを更新します。
- なるべく少ないスペース(最善は 1 つのアドレス範囲だけ)で、タグを組み合わせます。

有効なデータタイプ

以下のリストのデータタイプを、タグの設定に使用できます。

基本データタイプ

データタイプ	ビットアドレススペース
Bool	-
SInt	0-7
USInt	0-7
Int	0-15
UInt	0-15
DInt	0-31
UDInt	0-31
Real	-
String	-

配列

アドレス	有効なデータタイプ
配列	SInt、USInt、Int、UInt、DInt、UDInt、Real

アドレス指定

アドレス指定

タグは、コントローラのアドレスを使って WinCC で一意に参照されます。アドレスは PLC のタグ名と対応する必要があります。タグアドレスは、最大 128 文字の長さの文字列で定義されます。

文字を使用したアドレス指定

タグのアドレス指定に有効な文字

- 文字(a～z、A～Z)
- 数値(0～9)
- 下線(_)

タグアドレスは、PLC でタグの指定に使用されるタグ名と他の文字列で構成されます。

タグ名プロパティ:

- タグ名の最初の文字には下線を使用できますが、最後の文字には使用できません。
- 連続した下線や空白文字の入った文字列は無効です。
- アドレスの長さは 128 文字を超えることはできません。

注記

タグアドレス指定用に予約されている文字は、プログラム名やタグ名、または他のアドレスインスタンスには使用できません。

予約されている文字は次のとおりです。

予約文字	ファンクション
.	エレメント区切り文字
:	プログラムタグの定義
,	多次元配列のアドレス指定用区切り文字
/	ビットアドレス指定用予約文字
[]	配列エレメントまたは配列のアドレス指定

コントローラおよびプログラムタグ

"Allen-Bradley E/IP ControlLogix"は、PLC タグのアドレス指定(グローバルプロジェクトタグ)やプログラムタグ(グローバルプログラムタグ)を可能にします。プログラムタグは、コントローラのプログラム名や実際のタグ名によって宣言されます。コントローラ変数は、その名前によってアドレス指定されます。

注記

アドレス指定エラー

アドレス指定エラーは、タグ名とデータタイプが一致しない時に発生します。

WinCC のアドレスフィールドに定義されているタグ名と、コントローラのタグ名が一致する必要があります。WinCC とコントローラのタグのデータタイプは、一致する必要があります。

注記

例えば入力モジュールや出力モジュールのデータのように、モジュール固有の Tags は直接アドレス指定できません。代わりに、コントローラの Alias tag を使用します。

例：Local:3:O.データは、WinCC でアドレス指定できません。

コントローラの Local:3:O に別名"MyOut"が定義されている場合、WinCC で MyOut.Data を使ってアドレス指定できます。

アドレス指定構文

アドレスの表記

以下のテーブルは、個々のアドレス指定の書き込みの可能性を定義しています。

表 6-1 配列、基本データタイプおよび構造エレメントへのアクセス

データタイプ	タイプ	アドレス
基本データタイプ	PLC タグ	タグ名
	プログラムタグ	Programname:tagname
配列	PLC タグ	配列タグ
	プログラムタグ	プログラム名：配列タグ
ビット	PLC タグ	Tagname/bitnumber
	プログラムタグ	Programname:tagname/bitnumber

6.1 Allen Bradley - Ethernet IP

データタイプ	タイプ	アドレス
構造エレメント	PLC タグ	構造タグ構造エレメント
	プログラムタグ	プログラム名：構造タグ、構造エレメント

注記

Bool、Real、String のデータタイプのビットアドレス指定はできません。アドレス障害が発生します。

構文の説明

構文の説明：

```
(Programname:) tagname ([x(, y) (, z)]) { .tagname ([x(, y) (, z)]) } (/
bitnumber)
```

- "()"は、オプションの、式の一つのインスタンスを定義します。
- "{}"は、単一のインスタンスを複数含むオプションの式を定義します。

アドレス文字列の長さは、128 文字を超えることはできません。

アドレス指定のタイプ

配列エレメント

PLC の一次元、二次元、三次元配列のエレメントは、インデックスを設定し、タグエディタで表記を対応させることでインデックス付けが行われます。エレメントのアドレス指定に有効なすべての基本タイプ配列の配列アドレス指定は、エレメント"0"から始まります。読み書き操作は、配列全体ではなく、アドレスエレメントのみで実行されます。

ビットとビットタグ

ビットアクセスは、Bool、Real、String 以外のすべての基本データタイプで行えます。配列/構造エレメントでのビットアドレス指定も行えます。Bool データタイプは WinCC で、基本データタイプのビットやビットタグのアドレス指定用に定義されます。

1 プレースのビット番号は、"/x"または"/Ix" (x = ビット番号)でアドレス指定されます。ビット番号は、最高 2 桁まで定義できます。

注記

データタイプが SInt、Int および DInt の"Bool"データタイプでは、指定されたビットの変更後に、タグ全体が PLC に再書き込みされます。その間、タグのその他のビットが変更されたかどうかについては確認されません。そのために、PLC は指定されたタグへの読み取りアクセス権限しかない場合があります。

構造

ユーザー定義のデータタイプは、構造を使って作成されます。これらの構造は、異なるデータタイプのタグをグループ化します。構造は、基本タイプ、配列および他の構造で構成できます。WinCC では、構造エレメントとしてアドレス指定できるのは基本データタイプだけあり、構造全体はできません。

構造エレメント

構造エレメントは、構造名と必要な構造エレメントを使って、アドレス指定します。このアドレス指定はピリオドによって区切られます。基本データタイプに加え、構造エレメントは配列または他の構造を表すことができます。一次元の配列のみを、構造エレメントとして使用できます。

注記

構造の入れ子の深さは、アドレス最長限界の 128 文字のみで制限されます。

アドレス指定の例

アドレス指定の表の例

以下の表に、コントロール変数に対する基本アドレス指定の種類を示します。組み合わせることにより、他のアドレス指定の変異形を作ることが可能です。

タイプ	タイプ	アドレス
全般	PLC タグ	タグ名
	プログラムタグ	Program:tagname

6.1 Allen Bradley - Ethernet IP

タイプ	タイプ	アドレス
配列	二次元配列のエレメントへのアクセス	Arraytag[Dim1,Dim2]
	構造配列(一次元)のエレメント	Arraytag[Dim1].structureelement
	基本タイプ配列(二次元)のエレメントのビット	Arraytag[Dim1,Dim2]/Bit
構造	構造の配列	Structuretag.arraytag
	サブ構造の配列のエレメントのビット	Structuretag.structure2.arraytag[element]/bit

注記

プログラムタグは、PLC より抽出されたプログラム名をアドレスの先頭に付け、コロンで区切ることでアドレス指定できます。

例 : Programname:arraytag[Dim1,Dim2]

配列エレメントへのアクセス

タイプ	アドレス
PLC タグ	Arraytag[Dim1]
	Arraytag[Dim1,Dim2]
	Arraytag[Dim1,Dim2,Dim3]
プログラムタグ	Programname:arraytag[Dim1]
	Programname:arraytag[Dim1,Dim2]
	Programname:arraytag[Dim1,Dim2,Dim3]

Allen Bradley E/IP ControlLogix チャンネルユニットのタグの設定方法


はじめに

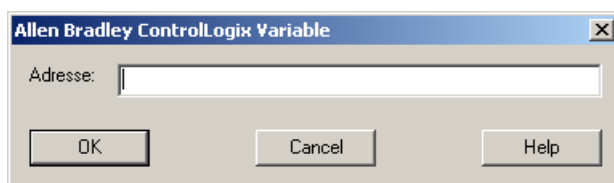
このセクションでは、オートメーションシステム(AS)のアドレス範囲におけるチャンネルユニット"Allen Bradley E/IP ControlLogix"のタグの設定方法を、示します。

必要条件

- チャンネル"Allen Bradley - Ethernet IP"を、このプロジェクトに統合しなければなりません。
- "Allen Bradley E/IP ControlLogix"チャンネルユニットに、接続を作成しなければなりません。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。 タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。
5. [Allen Bradley ControlLogix タグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. [アドレス]フィールドに、AS のタグのアドレスを入力します。

注記

アドレス指定エラー

アドレス指定エラーは、タグ名とデータタイプが一致しない時に発生します。

WinCC のアドレスフィールドに定義されているタグ名と、コントローラのタグ名が一致する必要があります。WinCC のタグのデータタイプとコントローラのデータタイプが、一致する必要があります。

7. [OK]ボタンをクリックして、ダイアログを閉じます。

Allen Bradley E/IP PLC5 または SLC50x 用の、ビット単位でアクセスするタグを設定する方法

はじめに


このセクションでは、オートメーションシステム(AS)のアドレスエリアにビット単位でアクセスするための、タグ設定方法を説明します。

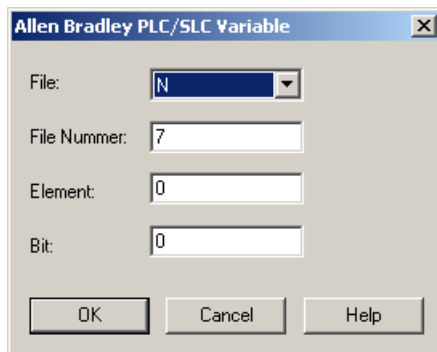
6.1 Allen Bradley - Ethernet IP

必要条件

- チャンネル"Allen Bradley - Ethernet IP"を、このプロジェクトに統合しなければなりません。
- "Allen Bradley E/IP PLC5"または"Allen Bradley E/IP SLC50x"チャンネルユニットで、接続を作成しなければなりません。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。
5. [Allen Bradley PLC/SLC タグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. [ファイル]フィールドでアドレス範囲を選択します。 エントリー N、R、C、T、B、S、I、O、D、A、ST が利用できます。
7. 示唆されているファイル番号と異なる番号の場合は、[ファイル番号]を入力します。
8. [要素]を入力します。
9. [ファイル]フィールドの設定によって、アドレスに"ビット"を定義するか、または"ビット(8進)"または"サブ"の値を選択します。
- 10.[OK]ボタンをクリックして、ダイアログを閉じます。

Allen Bradley E/IP PLC5 または SLC50x 用の、バイト単位でアクセスするタグを設定する方法


はじめに

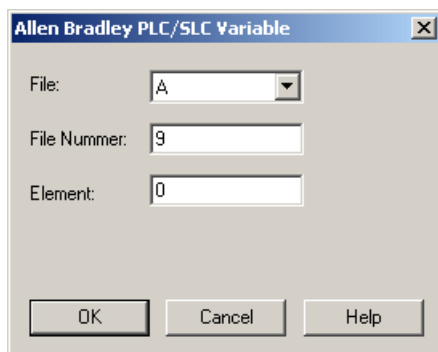
このセクションでは、オートメーションシステム(AS)のアドレスエリアへのバイト単位のアクセス用タグの設定方法を、説明します。

必要条件

- チャンネル"Allen Bradley - Ethernet IP"を、このプロジェクトに統合しなければなりません。
- "Allen Bradley E/IP PLC5"または"Allen Bradley E/IP SLC50x"チャンネルユニットで、接続を作成しなければなりません。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。
5. [Allen Bradley PLC/SLC タグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. A または ST アドレス範囲を、[ファイル]フィールドで選択します。
7. 示唆されているファイル番号と異なる番号の場合は、[ファイル番号]を入力します。
8. [要素]を入力します。
9. [OK]ボタンをクリックして、ダイアログを閉じます。

Allen Bradley E/IP PLC5 または SLC50x 用の、ワード単位でアクセスするタグを設定する方法

はじめに


このセクションでは、オートメーションシステム(AS)のアドレスエリアへのワード単位のアクセス用タグの設定方法を説明します。

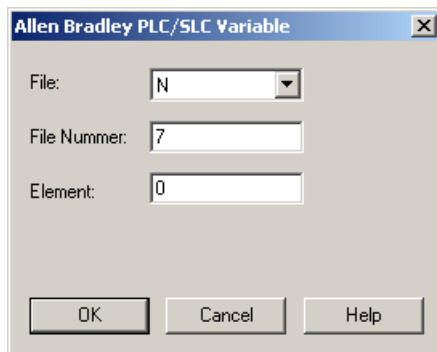
6.1 Allen Bradley - Ethernet IP

必要条件

- チャンネル"Allen Bradley - Ethernet IP"を、このプロジェクトに統合しなければなりません。
- "Allen Bradley E/IP PLC5"または"Allen Bradley E/IP SLC50x"チャンネルユニットで、接続を作成しなければなりません。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。
5. [Allen Bradley PLC/SLC タグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. [ファイル]フィールドでアドレス範囲を選択します。 エントリー N、R、C、T、B、S、I、O、D、A、ST が利用できます。
7. 示唆されているファイル番号と異なる番号の場合は、[ファイル番号]を入力します。 ファイル番号は設定"S"用に変更できません。
8. [要素]を入力します。
9. 表示されている場合、[サブ]フィールドの値を 1 つ選択します。 これは[ファイル]フィールドの設定に依存します。
- 10.[OK]ボタンをクリックして、ダイアログを閉じます。

Allen Bradley E/IP PLC5 または SLC50x のテキストタグの設定方法


はじめに

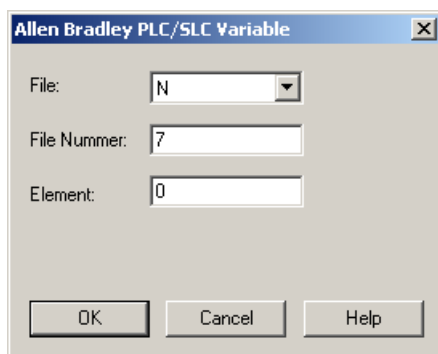
このセクションでは、オートメーションシステム(AS)のアドレスエリアへのワード単位のアクセス用タグの設定方法を説明します。

必要条件

- チャンネル"Allen Bradley - Ethernet IP"を、このプロジェクトに統合しなければなりません。
- "Allen Bradley E/IP PLC5"または"Allen Bradley E/IP SLC50x"チャンネルユニットで、接続を作成しなければなりません。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。
5. [Allen Bradley PLC/SLC タグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. A または ST アドレス範囲を、[ファイル]フィールドで選択します。
7. 示唆されているファイル番号と異なる番号の場合は、[ファイル番号]を入力します。ファイル番号は設定"S"用に変更できません。
8. [要素]を入力します。
9. 表示されている場合、[サブ]フィールドの値を 1 つ選択します。これは[ファイル]フィールドの設定に依存します。
10. [OK]ボタンをクリックして、ダイアログを閉じます。

6.2 三菱製 Ethernet

6.2.1 WinCC チャンネル"三菱製 Ethernet"

はじめに

「三菱製 Ethernet」チャンネルは、WinCC ステーションと三菱製コントローラの間の通信用です。

通信は MELSEC 通信プロトコル(MC プロトコル)によって行われます。

チャンネルユニット

WinCC チャンネル「三菱製 Ethernet」には、次のチャンネル単位があります。

- 三菱製 FX3U シリーズ
- 三菱製 Q シリーズ
- 三菱製 iQ-R シリーズ
- 三菱製 iQ-F シリーズ

6.2.2 サポートされるデータタイプ

はじめに

WinCC と接続されるコントローラの間の論理接続に必要なタグを定義します。

以下のデータタイプが、「三菱製 Ethernet」チャンネルによってサポートされます。

- 2 進タグ
- 符号付き 16 ビット値
- 符号なし 16 ビット値
- 符号付き 32 ビット値
- 符号なし 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- 浮動小数点数 64 ビット IEEE 754
- テキストタグ 8 ビット文字セット

- テキストタグ 16 ビット文字セット
- 未処理データタグ

6.2.3 チャンネルの設定

6.2.3.1 "三菱製 Ethernet"チャンネルの設定

はじめに

WinCC では、オートメーションシステム(AS)との WinCC の通信に、論理接続が必要です。
このセクションでは、「三菱製 Ethernet」チャンネルを設定する方法を説明します。

チャンネルの設定

"三菱製 Ethernet"チャンネルを設定するには、以下のステップが必要です:

1. 接続を設定します。
2. タグを設定します。

注記

コントローラの接続設定

三菱製コントローラを使用する場合、コントローラの接続も設定する必要があります。
これを行うには、対応する製造メーカー文書を使用します。

オンライン設定

「三菱製 Ethernet」チャンネルは、タグおよび接続のオンライン設定をサポートします。

サポートされる三菱製コントローラ

以下の三菱製コントローラで論理接続を設定できます。

- MELSEC FX3U シリーズ
- MELSEC システム Q
- MELSEC システム iQ-R
- MELSEC システム iQ-F

6.2 三菱製 Ethernet

接続およびタグを設定する場合、両方のコントローラファミリで手順は同じです。この設定は、特定のコントローラファミリの使用可能なアドレスタイプに関してのみ異なります。

情報のルーティングは、MELSEC システム Q と MELSEC システム iQ-R シリーズのモデルのみにサポートされています。

プロトコル

トランスポートプロトコルとして TCP/IP または UDP/IP プロトコルを使用して、AS との接続を確立できます。

「三菱製 Ethernet」チャンネルの論理接続を設定します。

- AS の IP アドレスと IP ポート番号を入力します
IP アドレスは、ドットで区切られた 4 つの数値で構成されます。数値は 0～255 の範囲でなければなりません。
- トランスポートプロトコルとして UDP または TCP を選択します

注記

タイムアウト動作

TCP/IP プロトコルの使用時には、中断された接続を即座には検出できません。
フィードバックが長くなる場合があります、オペレーティングシステムから独立です。

下記も参照

"三菱製 FX3U シリーズ"チャンネルユニット接続を設定する方法 (ページ 228)

"三菱製 Q シリーズ"チャンネルユニット接続を設定する方法 (ページ 230)

「三菱製 iQ シリーズ」チャンネルユニット接続を設定する方法 (ページ 231)

タグの設定 (ページ 234)

6.2.3.2 "三菱製 FX3U シリーズ"チャンネルユニット接続を設定する方法

はじめに

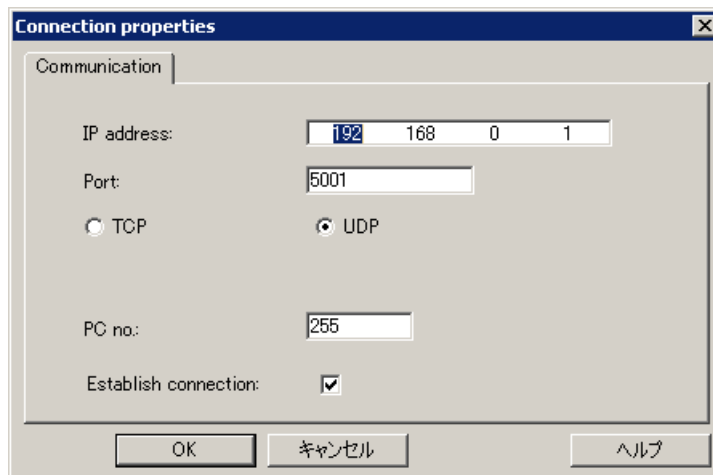
このセクションでは、"三菱製 FX3U シリーズ"チャンネルユニットの接続を設定する方法を説明します。

必要条件

- "三菱製 Ethernet"チャンネルの通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理のナビゲーションエリアで、"三菱製 Ethernet"通信ドライバのツリーの[三菱製 FX3U シリーズ]チャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。
4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。



5. コントローラの IP アドレスを入力します。
6. TCP/IP 接続に使用するポートを入力します。
値の有効範囲は 0～65535 です。
7. 使用するポート(「TCP」または「UDP」)を選択します。

注記

推奨事項:UDP

デフォルトのプロトコルである"UDP"を使用することを推奨いたします。このプロトコルの方がタイムアウト動作に優れています。

UDP を使用できない場合にのみ TCP を使用してください。

8. PC 番号を入力します。
PC 番号を入力したくない場合は、255 または 0 のどちらかの値を入力する必要があります。
9. 接続を確立するには、[接続の確立]を選択します。
10. どの場合も、[OK]を押してダイアログを閉じます。

6.2 三菱製 Ethernet

下記も参照

"三菱製 Ethernet"チャンネルの設定 (ページ 227)

タグの設定方法 (ページ 237)

6.2.3.3 "三菱製 Q シリーズ"チャンネルユニット接続を設定する方法

はじめに

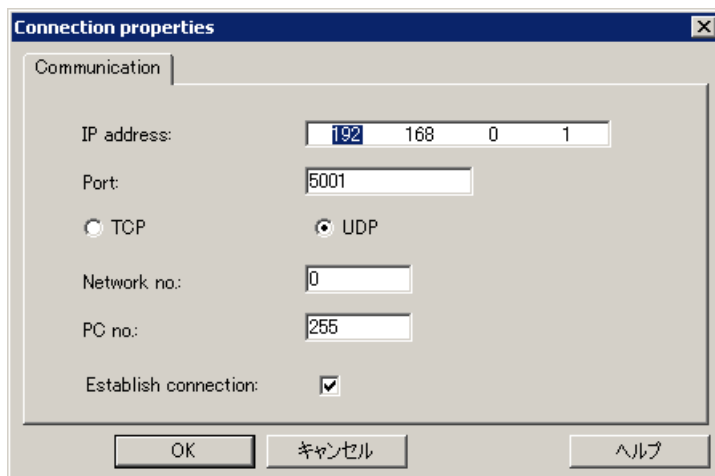
このセクションでは、"三菱製 Q シリーズ"チャンネルユニットの接続を設定する方法を説明します。

必要条件

- "三菱製 Ethernet"チャンネルの通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理のナビゲーションエリアで、「三菱製 Ethernet」通信ドライバのツリーの「三菱製 Q シリーズ」チャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。
4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。



5. コントローラの IP アドレスを入力します。
6. TCP/IP 接続に使用するポートを入力します。
値の有効範囲は 0～65535 です。

7. 使用するポート(「TCP」または「UDP」)を選択します。

注記**推奨事項:UDP**

デフォルトのプロトコルである"UDP"を使用することを推奨いたします。このプロトコルの方がタイムアウト動作に優れています。

UDP を使用できない場合にのみ TCP を使用してください。

8. ネットワーク番号を入力します。
デフォルト設定は 0 です。
9. PC 番号を入力します。
PC 番号を入力したくない場合は、255 または 0 のどちらかの値を入力する必要があります。
10. 接続を確立するには、[接続の確立]を選択します。
11. [OK]をクリックして、それぞれのダイアログボックスを閉じます。

下記も参照

"三菱製 Ethernet"チャンネルの設定 (ページ 227)

タグの設定方法 (ページ 237)

6.2.3.4 「三菱製 iQ シリーズ」チャンネルユニット接続を設定する方法**はじめに**

このセクションでは、「三菱製 iQ シリーズ」チャンネルユニットの接続を設定する方法を説明します。

必要条件

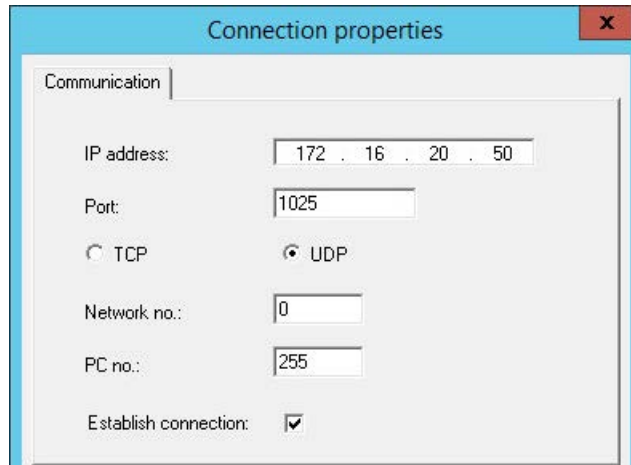
- "三菱製 Ethernet"チャンネルの通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理のナビゲーションエリアで、「三菱製 Ethernet」通信ドライバのツリーの「三菱製 iQ シリーズ」チャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。

6.2 三菱製 Ethernet

4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。



5. コントローラの IP アドレスを入力します。
6. TCP/IP 接続に使用するポートを入力します。
値の有効範囲は 0～65535 です。ポート 1025 はデフォルトで設定されています。
7. 使用するポート(「UDP」または「TCP」)を選択します。

注記**推奨事項:UDP**

デフォルトのプロトコルである"UDP"を使用することを推奨いたします。このプロトコルの方がタイムアウト動作に優れています。

UDP を使用できない場合にのみ TCP を使用してください。

8. ネットワーク番号を入力します。
デフォルト設定は 0 です。
9. PC 番号を入力します。
PC 番号を入力したくない場合は、255 または 0 のどちらかの値を入力する必要があります。
10. 接続を確立するには、[接続の確立]を選択します。
11. [OK]をクリックして、それぞれのダイアログボックスを閉じます。

下記も参照

"三菱製 Ethernet"チャンネルの設定 (ページ 227)

タグの設定方法 (ページ 237)

6.2.3.5 「三菱製 iQ-F シリーズ」チャンネルユニットの接続設定方法

はじめに

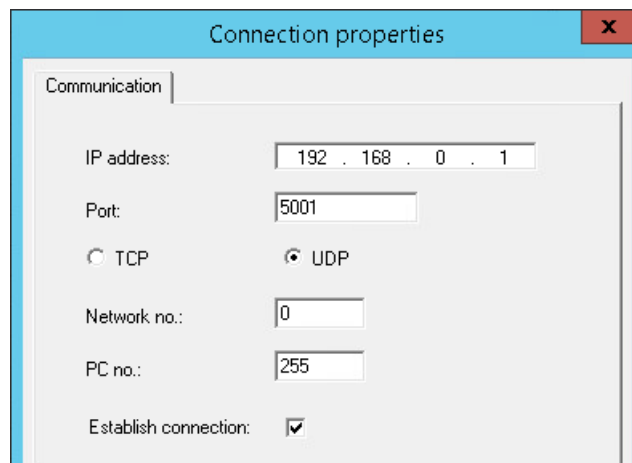
このセクションでは、「三菱製 iQ-F シリーズ」チャンネルユニットの接続を設定する方法を説明します。

必要条件

- "三菱製 Ethernet"チャンネルの通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理のナビゲーションエリアで、「三菱製 Ethernet」通信ドライバのツリーの「三菱製 iQ-F シリーズ」チャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。
4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。



5. コントローラの IP アドレスを入力します。
6. TCP/IP 接続に使用するポートを入力します。
値の有効範囲は 0～65535 です。ポート 1025 はデフォルトで設定されています。

6.2 三菱製 Ethernet

7. 使用するポート(「UDP」または「TCP」)を選択します。

注記

推奨事項:UDP

デフォルトのプロトコルである"UDP"を使用することを推奨いたします。このプロトコルの方がタイムアウト動作に優れています。

UDP を使用できない場合にのみ TCP を使用してください。

8. ネットワーク番号を入力します。
デフォルト設定は 0 です。
9. PC 番号を入力します。
PC 番号を入力したくない場合は、255 または 0 のどちらかの値を入力する必要があります。
10. 接続を確立するには、[接続の確立]を選択します。
11. [OK]をクリックして、それぞれのダイアログボックスを閉じます。

下記も参照

"三菱製 Ethernet"チャンネルの設定 (ページ 227)

タグの設定方法 (ページ 237)

6.2.3.6 タグの設定

タグの設定

概要

"三菱製 Ethernet"チャンネルによる WinCC と自動システムとの間の接続に対して、WinCC で異なるデータタイプのタグを作成できます。

以下のセクションでは、タグの設定方法について説明します。

アドレスタイプ、アドレス範囲およびデータタイプ

次の表は、タグおよび構造タグを設定するときに使用できるアドレスタイプ、アドレス範囲およびデータタイプを示します。

さらに、それぞれのアドレスタイプでサポートされているオートメーションシステム(AS)が指定されています。

- FX:MELSEC FX3U シリーズ
- Q:MELSEC システム Q

- iQR:MELSEC システム iQ-R
- iQF:MELSEC システム iQ-F

注記

WinCC データタイプはアドレスタイプに依存します

特定の WinCC データタイプを使用する場合は、選択したアドレスタイプを考慮しなければなりません。たとえば、アドレスタイプ"D"は WinCC データタイプ"バイナリ"をサポートしません。

アドレスタイプ	コード	表示	データタイプ	サポートされる AS
リレー				
リンクリレー	B	16 進数	ビット	Q、iQR、iQF
入力リレー(ダイレクト)	DX	16 進数	ビット	Q、iQR
出力リレー(ダイレクト)	DY	16 進数	ビット	Q、iQR
ラッチリレー	L	10 進数	ビット	Q、iQR、iQF
特殊なリンクリレー	SB	16 進数	ビット	Q、iQR、iQF
エッジリレー	V	10 進数	ビット	Q、iQR
入力リレー	X	16 進数(FX:8 進数)	ビット	FX、Q、iQR、iQF
出力リレー	Y	16 進数(FX:8 進数)	ビット	FX、Q、iQR、iQF
フラグ				
エラーフラグ (Annunciator)	F	10 進数	ビット	Q、iQR、iQF
フラグ(Internal relay)	M	10 進数	ビット	FX、Q、iQR、iQF
ステップフラグ(Step relay)	S	10 進数	ビット	Q、iQF
診断フラグ(Special relay)	SM	10 進数	ビット	Q、iQR、iQF
レジスタ				
データレジスタ	D	10 進数	ワード	FX、Q、iQR、iQF
拡張レジスタ(File register)	R	10 進数	16 ビット	FX、Q、iQF
拡張レジスタ	R	10 進数	ワード	iQR
データ更新レジスタ	RD	10 進数	ワード	iQR

6.2 三菱製 Ethernet

アドレスタイプ	コード	表示	データタイプ	サポートされる AS
診断レジスタ (Special register)	SD	10 進数	ワード	Q、iQR、iQF
特殊なリンクレジスタ	SW	16 進数	ワード	Q、iQR、iQF
リンクレジスタ	W	16 進数	ワード	Q、iQR、iQF
インデックスレジスタ	Z	10 進数	ワード	Q、iQR、iQF
倍長インデックスレジスタ	LZ	10 進数	倍長ワード	iQF
ファイルレジスタ (シリアル番号によるアクセス)	ZR	16 進数	ワード	Q、iQR
カウンタ				
カウンタ/コイル	CC	10 進数	ビット	Q、iQR、iQF
カウンタ/現在値	CN	10 進数	ワード	FX、Q、iQR、iQF
カウンタ/コンタクト	CS	10 進数	ビット	FX、Q、iQR、iQF
ロングカウンタ (コイル)	LCC	10 進数	ビット	iQR、iQF
ロングカウンタ (現在値)	LCN	10 進数	倍長ワード	iQR、iQF
ロングカウンタ (コンタクト)	LCS	10 進数	ビット	iQR、iQF
タイマ				
ロング保持タイマ (現在値)	LSTN	10 進数	倍長ワード	iQR
ロングタイマ (現在値)	LTN	10 進数	倍長ワード	iQR
保持タイマ (コイル)	SC	10 進数	ビット	Q、iQF
保持タイマ (現在値)	SN	10 進数	ワード	Q、iQF
保持タイマ (コンタクト)	SS	10 進数	ビット	Q、iQF
保持タイマ (コイル)	STC	10 進数	ビット	iQR
保持タイマ (現在値)	STN	10 進数	ワード	iQR
保持タイマ (コンタクト)	STS	10 進数	ビット	iQR
タイマ (OUT コイル)	TC	10 進数	ビット	Q、iQR、iQF
タイマ (現在値)	TN	10 進数	ワード	FX、Q、iQR、iQF
タイマ (コンタクト)	TS	10 進数	ビット	FX、Q、iQR、iQF

下記も参照

タグの設定方法 (ページ 237)

タグの設定方法

はじめに

このセクションでは、自動システム(AS)のアドレスエリアでタグアクセスを設定する方法を説明します。


必要条件

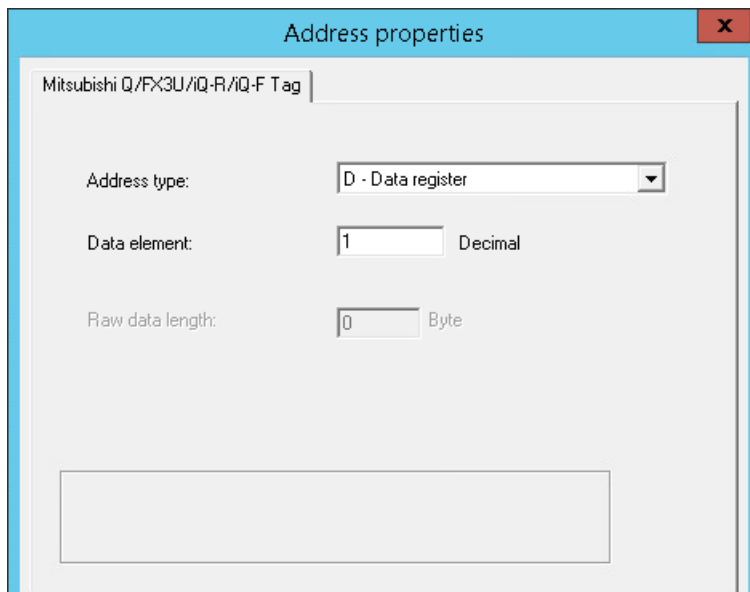
- "三菱製 Ethernet"チャンネルはプロジェクトに統合されます。
- チャンネルユニットの 1 つに接続が作成されます。
 - 三菱製 FX3U シリーズ
 - 三菱製 Q シリーズ
 - 三菱製 iQ-R シリーズ
 - 三菱製 iQ-F シリーズ

手順

1. タグを設定する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、希望するデータタイプを選択します。

6.2 三菱製 Ethernet

5. [アドレスプロパティ]ダイアログを開きます。
この目的で、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。



6. アドレスタイプを設定します。
7. 対応するデータ要素を入力します。
値はコントローラの設定に依存します。
8. 必要であれば、未処理のデータの長さを入力します。
9. [OK]をクリックして、[アドレスプロパティ]ダイアログを閉じます。
タグのアドレスは、[タグのプロパティ]ダイアログの[アドレス]フィールドに表示されます。
アドレスは、AS のデータフォーマットに調整されます。

下記も参照

"三菱製 FX3U シリーズ"チャンネルユニット接続を設定する方法 (ページ 228)

"三菱製 Q シリーズ"チャンネルユニット接続を設定する方法 (ページ 230)

「三菱製 iQ シリーズ」チャンネルユニット接続を設定する方法 (ページ 231)

タグの設定 (ページ 234)

6.3 Modbus TCP/IP

6.3.1 "Modbus TCP/IP"チャンネル

はじめに

"Modbus TCP/IP"チャンネルは、WinCC ステーションと、Ethernet 経由の Modbus をサポートする PLC の間の、通信用です。この通信は、Modbus TCP/IP プロトコルで処理されます。

チャンネルユニット

"Modbus TCP/IP"チャンネルは、"Modbus TCP/IP Unit #1"チャンネルユニットと一緒にあります。

6.3.2 サポートされるデータタイプ

はじめに

接続されるコントローラとの論理接続に必要なタグを定義します。以下のデータタイプが、"Modbus TCP/IP"チャンネルによってサポートされます。

- 2 進タグ
- 符号付き 16 ビット値
- 符号なし 16 ビット値
- 符号付き 32 ビット値
- 符号なし 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- テキストタグ 8 ビット文字セット
- テキストタグ 16 ビット文字セット

6.3 Modbus TCP/IP

6.3.3 チャンネルの設定

6.3.3.1 "Modbus TCP/IP"チャンネルの設定

はじめに

WinCC では、オートメーションシステム(AS)との WinCC の通信に、論理接続が必要です。このセクションでは、"Modbus TCP/IP Unit #1"チャンネルユニットとの通信について説明します。全ての接続固有のパラメータは、設定時に定義されます。

TCP/IP プロトコルの実装時に、論理接続用に AS の IP アドレスを定義します。IP アドレスは、ドットで区切られた 4 つの数値で構成されます。数値は 0~255 の範囲でなければなりません。

注記

タイムアウト動作

TCP/IP プロトコルの使用時には、中断された接続を即座には検出できません。メッセージのチェックバックに 1 分間かかります。

Modbus TCP/IP を使用する有効な通信方法

次の通信タイプはシステムでテスト済みで、承認されています。

- ポイントツーポイント通信
- オプションの量のコントローラを使用する WinCC ステーションからの複数ポイント接続。

注記

WinCC ステーションは Modbus マスターとして動作するので、Modbus ネットワークのブリッジ経由で WinCC ステーションを統合させることはできません。

オンライン設定

オンライン設定はサポートされません。

6.3.3.2 接続の設定方法

概要

チャンネル"Modbus TCP/IP"を設定するには、以下のステップが必要です。

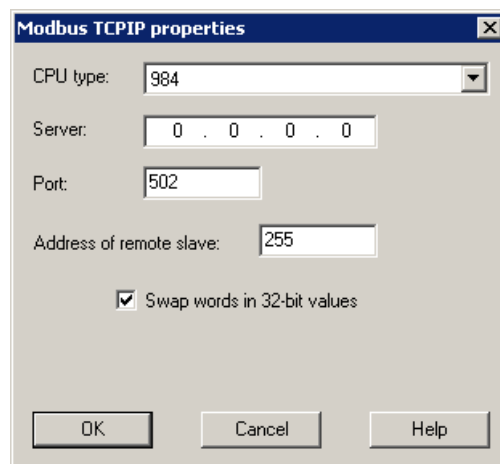
1. 接続の設定
2. タグの設定

必要条件

- チャンネル"Modbus TCP/IP"の通信ドライバがインストールされ、プロジェクトに統合されている。

手順

1. タグ管理のナビゲーションエリアで、"Modbus TCP/IP"通信ドライバのツリーの[Modbus TCP/IP ユニット#1]チャンネルユニットを選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。
4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[Modbus TCP/IP プロパティ]ダイアログが開かれます。



5. [CPU タイプ]で、接続された Modicon コントローラを選択します。
以下の CPU から選択できます。
 - 984
CPU 984 用 CPU タイプ(CPU 984A、984B、984X を除く)を使用します。
 - Modicon Compact、Modicon Quantum、Modicon Momentum
 - Modicon Premium、Modicon Micro
6. [サーバー]フィールドにコントローラの IP アドレスを入力します。

6.3 Modbus TCP/IP

7. TCP/IP 接続に使用されるポートを、[ポート]フィールドに入力します。
Modbus TCP/IP 接続のデフォルトのポートは 502 です。
8. ブリッジを使用している場合は、[リモートスレーブアドレス]フィールドにリモートコントローラのアドレスを入力します。
ブリッジが使用されていない場合は、デフォルト値の 255 または 0 をアドレスとして入力する必要があります。

注記

WinCC ステーションは Modbus マスタとして動作するので、Modbus ネットワークのブリッジ経由で WinCC ステーションを統合させることはできません。

9. 16 ビットレジスタの順番を 32 ビットプロセス値にスワップするには、[ワードを 32 ビット値にスワップ]を選択します。
- 10.[OK]をクリックしてダイアログを閉じます。

6.3.3.3 タグの設定

タグの設定

はじめに

チャンネル"Modbus TCP/IP"を経由する WinCC とオートメーションシステム(AS)の間の接続用に、異なるデータタイプのタグを WinCC で作成できます。

以下のセクションでは、タグの設定方法について説明します。AS でのデータ範囲のアドレス指定と WinCC タグのデータタイプは違います。

Modbus TCP/IP プロトコルでのタグの更新

複数のタグが同時に、コントローラからの画面で取得される場合、Modbus TCP/IP チャンネルは、更新を最適化しようと試みます。ただし、これは以下の条件を満たす場合にのみ達成されます。

- タグが同じアドレス範囲にある。
- アドレス範囲内で、複数のタグができるだけ互いの近くにある。

これらの推奨事項を遵守しない場合は、多量のタグが付いている画面の更新で、顕著な相違が発生します。特定の環境下で、取得サイクルを維持できなくなる可能性があります。

タグ設定時に以下の規則を守ると、接続のパフォーマンスが最高になります。

- 最大 2,000 タグを同時に更新します。
- なるべく少ないスペース(最善は 1 つのアドレス範囲だけ)で、タグを組み合わせます。

コントローラのデータタイプとアドレス範囲

次の表は、タグおよび構造タグを設定するときに使用できるデータタイプおよびアドレス範囲を示します。

名称	CPU Premium/ Micro による領域	CPU 984、 Compact、 Quantum、 Momentum による 領域	データタイプ
コイル (ディスクリート出力)	%M ¹⁾	0x	ビット
ディスクリート入力	(%I) – Premium/ Micro では実現しま せん	1x	ビット
入力レジスタ	(%IW) – Premium/ Micro では実現しま せん	3x	Bit, +/- Int, Int
保持レジスタ (出力)	%MW	4x	Bit ²⁾ 、 +/- Int、Int、 +/- Double、 Double、 Float、ASCII
拡張メモリ ("Quantum/Momentum" CPU だけで利用 可能)	--	6x	Bit ²⁾ 、 +/- Int、Int、 +/- Double、 Double、 Float、ASCII

1) 外部コントローラのシステム特性により、アドレス領域の最後の x ビットにはアクセスできません。

2) 書き込みアクセスの場合の注意：

"4x"、6x"、"%MW"領域の“ビット”データタイプの場合、指定されたビットを変更した後で、ワード全体がコントローラに書き戻されます。ワード内で他のビットが変更されたかを判断する確認は行われません。結果として、コントローラには、指定されたワードへの読み取りアクセス権限しかありません。

6.3 Modbus TCP/IP

984、Compact、Quantum、Momentum シリーズのコントローラで使用される標準的なビットカウント方法(16 LSB - 1 MSB)は、"bit"データタイプの"タグ"エディタでのみこれらのCPU に使用されます。 ビット位置は以下のように割り当てられます。

	左バイト								右バイト							
タグによるカウント	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

WinCC の他の場所へビット番号を入力する場合、WinCC のビット割当(0 LSB - 15 MSB)が適用されます。

ビット位置のカウント方法	左バイト								右バイト							
WinCC での設定 :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

このビットカウント方法は、Modicon Premium コントローラおよび Modicon Micro コントローラにも適用されます。

"符号付き"用フォーマット

プレースホルダ"+/-"は、データタイプ"Signed Int"および"Signed Double"を意味します。

下記も参照

ビット単位でアクセスするタグの設定方法 (ページ 244)

ワード単位でアクセスするタグの設定方法 (ページ 246)

テキストタグの設定方法 (ページ 247)

ビット単位でアクセスするタグの設定方法


はじめに

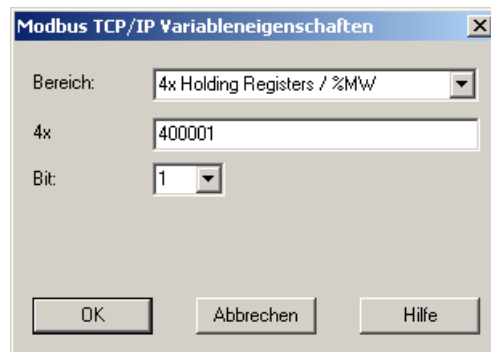
このセクションでは、オートメーションシステム(AS)のアドレスエリアにビット単位でアクセスするための、タグ設定方法を説明します。

必要条件

- チャンネル"Modbus TCP/IP"をプロジェクトに統合する必要があります。
- "Modbus TCP/IP Unit #1"チャンネルユニットに、接続を作成する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプとして[2 進タグ]を設定します。
5. [Modbus TCP/IP タグのプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. タグのアドレスを[4x]などの個々のアドレスフィールドに入力します。値はコントローラのコンフィグレーションに依存します。
7. 必要に応じて[ビット]フィールドのビットアドレスを入力します。エントリが可能か可能でないかは、[送信元エリア]フィールドの選択によります。
8. [値]フィールドに値[6x 拡張メモリ]を設定している場合、[ファイル]の値を選択します。
9. [OK]をクリックしてダイアログを閉じます。

注記

[Modbus TCP/IP タグプロパティ]ダイアログを閉じた後に、コントローラの内部タグアドレスが[タグプロパティ]ダイアログのフィールド[アドレス]に表示されます。このアドレスは AS データフォーマットに適合されていますので、入力済みアドレスと異なります。

下記も参照

テキストタグの設定方法 (ページ 247)

ワード単位でアクセスするタグの設定方法


はじめに

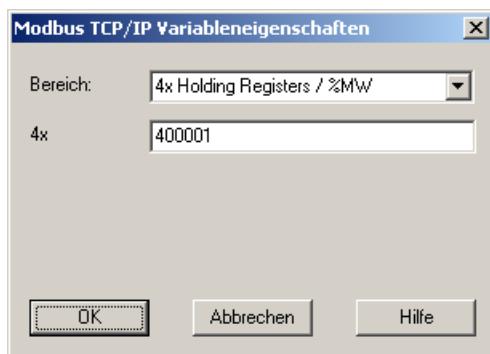
このセクションでは、オートメーションシステム(AS)のアドレスエリアへのワード単位のアクセス用タグの設定方法を説明します。

必要条件

- チャンネル"Modbus TCP/IP"をプロジェクトに統合する必要があります。
- "Modbus TCP/IP Unit #1"チャンネルユニットに、接続を作成する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプ[符号なし 16 ビット値]を定義します。
5. [Modbus TCP/IP タグのプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、ボタンをクリックしてください。



6. タグのアドレスを[4x]などの個々のアドレスフィールドに入力します。値はコントローラのコンフィグレーションに依存します。
7. [OK]ボタンをクリックして、両方のダイアログを閉じます。

注記

[Modbus TCP/IP タグプロパティ]ダイアログを閉じた後に、コントローラの内部タグアドレスが[タグプロパティ]ダイアログのフィールド[アドレス]に表示されます。このアドレスは AS データフォーマットに適合されていますので、入力済みアドレスと異なります。

テキストタグの設定方法


はじめに

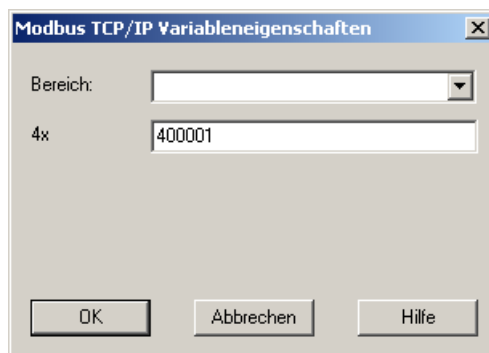
このセクションでは、オートメーションシステム(AS)のアドレスエリアへのワード単位のアクセス用タグの設定方法を説明します。

必要条件

- チャンネル"Modbus TCP/IP"をプロジェクトに統合する必要があります。
- "Modbus TCP/IP Unit #1"チャンネルユニットに、接続を作成する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプとして[テキストタグ 8 ビット文字セット]を設定します。
5. [Modbus TCP/IP タグのプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. アドレスフィールド[4x]にタグのアドレスを入力します。値はコントローラのコンフィグレーションに依存します。
7. [OK]ボタンをクリックして、両方のダイアログを閉じます。

6.4 OPC チャンネル

6.4.1 WinCC OPC チャンネル

はじめに

WinCC は、OPC サーバーとしても OPC クライアントとしても使用できます。OPC チャンネルは、WinCC の OPC クライアントアプリケーションです。

OPC 通信ドライバは、OPC DA クライアント、OPC XML クライアントおよび OPC UA クライアントとして使用できます。OPC UA クライアントのマニュアルは「OPC UA チャンネル」にあります。

以下の OPC コンポーネントが自動的にインストールされます。

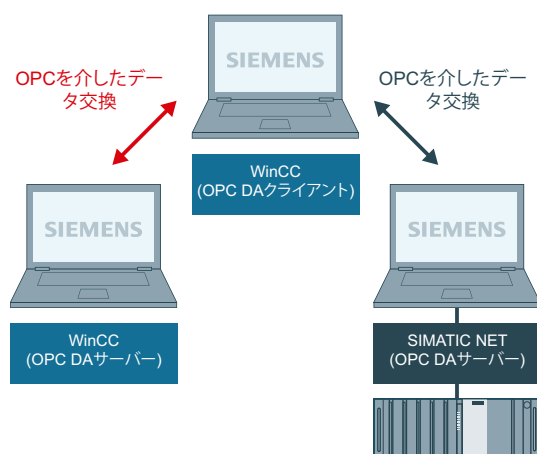
- OPC 通信ドライバ
- OPC 項目マネージャ

可能なアプリケーション

OPC DA クライアントとしての WinCC

WinCC を OPC DA クライアントとして使用する場合、WinCC プロジェクトに OPC チャンネルを追加する必要があります。WinCC OPC DA クライアントの WinCC プロジェクトで、データ交換用接続が作成されます。この接続を使用して、OPC DA サーバーの WinCC タグへのアクセスを処理します。

工程を簡単にするために、OPC 項目マネージャを使用します。WinCC OPC DA クライアントは複数の OPC DA サーバーにアクセスできます。この操作をするには、OPC サーバーごとに接続を 1 つ作成する必要があります。このように、WinCC OPC DA クライアントを、中央操作およびモニタステーションとして使用できます。



注記

WinCC OPC チャンネルは、"OPC_STATUS_RUNNING"ステータスの OPC サーバーへの接続だけを確立します。

注記

[OPC]チャンネル

接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

6.4.2 OPC 項目マネージャ

はじめに

接続および WinCC タグは、OPC サーバーのタグへのアクセスを有効にするために、WinCC OPC クライアントの WinCC プロジェクトで設定されます。OPC 項目マネージャは、このプロセスを簡素化します。OPC 項目マネージャは、WinCC で自動的にインストールされます。

注記

[OPC]チャンネル

接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

必要条件

OPC 項目マネージャを使用して設定するには、以下の必要条件に対応している必要があります。

- OPC サーバーは、OPC DA サーバーまたは OPC XML サーバーです。
- OPC サーバー上でタグが既に設定されています。
- WinCC を OPC サーバーとして使用する場合、WinCC OPC サーバーの WinCC プロジェクトが有効になっている必要があります。有効でない場合は、OPC 項目マネージャは WinCC OPC サーバーにアクセスできません。
- IP アドレスまたは HTTP を使用して、OPC サーバーのコンピュータにアクセスする必要があります。
- OPC サーバーがブラウザ機能をサポートしている必要があります。サポートしていない場合は、OPC サーバーのタグを手動で設定する必要があります。

注記

OPC 項目マネージャが開いているときに WinCC エクスプローラの言語を変更する場合は、[サーバーの参照]ボタンをクリックしたときにタグが表示されません。言語を変更する前に、OPC 項目マネージャを終了します。

OPC 項目マネージャのタスク

OPC 項目マネージャでは、以下のタスクが想定されています。

- OPC サーバーの選択
- 接続の作成
- タグの選択
- タグの追加

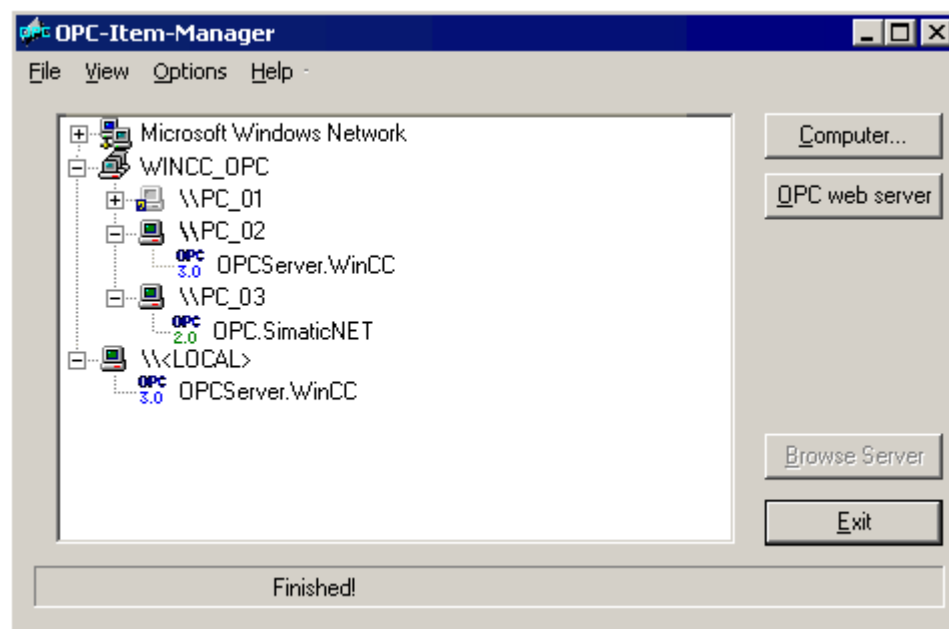
OPC サーバーの選択






OPC DA サーバー

OPC 項目マネージャを使用すれば、ネットワークで OPC DA サーバーの名前を決定できます。これらの OPC DA サーバーは、同じコンピュータ上、または接続したネットワーク環境内の異なるコンピュータ上で実行できます。詳細については、「WinCC OPC DA クライアント」を参照してください。

OPC XML サーバー

OPC XML サーバーの名前をクエリするには OPC 項目マネージャを使用できません。[OPC Web サーバー]ボタンを使用すれば、OPC XML サーバーを追加できます。詳細については、「WinCC OPC XML クライアント」を参照してください。



OPC 項目マネージャのアイコン	説明
	インストールされた OPC DA サーバーでは、ネットワーク上のコンピュータはまだ検索されていません。
	ネットワークでコンピュータが見つかりません。または、コンピュータにアクセスできません。
	インストールされた OPC DA サーバーでは、ネットワーク上のコンピュータが検索されています。
	ネットワーク上のコンピュータには、OPC シンボルが指定された OPC DA サーバーが格納されています。番号は、使用する WinCC OPC DA クライアントの OPC DA 仕様を示しています。
\\<LOCAL>	OPC 項目マネージャを実行しているコンピュータを参照します。
 http://	OPC XML サーバーの名前です。 [OPC Web サーバー]ボタンを使用すれば、OPC XML サーバーを追加できます。

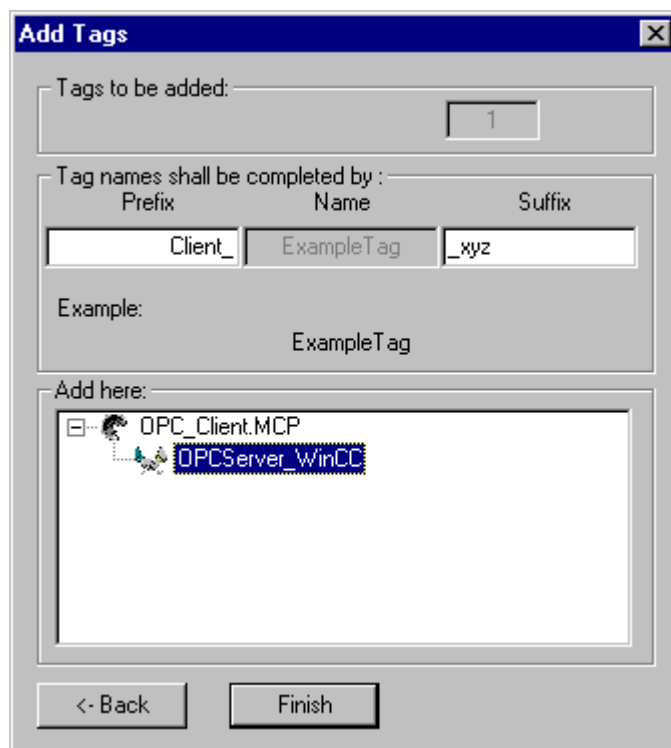
接続の作成

接続の作成時、OPC 項目マネージャによって、必要な設定がすべて設定されます。OPC サーバーとの接続がすでに作成されている場合、この機能は使用できません。

タグの選択

[タグの選択]ダイアログを使用すれば、WinCC OPC クライアントがアクセスする OPC サーバー上でタグを 1 つ以上選択できます。フィルタ規準を使用すれば、[タグの選択]ダイアログ内の選択を制限できます。

タグの追加



[タグの追加]ダイアログで、OPC サーバー上でタグにアクセスする WinCC タグ名を設定できます。

WinCC タグ名は、"接頭語"、"名前"、および"接尾語"で構成されます。"名前"フィールドは、"ExampleTag"テキストで事前設定されています。"ExampleTag"は、WinCC OPC サーバー上の WinCC タグ名を表します。

WinCC OPC クライアント上の WinCC タグ名を、WinCC OPC サーバー上の WinCC タグ名と区別するために、接頭辞または接尾辞を割り付けることができます。プロジェクトモニタを設定する場合、接頭語または接尾語を割り付ける必要があります。

所定の WinCC プロジェクトごとに、タグ名は一度のみ割り付けできます。

例

WinCC OPC DA サーバー上の WinCC タグ名は、"OPC_Server_Tag"と呼ばれます。[接頭語]フィールドに"Client_"と入力し、[接尾語]フィールドに"_xyz"と入力します。WinCC OPC DA クライアントの WinCC プロジェクトで、WinCC タグ"Client_OPC_Server_Tag_xyz"が作成されます。

OPC サーバー上のタグ名に特殊文字が指定されている場合、これらの特殊文字はアンダースコア"_"に置換されます。これは、OPC 項目マネージャでは、タグ名に指定される特殊文字の一部がサポートされていないためです。

6.4 OPC チャンネル

[終了]ボタンをクリックすると、WinCC OPC DA クライアントの WinCC プロジェクトに WinCC タグが追加されます。OPC 項目マネージャは、WinCC タグのデータタイプ、名前、およびアドレスパラメータを自動的に設定します。

下記も参照

OPC 項目マネージャによる WinCC タグへのアクセス方法 (ページ 276)

OPC 項目マネージャによる WinCC タグへのアクセス方法 (ページ 257)

6.4.3 サポートされている WinCC データタイプの概要

次のリストでは、WinCC OPC DA クライアントおよび WinCC OPC DA サーバーがサポートするデータタイプについてまとめています。

- 2 進タグ
- 符号付き 8 ビット値
- 符号なし 8 ビット値
- 符号付き 16 ビット値
- 符号なし 16 ビット値
- 符号付き 32 ビット値
- 符号なし 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- 浮動小数点数 64 ビット IEEE 754
- テキストタグ、8 ビット文字セット
- テキストタグ、16 ビット文字セット
- 未処理データタイプ
- 構造タイプ
- テキスト参照
- 日付/時刻

注記**構造タイプ**

構造体タイプでは、構造体エレメントのみがサポートされ、構造体自体はサポートされません。ただし、後ほど構造体を設定できます。詳細については、「WinCC OPC DA クライアント上での構造体の使用」トピックを参照してください。

テキスト参照

OPC 項目マネージャを使用してテキストタグを作成するとこのテキストタグには、160 文字長が割り付けられます。この長さは、どの長さにも変更できます。

下記も参照

WinCC OPC DA クライアント上での構造体の使用方法 (ページ 265)

6.4.4 WinCC OPC DA クライアント**6.4.4.1 WinCC OPC DA クライアントの機能****はじめに**

OPC チャンネルには、別個の通信モジュールは必要ありません。OPC チャンネルは、OPC ソフトウェアインターフェースを利用して OPC DA サーバーを使用してプロセスデータにアクセスするアプリケーションです。

WinCC を OPC DA クライアントとして使用する場合、WinCC プロジェクトに OPC チャンネルを追加する必要があります。

WinCC OPC DA サーバーとの通信が確立すると、WinCC タグの値が交換されます。この操作を実行するため、WinCC OPC DA クライアントの WinCC プロジェクトで接続が設定されます。この接続を使用して WinCC OPC DA サーバーへのアクセスを処理します。

WinCC OPC DA クライアントが複数の OPC DA サーバーにアクセスする場合には、WinCC プロジェクトで、各 OPC DA サーバーの接続を設定する必要があります。チャンネルとタ

6.4 OPC チャンネル

グのトラブルシューティングの詳細については、「トラブルシューティング」を参照してください。

注記

WinCC OPC チャンネルは、"OPC_STATUS_RUNNING"ステータスの OPC サーバーへの接続だけを確立します。

注記**[OPC]チャンネル**

接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

接続のモニタ

WinCC OPC-DA クライアントでは、接続のモニタのために 3 種類のメカニズムが統合されます。この結果、OPC DA サーバーでネットワークエラーや誤動作が発生した場合、可能な限り適切な対策を実行できます。

1. DCOM 有効化の処理期間が、警告値の 5 秒を超えると、タグに値"アドレス指定エラー"が割り付けられます。この処理期間が、キャンセル値の 10 秒を超えると、OPC DA サーバーとの接続が割り込まれます。WinCC エクスプローラの[接続ステータス]ダイアログに、この旨が表示されます。

OPC DA 仕様 3.00 では、"キープアライブ"機能が実現されています。OPC DA サーバーが OPC DA 仕様 3.00 をサポートしている場合、この機能が使用されます。この機能を使用すれば、OPC DA サーバーは、タグ値を変更しなくても周期的更新(OnDataChange の呼び出し)を自動的にトリガできます。この定期更新を無効にすると、WinCC OPC DA クライアントは接続を終了します。

OPC DA サーバーが OPC DA 仕様 2.05a をサポートしている場合、同じ動作が適用されます。OPC DA サーバーとの接続をチェックするには、10 秒ごとに周期的に WinCC OPC DA クライアントはステータスを要求します。この定期更新を無効にすると、WinCC OPC DA クライアントは接続を終了します。

接続が機能できない場合、通常、WinCC OPC DA クライアントは、OPC DA サーバーとの接続を終了します。WinCC OPC DA クライアントは、10 秒ごとに自動的に繰り返し接続を再確立しようとします。

下記も参照

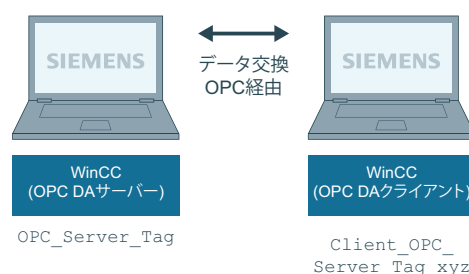
WinCC OPC DA クライアント上での構造体の使用方法 (ページ 265)
OPC 項目マネージャによらない WinCC タグへのアクセス (ページ 263)
OPC 項目マネージャによるアクセスの設定 (ページ 259)
サポートされている WinCC データタイプの概要 (ページ 254)
OPC 項目マネージャ (ページ 250)
チャンネルおよびタグの診断 (ページ 615)
OPC 仕様と互換性 (ページ 699)
WinCC OPC DA サーバーの機能 (ページ 709)

6.4.4.2 OPC 項目マネージャによる WinCC タグへのアクセス方法

OPC 項目マネージャによる WinCC タグへのアクセス方法

はじめに

WinCC 間に OPC 接続を確立する場合、WinCC タグを使用してデータ交換を実行します。WinCC OPC DA クライアントは、OPC 接続を使用して、WinCC OPC DA サーバー上で WinCC タグ"OPC_Server_Tag"を読み取ります。プロセスを簡単にするために、OPC 項目マネージャを使用します。



必要条件

- WinCC プロジェクトをインストールしたコンピュータが 2 台あること。
- IP アドレスを使用して、どちらのコンピュータにもアクセスできる必要があります。

コンフィグレーションの手順

WinCC OPC DA クライアントの WinCC プロジェクトでは、以下のコンフィグレーションが必要です。

- 接続の作成
- WinCC OPCXML クライアント上での、WinCC OPC DA サーバーの WinCC タグにアクセスする WinCC タグ"XMLClient_OPC_Var1_xyz"のコンフィグレーション。

下記も参照

OPC 項目マネージャによるアクセスの設定 (ページ 259)

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

OPC 項目マネージャによるアクセスの設定 (ページ 277)

WinCC 間の接続例 (ページ 713)

WinCC OPC DA クライアント上での OPC チャンネルの設定

はじめに

OPC を使用してデータ交換を実行するには、WinCC プロジェクトで OPC チャンネルを設定する必要があります。

手順

1. WinCC OPC DA クライアント上の WinCC Explorer のナビゲーションウィンドウで[タグ管理]アイコンをクリックします。
2. [タグ管理]のショートカットメニューで[新規ドライバの追加]を選択します。[新規ドライバの追加]ダイアログを開きます。
3. [OPC チャンネル]ドライバを選択した後、[開く]ボタンをクリックします。チャンネルが作成され、[タグ管理]内に通信ドライバが表示されます。

下記も参照

OPC 項目マネージャによるアクセスの設定 (ページ 259)

OPC 項目マネージャによるアクセスの設定

はじめに

このセクションでは、OPC 項目マネージャを使用して WinCC OPC DA サーバーの WinCC タグ"OPC_Server_Tag"へのアクセスを設定する方法を説明します。

必要条件

- WinCC OPC DA サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"OPC_Server_Tag"内部タグが設定されていること。
- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっていること。
- WinCC OPC DA クライアントの WinCC プロジェクトに"OPC"チャンネルが追加されていること。

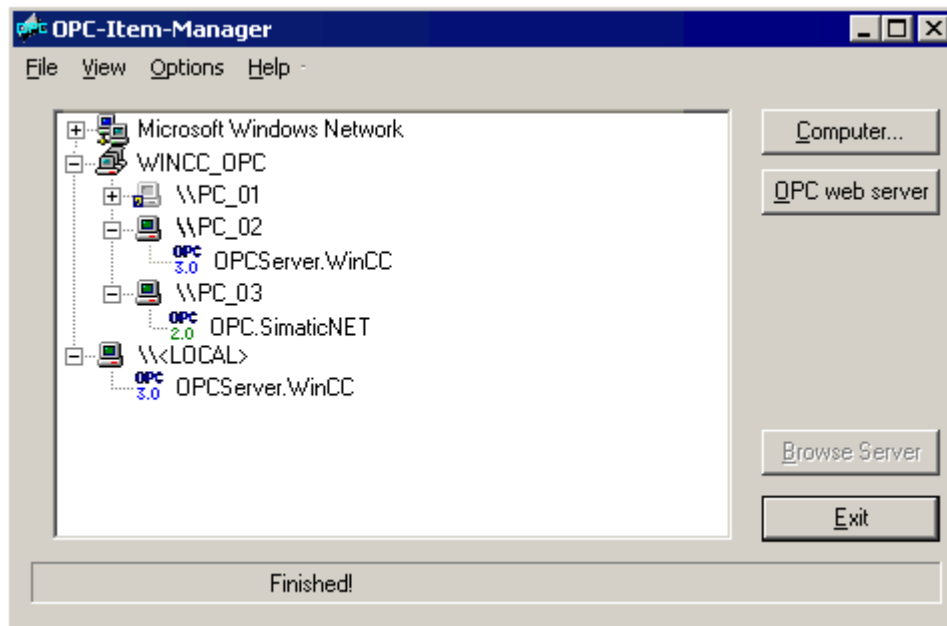
注記

[OPC]チャンネル

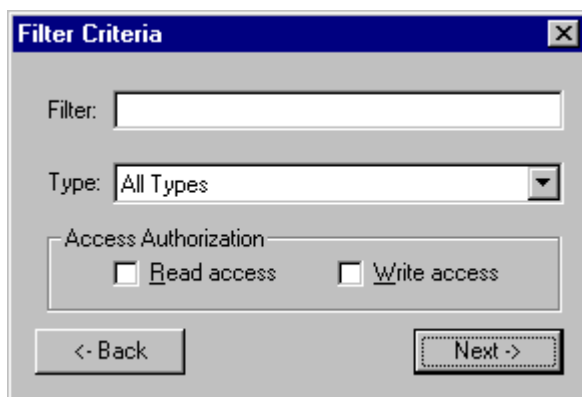
接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

手順

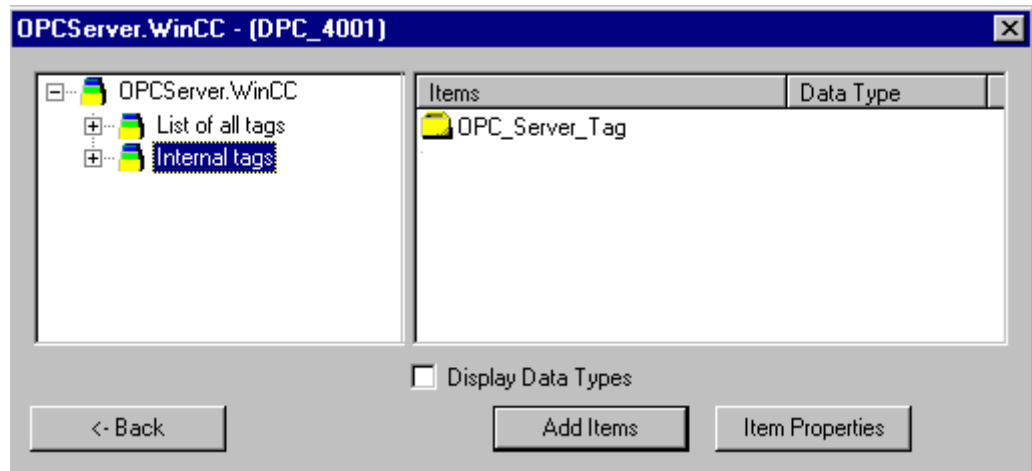
1. WinCC OPC DA クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[システムパラメータ]を選択します。[OPC 項目マネージャ]が開きます。



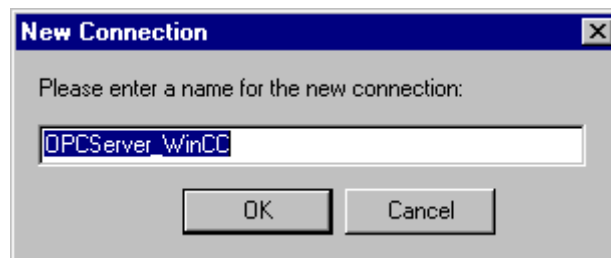
2. 選択ダイアログで、WinCC OPC DA サーバーとして使用するコンピュータの名前を選択します。
表示されたリストで、[OPCServer.WinCC]を選択します。
3. [サーバーの参照]ボタンをクリックします。
[フィルタ規準]ダイアログが開きます。



4. [次へ]をクリックします。
[OPCServer.WinCC ...]ダイアログが開きます。

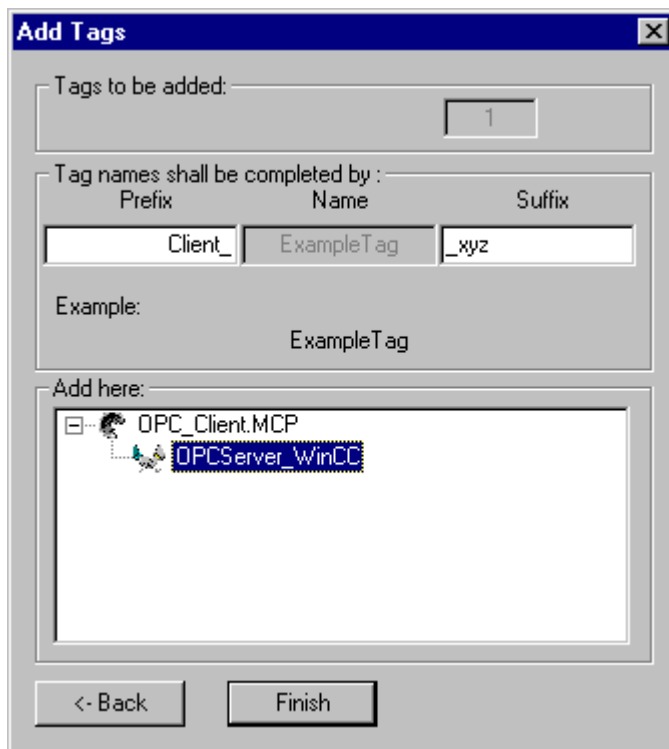


5. WinCC タグの[OPC_Server_Tag]を選択します。
[項目の追加]ボタンをクリックします。
6. WinCC OPC DA サーバーとの接続がすでに存在している場合には、引き続き手順 6 に進みます。
接続が作成されていない場合は、メッセージが表示されます。
[はい]ボタンをクリックします。 [新規接続]ダイアログが開きます。



6.4 OPC チャンネル

7. この接続名として"OPCServer_WinCC"を入力します。[OK]をクリックします。
[タグの追加]ダイアログが開きます。



8. [接頭語]フィールドに"Client_"を入力し、[接尾語]フィールドには"_xyz"を入力します。
9. 接続"OPCServer_WinCC"を選択します。
[終了]をクリックします。
10. [OPCServer.WinCC ...]ダイアログで[戻る]ボタンをクリックします。
[終了]をクリックして OPC 項目マネージャを閉じます。

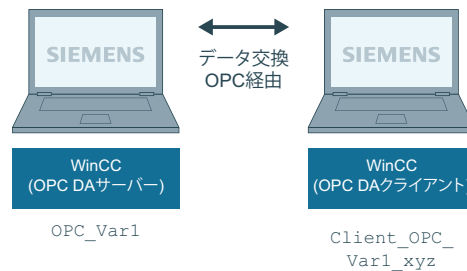
下記も参照

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

6.4.4.3 OPC 項目マネージャによらない WinCC タグへのアクセス

はじめに

ブラウザ機能をサポートしていない OPC サーバーでは、アクセスを手動で設定する必要があります。WinCC 間の OPC 接続例を使用して、WinCC OPC DA クライアント上の WinCC タグのコンフィグレーションを示します。



注記

OPC 項目マネージャを使用せずに WinCC タグにアクセスするには、項目 ID を手動で設定する必要があります。WinCC タグをアドレス指定する場合、シンボルコンピュータ名(サーバー接頭語)も指定できます。項目 ID では、以下の構文が使用されます。サーバー接頭語::WinCC tag。ローカル WinCC プロジェクトの WinCC タグをアドレス指定する場合には、サーバー接頭語を省略します。

WinCC OPC DA クライアントの WinCC プロジェクトでは、以下のコンフィグレーションが必要です。

1. アクセスする WinCC タグ"OPC_Var1"の選択
2. 接続の作成
3. WinCC OPC DA サーバーの WinCC タグにアクセスする WinCC タグ"Client_OPC_Var1_xyz"のコンフィグレーション

必要条件

- WinCC プロジェクトをインストールしたコンピュータが 2 台あること。
- IP アドレスを使用して、どちらのコンピュータにもアクセスできる必要があります。
- WinCC OPC DA サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"OPC_Var1"内部タグが設定されていること。

6.4 OPC チャンネル

- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっていること。
- WinCC OPC DA クライアントの WinCC プロジェクトに OPC チャンネルが追加されていること。

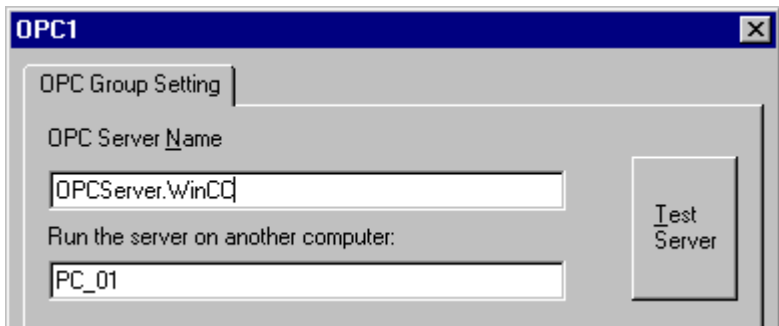
注記

[OPC]チャンネル

接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

手順

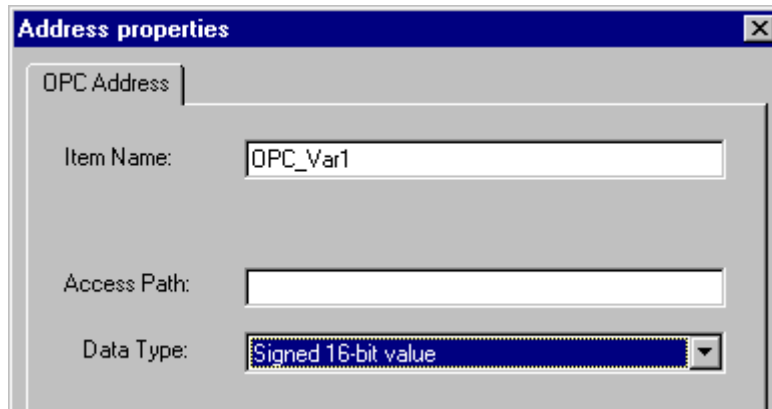
1. WinCC OPC DA クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[新規接続]を選択します。[接続プロパティ]ダイアログが開きます。対応するフィールドにこの接続の名前を入力します。
2. [プロパティ]ボタンをクリックします。タイトルにその接続名が表示されたダイアログが表示されます。



WinCC V6x に接続する場合は、[OPC サーバー名]フィールドに"OPCServer.Wi nCC"エントリを入力する必要があります。

3. [このコンピュータ上でサーバーを起動する]フィールドに、OPC DA サーバーとして使用するコンピュータの名前を入力します。[テストサーバー]ボタンをクリックして、WinCC OPC DA サーバーとの接続を確認します。
4. 接続のショートカットメニューで[タグの新規作成]を選択します。[タグのプロパティ]ダイアログが開きます。
5. [タグ]フィールドに名前"Client_OPC_Var1_xyz"を入力します。データタイプを[符号付き 16 ビット]に設定します。

6. [タグのプロパティ]ダイアログで[選択]ボタンをクリックします。[アドレスのプロパティ]ダイアログが開きます。



[項目名]フィールドに WinCC OPC DA サーバーの WinCC タグの名前を入力します。[アクセスパス]フィールド内のエントリはそのままにしておきます。データタイプを[符号付き 16 ビット]に設定します。

7. [OK]をクリックして、開いているダイアログをすべて閉じます。

6.4.4.4 WinCC OPC DA クライアント上での構造体の使用

WinCC OPC DA クライアント上での構造体の使用方法

はじめに

構造体を使用して、タグとタグタイプを編成して 1 つの論理ユニットを形成します。この結果、単一の論理名を使用してタグとタグタイプを参照できます。

構造体は、OPC DA 仕様でサポートされていません。この結果、OPC 項目マネージャを使用して構造体を設定することはできません。設定できるのは、構造体内の個々のタグのみです。それにもかかわらず、WinCC OPC DA クライアント上で構造体を使用する場合には、サーバータグの関連項目名を使用できるように、WinCC OPC DA クライアントの WinCC プロジェクトで引き続きデータ構造体を設定する必要があります。

必要条件

- WinCC プロジェクトを持つコンピュータが 2 台あること。
- IP アドレスを使用して、どちらのコンピュータにもアクセスできる必要があります。

設定の手順

WinCC OPC DA クライアント上で構造体を使用するには、以下の設定の手順が必要です。

- WinCC OPC DA サーバー上での構造体および構造体タグの設定
- WinCC プロジェクトにおける WinCC OPC DA クライアント上での構造体の使用

下記も参照

WinCC OPC DA クライアント上での構造体の設定方法 (ページ 268)

WinCC OPC DA サーバー上での構造体および構造体タグの設定 (ページ 266)

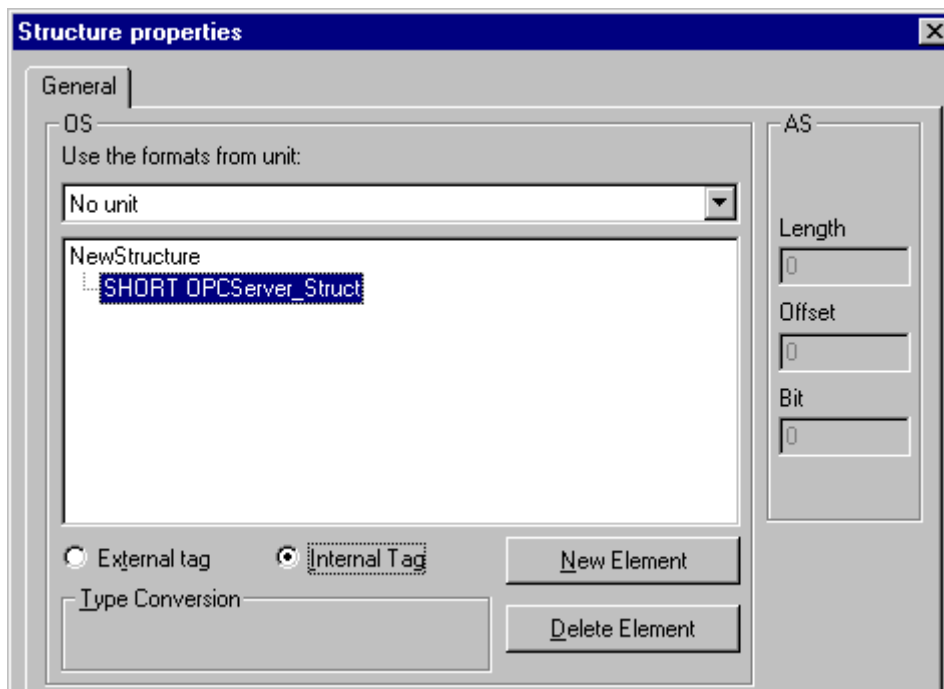
WinCC OPC DA サーバー上での構造体および構造体タグの設定

はじめに

このセクションでは、OPC DA サーバーの WinCC プロジェクトで、構造体および構造体タグを作成します。OPC DA クライアントが構造体タグにアクセスするには、この設定が必要です。

手順

1. WinCC OPC DA サーバーで、構造体タイプのショートカットメニューの[構造体タイプの新規作成]を選択します。[構造体プロパティ]ダイアログが表示されます。
2. [エレメントの新規作成]をクリックして、SHORT データタイプの内部タグ "OPCServer_Struct"を作成します。



[OK]をクリックしてダイアログを閉じます。

3. ナビゲーションウィンドウで、タグ管理用アイコンの前に表示されているプラス記号をクリックします。内部タグのショートカットメニューで[タグの新規作成]を選択します。この構造体タイプを指定して"Var"WinCC タグを作成します。
4. WinCC Explorer のデータフレームに、単一タグ"Var"と構造体タグ"Var.OPCServer_Struct"が表示されます。
5. WinCC プロジェクトを有効にすること。

下記も参照

WinCC OPC DA クライアント上での構造体の設定方法 (ページ 268)

WinCC OPC DA クライアント上での構造体の設定方法

はじめに

構造体は、OPC DA 仕様でサポートされていません。この結果、OPC 項目マネージャを使用して構造体を設定することはできません。このセクションでは、WinCC OPC DA サーバーの WinCC プロジェクトにすでに存在している構造体を、WinCC OPC DA クライアントの WinCC プロジェクトに対応して設定します。WinCC OPC DA クライアント上で、WinCC OPC DA サーバー上の既存の構造体タグにアクセスする WinCC タグを設定します。

必要条件

- WinCC OPC DA サーバーの WinCC プロジェクトで、"Var.OPCServer_Struct"構造体および構造体タグが作成されていること。
- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっていること。
- WinCC OPC DA クライアントの WinCC プロジェクトに OPC チャンネルが追加されていること。

注記

[OPC]チャンネル

接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

手順

1. WinCC OPC DA クライアントで、構造体タイプのショートカットメニューの[構造体タイプの新規作成]を選択します。[構造体プロパティ]ダイアログが表示されます。
2. [エレメントの新規作成]ボタンをクリックして、外部タグを設定します。OPC-DA サーバーの WinCC プロジェクトで付けたのとまったく同じようにエレメントに名前を付けます。[OK]をクリックして、[構造体プロパティ]ダイアログを閉じます。
3. OPC DA サーバーとの接続がすでに存在している場合には、引き続き手順 6 に進みます。接続が作成されていない場合、チャンネルユニット[OPC]のショートカットメニューで[新規接続]を選択します。[接続プロパティ]ダイアログが開きます。対応するフィールドにこの接続の名前を入力します。
4. [プロパティ]ボタンをクリックします。タイトルにその接続名が表示されたダイアログが表示されます。WinCC V6x に接続する場合は、[OPC サーバー名]フィールドに "OPCServer.WinCC" エントリを入力する必要があります。

5. [このコンピュータ上でサーバーを起動する]フィールドに、WinCC OPC DA サーバーとして使用するコンピュータの名前を入力します。[テストサーバー]ボタンをクリックして、WinCC OPC DA サーバーとの接続を確認します。[OK]をクリックしてダイアログを閉じます。
6. 接続のショートカットメニューで[タグの新規作成]を選択します。[タグのプロパティ]ダイアログが開きます。データタイプとして、新たに作成した構造体タイプを選択します。
7. [タグのプロパティ]ダイアログで[選択]ボタンをクリックします。[アドレスのプロパティ]ダイアログが開きます。
[項目名]フィールドに、WinCC OPC DA サーバーの構造タグの名前「Var.OPCServer_Struct」を入力します。[アクセスパス]フィールド内のエントリはそのままにしておきます。
8. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

WinCC OPC DA サーバー上での構造体および構造体タグの設定 (ページ 266)

6.4.4.5 OPC DA 通信に障害が発生した場合のエラー処理

OPC DA 通信に障害が発生した場合のエラー処理

はじめに

通信テストの手順は、WinCC の使用方法に関係ありません。

OPC DA サーバーとして使用される WinCC

WinCC OPC DA クライアント上でチャンネル診断機能を使用して、OPC DA サーバーとの接続を確立できるかどうかを決定します。チャンネルの問題分析の詳細については、「トラブルシューティング」を参照してください。

OPC DA クライアントとして使用される WinCC

WinCC OPC DA クライアント上でチャンネル診断機能を使用して、OPC DA サーバーとの接続を確立できるかどうかを決定します。チャンネルの問題分析の詳細については、「トラブルシューティング」を参照してください。

6.4 OPC チャンネル

下記も参照

WinCC は OPC DA クライアントとして使用され、接続が確立されませんでした。(ページ 274)

WinCC は OPC DA クライアントとして使用され、接続が確立されました。(ページ 273)

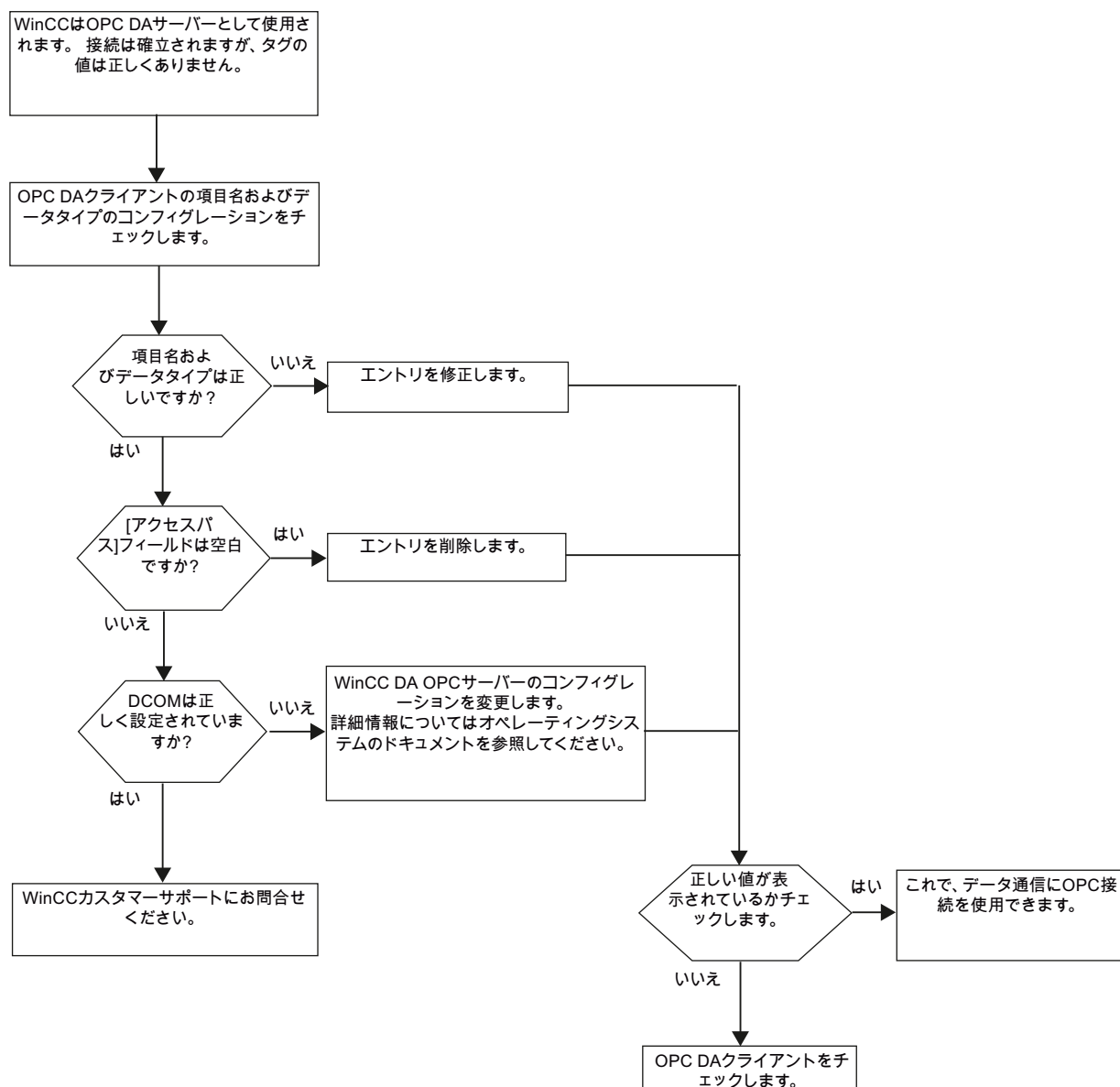
WinCC は OPC DA サーバーとして使用され、接続が確立されませんでした。(ページ 272)

WinCC は OPC DA サーバーとして使用され、接続が問題なく確立されました。(ページ 271)

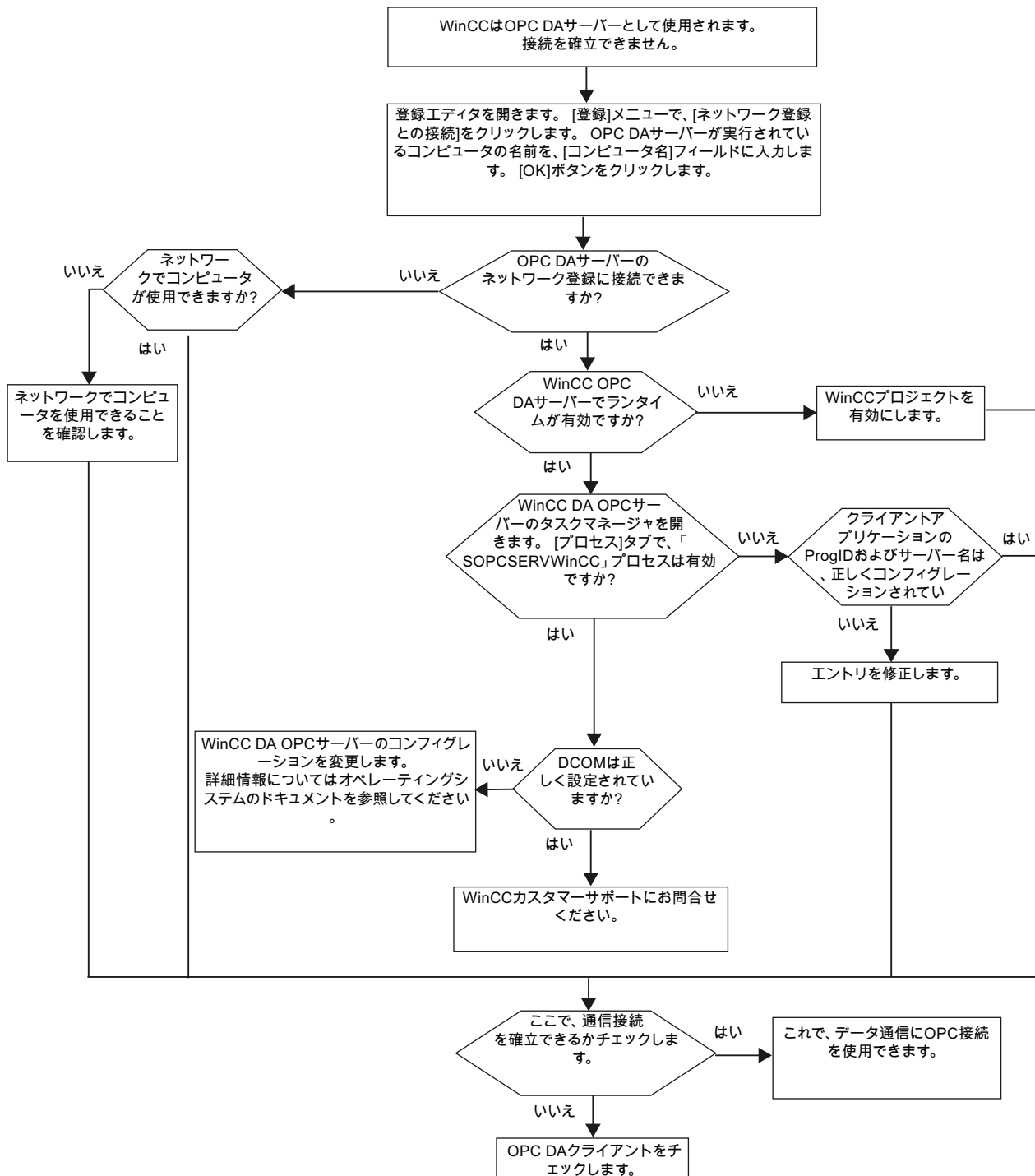
"OPC"チャンネルの診断のための機能 (ページ 669)

OPC DA サーバーとしての WinCC

WinCC は OPC DA サーバーとして使用され、接続が問題なく確立されました。

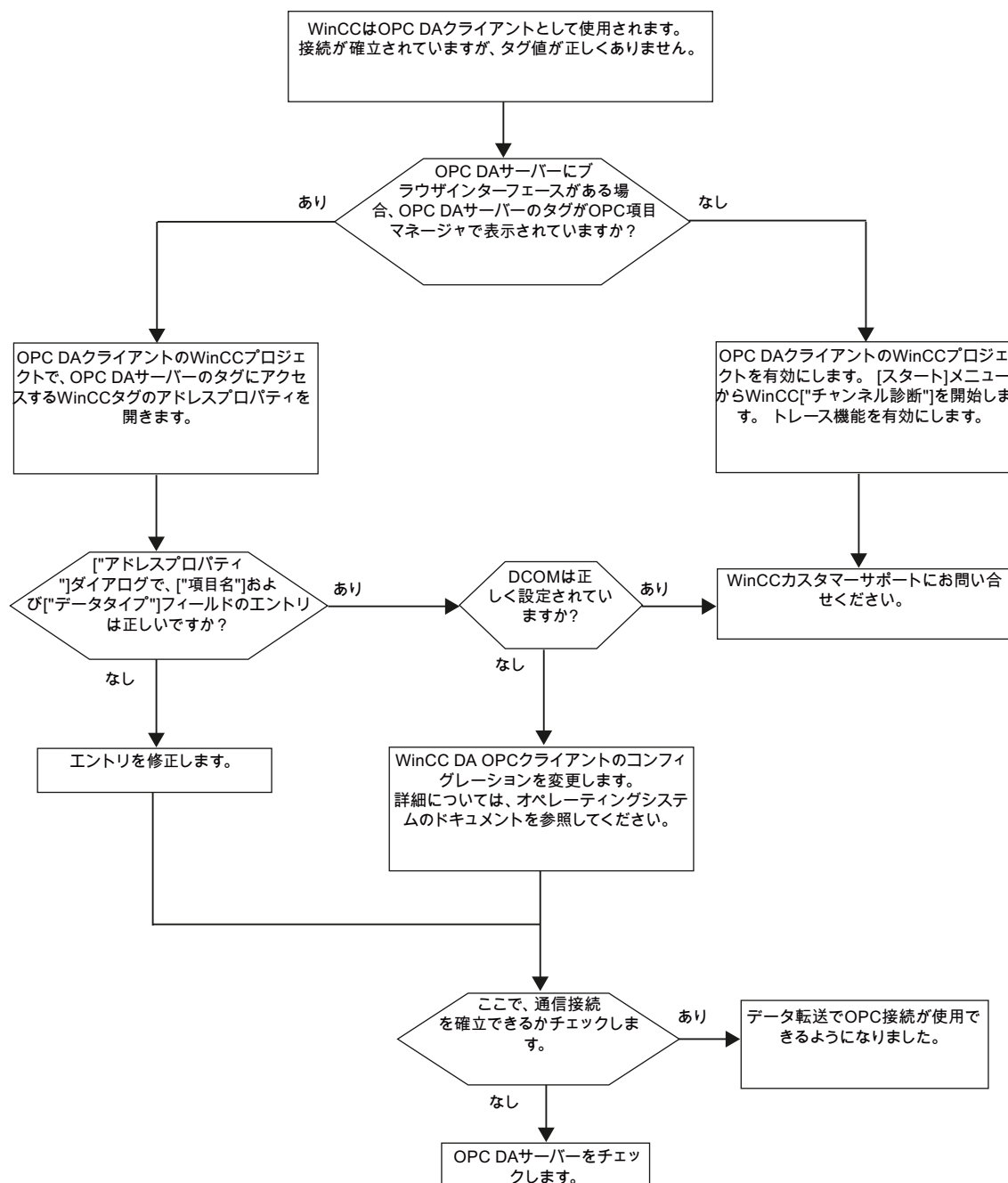


WinCC は OPC DA サーバーとして使用され、接続が確立されませんでした。

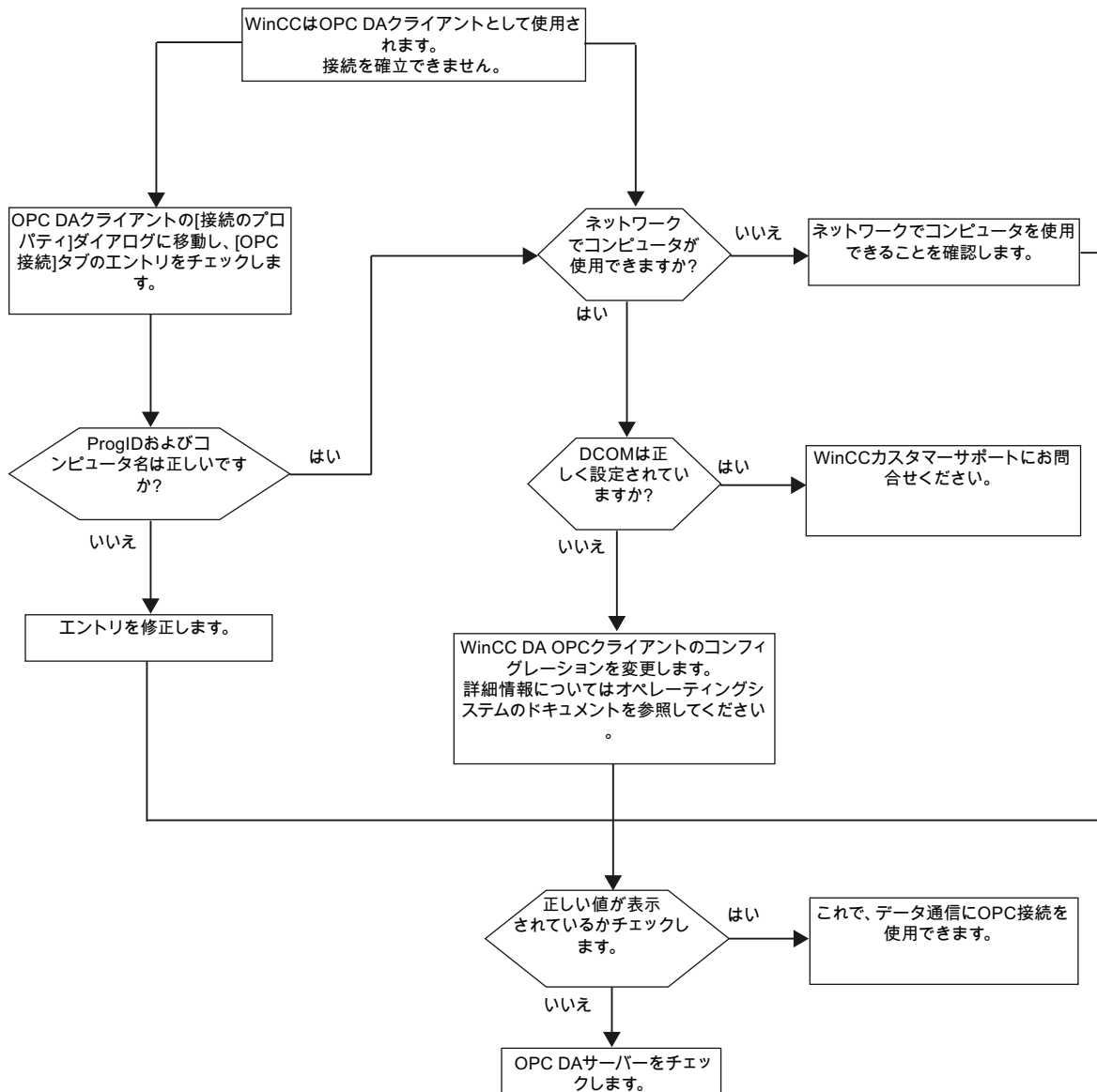


OPC DA クライアントとしての WinCC

WinCC は OPC DA クライアントとして使用され、接続が確立されました。



WinCCはOPC DAクライアントとして使用され、接続が確立されませんでした。



6.4.5 WinCC OPC XML クライアント

6.4.5.1 WinCC OPC XML クライアントの機能

はじめに

OPC チャンネルには、別個の通信モジュールは必要ありません。OPC 通信ドライバは、OPC XML クライアントとしてインプリメントできます。

WinCC を WinCC OPC XML クライアントとして使用するには、WinCC プロジェクトに OPC チャンネルを追加する必要があります。

WinCC OPC XML クライアントは、OPC XML サーバーに Web ページ形式で OPC プロセスデータを提供します。HTTP を使用すれば、インターネット/イントラネットに Web ページにアクセスできます。WinCC OPC XML クライアントがデータを要求すると、Web サーバーにより Web サービスが自動的に開始されます。

WinCC OPC XML クライアントが複数の OPC XML サーバーにアクセスできるには、WinCC プロジェクトで、各 OPC XML サーバーと接続する必要があります。

WinCC OPC XML サーバーとの通信が確立すると、WinCC タグの値が交換されます。WinCC OPC XML サーバーにアクセスする WinCC OPC XML クライアントの WinCC プロジェクトで接続が設定されます。WinCC OPC XML クライアントの場合、接続モニタは実行されません。

通知

複数のネットワークアダプタおよび有効な TCP/IP を使った操作

複数のネットワークアダプタおよび有効な TCP/IP プロトコルを使用して作業する場合、「WinCC リリースノート/操作上の注記/ネットワークテクノロジーおよび UPS」および「コンフィグレーション/分散システム/ランタイムでのシステム動作/複数のネットワークアダプタを備えたサーバーを使用した通信の特殊機能」セクションの情報を遵守してください。

注記

WinCC OPC チャンネルは、"OPC_STATUS_RUNNING"ステータスの OPC サーバーへの接続だけを確立します。

注記**[OPC]チャンネル**

接続名で **Unicode** はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、**Unicode** がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

下記も参照

サポートされている WinCC データタイプの概要 (ページ 254)

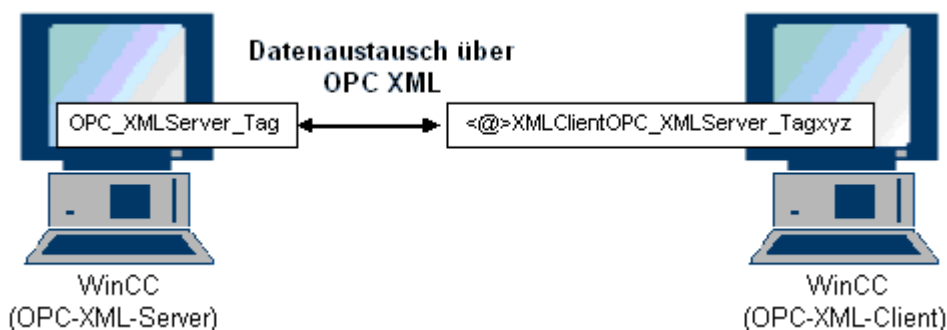
OPC 項目マネージャ (ページ 250)

OPC 仕様と互換性 (ページ 699)

WinCC OPC XML DA サーバーの機能 (ページ 703)

6.4.5.2 OPC 項目マネージャによる WinCC タグへのアクセス方法**OPC 項目マネージャによる WinCC タグへのアクセス方法****はじめに**

WinCC 間に OPC 接続を確立する場合、WinCC タグを使用してデータが交換されます。WinCC OPC DA クライアントは、OPC 接続を使用して、WinCC OPC XML サーバー上で WinCC タグ"OPC_Server_Tag"を読み取ります。プロセスを簡単にするために、OPC 項目マネージャを使用します。



必要条件

- WinCC プロジェクトをインストールしたコンピュータが 2 台あること。
- HTTP を使用して両方のコンピュータにアクセスできること。

設定の手順

WinCC OPC XML クライアントの WinCC プロジェクトでは、以下の設定が必要です。

下記も参照

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

OPC 項目マネージャによるアクセスの設定 (ページ 277)

OPC 項目マネージャによるアクセスの設定

はじめに

このセクションでは、OPC 項目マネージャを使用して WinCC OPC XML サーバーの WinCC タグ"OPC_XMLServer_Tag"へのアクセスを設定する方法について説明します。

必要条件

- WinCC OPC XML サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"OPC_XMLServer_Tag"内部タグが設定されていること。
- WinCC OPC XML サーバーの WinCC プロジェクトが有効になっていること。
- WinCC OPC XML クライアントの WinCC プロジェクトに OPC チャンネルを追加する必要がある。

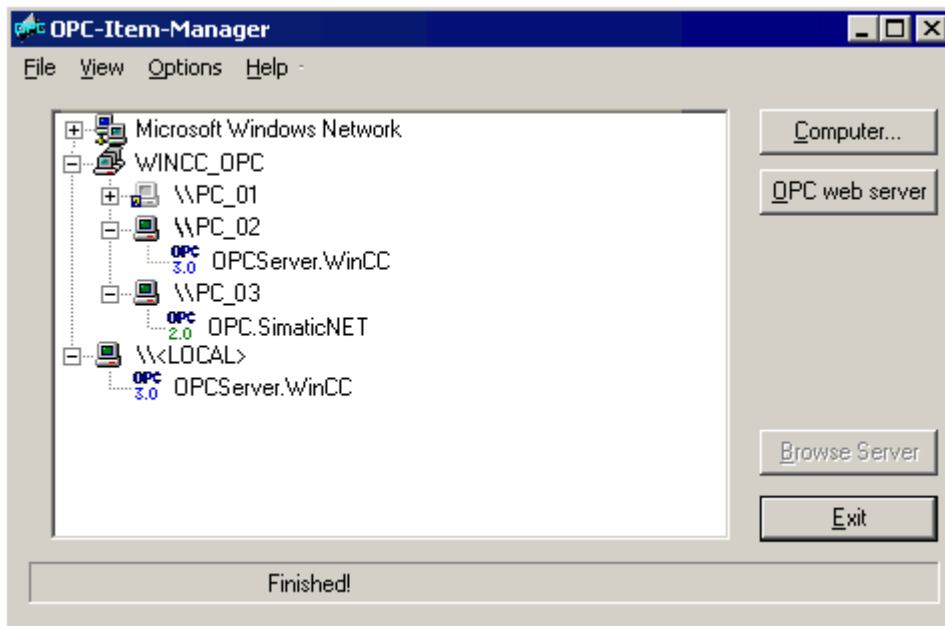
注記

[OPC]チャンネル

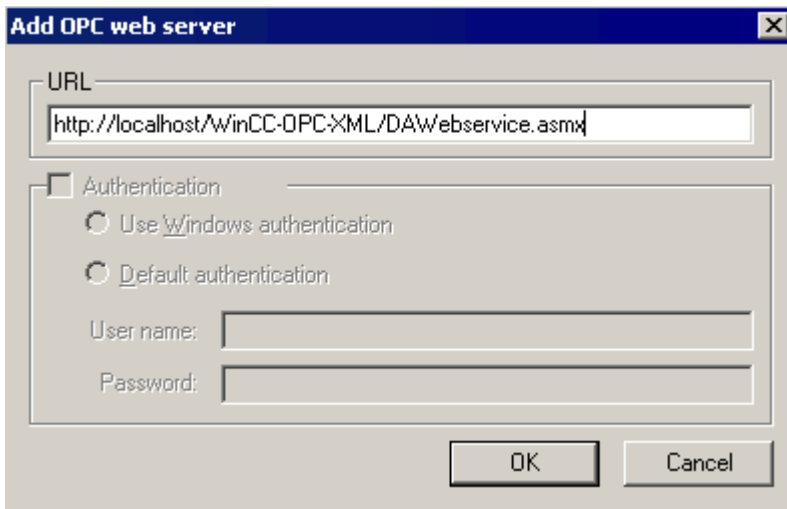
接続名で Unicode はサポートされていません。プロジェクト内のすべての接続に対して、必ず同じ言語で名前を付けているようにします。コンピュータのコントロールパネルを開き、Unicode がサポートされていないプログラムで使用するために、この言語のコードページを設定します。

手順

1. WinCC OPC XML クライアント上の[OPC グループ(OPCHN Unit#1)]チャンネルユニットのショートカットメニューで[システムパラメータ]を選択します。[OPC 項目マネージャ]が開きます。

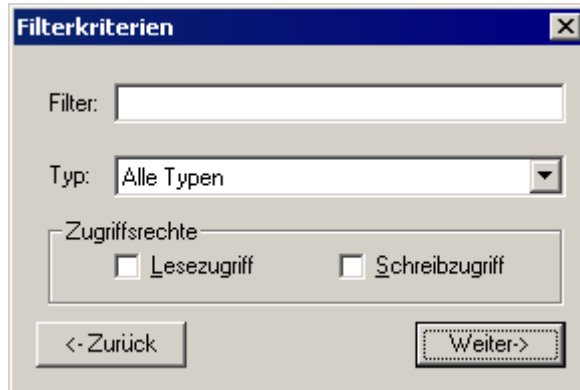


2. [OPC Web サーバー]ボタンをクリックします。[OPC Web サーバーの追加]ダイアログが開きます。以下の形式で、[URL]フィールドに、WinCC OPC XML サーバーの URL を入力します。<http://<xxx>/WinCC-OPC-XML/DAWebService.asmx>。xxx の部分は IP アドレスまたは OPC XML Web サービスを実行しているコンピュータ名です。

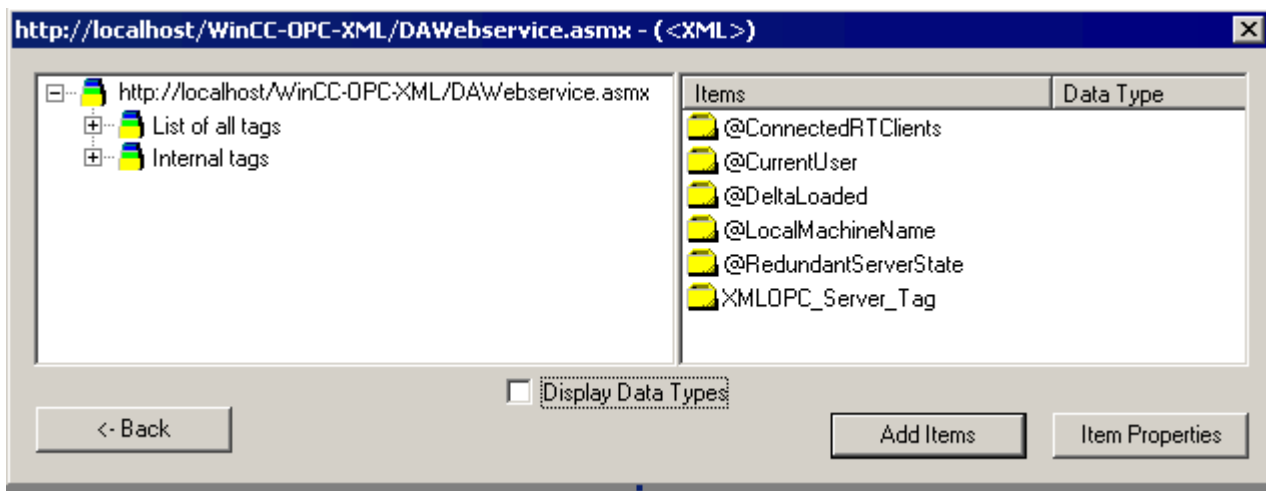


[OK]をクリックしてダイアログを閉じます。

3. リストが表示されるので、このリストから"/WinCC-OPC-XML/DAWebservice.asmx>"を選択します。[サーバーの参照]ボタンをクリックします。[フィルタ規準]ダイアログが開きます。



4. [フィルタ規準]ダイアログで、[次へ->]ボタンをクリックします。[http://...]ダイアログが開きます。



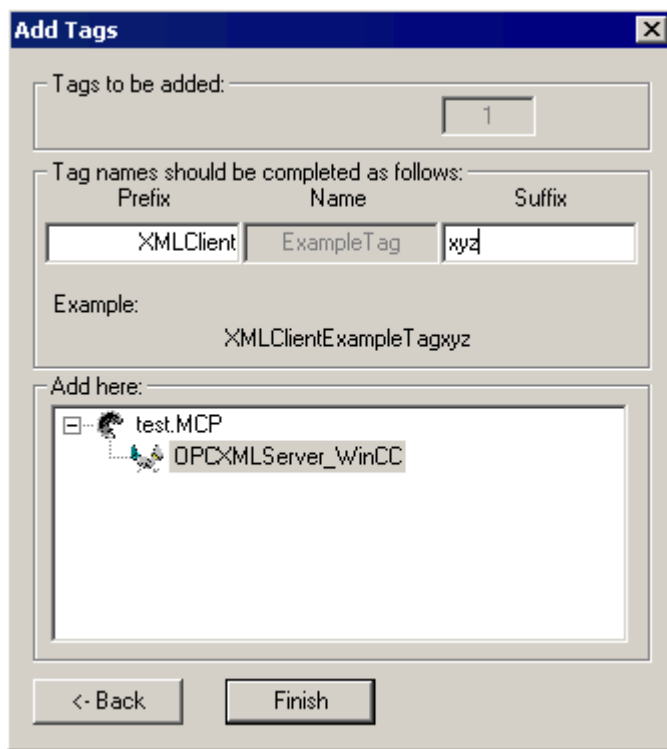
5. [http://...]ダイアログで[XML OPC_Server_Tag] WinCC タグを選択します。[項目の追加]ボタンをクリックします。
6. WinCC OPC XML サーバーとの接続がすでに存在している場合には、引き続き手順 7 に進みます。接続が設定されていない場合、対応するメッセージが出力されます。[はい]をクリックします。[新規接続]ダイアログが開きます。



この接続の名前"OPCXMLServer_WinCC"を入力します。[OK]をクリックします。

6.4 OPC チャンネル

7. [タグの追加]ダイアログが開きます。
[接頭語]フィールドに文字列"XMLClient_"を入力し、[接尾語]フィールドに"_xyz"を入力します。[OPCXMLServer_WinCC]接続を選択します。[終了]をクリックします。



8. [http://...]ダイアログで、[<戻る]ボタンをクリックします。[終了]をクリックして OPC 項目マネージャを閉じます。

下記も参照

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

6.4.5.3 OPC 項目マネージャを使用しない WinCC タグへのアクセス

はじめに

ブラウザ機能をサポートしていない OPC サーバーでは、アクセスを手動で設定する必要があります。WinCC 間の OPC 接続例を使用して、WinCC OPC XML クライアント上の WinCC タグの設定を示します。

注記

OPC 項目マネージャを使用せずに WinCC タグにアクセスするには、項目 ID を手動で設定する必要があります。WinCC タグをアドレス指定する場合、シンボルコンピュータ名(サーバー接頭語)も指定できます。項目 ID では、以下の構文が使用されます。
server_prefix::<@>WinCC タグ。ローカル WinCC プロジェクトの WinCC タグをアドレス指定する場合、項目 ID の構文は、以下のとおりです。<@>WinCC タグ。

設定の手順

WinCC OPC XML クライアントの WinCC プロジェクトでは、以下の設定が必要です。

1. 接続の作成
2. WinCC OPC XML クライアント上で、WinCC OPC DA サーバーの WinCC タグにアクセスする WinCC タグ"XMLClient_OPC_Var1_xyz"を設定する。

前提条件

- WinCC プロジェクトを持つコンピュータが 2 台あること。
- HTTP を使用してコンピュータにアクセスできることが必要。
- WinCC OPC XML サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"XMLOPC_Server_Tag"内部タグが設定されていること。
- WinCC OPC XML サーバーの WinCC プロジェクトが有効になっていること。
- WinCC OPC XML クライアントの WinCC プロジェクトに OPC チャンネルを追加する必要がある。

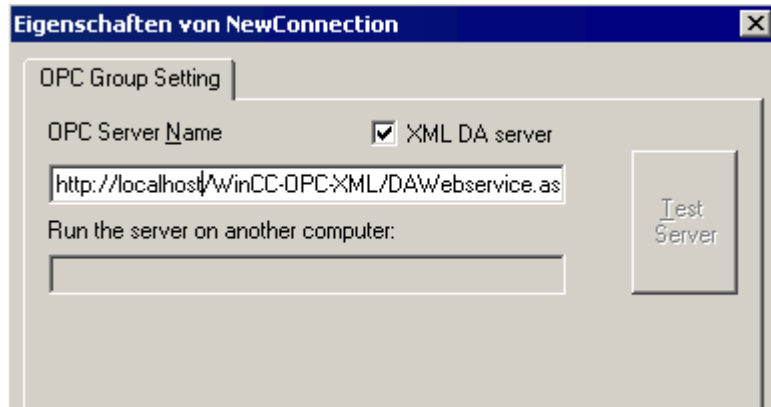
注記

OPC チャンネルで外部タグを設定する場合、[タグプロパティ]ダイアログの[タイプ変換]フィールドで、WinCC の事前設定値を変更しないでください。プロセス内のタグのデータタイプは、[アドレスプロパティ]ダイアログの[データタイプ]フィールドで定義します。

6.4 OPC チャンネル

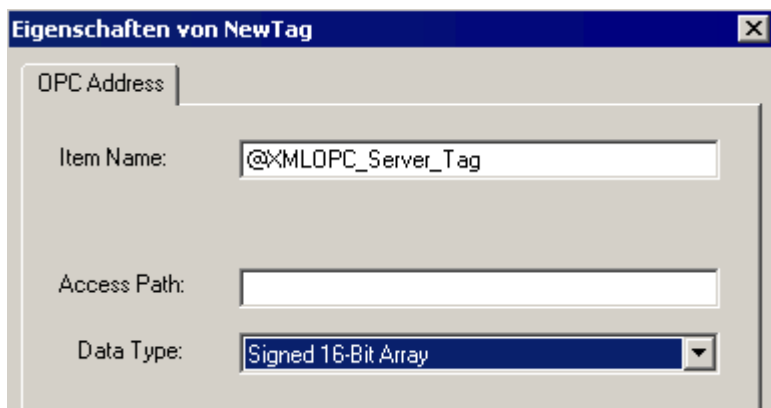
手順

1. WinCC OPC XML クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[新規接続]を選択します。[接続プロパティ]ダイアログが開きます。対応するフィールドにこの接続の名前を入力します。
2. [プロパティ]ボタンをクリックします。タイトルにその接続名が表示されたダイアログが表示されます。



[XML DA サーバー]チェックボックスを選択します。WinCC OPC XML サーバーと接続する場合、[OPC サーバー名]フィールドに、WinCC OPC XML サーバーの URL を指定する必要があります。URL では、以下の構文が使用されます。"http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx"。xxx の部分は IP アドレスまたは OPC-XML Web サービスを実行しているコンピュータ名です。

3. 接続のショートカットメニューで[タグの新規作成]を選択します。[タグのプロパティ]ダイアログが開きます。
4. [タグ]フィールドに名前"XMLClient_OPC_Var1_xyz"を入力します。データタイプを[符号付き 16 ビット]に設定します。
5. [タグプロパティ]ダイアログで[選択]ボタンをクリックします。タイトルにそのタグ名が表示されたダイアログが表示されます。



[項目名]フィールドに、WinCC OPC XML サーバーのシンボル"<@>"と WinCC タグ名を入力します。[アクセスパス]フィールド内のエントリはそのままにしておきます。データタイプを[符号付き 16 ビット]に設定します。

6. [OK]をクリックして、開いているダイアログをすべて閉じます。

6.5 OPC UA WinCC チャンネル

6.5.1 WinCC チャンネル「OPC UA WinCC チャンネル」

はじめに

WinCC は、OPC UA サーバーとしても OPC UA クライアントとしても使用できます。「OPC UA WinCC Channel」チャンネルは、WinCC の OPC UA クライアントアプリケーションです。

OPC UA (ユニファイドアーキテクチャ)で、WinCC は、OPC のプラットフォーム独立フォローアップテクノロジーをサポートします。OPC UA 仕様および WinCC Information System の[インターフェース]>[OPC - オープン接続性]>[WinCC OPC UA サーバー]の下に詳細が記載されています。設定には OPC UA の基本的な知識が必要です。

WinCC Information System の[通信]>[OPC チャンネル]の下に OPC-DA クライアントおよび OPC-XML クライアントのマニュアルがあります。

WinCC OPC UA クライアントの機能

WinCC OPC UA クライアントは、OPC ユニファイドアーキテクチャ仕様に準拠して、すべての OPC UA サーバーへのデータアクセスを可能にします。

OPC Unified Architecture (OPC UA)は、関係するパートナー間の安全な通信の認証および暗号化などの追加機構も提供します。

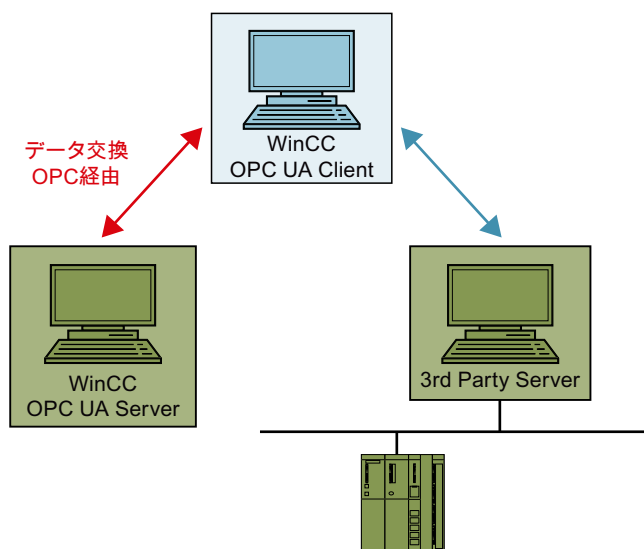
OPC UA による通信

WinCC を WinCC OPC UA クライアントとして使用するために、OPC UA 通信ドライバ「OPC UA WinCC Channel」を WinCC プロジェクトに挿入します。別の通信モジュールは必要ありません。

WinCC OPC UA クライアントは複数の OPC UA サーバーにアクセスできます。このために各 OPC UA サーバーへの接続を設定できます。WinCC OPC UA クライアントを集中型オペレータ制御および監視ステーションとして使用できます。

WinCC Configuration Studio のタグ管理で設定を実行します。

6.5 OPC UA WinCC チャンネル



6.5.2 サポートされているデータタイプの概要

はじめに

データ通信のために、WinCC OPC UA クライアントの WinCC プロジェクトの中の OPC UA サーバーのタグへのアクセスを構成します。

こうするには、タグ管理で OPC UA ノードを WinCC タグとしてインポートします。

サポートされるデータタイプ

WinCC OPC UA クライアントと WinCC OPC UA サーバーは以下の WinCC データタイプをサポートしています。

- 2 進タグ
- 符号付き 8 ビット値
- 符号なし 8 ビット値
- 符号付き 16 ビット値
- 符号なし 16 ビット値
- 符号付き 32 ビット値
- 符号なし 32 ビット値
- 浮動小数点数 32 ビット IEEE 754

- 浮動小数点数 64 ビット IEEE 754
- テキストタグ、8 ビットフォント
- テキストタグ、16 ビット文字セット
- 未処理データタイプ
- 構造タイプ
- 日付/時刻

6.5.3 OPC UA チャンネルの設定

6.5.3.1 全般シーケンス

はじめに

WinCC と OPC UA サーバーの間の OPC UA リンクにより、WinCC OPC UA クライアントは、セキュア接続を介して OPC UA サーバーのタグにアクセスします。OPC UA サーバーのデータは、この目的のために WinCC タグにマップされます。

タグ管理で設定を実行します。

必要条件

- OPC UA サーバーが有効であること。
- コンピュータを TCP 経由で接続する必要があること。
- 通信はファイアウォールでブロックしてはいけないこと。
- OPC UA サーバーのポート番号を有効にする必要があること。

設定手順

- WinCC プロジェクトへの OPC UA WinCC チャンネルの挿入 (ページ 286)
- OPC UA サーバーへの接続作成 (ページ 286)
- 証明書による認証の設定。(ページ 293)
- OPC UA タグの設定 (ページ 303)
- WinCC における OPC UA アラームの使用 (ページ 317)

6.5 OPC UA WinCC チャンネル

6.5.3.2 WinCC プロジェクトへの OPC UA WinCC チャンネルの挿入

必要条件

- WinCC プロジェクトが作成されていること。
- タグ管理が開いていること。

手順

1. ナビゲーションエリアのタグ管理のショートカットメニューを開きます。
2. [新規ドライバの追加] > [OPC UA WinCC Channel]を選択します。

結果

チャンネル OPC UA WinCC Channel が、タグ管理に追加されます。

チャンネルユニット「OPC UA Connections」がチャンネルに作成されます。このチャンネルユニットに、1 つ以上の OPC UA サーバーへの接続を設定します。

6.5.3.3 OPC UA サーバーへの接続作成

OPC UA サーバーに接続するには、サーバーおよびセキュリティ設定に関する情報が必要です。

それぞれの OPC UA サーバーのために WinCC プロジェクトで 1 つの接続を作成できます。

タグ管理で、接続のショートカットメニューのエントリを使って[接続パラメータ]ダイアログを開きます。

接続パラメータ

[UA サーバブラウザ]タブで、OPC UA サーバとの通信に必要な設定を設定します。

The screenshot shows the 'Line1' dialog box with the 'UA Server Browser' tab selected. The tree view on the left shows 'UA Server Discovery' selected. The 'Settings' section on the right contains the following fields:

- 接続名: (text input)
- サーバURI: (text input)
- セキュリティ設定:
 - セキュリティポリシー: (dropdown menu)
 - セキュリティモード: (dropdown menu)
- 認証設定:
 - ☐ 匿名
 - ☐ ユーザー名/パスワード: (text input)

At the bottom, there is a log table with the following data:

タイムスタンプ	タイプ	メッセージ
2017-06-29 10:00:08.994	Information	Discovery of Ua Servers Succeeded.
2017-06-29 10:00:08.994	Information	Browsing server succeeded - No Server(s) found.
2017-06-29 10:00:08.494	Information	Browsing local servers initiated.
2017-06-29 10:00:08.463	Information	Successfully created Connection parameter.

Buttons at the bottom: 証明書, OK, キャンセル, ヘルプ

フィールド/設定	内容
UA Server Discovery	UA Discovery Server は、使用可能な OPC UA サーバのリストを提供します。
Local Discovery	Local Discovery は、ローカルコンピュータ上のすべての OPC UA Server を一覧表示します。 これらは Local Discovery Server (LDS)に登録されています。 必要条件:LDS がローカルコンピュータにインストールされていること。

6.5 OPC UA WinCC チャンネル

フィールド/設定	内容
Custom Discovery	<p>Custom Discovery では、ユーザーが接続名を使用して OPC UA Server を手動で入力できます。</p> <p>これは特に、OPC UA Server がリモートコンピュータ上にある場合に必要です。</p> <p>OPC UA Server が Discovery Server に登録されていない場合は、以下の形式で OPC UA Server のディスカバリアドレスを入力します。</p> <ul style="list-style-type: none"> • <opc.tcp://Discovery Server のアドレス:ポート番号> <p>サーバーリストのショートカットメニューで[参照]コマンドを使用してサーバー表示を更新します。</p>
接続名	<p>OPC UA サーバーの接続名です。</p> <p>OPC UA サーバーの名前は、このフィールドの影響を受けません。</p>
サーバー URI	OPC UA サーバーのアドレス。
セキュリティポリシー	<p>OPC UA サーバーが提供するセキュリティポリシーの 1 つを選択します。</p> <ul style="list-style-type: none"> • なし • Basic128Rsa15 • Basic256 • Basic256Sha256 • 最もセキュアなオプションを自動的に選択
セキュリティモード	<p>希望するセキュリティモードを選択します。</p> <ul style="list-style-type: none"> • 署名 • 署名と暗号化
認証設定	<p>接続にユーザー ID が必要であるか、匿名アクセスを許可するかを選択します。</p> <p>ユーザー ID を設定する場合は、WinCC OPC UA クライアントが OPC UA サーバーにアクセスするユーザー名およびパスワードを入力します。</p> <p>オーソリゼーションが OPC UA サーバーによりチェックされます。</p> <p>WinCC OPC UA サーバーを使用して、OPC UA サーバーが実行されているパソコンの Window ユーザー管理を介して、オーソリゼーションは設定されます。</p>
証明書	インストールディレクトリで「PKI」フォルダを開きます。

[設定]タブで、OPC UA サーバーとの通信に必要な接続設定を設定します。

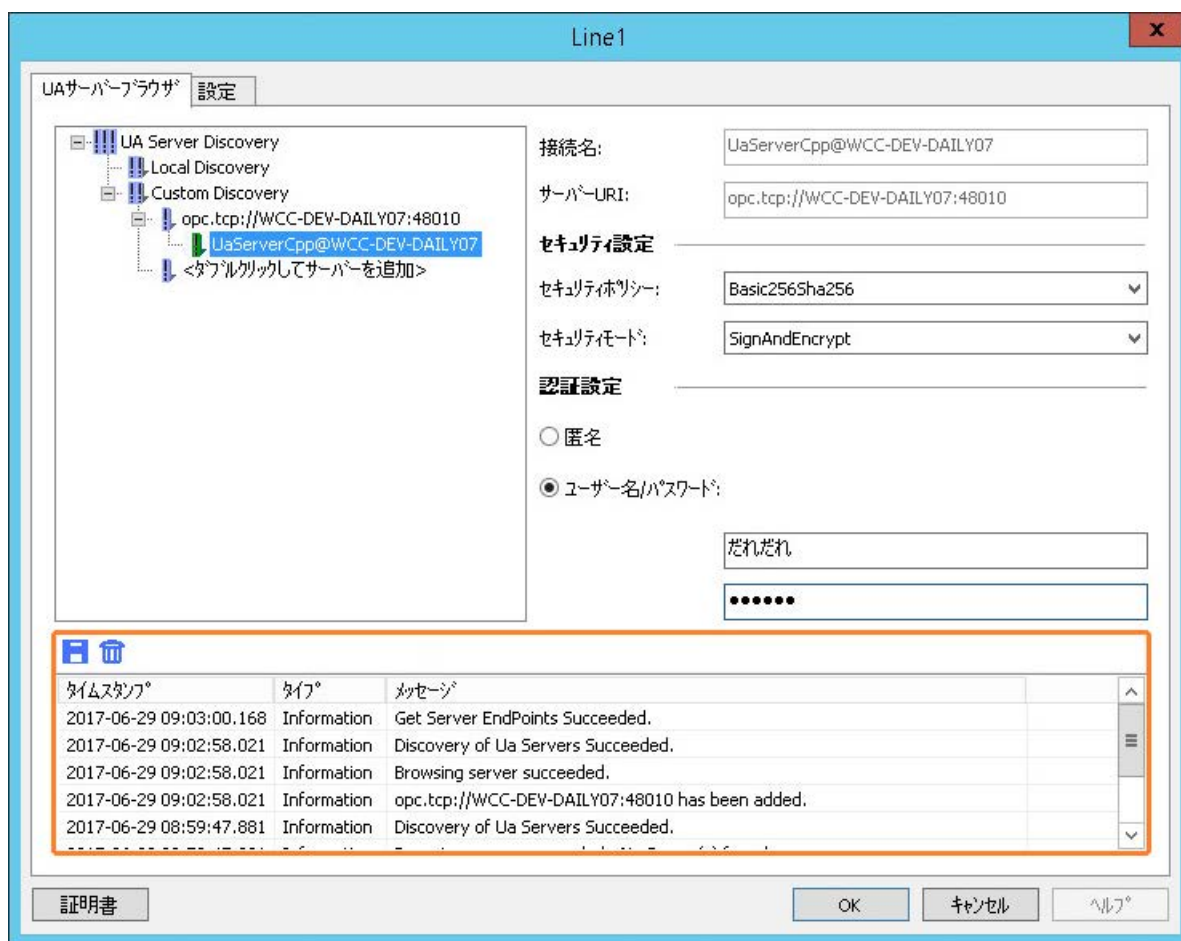
フィールド/設定	内容
CallTimeout (ms)	読み取り、書き込み、参照、接続、すべての OPC UA 操作のタイムアウト。
BrowseTimeout (ms)	タグなど、OPC UA オブジェクトの参照のタイムアウト。 サイズの大きなオブジェクトツリーの参照には数秒かかる場合があるため、BrowseTimeout は、参照に関して CallTimeout を上書きします。

6.5 OPC UA WinCC チャンネル


フィールド/設定	内容
ConnectTimeout (ms)	接続確立のタイムアウト。 サーバーは別のパソコンで実行する場合があります、リモートアクセスが可能なため、ConnectTimeout は接続に関して CallTimeout を上書きします。
IndexRange を使用しない配列エントリの書き込み	リモートアクセスのための IndexRange サポートを有効化または無効化します。

コンソール

[コンソール]出力ウィンドウには、ステータス情報およびエラーメッセージがタイムスタンプとメッセージのタイプと共に含まれます。




メッセージの削除

次のボタンはすべてのエントリを削除します。 

通知
<p>削除:[元に戻す]は使用できません</p> <p>削除したエントリは回復できません。</p>

メッセージの保存

次のボタンをクリックすると、メッセージをログファイルに保存できます。 

ここで開いたダイアログから、保存場所を選択します。

メッセージのショートカットメニューを使って、個別のメッセージをクリップボードにコピーできます。

タイムスタンプ	タイプ	メッセージ
2017-06-29 10:04:48.000	Information	Get Server EndPoints Succeeded.
2017-06-29 10:04:48.000	Information	Clearing port to 11000. EV-DAILY07 rebrowsing initiated.

下記も参照

OPC UA サーバーへの接続の設定方法 (ページ 300)

NAT ルータ経由の OPC UA サーバーへの接続作成 (ページ 291)

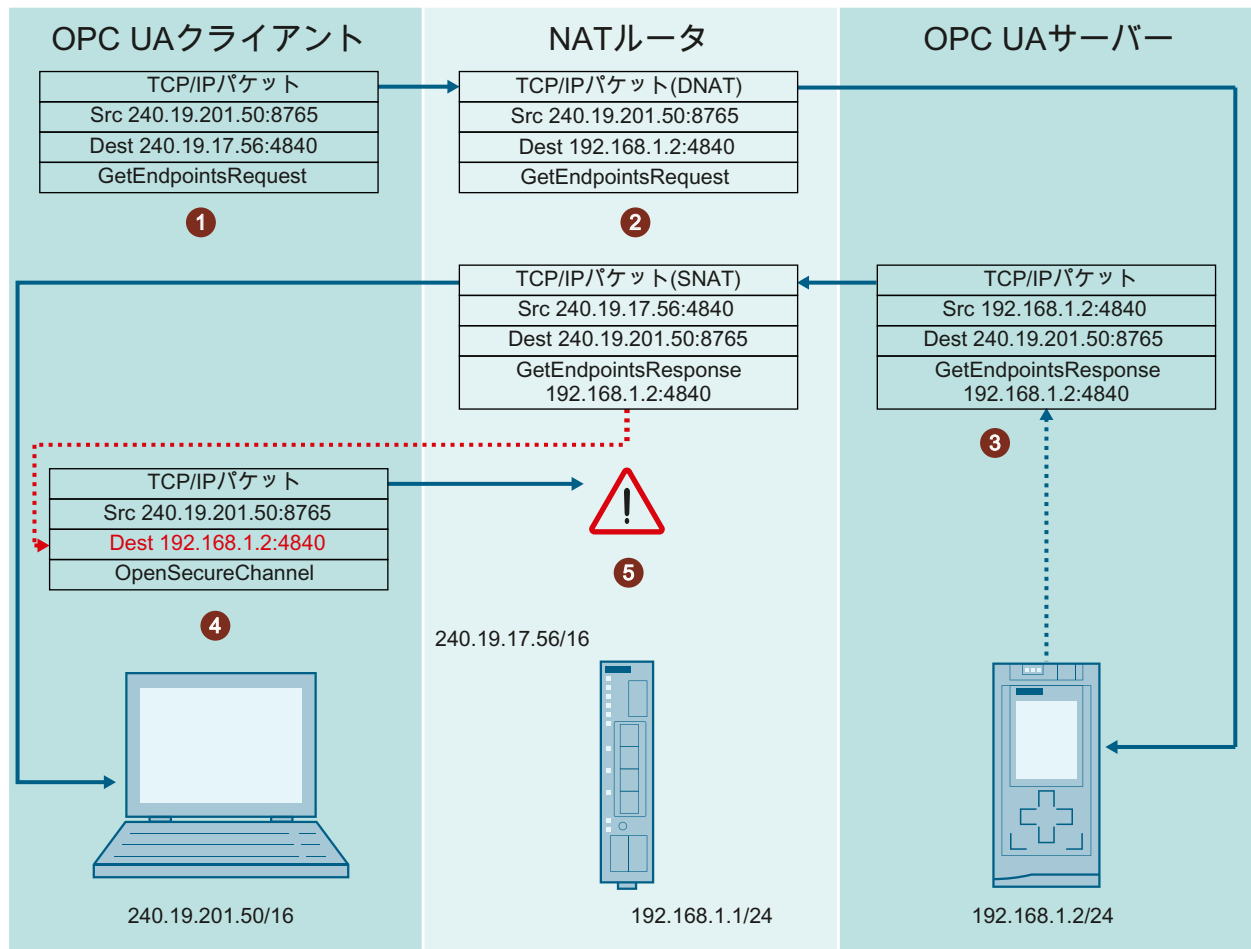
6.5.3.4 NAT ルータ経由の OPC UA サーバーへの接続作成

NAT ルータを介して OPC UA サーバーに接続を確立しようとする時、エラーメッセージ「BadCommunicationError」または「BadNotConnected」が表示されて接続が失敗することがあります。

NAT システムでは、ルータによって IPv4 パケットが操作されます。その結果、パケットの送信元 IP (「送信元 NAT」)または宛先 IP (「宛先 NAT」)は、(宛先ポートに基づいて)ルータに設定されている IP アドレスに置き換えられます。

問題のある接続確立

次の例では、NAT ルータを介した、問題のある接続確立を示しています。



1. OPC UA クライアントが、宛先 IP アドレス(DNAT) 240.19.17.56 を指定して「GetEndpointsRequest」を送信します。
2. NAT ルータが、このパケットを OPC UA サーバーに送信するため、パケットの宛先 IP を 192.168.1.2 に変更します。
3. OPC UA サーバーが、その独自の IP アドレス(192.168.1.2)などの情報を含む「GetEndpointResponse」を生成し、NAT ルータを介してこのテレグラムをクライアントに送り返します。
クライアントがこのテレグラムを割り付けることができるようにするため、ルータはパケットの送信元 IP アドレス(SNAT)を 240.19.17.56 に調整する必要があります。
4. クライアントが、サーバーの「GetEndpointResponse」に含まれる IP アドレスへの接続(「OpenSecureChannel」)の確立を開始します。
NAT ルータは、サーバーの「GetEndpointResponse」内の IP アドレスを操作できません。
5. しかし、クライアントから応答テレグラムの IP アドレスに直接到達することは不可能です。このため、この接続は確立できません。

解決策:OPC UA クライアントが IP アドレスを比較する

OPC UA クライアントが、「GetEndpointRequest」の IP アドレスを、OPC UA サーバーによって送信された「GetEndpointResponse」の IP アドレスと比較します。

それらの IP アドレスが異なる場合、OPC UA クライアントは「GetEndpointResponse」の IP アドレスを「GetEndpointRequest」の IP アドレスに置き換えます。

これにより、OPC UA クライアントは操作された「GetEndpointResponse」の情報に接続できるようになります。

証明書検証中のトラブルシューティング

エラーメッセージ「BadCertificateHostNameInvalid」が表示されて接続試行が終了する場合、「ValidationOptions」を使用して設定クライアントおよびランタイムクライアントの設定ファイル内のエラーを抑止することができます。

「TrustedCertificateStore」と「IssuerCertificateStore」に「SuppressHostNameInvalid」フラグを設定する必要があります。

詳細情報は、「証明書による認証の設定。(ページ 293)」を参照してください。

下記も参照

OPC UA サーバーへの接続作成 (ページ 286)

6.5.3.5 証明書による認証の設定。

サーバー証明書およびクライアント証明書

設定するとき、クライアントとサーバーの認証を区別します。安全な通信は、クライアントとサーバーが互いの認証を認識した場合にのみ可能です。

認証は、それぞれのコンピュータにリンクされます。WinCC プロジェクトを別のコンピュータに移動、コピー、または複製した後、個々のコンピュータが他のコンピュータの認証を確認できるように、この手順を繰り返します。

診断

コンソールおよび WinCC チャンネル診断を分析のために使用します。

詳細情報は、以下を参照してください。

- [インターフェース]>[OPC - オープン接続性]>[WinCC OPC UA サーバー]>[OPC UA のセキュリティコンセプト (ページ 783)]
- [通信]>[通信診断]>[診断チャンネル「OPC」 (ページ 669)]

証明書の取り扱い

インストール時に WinCC OPC UA Client の自己署名証明書が作成されます。

サーバーがこのクライアント証明書が信頼できると認める場合のみ、OPC UA Client は、OPC UA Server に接続できます。

新しい接続を作成するとき、OPC UA Server は、クライアント証明書を確認します。WinCC チャンネル「OPC UA WinCC Channel」を介する通信では、サーバーは、設定クライアントおよびランタイムクライアントの証明書を信頼できるものと認識する必要があります。

証明書は、WinCC インストールパスの以下のフォルダにあります。

証明書	opc\UAClient\PKI\OPCUA\certs
<ul style="list-style-type: none">• Siemens OPC UA Tag Importer for WinCC• Siemens OPC UA Client for WinCC Runtime	
秘密鍵	opc\UAClient\PKI\OPCUA\private

拒否された証明書

サーバーがクライアント証明書を信頼できると認めない場合は、接続は拒否され赤色でマークされます。

以下のように、メッセージがコンソールで生成されるか、ログファイルに入力されます。
例:

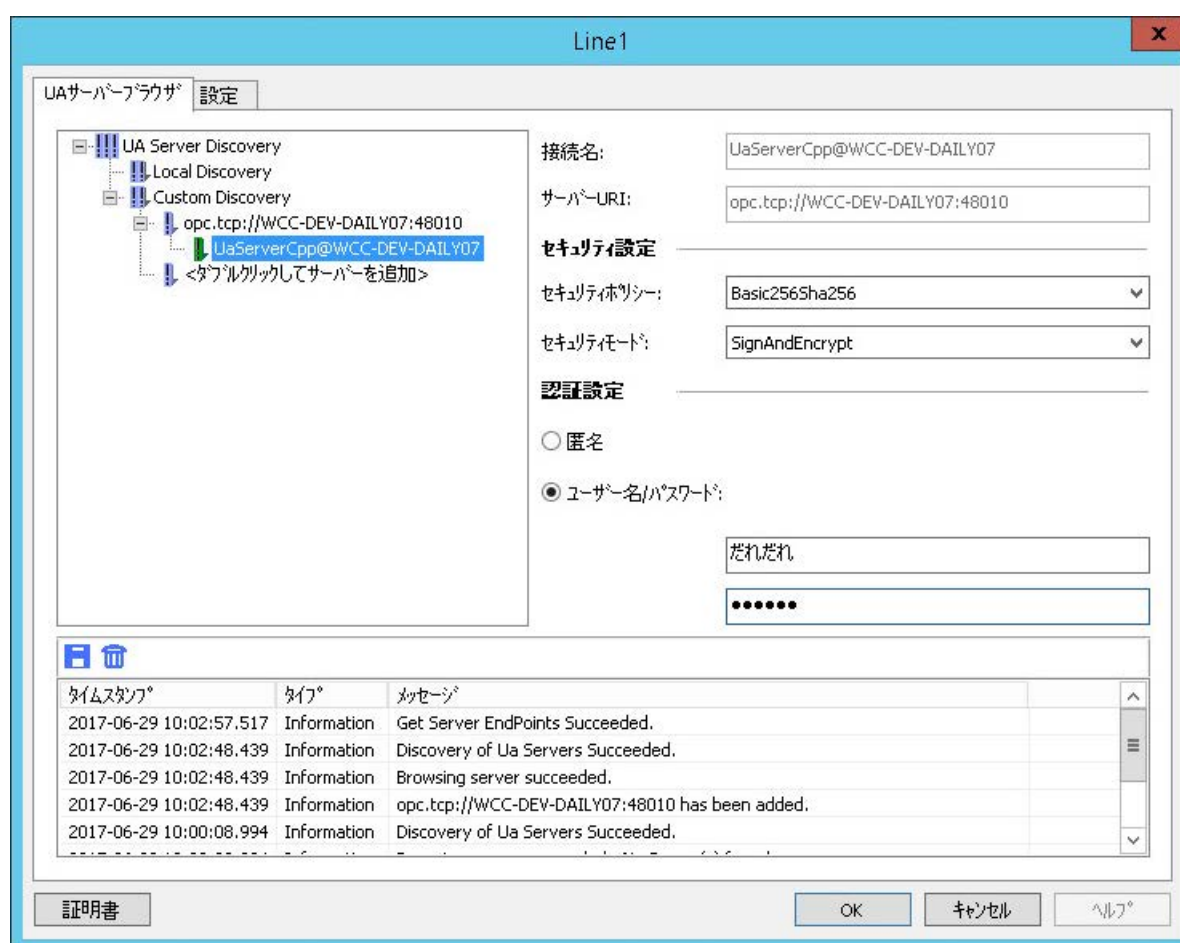
- Discovery of UA Server failed - The Certificate is not trusted.

拒否された各証明書は、「PKI\...\Rejected\Certs」フォルダに保存されます。

証明書を信頼する

証明書が信頼されていることを指定するには、証明書を「Trusted\Certs」フォルダに移動します。

これを行うには、[UA サーバブラウザ]タブで[証明書]ボタンをクリックし、「PKI」フォルダを開きます。



信頼すべきでない認証機関の証明書を「Issuer\certs」フォルダに(可能であれば、対応する証明書失効リストの「Issuer\crl」フォルダに)移します。

設定ファイル

説明	アプリケーション	設定ファイル
設定クライアント 証明書: Siemens OPC UA Tag Importer for WinCC 有効なクライアント証明書が見つからない場合は、接続を確立しようとする試みを中止します。 保存パス: <ul style="list-style-type: none"> opclUAClient\UaConfigServer 	CCOpCuaImporter.exe	CCOpCuaImporter.xml
ランタイムクライアント 証明書: Siemens OPC UA Client for WinCC Runtime 有効なランタイム証明書がない場合は、現在の値はランタイムに表示されません。 保存パス: <ul style="list-style-type: none"> opclUAClient\UaDAS 	CcUaDAS.exe	CcUaDAS.xml

2 つのクライアントの設定ファイルでは、証明書パラメータは<CertificateDescriptor>セクションの<ClientConfiguration>で参照できます。

例:証明書の制御のためのパラメータ

```

<ClientConfiguration>
  <CertificateDescriptor>
    <OrganizationUnit>DF PL DER HMI</OrganizationUnit>
    <Organization>Siemens AG</Organization>
    <Country>DE</Country>
    <KeyLength>2048</KeyLength>
    <SignatureAlgorithm>SHA256</SignatureAlgorithm>
    <LifetimeInMonths>60</LifetimeInMonths>
  </CertificateDescriptor>

```


説明

パラメータ	意味
OrganizationUnit Organization Country	説明エレメント パラメータは変更でき、アプリケーションの機能には影響はありません。
KeyLength	証明書を作成するときに使われた秘密鍵の長さ 長さは署名のアルゴリズムに依存します。 <ul style="list-style-type: none"> 1024:OPC UA を介した安全な通信のための最短の長さ 2048:Sha256 が使用されるとき最短の長さ¹⁾
SignatureAlgorithm	証明書に署名するために使用される署名アルゴリズム <ul style="list-style-type: none"> 可能な値:Sha1、Sha224、Sha256、Sha384、Sha512 通常値:Sha1、Sha256 デフォルト値:鍵の長さが 2048 の Sha256¹⁾
LifetimeInMonths	証明書の有効期間(月単位) 指定された時間が経過した後、クライアントはこの証明書を使用して操作できなくなります。 <ul style="list-style-type: none"> デフォルト値:60

1) セキュリティポリシー「Basic256Sha256」を使用する OPC UA サーバーへのセキュア接続を確立するには、このサーバーおよび OPC UA クライアントには次の値を持つ証明書が必要です。

- KeyLength:最低 2048
- SignatureAlgorithm:Sha256

新しいクライアント証明書の作成

OPC UA クライアントに対して新しい証明書を作成するには、管理者権限が必要です。

新しい証明書を作成するとき、信頼設定がリセットされます。以前に信頼できると認識された証明書は信頼されなくなります。

1. バックアップを作成します。
2. 該当するフォルダにある既存の証明書と関連する秘密鍵を削除します。
3. 設定ファイルで、証明書パラメータを更新し、XML ファイルを保存します。
4. Windows で管理者権限を使用して DOS ウィンドウ「cmd.exe」を開きます。

5. 証明書を作成するには、対応する OPC UA アプリケーションのインストールパスに進みます。
6. 対応する呼び出しを入力します。
 - CCOpcUaImporter.exe/CreateCertificate
 - CcUaDAS.exe/CreateCertificate保存パスに新しい証明書と秘密鍵が作成されます。
新しい証明書が信頼されていることを指定します。

クライアント証明書の自動作成

アプリケーション証明書ストアに証明書が見つからず、設定ファイルで「ForceCertificateCreation」フラグが「true」に設定されている場合、クライアントはその独自のインスタンス証明書を作成します。

このインスタンス証明書は、クライアントが起動するとすぐに作成されます。

```
- <ClientConfiguration>  
  + <CertificateDescriptor>  
    <ForceCertificateCreation>true</ForceCertificateCreation>
```

証明書チェックエラーの抑止

注記

証明書チェックで抑止されるエラーは、オペレータに通知する必要があります。

証明書は厳密にチェックされます。

クライアントの両方のアプリケーションのいくつかの既知のエラーを抑止することは依然として可能です。

クライアントの独自のインスタンス証明書

クライアントの起動時には、その独自のインスタンス証明書がチェックされます。

その証明書が無効であるか、有効期限が切れている場合は、サーバーへの接続を確立することはできません。

有効な証明書をインストールできます。

別の方法として、クライアントの設定ファイルで発生するエラーを抑止することもできます。

このためには、「ApplicationCertificate」で「ValidationOptions」を適合させます。

信頼される証明書と認証機関の証明書

サーバー証明書のチェックは、無効な証明書が原因となって失敗することがあります。クライアントの設定ファイルで発生するエラーは抑止できます。

このためには、「TrustedCertificateStore」と「IssuerCertificateStore」で「ValidationOptions」を適合させます。

注記

[ValidationOptions]のフラグは、[TrustedCertificateStore]と[IssuerCertificateStore]で同一でなければなりません。

「ValidationOptions」

フラグ	説明	コメント
SuppressCertificateExpired	証明書または認証機関の有効期間の満了が原因で発生するエラーを無視します。	
SuppressHostNameInvalid	アプリケーションのホスト名または Uniform Resource Identifier (URI)との間の一致の欠落を無視します。	このチェックは、サーバー側の検証にはスキップされます。
SuppressRevocationStatusUnknown	認証機関の証明書失効リストが存在しないために発生するエラーを無視します。	このエラーはデフォルトでは抑止されています。
CheckRevocationStatusOnline	失効ステータスをオンラインでチェックします。	この機能はサポートされていません。
CheckRevocationStatusOffline	失効ステータスをオフラインでチェックします。	このチェックはスキップされます。
UseDefaultOptions	標準のオプションを使用:有効期間とホスト名がチェックされます。	

例: 証明書の[ValidationOptions]

```

<StoreType>Directory</StoreType>
<StorePath>[ApplicationPath]\..\PKI\OPCUA</StorePath>
<SubjectName>Siemens OPC UA Client for WinCC Runtime</SubjectName>
<Thumbprint/>
<RawData/>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid-->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
  <OfflineRevocationList/>
  <OnlineRevocationList/>
</ApplicationCertificate>
- <TrustedCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\..\PKI\Trusted</StorePath>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid,
    SuppressRevocationStatusUnknown, CheckRevocationStatusOnline, CheckRevocationStatusOffline -->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
</TrustedCertificateStore>
+ <TrustedCertificates>
- <IssuerCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\..\PKI\Issuers</StorePath>
  <!--One or more of: UseDefaultOptions, SuppressCertificateExpired, SuppressHostNameInvalid,
    SuppressRevocationStatusUnknown, CheckRevocationStatusOnline, CheckRevocationStatusOffline -->
  <ValidationOptions>UseDefaultOptions</ValidationOptions>
</IssuerCertificateStore>
- <IssuerCertificates>

```

下記も参照

OPC UA のセキュリティ概念 (ページ 783)

OPC UA サーバーへの接続の設定方法 (ページ 300)

NAT ルータ経由の OPC UA サーバーへの接続作成 (ページ 291)

"OPC"チャンネルの診断 (ページ 669)

6.5.3.6 OPC UA サーバーへの接続の設定方法

はじめに

このセクションでは、OPC UA サーバーへの接続方法を説明します。

必要条件

- OPC UA サーバーが有効であること。
- 通信はファイアウォールでブロックしてはいけないこと。
OPC UA サーバーのポート番号を有効にする必要があること。

- WinCC パソコンから IP アドレスを使用して、OPC サーバーのパソコンにアクセスできる必要があること。
- OPC UA サーバーがクライアント証明書を信頼すること。
- 「OPC UA WinCC Channel」チャンネルが WinCC OPC UA クライアントの WinCC プロジェクトに追加されていること。

手順

1. WinCC Configuration Studio で WinCC タグ管理を開きます。
2. [OPC UA Connections]のショートカットメニューを使用して新しい接続を作成します。
3. 接続に名前を付けます。

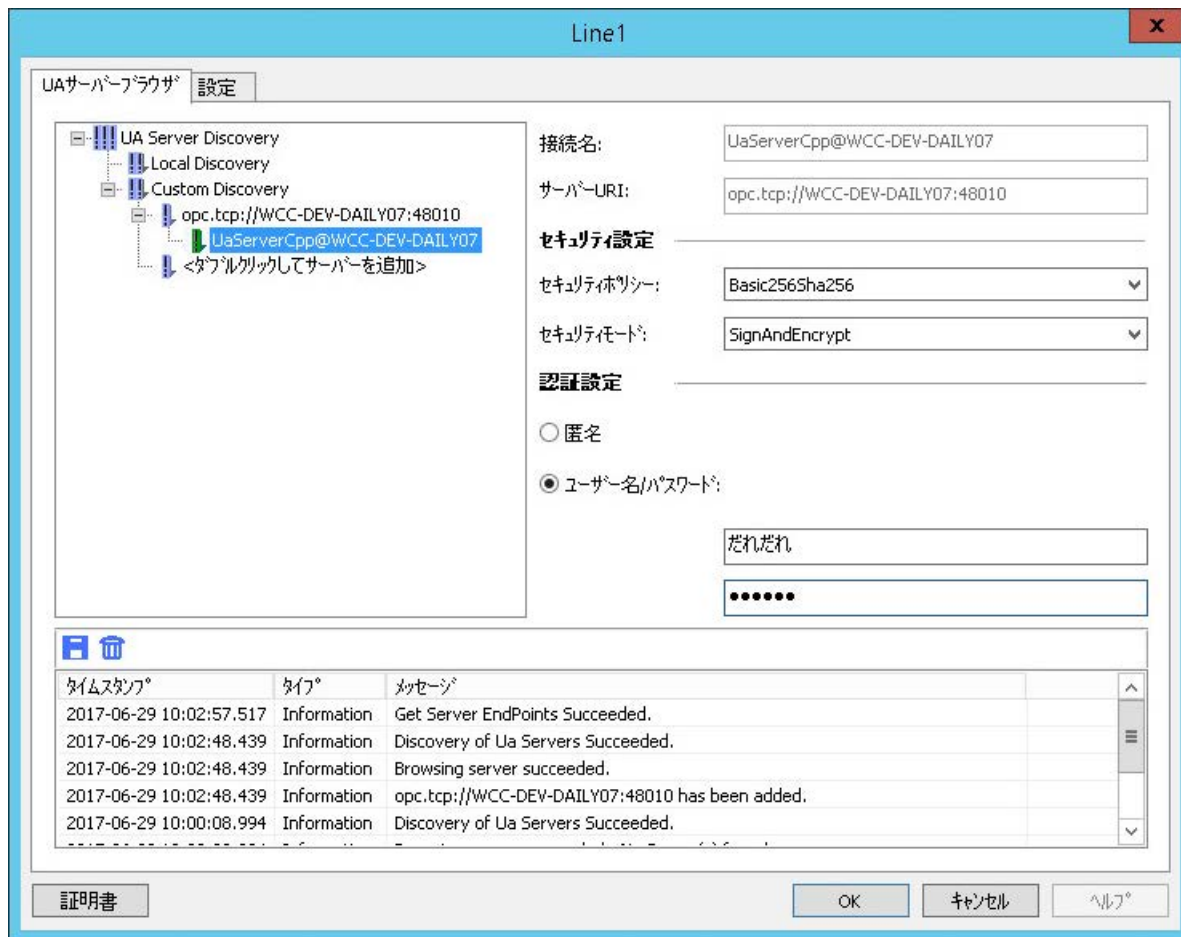
注記

接続の名前はランタイムが無効になったときにのみ変更できます。

4. 作成した接続のショートカットメニューから[接続パラメータ]ダイアログを開きます。
5. サーバーを選択します。
 - ローカル OPC UA サーバーの表示を更新するには、[Local Discovery]のショートカットメニューで[参照]を選択します。
 - OPC UA サーバーの URL を入力するには、[Custom Discovery]の行をダブルクリックします。
次の形式で IP アドレスを入力します。
- opc.tcp://<OPC-UA-Server_Address:Port_Number>

6.5 OPC UA WinCC チャンネル

6. 希望の設定を指定し、[OK]を押して入力を確認します。
OPC UA サーバーへの接続が確立され、[UA サーバーブラウザ]タブに緑色でマークされます。



7. 接続確立と接続ステータスのシステムタグを作成するには、接続のショートカットメニューで[有効化/無効化タグの作成]エントリを選択します。
内部タググループ「ConnectionStates」に以下のタグが作成されます。
- @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx

結果

ランタイムが有効な場合、接続はタグ管理に緑色のチェックマークでマークされます。

クライアント証明書

OPC UA サーバーがクライアント証明書を認識しない場合は、接続は確立されません。

接続が[UA サーバーブラウザ]タブに赤色でマークされます。接続はタグ管理のプロジェクトツリーに赤色の感嘆符でマークされます。

OPC UA サーバーがクライアント証明書を受け入れることを確認してください。

下記も参照

[シンボル]の表示 (ページ 306)

OPC UA タグ (ページ 303)

証明書による認証の設定。(ページ 293)

6.5.3.7 OPC UA タグの設定

OPC UA タグ

サポートされる OPC UA ノード


OPC UA サーバーへの接続が確立されると、OPC UA サーバーのオブジェクトおよび OPC UA ノードが、[シンボル]ビューにロードされます。

データタイプ

以下のデータタイプのあるタグがサポートされています。

OPC UA ノード	WinCC タグタイプ
Binary Tag	2 進タグ
Byte	符号付き 8 ビット値または符号なし 8 ビット値
Int16	符号付き 16 ビット値
UInt16	符号なし 16 ビット値
Int32	符号付き 32 ビット値
UInt32	符号なし 32 ビット値
Float	浮動小数点数 32 ビット IEEE 754 または浮動小数点数 64 ビット IEEE 754
String	テキストタグ 8 ビット文字セットまたはテキストタグ 16 ビット文字セット
ByteString	未処理データタグ
DateTime	日付/時刻
Enumerations	符号付き 32 ビット値

OPC UA ノードを WinCC タグとしてインポート

WinCC タグとしてインポートできる OPC UA ノードは、データ記憶装置エリアの[アクセス]列で  アイコンを使用してアクセスできます。サポートされないデータタイプに対してこのフィールドは無効になります。

どの WinCC データタイプとも適合しない OPC UA ノードをインポートするには、適合する WinCC データタイプを個別の OPC UA ノードに手動で割り付けます。

OPC UA ノードがインポートできるのは 1 回だけです。

追加情報は、「OPC UA ノードを WinCC タグとしてインポートする方法 (ページ 308)」を参照してください。

WinCC タグ名

OPC UA ノードをインポートするとき、WinCC タグの名前が自動的に割り付けられます。

OPC UA ノードのロード後に各接続のプロパティのタグ名を設定します。

OPC UA サーバー上のタグ名に特殊文字が指定されている場合、これらの特殊文字はアンダースコア「_」に置換されます。

名前生成の設定

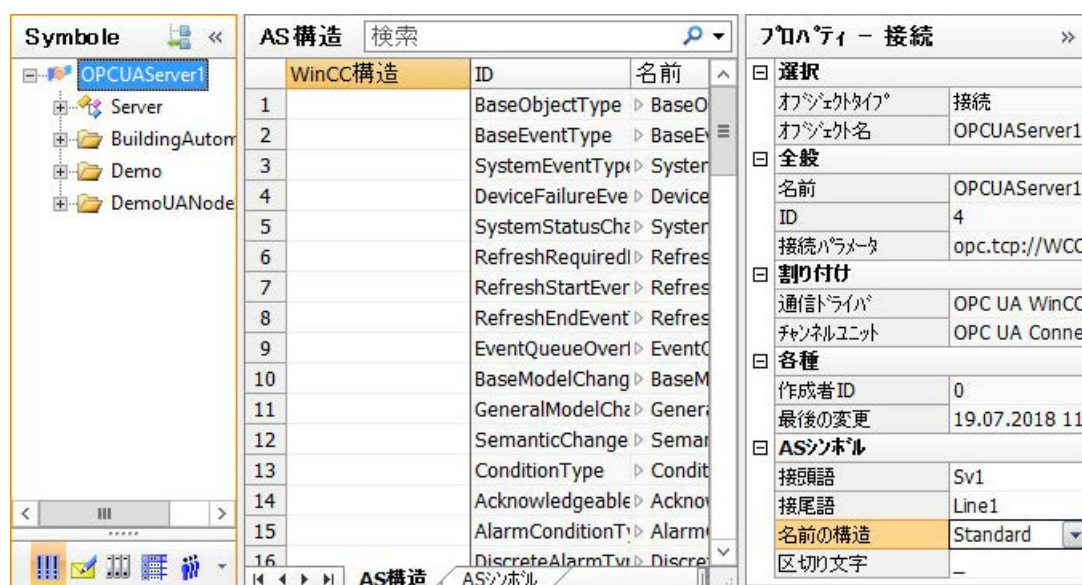
[AS シンボル]プロパティグループの設定を表示するには、[シンボル]ビューで接続名をクリックします。

以下の設定を行うことができます。

- 名前として、OPC UA ノードのパスが適用されます。
- OPC UA ノードの名前が適用され、必要に応じて接頭語または接尾語が補足されます。
- OPC UA ノードのパスが適用され、必要に応じて接頭語または接尾語が補足されます。

[接頭語]または[接尾語]オプションで、指定した文字列がタグ名に追加されます。

名前の構成部分がセパレータにより接続されます。アンダースコアがデフォルトで使用されます。



例

パス「Spectrometer/Channel_0/ChannelStateMachine」に OPC UA サーバーの「CurrentState」タグがあります。

「Prefix_」が[接頭語]フィールドに入力され、[_Suffix]が[接尾語]フィールドに入力されています。

以下の WinCC タグが、WinCC OPC UA クライアントの WinCC プロジェクトに作成されています。

設定	WinCC タグ名
接頭語および接尾語のないパス名	Spectrometer/Channel_0/ChannelStateMachine/CurrentState
OPC UA ノードの名	Prefix_CurrentState_Suffix
パス名	Prefix_Spectrometer/Channel_0/ChannelStateMachine/CurrentState_Suffix

WinCC タグの削除

OPC UA サーバーへの有効な接続は、WinCC OPC UA タグを削除するためには不要です。

WinCC タグ管理でインポートされた WinCC タグを削除するには、WinCC タグのコンテキストメニューから[削除]エントリを選択するか、またはキーを使います。

タググループの作成

接続の下にタググループを作成するには、接続のコンテキストメニューで[新規グループ]を選択します。

名前を変更するには、グループ名をクリックします。

WinCC タグの作成時には、次の操作に従ってください。

- プロジェクトナビゲーションウィンドウで接続が選択されます。
 - 接続のすぐ下に WinCC タグが作成されます。
 - [タグ]データ記憶装置エリアには、タグユーザーグループに割り付けられている場合も、すべてのタグが表示されます。
[グループ]フィールドには、割り付けられているタググループが含まれることがあります。ドロップダウンリストボックスを通じてグループの割り付けを変更できます。
- タググループがプロジェクトナビゲーションウィンドウで選択されます。
 - WinCC タグがタググループで作成されます。
 - [タグ]データエリアには、タググループで作成されたタグだけが表示されます。

WinCC プロジェクトの移行

WinCC V7.4 以前では、WinCC OPC UA 接続が OPC チャンネルで作成されました。

WinCC プロジェクトへの移行では、WinCC OPC UA クライアントの接続およびタグも変更された構造に移行されます。

エクスポートした WinCC OPC UA タグがある場合は、次の順序に従ってください。

1. エクスポートされた WinCC OPC UA タグをインポートします。
2. WinCC プロジェクトを移行します。

下記も参照

[シンボル]の表示 (ページ 306)

OPC UA ノードを WinCC タグとしてインポートする方法 (ページ 308)

[シンボル]の表示

はじめに

接続の設定が正常に行われた後には、OPC UA サーバーの OPC UA ノードにアクセスできます。

WinCC タグを作成するには、タグ管理の OPC UA ノードを[シンボル]の表示にロードします。

シンボルの表示

データ構造の表示

構造ツリーのデータの表示は、OPC UA サーバーの階層に対応しています。

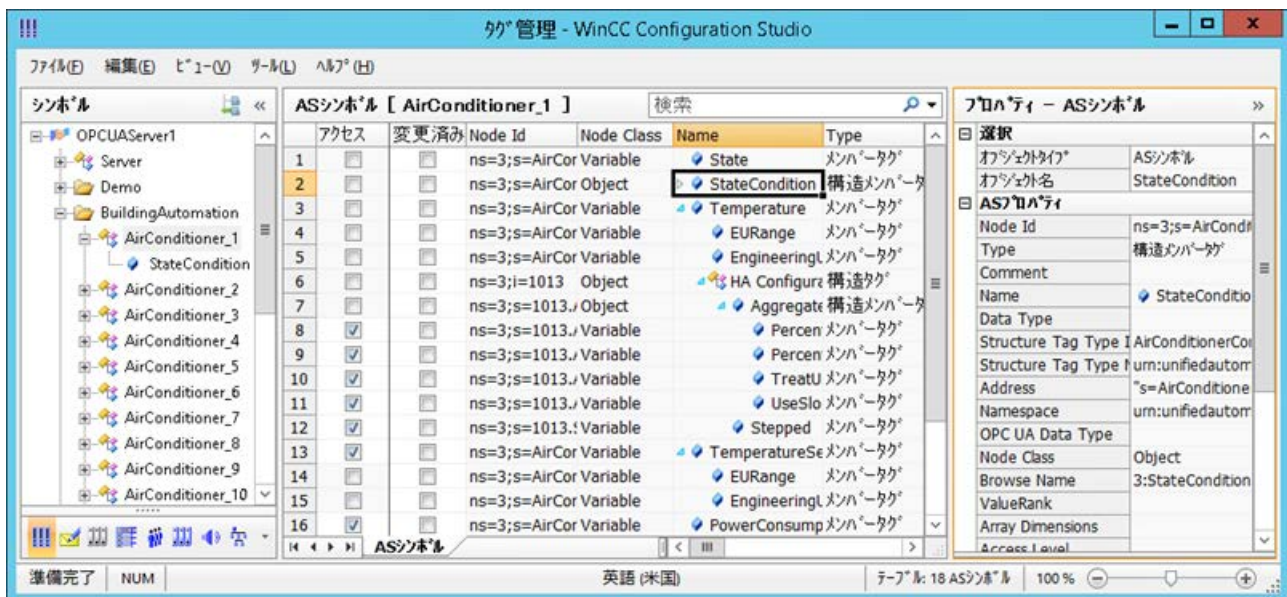
複雑なデータ構造がデータ領域に表示されます。

データ領域の下位レベルの階層を表示するには、シンボルをクリックして[名前]列で展開します：▶。

WinCC タグとのリンク付け

[AS シンボル]タブと[AS 構造]タブで、どの OPC UA ノードを WinCC タグとリンクするかを指定します。

OPC UA サーバー上の OPC UA ノードのプロパティのみを変更できます。



[AS 構造]タブ

ロードされているデータが構造も含んでいる場合、[AS 構造]タブが追加で表示されます。


タブは、接続名がナビゲーションエリアで選択されているときに表示されます。

タグ管理のデフォルト表示で構造割り付けも設定できます。

6.5 OPC UA WinCC チャンネル

通信チャンネル[OPC UA WinCC Channel]の接続をクリックして、[AS 構造]タブを表示します。

表示を次へ変更

次のボタンを使用して、タグ管理をデフォルト表示と[シンボル]の表示の間で切り替えます。

ボタンは、データレコードがロードされた後にのみ使用可能になります。

WinCC Configuration Studio が閉じられた後には、[AS シンボル]タブと[AS 構造]タブのある[シンボル]の表示が再度非表示になります。

デフォルト表示では、OPC UA ノードが再度ロードされないと、[AS 構造]タブは再度表示されません。

下記も参照

自動的に生成された構造タイプの使用方法 (ページ 313)

OPC UA ノードを WinCC タグとしてインポートする方法 (ページ 308)

OPC UA サーバーへの接続の設定方法 (ページ 300)

OPC UA タグ (ページ 303)

OPC UA ノードを WinCC タグとしてインポートする方法

はじめに

このセクションでは、OPC UA ノードを WinCC タグとして WinCC タグ管理にインポートする方法を説明します。

OPC UA WinCC チャンネルのタグは、WinCC Configuration Studio の[OPC UA Connections]チャンネルユニットの下に作成されます。

データタイプの割り付け

OPC UA ノードのサポートされるタイプは、対応する WinCC タグタイプに自動的に割り付けられます。「OPC UA タグ (ページ 303)」で割り付けテーブルを検索できます。

サポートされていない OPC UA ノードの場合、[アクセス]オプションは初期設定で無効にされています。これらのノードは、タグ管理にインポートされません。

どの WinCC データタイプとも適合しない OPC UA ノードもインポートするには、適合する WinCC データタイプを手動で割り付けます。

必要条件

- OPC UA サーバーへの接続が確立されていること。

手順

1. [OPC UA Connections]のナビゲーションエリアで設定済みの接続を選択します。
2. 接続のコンテキストメニューから[OPC サーバーを参照]エントリを選択します。
OPC UA サーバーの利用可能なデータがロードされ、[シンボル]の表示が開かれます。
ロードされているデータが、[AS シンボル]タブのテーブルエリアに表示されます。
ロードされているデータが構造も含んでいる場合、[AS 構造]タブが追加で表示されます。
3. [シンボル]の表示のナビゲーションエリアで接続を選択します。
4. [AS シンボル]グループの[特性-接続]セクションにある WinCC タグ名のオプションを選択します。
 - 名前の構成
 - セパレータ
 - 適切な場合は接頭語および接尾語
5. ナビゲーションエリアで必要なエントリを選択します。
それぞれ含まれているロード済みの OPC UA ノードが[AS シンボル]タブに表示されます。
6. 必要な AS シンボルに対して WinCC タグを作成するには、それぞれの場合で[アクセス]列を有効にします。
WinCC で選択されたオブジェクトのすべてのサポート対象の OPC UA ノードをインポートするには、[アクセス]列のコンテキストメニューから[すべて選択]を選択します。
7. サポートされない OPC UA ノードをインポートするには、[Data Type]列で適合する WinCC データタイプを選択します。
OPC UA ノードの[アクセス]フィールドを選択できるようになりました。
[アクセス]オプションを有効にすると、OPC UA ノードが、選択されたデータタイプで WinCC タグとしてインポートされます。

結果

WinCC タグ管理に新たに設定された WinCC タグが表示されます。

ただし、OPC UA サーバーのプロパティタグのみを変更します。

WinCC タグと OPC UA サーバーとの同期化

コントローラまたはファイルをロードした後、タグ管理は AS シンボルのプロパティをリンクされている WinCC タグと比較します。

6.5 OPC UA WinCC チャンネル

シンボルのプロパティが一致しないと、[AS シンボル]タブの[変更済み]フィールドが有効化されます。

それぞれのプロパティフィールドが赤で強調表示されます。フィールドのヒントには追加情報が記載されています。

WinCC タグの更新

OPC UA ノードの現在のプロパティを適用するには、[変更済み]フィールドを無効化します。
または、[アクセス]フィールドを無効化し、WinCC タグを再度作成するために再度有効化します。

下記も参照

自動的に生成された構造タイプの使用方法 (ページ 313)

[シンボル]の表示 (ページ 306)

6.5.3.8 WinCC における OPC UA タイプの使用

OPC UA タイプを WinCC 構造タイプとしてインポート

はじめに

このセクションでは、OPC UA サーバーのオブジェクトタイプまたはオブジェクトを WinCC タグ管理にインポートする方法を説明します。

目的は、OPC UA オブジェクトを WinCC に構造タグとして簡単に設定することです。

概要:基本手順

- 1. オブジェクトタイプの割り付け
- 2. オブジェクトの設定

インポートされたオブジェクトは WinCC 構造または構造タグとして作成され、次のようにマッピングされます。

OPC UA	WinCC
OPC UA オブジェクトタイプ	構造タイプ
Properties/OPC UA オブジェクトタイプのタグ	構造タイプ要素

OPC UA	WinCC
OPC UA オブジェクト	構造タグ
Properties/OPC UA オブジェクトのタグ	構造タグ要素

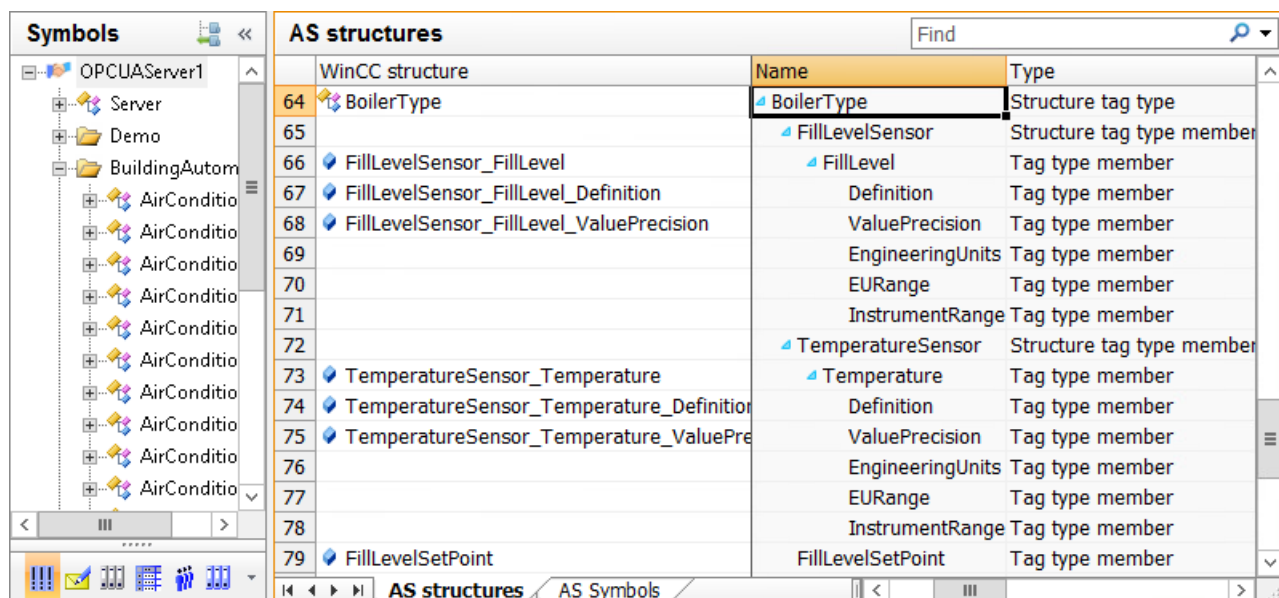
設定ステップ 1:オブジェクトタイプの割り付け

接続名が[シンボル]ビューのナビゲーションエリアで選択されている場合、[AS 構造]タブが表示されます。

[AS 構造]タブでは、OPC UA オブジェクトタイプを WinCC 構造タイプとリンクします。

OPC UA オブジェクトタイプのプロパティとタグが構造タイプ要素とリンクされます。

WinCC 構造タイプと構造タイプ要素を自動的に作成したり、すでに作成されている WinCC 構造タイプを割り付けたりすることができます。



自動割り付け

WinCC 構造タイプと構造タイプ要素を自動的に作成しています。

OPC UA 構造の名前のある構造タイプが WinCC タグ管理で作成されます。

WinCC でマッピングできる OPC UA オブジェクトタイプのプロパティとタグに対して、構造タイプ要素が作成されます。

6.5 OPC UA WinCC チャンネル

OPC UA オブジェクトタイプの階層は、「FillLevelSensor_FillLevel_Definition」のように、構造タイプ要素の名前を通じてマッピングされます。

注記
タグ名の最大長

WinCC タグの名前の最大長は 128 文字ですので注意してください。

構造タグ要素の場合、次の全体の長さがこの制限以内になるように指定します。

- 構造タグ名+ピリオド+構造タイプ要素名。
-

手動割り付け

構造タイプと構造タイプ要素を WinCC タグ管理で作成します。構造タイプ要素のデータタイプと対応するプロパティまたはタグの **DataType** が常に同じであるようにしてください。

読み込まれる OPC UA オブジェクトタイプを作成される WinCC 構造タイプとリンクします。

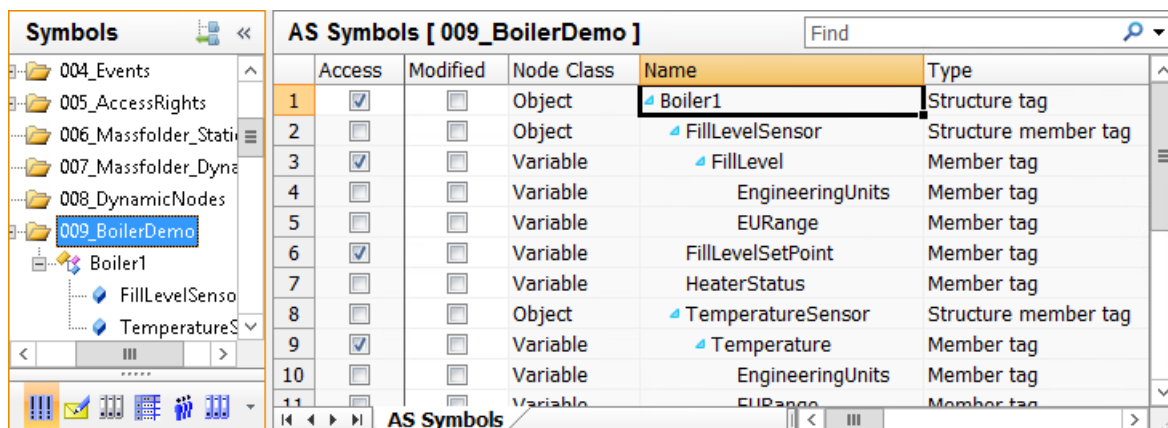
構造タイプ要素とプロパティまたはタグが同じ名前とデータタイプの場合、自動的に割り付けられます。

または、構造タイプ要素をプロパティとタグへ個別に割り付けます。

設定ステップ 2:オブジェクトの割り付け

[AS シンボル]タブで OPC UA オブジェクトを WinCC 構造タグとして設定します。

OPC UA オブジェクトへのアクセスを有効にすると、構造タグと構造タグ要素は自動的に作成されます。



自動的に生成された構造タイプの使用方法

はじめに

このアプローチでは、OPC UA オブジェクトのインポート時に、WinCC 構造タイプと構造タイプ要素を自動的に作成します。

必要条件

- OPC UA サーバーへの接続が確立されていること。



手順

1. OPC UA 接続のコンテキストメニューから[OPC サーバーを参照]エントリを選択します。
OPC UA サーバーの利用可能なデータがロードされます。
[AS 構造]タブが、OPC UA オブジェクトタイプと共に[シンボル]ビューに表示されます。タブは、接続名がナビゲーションエリアで選択されているときに表示されます。
オブジェクトタイプの下の要素を表示するには、[名前]フィールドのオブジェクトタイプ名の前にある矢印をクリックします。
2. OPC UA オブジェクトタイプを選択するには、行番号をクリックします。
複数選択できます。

6.5 OPC UA WinCC チャンネル

3. 行のコンテキストメニューで[構造を作成]エントリを選択します。
 - OPC UA オブジェクトタイプの名前のある構造タイプが WinCC タグ管理で作成されます。
 - マッピングできる OPC UA オブジェクトタイプのすべてのプロパティとタグに対して、1 つの構造タイプ要素がそれぞれに作成されます。
 - 階層は構造タイプ要素の名前を通じてマッピングされます。
WinCC タグの名前の最大長は 128 文字ですので注意してください。
必要に応じて、構造タグの作成前に、構造タイプ要素の名前を短くします。

Symbol		AS 構造		検索
OPCUAServer1		WinCC 構造	名前 ▲	タイプ
+	Server	1	▶ AccessPermissionObjectType	構造タグタイプ
+	Demo	2	▶ AcknowledgeableConditionType	構造タグタイプ
+	BuildingAutomation	3	▶ AddressSpaceFileType	構造タグタイプ
+	DemoUANode	4	▶ AggregateConfigurationType	構造タグタイプ
		5	▶ AggregateFunctionType	構造タグタイプ
		6	▶ AirConditionerControllerType	構造タグタイプ
		7	▶ Humidity	タグタイプメンバー
		8	▶ HumiditySetpoint	タグタイプメンバー
		9	▶ PowerConsumption	タグタイプメンバー
		10	▶ State	タグタイプメンバー
		11	▶ StateCondition	構造タグタイプメンバー
		12	▶ Temperature	タグタイプメンバー
		13	▶ EngineeringUnits	タグタイプメンバー
		14	▶ EURange	タグタイプメンバー
		15	▶ HA Configuration	構造タグタイプメンバー
		16	▶ AggregateConfiguration	構造タグタイプメンバー
		17	▶ Stepped	タグタイプメンバー
		18	▶ TemperatureSetPoint	タグタイプメンバー

4. [タグ管理]ビューで構造タイプと構造タイプ要素を編集するには、次のシンボル  をクリックします。
必要に応じて、[構造タグ]で構造タイプまたは構造タイプ要素の名前を変更します。
必要な場合に、必要のない構造タイプ要素を削除します。
変更点は OPC UA 接続の構造タグに適用されます。
5. [シンボル]ビューをもう一度表示するには、OPC UA 接続と次のシンボル  をクリックします。
6. OPC UA オブジェクトを表示するには、希望のノードをナビゲーションで選択します。
7. [AS シンボル]タブで OPC UA オブジェクトの[アクセス]フィールドを有効にします。

Symbol		AS シンボル [BuildingAutomation]				
OPCUAServer1		アクセス	変更済み	Node Class	名前	タイプ
+	Server	1	<input type="checkbox"/>	Variable	ControllerConfigurat	タグ
+	Demo	2	<input checked="" type="checkbox"/>	Object	▶ AirConditioner_1	構造タグ
+	BuildingAutomation	3	<input checked="" type="checkbox"/>	Object	▶ AirConditioner_2	構造タグ
+	AirConditioner_1	4	<input checked="" type="checkbox"/>	Object	▶ AirConditioner_3	構造タグ
+	AirConditioner_2	5	<input type="checkbox"/>	Object	▶ AirConditioner_4	構造タグ

構造タグが OPC UA オブジェクトのリンクされた構造タイプで作成されます。
OPC UA オブジェクトのプロパティとタグが構造タイプ要素とマッピングされます。

下記も参照

OPC UA ノードを WinCC タグとしてインポートする方法 (ページ 308)

[シンボル]の表示 (ページ 306)

手動で生成された構造タイプの使用方法

はじめに

このアプローチでは、OPC UA オブジェクトのインポートのために WinCC タグ管理で作成された構造タイプと構造タイプ要素を使用します。

必要条件

- OPC UA サーバーへの接続が確立されていること。
- WinCC タグ管理で構造タイプが作成されました。
- 次のプロパティの構造タイプ要素が、構造タイプで設定されます。
 - 外部:有効
 - データタイプ:OPC UA オブジェクトタイプの対応するプロパティまたはタグの DataType

手順

1. OPC UA 接続のコンテキストメニューから[OPC サーバーを参照]エントリを選択します。OPC UA サーバーの利用可能なデータがロードされます。
[AS 構造]タブが、OPC UA オブジェクトタイプと共に[シンボル]ビューに表示されます。タブは、接続名がナビゲーションエリアで選択されているときに表示されます。
オブジェクトタイプの下の要素を表示するには、[名前]フィールドのオブジェクトタイプ名の前にある矢印をクリックします。
2. [WinCC 構造]フィールドで、OPC UA オブジェクトタイプに割り付ける作成済みの構造タイプを選択します。
オブジェクトタイプのプロパティまたはタグと同じ名前とデータタイプを保持する構造タイプ要素が、自動的に割り付けられます。

6.5 OPC UA WinCC チャンネル

3. 別の名前の構造タイプ要素をプロパティまたはタグに割り付けるには、[WinCC 構造]フィールドで要素を選択します。
リストには、まだ割り付けられていない状態で、プロパティまたはタグと同じデータタイプを保持しているすべての構造タイプ要素が含まれています。

Symbol

OPCUAServer1

Server

Demo

BuildingAutomation

DemoUANode

AS 構造		
WinCC構造	名前	タイプ
5	AggregateFunctionType	構造タグタイプ
6	AirConditionerControllerType	構造タグタイプ
7	Humidity	タグタイプメンバ
8	HumiditySetpoint	タグタイプメンバ
9	PowerConsumption	タグタイプメンバ
10	State	タグタイプメンバ
11	StateCondition	構造タグタイプメンバ
12	Temperature	タグタイプメンバ
13	EngineeringUnits	タグタイプメンバ
14	EURange	タグタイプメンバ
15	HA Configuration	構造タグタイプメンバ
16	AggregateConfiguration	構造タグタイプメンバ
17	PercentDataBad	タグタイプメンバ
18	PercentDataGood	タグタイプメンバ
19	TreatUncertainAsBad	タグタイプメンバ
20	UseSlopedExtrapolation	タグタイプメンバ
21	Stepped	タグタイプメンバ
22	TemperatureSetPoint	タグタイプメンバ

4. OPC UA オブジェクトを表示するには、希望のノードをナビゲーションで選択します。
5. [AS シンボル]タブで OPC UA オブジェクトの[アクセス]フィールドを有効にします。

Symbol

OPCUAServer1

Server

Demo

BuildingAutomation

AirConditioner_1

AirConditioner_2

ASシンボル [BuildingAutomation]					
	アクセス	変更済み	Node Class	名前	タイプ
1	<input type="checkbox"/>	<input type="checkbox"/>	Variable	ControllerConfigurat	タグ
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	AirConditioner_1	構造タグ
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	AirConditioner_2	構造タグ
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Object	AirConditioner_3	構造タグ
5	<input type="checkbox"/>	<input type="checkbox"/>	Object	AirConditioner_4	構造タグ

- 構造タグが OPC UA オブジェクトのリンクされた構造タイプで作成されます。
OPC UA オブジェクトのプロパティとタグが構造タイプ要素とマッピングされます。
6. [タグ管理]の表示で構造タグを編集するには、次のシンボルをクリックします。
必要に応じて、[構造タグ]で構造タイプ、構造タイプ要素、構造タグの名前を変更します。
変更点は OPC UA 接続の構造タグに適用されます。

6.5.3.9 WinCC における OPC UA アラームの使用

[監視オブジェクト]の表示

Event Notifier とアラーム

接続の設定が正常に行われた後には、OPC UA サーバーの Event Notifier にアクセスできます。

Event Notifier が、出力を WinCC メッセージとして保持できるアラームまたはイベントをトリガします。

WinCC メッセージを OPC UA アラームに対して設定するには、アラームロギングの Event Notifier を[監視オブジェクト]の表示にロードします。

通知
<p>ローカルの WinCC OPC UA サーバーは許可されません。</p> <p>この機能はローカルの WinCC OPC UA サーバーで有効ではありません。</p> <p>WinCC メッセージをローカルの WinCC OPC UA サーバーの Event Notifier とリンクすると、設定が不適切な場合に、アラームロギングの無限ループが発生することがあります。</p>

WinCC アラームロギングの OPC Event Notifier

[OPC UA WinCC Channel]通信チャンネルで接続を作成すると、すぐに[OPC メッセージ]エントリが[アラームロギング]エディタで作成されます。

作成済みの接続がエントリの下にリストされます。

接続済みの OPC UA サーバーの Event Notifier を[監視オブジェクト]の表示にロードできます。

WinCC メッセージのトリガ

Event Notifier が WinCC プロジェクトでトリガされることを通知するアラームを表示するには、Event Notifier を WinCC メッセージとリンクします。

次いで、WinCC メッセージが、Event Notifier によってトリガされるすべてのアラーム、ならびにその階層的に従属しているノードによりトリガされます。これは、メッセージの数が激増する可能性があることを意味しています。

どの OPC UA アラームが WinCC メッセージをトリガするかを特定するには、フィルタを使用できます。

これにより、トリガされるメッセージの数が減り、関連のイベントに対してのみアラームを受信できます。

注記

システムパフォーマンス: **Event Notifier 「Server」** を避けてください。

WinCC メッセージを上位レベルの Event Notifier 「サーバーオブジェクト」とリンクすると、多数のメッセージが生成されることがあります。

OPC UA イベントの数を減らすフィルタが使用されている場合でも、このような手順はパフォーマンスに悪影響を及ぼすことがあります。

WinCC メッセージの割り付け

Event Notifier と WinCC メッセージは、メッセージ番号を通じてリンクします。

メッセージ番号がアラームロギングですでに作成されている場合、このメッセージはリンクされます。それ以外の場合は、指定の番号のあるメッセージがアラームロギングで作成されます。

同じメッセージを複数の Event Notifier とリンクできます。

しかし、1 つの WinCC メッセージは常に 1 つの OPC UA 接続に対してしか使用できません。複数の OPC UA 接続を作成していると、各接続は別の WinCC メッセージを使用します。

複数の WinCC メッセージの割り付け

1 つの Event Notifier を複数の WinCC メッセージにリンクできます。

アラームがトリガされると、メッセージとフィルタは上から下まで処理されます。メッセージ番号は、WinCC メッセージがトリガされるシーケンスには影響を与えません。

メッセージの順番を変更するには、フィルタを折りたたんで、行のショートカットメニューで[上へ移動]または[下へ移動]を選択します。

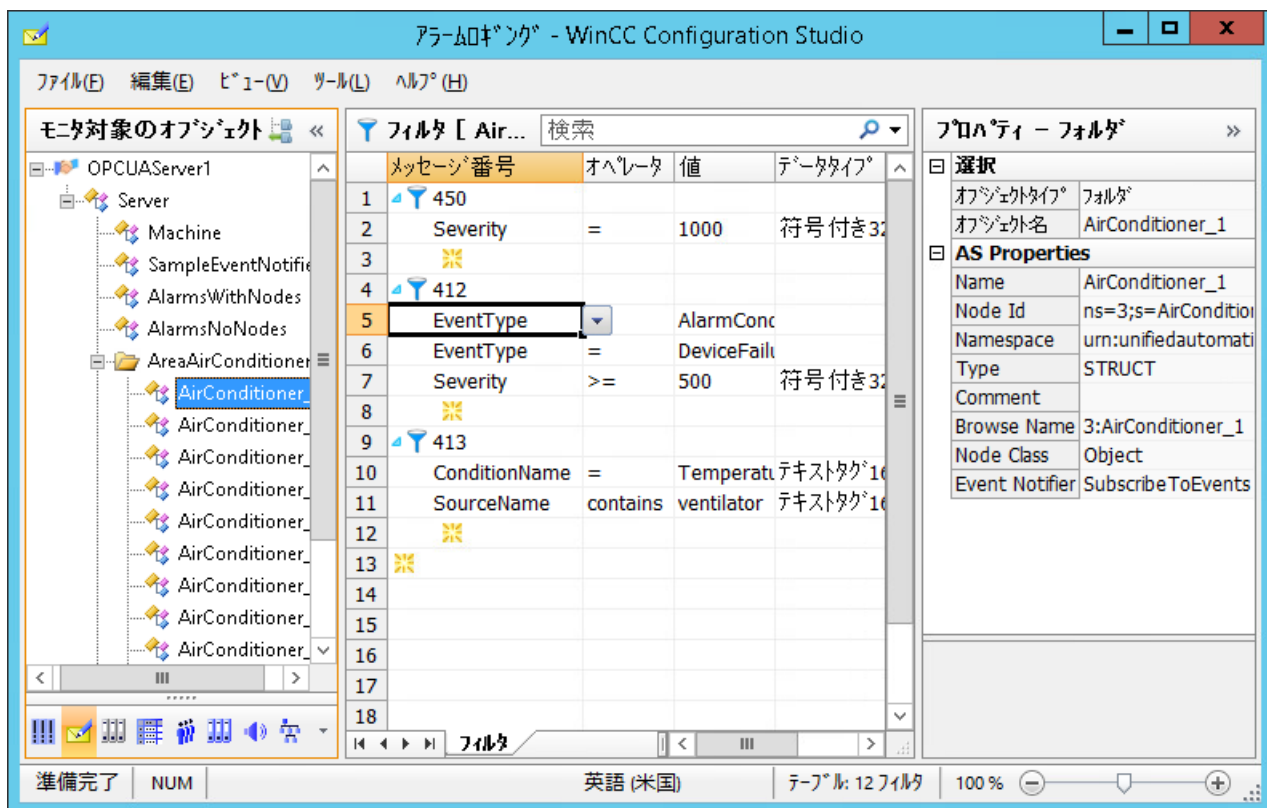
シンボルの表示

構造ツリーのデータの表示は、OPC UA サーバーの階層に対応しています。

[フィルタ]タブで、どの WinCC メッセージを Event Notifier とリンクするかを指定します。

各メッセージに対して、トリガされるアラームのために 1 つ以上のフィルタを定義できます。

選択された Event Notifier のプロパティは、[特性-フォルダ]エリアに表示されます。OPC UA サーバーのプロパティしか変更できません。



表示を次へ変更

次のボタンを使用して、アラームロギングでデフォルト表示と[監視オブジェクト]の表示の間を切り替えます。

ボタンは、データレコードがロードされた後にのみ使用可能になります。

WinCC Configuration Studio が閉じられた後には、[監視オブジェクト]の表示が再度非表示になります。

デフォルト表示の[アラームロギング]では、[監視オブジェクト]タブは表示されたままです。

アラームロギング:[割り付け]タブ

[OPC メッセージ]をクリックして、アラームロギングのデフォルト表示の[割り付け]タブを表示します。

ここでは、OPC UA アラームの属性に対して割り付けを設定します。

設定済みのルールはすべての OPC UA 接続に適用されます。

割り付けルールの設定

トリガされた OPC UA アラームが WinCC メッセージにリンクされていると、属性がプロセス値ブロック 1～10 に適用されます。

この目的のために、プロセス値ブロックをアラームロギングの[メッセージブロック]で[使用済み]として設定します。

プロセス値ブロック 1 には常に OPC UA アラームのメッセージテキストが含まれています。

プロセス値ブロック 2～10 のそれぞれに対して、ドロップダウンリストボックスで希望する属性を選択します。

デフォルトルール[Default]を変更することはできません。

アラームロギング:[監視オブジェクト]タブ

アラームロギングのデフォルト表示で[監視オブジェクト]タブを表示するには、[OPC メッセージ]の接続をクリックします。

WinCC メッセージとそのフィルタにリンクされている Event Notifier が表示されます。

割り付けのルールの割り付け

このタブで、Event Notifier の割り付けルールを選択します。[Default]ルールがデフォルトで割り付けられます。

同じ割り付けルールが Event Notifier の全アラームとその階層上で従属しているノードに使用されます。

デフォルト表示でフィルタを編集

[監視オブジェクト]タブでフィルタを編集することもできます。

変更されたフィルタは、[フィルタ]タブの[監視オブジェクト]の表示で使用されます。

しかし、デフォルト表示では、フィルタ基準が一貫性あるか、あるいは正確に入力されているかはチェックされません。

変更されたフィルタをテストして、必要に応じて[監視オブジェクト]の表示でフィルタを修正します。

下記も参照

Event Notifier を WinCC メッセージとしてインポートする方法 (ページ 325)

OPC UA アラームのフィルタ (ページ 321)

OPC UA アラームのフィルタ

OPC UA アラームのフィルタリング

Event Notifier とリンクするそれぞれの WinCC メッセージに対して 1 つ以上のフィルタを指定できます。

Event Notifier は、複数のアラームまたはイベントをトリガし、通常は WinCC メッセージに対して一部のアラームまたはイベントのみが必要です。

フィルタにより、ユーザーはトリガされたメッセージを関連イベントに絞り込むことができます。

注記

システムパフォーマンス: Event Notifier 「Server」を避けてください。

WinCC メッセージを上位レベルの Event Notifier 「サーバーオブジェクト」とリンクすると、多数のメッセージが生成されることがあります。

OPC UA イベントの数を減らすフィルタが使用されている場合でも、このような手順はパフォーマンスに悪影響を及ぼすことがあります。

WinCC メッセージ: 固有の割り付け

OPC UA アラームまたはイベント、ならびに WinCC メッセージをできるだけ明確に割り付ける特定のフィルタを定義してください。

WinCC メッセージは、アラームまたはイベントのプロパティとマッピングされるように設定する必要があります。例えば、確認のセオリーとメッセージソース(Source)というようにです。

フィルタの設定

フィルタを使用して、どの OPC UA アラームまたは OPC UA イベントが WinCC メッセージをトリガするかを定義することができます。

1 つのメッセージを複数の Event Notifier にリンクできますが、別々のアラームでフィルタします。

[フィルタ] タブの [監視オブジェクト] の表示でフィルタを設定します。

フィルタを作成するには、メッセージ番号の前にある矢印をクリックします。表示される行で、フィルタ基準、演算子、値を選択します。

フィルタ基準のデータタイプは自動的に追加され、変更することはできません。

オンライン設定

ランタイムでフィルタを変更すると、すぐに適用されます。

フィルタ基準と演算子

フィルタ基準により、Event Notifier のアラームのどの条件を満たし、リンクされたメッセージをトリガできるようにする必要があるかを、ユーザーは特定できます。

演算子は選択されているフィルタ基準により異なります。

フィルタ基準	演算子		説明
EventType	=		タイプのドロップダウンリスト [BaseEventType]値: <ul style="list-style-type: none"> すべての OPC UA アラームまたは OPC UA イベントが返されます(フィルタなし)。
ConditionName SourceName	= contains		フリーテキスト入力 大文字小文字を考慮する必要があります。 演算子「contains」: <ul style="list-style-type: none"> 入力されたテキストが含まれています。プレースホルダは使用されません。
Severity	=	～に等しい	数値入力 値の範囲: <ul style="list-style-type: none"> 1～1000 WinCC メッセージでのマッピング: <ul style="list-style-type: none"> 優先度 0 = Severity 1 優先度 1～15 = 0 と 1000 の間の線形補間 優先度 16 = Severity 1000 例「between」: <ul style="list-style-type: none"> 100、200 100～200 の Severity に相当 (それぞれ指定された値を含む)
	!=	～に等しくない	
	>	より大きい	
	<	未満	
	>=	～以上	
	<=	～以下	
	between	範囲(開始、終了)	

フィルタ基準の組み合わせ

フィルタのフィルタ基準を組み合わせたり、同じフィルタ基準を複数回使用したりできます。

- 別のフィルタ基準は「AND」でリンクされます。
- 同じフィルタ基準は「OR」でリンクされます。
- フィルタの階層はありません。入力されるフィルタ基準の順番はフィルタの適用に影響を与えません。

例:[アラームロギング]の表示

アラームロギングのデフォルト表示で、WinCC メッセージとリンクされている Event Notifier が各接続に対してリストされます。

Event Notifier のノードで、メッセージ番号が表示されます。それぞれの場合で、下にフィルタが表示されます。

この表示を使用して、複数の Event Notifier のフィルタ間で同期をとることもできます。

この例では、[OPCUAServer1]接続の設定済みの Event Notifier を確認できます。

6.5 OPC UA WinCC チャンネル

アラームロギング		モニタ対象のオブジェクト [OPCUAServer1]			
		メッセージ番号	オペレータ	値	データタイプ
<div> <div>メッセージ</div> <div>メッセージアロタク</div> <div>メッセージグループ</div> <div>システムメッセージ</div> <div>限界モニタリング</div> <div>ASメッセージ</div> <div>OPCメッセージ</div> <div>OPCUAServer1</div> </div>		1	AirConditioner_1		
		2	450		
		3	Severity	= 1000	符号付き32ビット
		4			
		5	412		
		6	EventType	= AlarmConditionType	
		7	EventType	= DeviceFailureEventType	
		8	Severity	>= 500	符号付き32ビット
		9			
		10	413		
		11	SourceName	contains ventilator	テキストタグ16ビット
		12	ConditionName	= TemperatureHigh	テキストタグ16ビット
		13	SourceName	contains cool	テキストタグ16ビット
		14	EventType	= AlarmConditionType	
		15			
		16			
		17	Furnace_1		
		18	450		
		19	412		
		20			
		21	Furnace_2		
		22	450		
		23	Severity	= 1000	符号付き32ビット
		24			
		25	412		
		26	EventType	= AlarmConditionType	
		27	EventType	= DeviceFailureEventType	
		28			
		29	621		
		30	EventType	= AcknowledgeableCondi	
		31	ConditionName	contains heating	テキストタグ16ビット
		32	SourceName	= torch	テキストタグ16ビット
		33	SourceName	= flame	テキストタグ16ビット

フィルタ例

- EventType = AlarmConditionType
- EventType = DeviceFailureEventType
- Severity >= 500

この例は、次の条件に相当します。

- (EventType=AlarmConditionType OR EventType=DeviceFailureEventType) AND Severity>=500

下記も参照

[監視オブジェクト]の表示 (ページ 317)

Event Notifier を WinCC メッセージとしてインポートする方法 (ページ 325)

Event Notifier を WinCC メッセージとしてインポートする方法

はじめに

このセクションでは、OPC UA サーバーの Event Notifier を WinCC メッセージと接続する方法を説明します。

Event Notifier の OPC UA アラームは、ランタイムの WinCC アラームロギングでメッセージをトリガします。これらは、WinCC AlarmControl にアーカイブされ、表示されます。

概要:基本手順

1. WinCC アラームロギングに OPC UA Event Notifier をロードします。
2. Event Notifier を WinCC メッセージ番号とリンクします。
3. オプション:Event Notifier のトリガされたアラームに対するフィルタを特定します。
4. オプション:割り付けルールを定義
5. それぞれの Event Notifier に対して割り付けルールを指定します。
6. オプション:WinCC メッセージのプロパティ、表示、アーカイブを設定します。

注記

システムパフォーマンス:Event Notifier 「Server」を避けてください。

WinCC メッセージを上位レベルの Event Notifier 「サーバーオブジェクト」とリンクすると、多数のメッセージが生成されることがあります。

OPC UA イベントの数を減らすフィルタが使用されている場合でも、このような手順はパフォーマンスに悪影響を及ぼすことがあります。

WinCC メッセージ:固有の割り付け

OPC UA のアラームまたはイベント、ならびに WinCC メッセージを割り付ける特定のフィルタを、できるだけ明確に定義してください。

WinCC メッセージは、アラームまたはイベントのプロパティとマッピングされるように設定する必要があります。例えば、確認のポリシーとメッセージソース(Source)というようにです。

通知

ローカルの WinCC OPC UA サーバーは許可されません。
この機能はローカルの WinCC OPC UA サーバーで有効ではありません。
WinCC メッセージをローカルの WinCC OPC UA サーバーの Event Notifier とリンクすると、設定が不適切な場合に、アラームロギングの無限ループが発生することがあります。

必要条件

- OPC UA サーバーへの接続が確立されていること。
- プロセス値ブロックが使用のために有効化されます。

手順

1. [OPC メッセージ]の[アラームロギング]エディタで設定済みの接続を選択します。
2. OPC UA 接続のコンテキストメニューから[OPC サーバーを参照]エントリを選択します。
OPC UA サーバーの利用可能なデータがロードされます。[監視オブジェクト]の表示が開かれます。
ナビゲーションエリアで、Event Notifier が接続名の下に表示されます。
3. ナビゲーションエリアで Event Notifier を選択します。
4. データエリアで 1 つ以上の WinCC メッセージ番号を入力します。
すべての従属する Event Notifier もリンクされたメッセージをトリガするため、階層の最高レベル[Server]のリンクは避けてください。トリガされるメッセージの多数のパフォーマンスに悪影響を及ぼすことがあります。

モニタ対象のオブジェクト

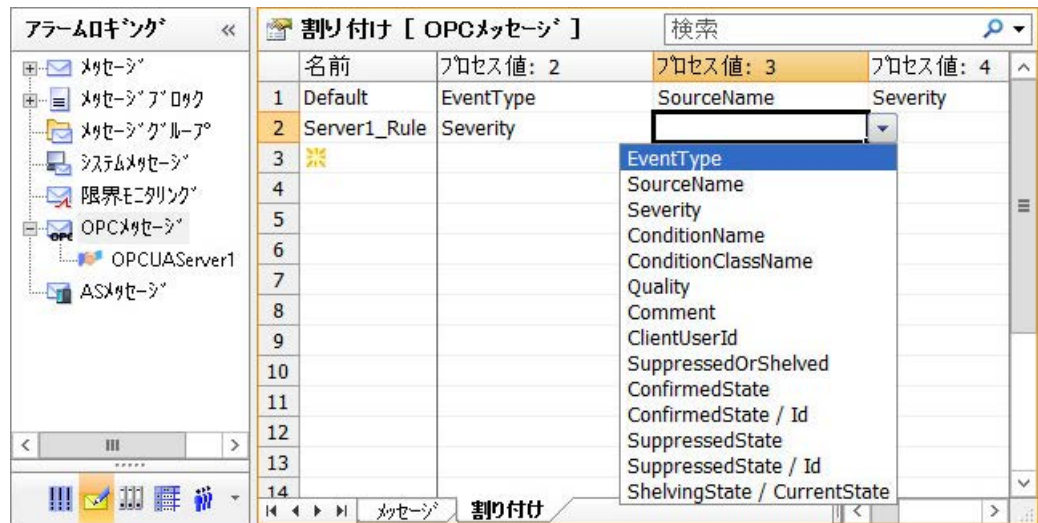
OPCUAServer1
Server
Machine
SampleEventNotifier
AlarmsWithNodes
AlarmsNoNodes
AreaAirConditioner
AirConditioner_1
AirConditioner_2
AirConditioner_3
AirConditioner_4
AirConditioner_5
AirConditioner_6
AirConditioner_7
AirConditioner_8

フィルタ [AirConditioner_1]

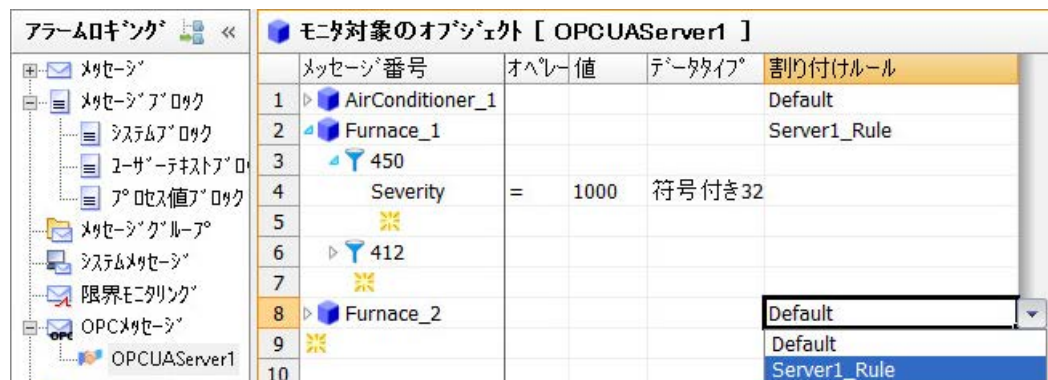
	メッセージ番号	オペレータ	値	データタイプ
1	450			
2	Severity	=	1000	符号付き32
3				
4	412			
5	EventType	=	AlarmConditionType	
6	EventType	=	DeviceFailureEventType	
7				
8	413			
9	SourceName	contains	ventilator	テキストタグ16
10	ConditionName	=	TemperatureHigh	テキストタグ16
11				
12	ConditionName			
13	EventType			
14	Severity			
15	SourceName			

5. フィルタを定義するには、メッセージ番号の前にある矢印をクリックします。
表示される行で、フィルタ基準、演算子、値を選択します。
フィルタ基準のデータタイプは自動的に追加され、変更することはできません。

6. [アラームロギング]の表示でメッセージを編集するには、次のシンボルをクリックします。
7. 割り付けルールを表示するには、[OPC メッセージ]をクリックし、[割り付け]タブを選択します。
8. [デフォルト]ルールの[名前]フィールドに新しいルールの名前を入力します。



9. それぞれの場合で、Event Notifier の希望の属性を、プロセス値フィールドのドロップダウンリストから選択します。
属性は対応するプロセス値ブロックにリンクされます。
10. [OPC メッセージ]の接続名をクリックします。
OPC UA 接続のメッセージとフィルタが表示されます。
11. Event Notifier に対してそれぞれの割り付けのルールを選択します。
[デフォルト]ルールがデフォルトでリンクされます。



6.5 OPC UA WinCC チャンネル

12. WinCC メッセージを表示するには、[OPC メッセージ]をクリックします。
メッセージとそのプロパティが、[メッセージ]タブに表示されます。
13. メッセージクラス、メッセージのタイプ、アーカイブなど、WinCC メッセージのプロパティを設定します。
ユーザーテキストブロックのプロセス値ブロックの内容にアクセスするには、「@1%」フォーマットを使用します。
プロセス値ブロックの詳細については、WinCC 情報システムの「WinCC での作業」>「メッセージシステムのセットアップ」>「メッセージシステムの設定」>「メッセージでの作業」を参照してください。
 - 「メッセージのテキストの指定方法」
 - 「ユーザーテキストブロックでのプロセス値の挿入方法」

下記も参照

[監視オブジェクト]の表示 (ページ 317)

OPC UA アラームのフィルタ (ページ 321)

6.5.4 OPC UA WinCC チャンネルの OPC UA 配列

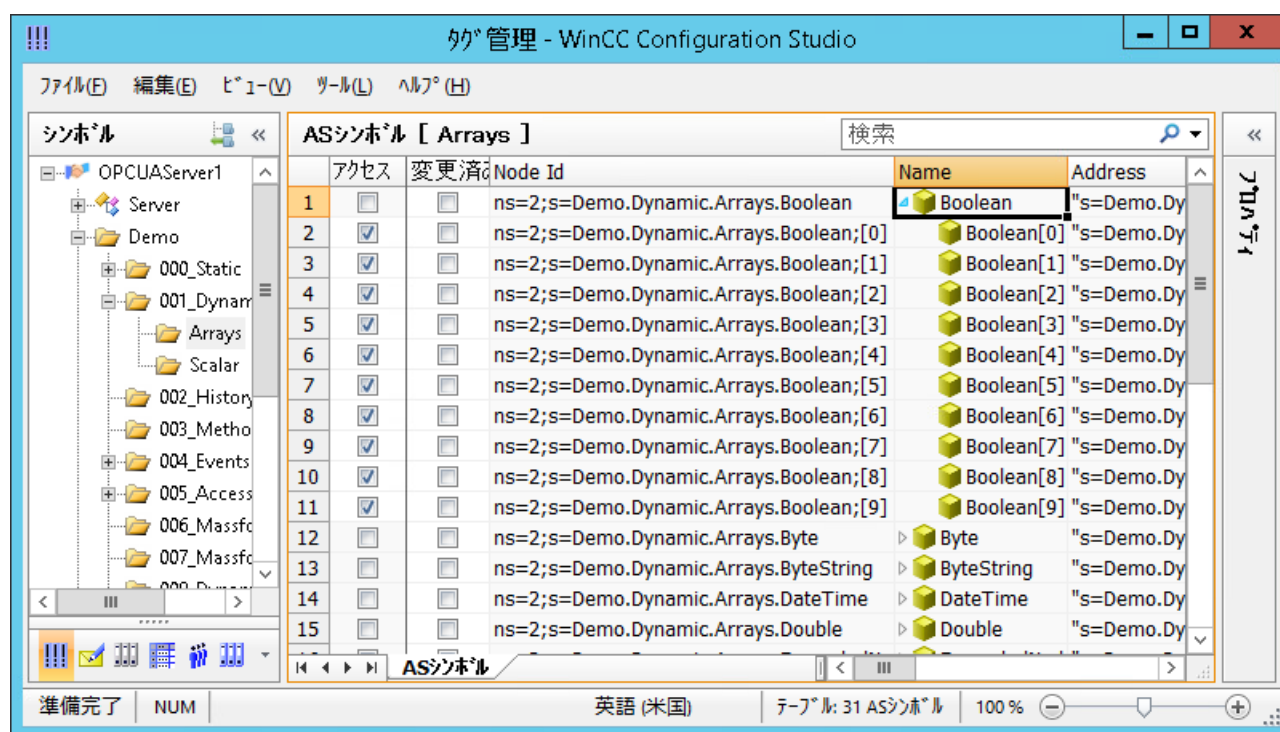
6.5.4.1 WinCC の配列

WinCC は OPC UA 配列の設定をサポートします。この文脈では、WinCC タグは単一の配列エレメントにのみ対応できます。つまり、完全な OPC UA 配列値を 1 つのみの WinCC タグにマッピングできません。

WinCC Configuration Studio の配列の表示

WinCC Configuration Studio の「AS シンボル」作業エリアでデータエントリを展開するときに配列が表示されます。配列タイプおよび寸法が OPC UA DataType 列に表示されます。

[名前]列で対応する配列シンボルをクリックして、個別の配列エレメントへのアクセスを取得します。



WinCC タグとして配列エレメントをマッピングするため、[アクセス]列でチェックボックスを有効化します。

データタイプ

WinCC は任意の寸法の、次の OPC UA 配列データタイプをサポートします。

- Boolean
- SByte
- Byte
- Int16
- UInt16
- Int32
- UInt32
- Float
- Double
- String

- DateTime
- ByteString

注記

サーバーがサポートされない配列データタイプを出力する場合、WinCC Configuration Studio でタグを設定できません。

IndexRange を使用しない配列エントリの書き込み

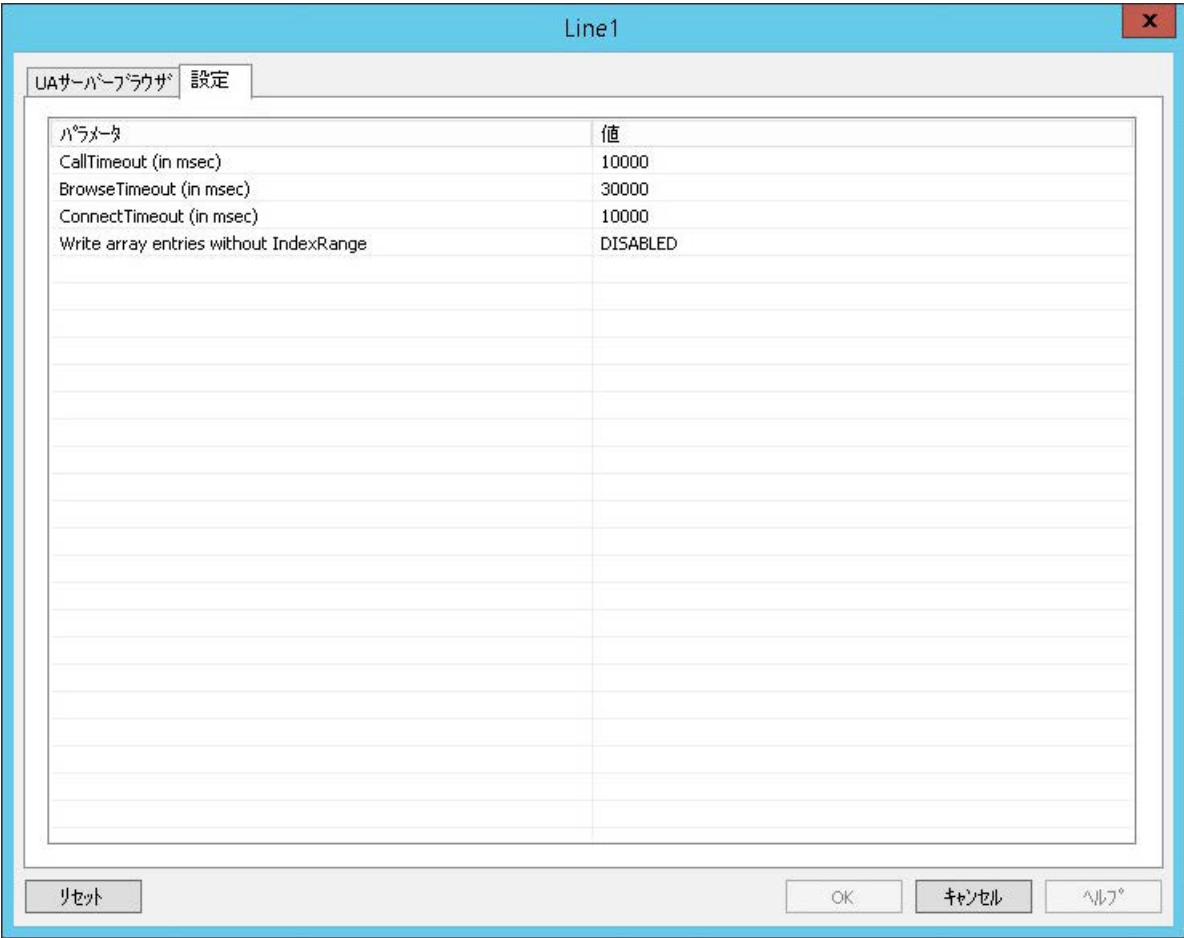
OPC UA 配列を読み取りおよび書き込みするとき、使用される OPC-UA 基準で指定される IndexRange ファンクションによって、OPC UA クライアントは配列全体またはその一部のみにアクセスすることができます。

たとえば、クライアントは、インデックス範囲として、「3,10」を指定することで、配列のエレメント 3～10 のみを読み取りまたは書き込みすることができます。または、IndexRange として「5」を指定することで、配列エレメント 5 のみを読み取りおよび書き込みします。

OPC UA 仕様に一致する OPC UA サーバーは、配列を読み取るときに IndexRange をサポートする必要があります。逆に、IndexRange を使用した配列の書き込みはサポートできません。

WinCC は、常に、IndexRange を使用して配列エレメントの読み取りまたは書き込みを試みます。配列エレメントを IndexRange を使用して読み取れない場合、WinCC は配列全体の読み取りおよび WinCC で設定された配列エレメントの抽出を試みます。配列エレメントを IndexRange を使用して書き込めない場合、WinCC はフォールバック戦略[IndexRange を使用しない配列エントリの書き込み]を使用します。これはデータの不一致を引き起こすことがあるため、デフォルトでは無効にされています。

[接続パラメータ]ダイアログの[設定]タブで[IndexRange を使用しない配列エントリの書き込み]機能を有効にします。



通知
<p>データの不一致</p> <p>この機能を有効にすると、データの不一致を引き起こすことがあります。</p>

制限事項

- データの一貫性**
- ランタイムでは、WinCC で設定されている OPC UA 配列値のデータの一貫性を保証することはできません。
- 配列データは、特に、次の場合に一貫性を失うことがあります。
- 異なる読み取りサイクルのいくつかの配列エレメントを読み取るとき。
 - 配列全体を読み取りおよび更新するとき ([IndexRange を使用しない配列エントリの書き込み]セクションを参照)。

パフォーマンス

WinCC を配列値全体を読み取り、設定された配列エレメントを抽出する必要があるため、OPC サーバーが **IndexRange** をサポートしない場合、配列のいくつかのエレメントを読み取るときにパフォーマンスの問題が発生することがあります。

ダイナミック配列

OPC-UA 配列の長さは、動的にすることができ、ランタイム中に変更できます。たとえば、配列値が期待されたよりも短い場合、WinCC タグ値の品質が低下することがあります。

互換性

一部の OPC UA サーバーが OPC UA 基準との完全な互換性がないため、結果的にタグ管理が配列エレメントを表示できなくなることがあります。この場合、配列タグを手動で作成して、配列アドレスを変更する必要があります。

そのため、それらのサーバーの配列を設定できるようにするには、サーバー固有の知識が必要です。

手動設定に関する追加情報は、「OPC UA 配列を手動で作成および設定する方法 (ページ 332)」セクションで参照できます。

6.5.4.2 OPC UA 配列を手動で作成および設定する方法

OPC UA サーバーが基準に従って配列を表示しない場合、WinCC が OPC UA 配列の配列エレメントを表示できなくなることがあります。この場合、WinCC タグ管理で、配列エレメントを手動で作成および設定する必要があります。この目的で、サーバー固有の知識が必要です。

必要条件

- OPC UA サーバーへの接続が確立されていること。
- タグ管理が開いていること。

手順

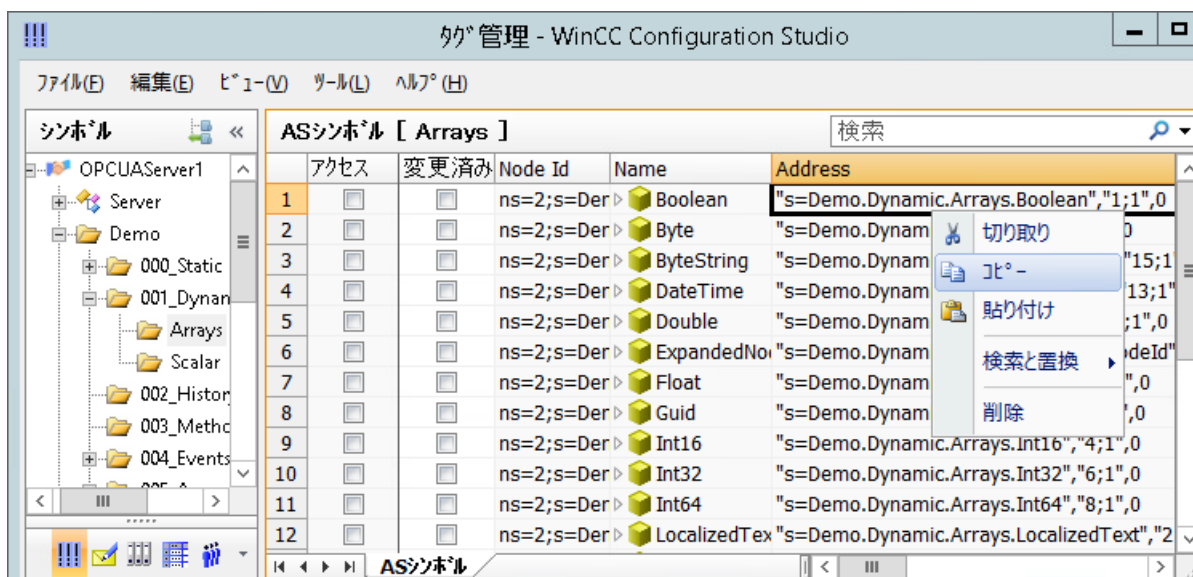
1. [タグ]作業エリアのショートカットメニューを介して既存のタグをコピーおよび挿入することで新しいタグを作成します。

注記

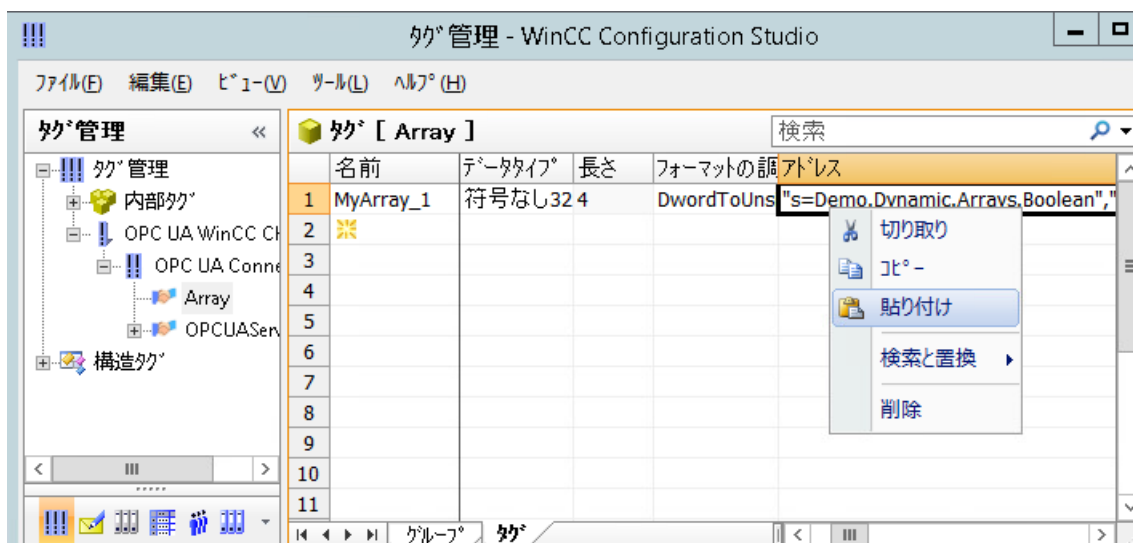
タグのコピーと挿入

すべてのプロパティと一緒にタグをコピーするには、タグ名のみでなく、行全体にマークを付ける必要があります。この目的で、先頭の番号をクリックします。


2. 該当する場合、新たに作成されたタグの名前を変更します。
3. [AS シンボル]作業エリアの対応する配列値のアドレスをクリップボードにコピーします。

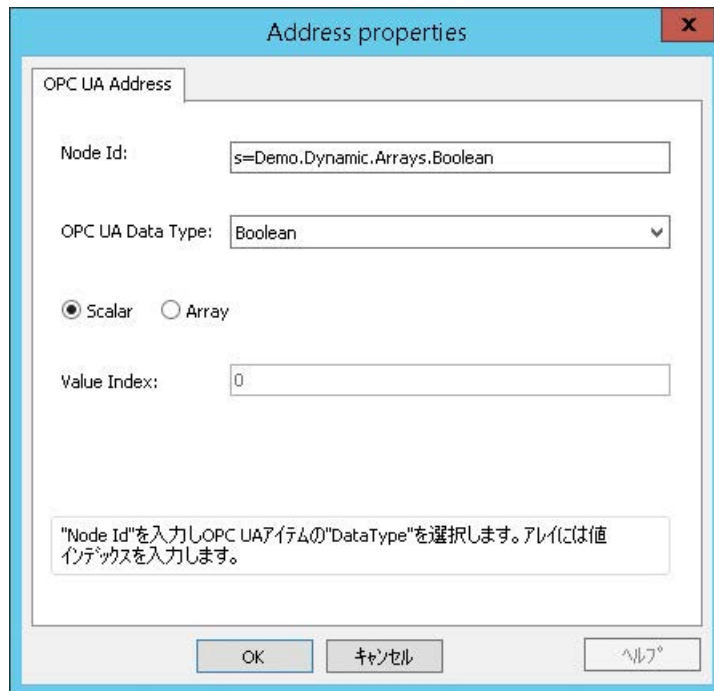


4. [AS シンボル]作業エリアの対応する配列値のアドレスをコピーされたタグの[アドレス]列に挿入します。



6.5 OPC UA WinCC チャンネル

5.  ボタンを使用して、タグのプロパティで[アドレスプロパティ]ダイアログを開きます。



6. 必要な設定を実行し、[OK]で確定します。

6.5.5 エラー処理

6.5.5.1 OPC DA 通信に障害が発生した場合のエラー処理

はじめに

通信テストの手順は、WinCC の使用方法に関係ありません。

チャンネル診断の詳細は、[通信] > [通信診断]を参照してください。

OPC UA サーバーとして使用する WinCC

WinCC OPC UA クライアントでチャンネル診断を使用して、接続が WinCC OPC UA サーバーへ確立できるか確認します。

OPC UA クライアントとして使用する WinCC

WinCC OPC UA クライアントでチャンネル診断を使用して、接続が WinCC OPC UA サーバーへ確立できるか確認します。

6.6 PROFIBUS DP

6.6.1 WinCC チャンネル"PROFIBUS DP"

内容

[PROFIBUS DP]チャンネルは PROFIBUS DP-Master としての WinCC ステーションと、ET200 などの対応する周辺アセンブリの間の通信に使用されます。

通信には PROFIBUS DP プロトコルが使用されます。

この章では、以下のトピックについて説明します。

- [PROFIBUS DP]チャンネルによるデータ転送を設定する方法
- 接続およびタグの設定方法

現行バージョンの PROFIBUS DP における変更点

提供されている文書とは異なり、現在のバージョンの PROFIBUS DP には次の変更点があります。

- PROFIBUS DP Master はアプリケーション OPC サーバーとして使用されます。

6.6.2 WinCC ドライバ Profibus DP のプロパティ

プロパティ

WinCC ドライバ Profibus DP には次のプロパティがあります。

- 通信プロセッサ(= CP)を使用する WinCC PC は、Profibus の DP マスタです。
- すべての DP 標準スレーブをアドレス指定できます。
- 最大 4 つの CP カードを初期化して、CP モジュールごとに最大 123 個の DP スレーブステーションと設定できます。

制限値は、より新しいバージョンで変更可能であるため、コミッショニング前にチェックされる必要があります。

6.6 PROFIBUS DP

他の Profibus バスパートナーへの通信は、ドライバがこれを許可する場合に、他のプロトコルを使用して行うことができます。

注記

1 つの DP マスタのみ

同一のスレーブにアドレス指定している他の DP マスタを Profibus バスで接続することはできません。

6.6.3 「Profibus DP」ドライバの統合

規格

Profibus 分散 I/O (DP)規格に準拠しています。

- DIN 19245-3、または
- pr EN 50170

必要条件

ハードウェア:

- WinCC ドライバの Profibus DP を使用するには、Profibus の接続用の通信プロセッサ CP 5613 (A3)または CP 5612 が必要です。
これを使用して、すべての DP 標準スレーブをアドレス指定できます。
- 使用される通信プロセッサの数は、PC の空きの割り込みに依存しています。

ソフトウェア:

- CP を設置し、設定するには、ドライバおよびプロジェクト設計ソフトウェアが必要です。
これは SIMATIC NET CD で入手可能です。

手順

1. タグ管理のナビゲーションエリアで、[Profibus DP]通信ドライバを選択します。
チャンネルユニットが作成されます。
2. チャンネルユニットのコンテキストメニューで、[システムパラメータ]を選択します。
設定ダイアログが開きます。
3. CP 基板番号およびモニタリング時間を指定します。

4. チャンネルユニットのコンテキストメニューで、[新規接続]エントリを選択します。
5. 接続の名前を入力します。
6. 接続のコンテキストメニューから[接続パラメータ]エントリを選択します。
設定ダイアログが開きます。
7. スレーブアドレスを選択し、[OK]で確定します。

6.6.4 「PROFIBUS DP」ドライバの設定

システムパラメータの設定

CP 基板番号

PC での CP カードの番号です(設定ツールから)。

値の範囲:

- 1～4
- 0 = インストールされていない

ウォッチドッグ時間

通信カードの WinCC のモニタリング時間の係数を入力します。モニタリング時間は、どの場合でも 0.4 秒の倍数です。

この機能は出力データと一緒に提供可能なスレーブに対してのみ有効です。

- 入力 0:
モニタリングが無効にされます。
- 入力 > 0:
これ以上書き込みアクセスがない場合、時間が経過するにつれて、出力が 0 に設定されます。
これは、適切な WinCC 設定によって確認される必要があります。

接続パラメータの設定

スレーブアドレス


読み取りまたは書き込み対象のスレーブのアドレス

値の範囲:

- 1～127

タグアドレスの設定

タグアドレスを設定するには、[プロパティ - タグ]ウィンドウの空の[アドレス]フィールドをクリックします。

次のシンボルを使用して設定ダイアログを開きます。 

注記

接続のパフォーマンス

電源やスレーブットが正常に機能しない場合、次に注意します。

アクセスが Profibus の全体の DP デバイスにのみ可能であるため、タグの相互接続の更新時間が接続に影響します。

プロセスタグのプロパティ

フィールド	意味
入力	スレーブの入力範囲
出力	スレーブの出力範囲
長さ(ビット)	タグサイズの表示(ビット単位) 値は、以前に選択されたデータタイプに基づいています。 例外:未処理データタグ
バイトオフセット	タグのコンテンツが保存された後のバイト数 値の範囲:0 から長さ-1
ビットオフセット	[2 進タグ]データタイプを使用した場合のみ有効 2 進タグが入力された、上記で指定されたバイトのビット数 値の範囲:0~7 より大きい値は、バッファの長さが超過しない場合に可能です。
バイト配列の変更	無効:リトルエンディアン(初期設定) アクティブ:ビッグエンディアン

未処理データタグのプロパティ

フィールド	意味
入力	スレーブの入力範囲
出力	スレーブの出力範囲
長さ(ビット)	無効

フィールド	意味
バイトオフ セット	タグのコンテンツが保存された後のバイト数 値の範囲:0 から長さ-1
ビットオフ セット	無効
長さ(バイ ト)	このフィールドに必要なブロック長を入力します。 このフィールドの長さの単位はバイトです。
ブロックの 送信/受信	定義されたデータブロックは、WinCC からのリクエスト後に送信または受 信されます。

6.7 S5 Ethernet 第4層

6.7.1 WinCC チャンネル"SIMATIC S5 Ethernet 第4層"

はじめに

通信ドライバは、例えばオートメーションシステム SIMATIC S5-115U/H、SIMATIC S5-135U、SIMATIC S5-155U/H を ISO トランスポートプロトコルまたは TCP/IP プロトコルで接続するために、使用されます。

使用される通信プロトコルによって異なりますが、以下の通信パートナーが実装されます。

通信プロトコル	WinCC 側	SIMATIC S5 側
ISO トランスポートプロトコル	CP1612 A2 (3Com 互換) CP1613 A2 CP1623	CP1430 TF
TCP/IP (RFC1006 に適合)	CP1612 A2 (3Com 互換) CP1613 A2 CP1623	CP1430 TCP

このチャンネルを使う場合、ローカルデータベースは不要です。

チャンネルユニット

通信ドライバには、最大 2 つのチャンネルユニット CP1612 A2、CP1613 A2 または CP1623 を操作できる 2 つのチャンネルユニット「CP1413-x」があります。チャンネルユニットの機能は同じです。2 つの CP の論理デバイス名が異なるだけです。CP1623 は CP1613 A2 と同じですが、PCI Express で動作します。

3 番目のチャンネルユニット「TCP/IP」を使用して、CP1612 A2、CP1613 A2 または CP1623 で TCP/IP プロトコルを介して通信を確立できます。

論理デバイス名は、チャンネルユニットのシステムパラメータで変更できます。ここで、使用されるプロトコルのパラメータを設定することもできます。

以下のアプリケーション機能があります。

- SIMATIC Industrial Ethernet (CP 1612 A2/1613 A2/1623)の通信モジュール用チャンネルユニット「S5-Transport (CP 1413-1)」。
- SIMATIC Industrial Ethernet (CP 1612 A2/1613 A2/1623)の通信モジュール用チャンネルユニット「S5-Transport (CP 1413-2)」。
- SIMATIC Industrial Ethernet (CP 1612 A2/1613 A2/1623)の通信モジュール用のチャンネルユニット「S5-Transport (TCP/IP)」。

6.7.2 タグのデータタイプ

はじめに

論理接続に必要なタグを定義します。WinCC の観点から、以下のデータタイプにアクセスできます。

- 2進タグ
- 符号なし 8 ビット値
- 符号付き 8 ビット値
- 符号なし 16 ビット値
- 符号付き 16 ビット値
- 符号なし 32 ビット値
- 符号付き 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- テキストタグ、8 ビット文字セット
- 未処理データタイプ

6.7 S5 Ethernet 第4層

6.7.3 チャンネルの設定

6.7.3.1 チャンネル"SIMATIC S5 Ethernet 第4層"の設定

はじめに

チャンネル"SIMATIC S5 Ethernet 第4層"の設定には、以下のステップが必要です。

1. 接続の設定
2. タグの設定
3. システムパラメータの設定

6.7.3.2 接続の設定方法

はじめに

接続パラメータは、使用するすべてのプロトコルでほとんど同一です。以下の例のように、チャンネルユニット"CP1413-x"と ISO トランスポートプロトコルを使用して、通信が説明されます。

TCP/IP プロトコルを実装するとき、Ethernet アドレスの代わりに AS の IP アドレスが入力されます。IP アドレスは、ドットで区切られた 4 つの数値で構成されます。数値は 0～255 の範囲でなければなりません。

論理接続に対して、WinCC は、1 つの読み取り用接続("READ ファンクション"エリア)と 1 つの書き込み用接続("WRITE ファンクション"エリア)を、トランスポートレイヤに確立します。両方のファンクションのアドレスパラメータが、ダイアログで定義されます。両方の接続が確立された場合のみ、論理接続も"確立済"と表示されます。

READ ファンクションの割り付け

WinCC 側	SIMATIC S5 側
FETCH-アクティブ ([READ-アクティブ]要求)	READ-パッシブ ([READ-パッシブ]要求)
FETCH-パッシブ ([WRITE-パッシブ]要求)	WRITE-アクティブ ([WRITE-アクティブ]要求)

注記

AS からデータがアクティブで送信され、接続パラメータの READ ファンクションに[FETCH パッシブ]が設定されている場合、AS のデータエリアで 2 進変数もバイト変数も書き込みできません。

少なくとも 1 つのテレグラムが AS から WinCC へ送信された場合のみ、[FETCH パッシブ]接続に[OK]ステータスが割り付けられます。

WRITE ファンクションの割り付け

WinCC 側	SIMATIC S5 側
[WRITE アクティブ]要求	[WRITE パッシブ]要求

手順

1. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。



2. [Ethernet アドレス AG]フィールドで、産業用 Ethernet バスの SIMATIC S5 のステーションアドレスを入力します。TCP/IP プロトコルを実装するとき、ここで[IP アドレス AG]フィールドに IP アドレスを入力します。
3. WinCC システムの READ ファンクションのパラメータを定義します。これらは SIMATIC S5 で使用する要求とは、無関係です。
4. 次に、CP1430 TF の設定中に[リモートパラメータ]で"TSAP"として設定された、割り付け済フィールド[自分の TSAP]に、値を入力します。
5. そこで、CP1430 TF の設定中に[ローカルパラメータ]で"TSAP"として設定された、割り付け済フィールド[リモート TSAP]に、値を入力します。
6. WRITE ファンクションについて、パラメータ[自分の TSAP]および[リモート TSAP]を、適切に定義します。

注記

"TSAP"の入力では、空白は使えません。

6.7.3.3 タグの設定

タグの設定

はじめに

チャンネル"SIMATIC S5 Ethernet 第4層"経由での WinCC と AS の接続用に、異なるデータタイプのタグを WinCC で作成できます。以下に、これらのデータタイプのタグの設定方法を説明します。

- タグのアドレス
- ビット単位でアクセスするタグの設定
- バイト単位でアクセスするタグの設定
- ワード単位でアクセスするタグの設定
- 未処理データタグの設定

タグのアドレス

はじめに

SIMATIC S5 のアドレス構造に準じてタグアドレスを入力します。

タグタイプによって、AS でのメモリ領域へのアクセスはビット単位、バイト単位あるいはワード単位となります。この目的で、書込み要求が出されるたびにアドレス指定されたメモリ領域が AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

注記

読み取りデータ領域で AS によって行なわれた変更は、データ領域に書き戻された時点で上書きされます。

AS からデータがアクティブで送信され、接続パラメータの READ ファンクションに[FETCH パッシブ]が設定されている場合、AS のデータ領域で 2 進変数もバイト変数も書き込みできません。


タグのアドレスの設定は、タグタイプとは独立して行なわれます:

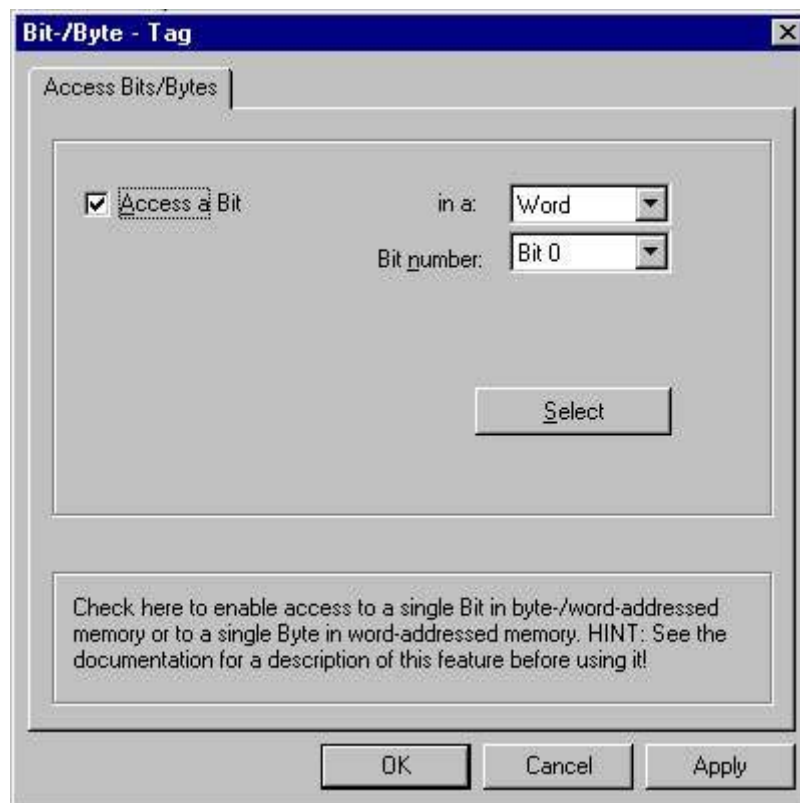
- [2進]や[8ビット値]のタグの場合、最初に[ビットタグ/バイトタグ]ダイアログが開き、そこで AS のメモリ領域へのアクセスがビット単位かバイト単位かを定義します。その後、AS メモリのタグのアドレスは、[アドレスプロパティ]ダイアログで定義されます。
- ワード指向のタグでは、AS メモリのタグのアドレスは、[アドレスプロパティ]ダイアログで定義されます。

AS メモリへのアクセスがワード単位なので、[ビットタグ/バイトタグ]ダイアログは開きません。

ビット単位でアクセスするタグの設定方法

手順

1. 接続を選択して、ショートカットメニューから[ビットタグ/バイトタグ]ダイアログウィンドウを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。
2. [選択]ボタンをクリックします。[ビットタグ/バイトタグ]ダイアログが開きます。



3. チェックボックスを使って、メモリエリアの特定のビットを読み書きするために、アクセスを有効にするべきかどうかを定義します。
4. [選択]フィールドで、AS メモリのアドレス指定方法を、例えば"ワード"または"バイト"のように選択します。
5. [選択]フィールドで変更するビット数を選択します。
6. [選択]ボタンを使用して、AS のタグアドレスを定義する[アドレスプロパティ]ダイアログを開きます。

注記


S5 の場合、フラグ、入力、出力はバイト単位でアドレス指定し、データブロック(DB、DX)はワード単位でアドレス指定します。

[ビットにアクセス]チェックボックスを有効にすると、[アドレスプロパティ]ダイアログのフィールドの表示に影響します。

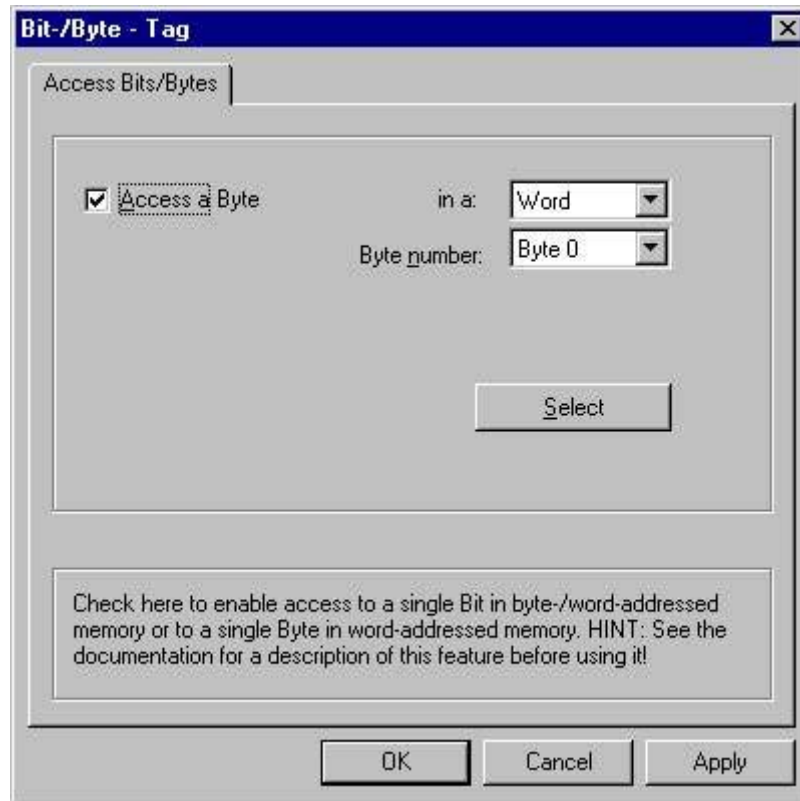
ワード指向タグの場合、タグアドレスと PLC メモリへのアクセスがワード単位なので、上記の[ビットタグ/バイトタグ]は開きません。

バイト単位でアクセスするタグの設定方法

手順

1. タグを選択してから、[データタイプ]フィールドで、データタイプ[符号なし 8 ビット値]または[符号付き 8 ビット値]を選択します。
2. 接続を選択して、ショートカットメニューから[ビットタグ/バイトタグ]ダイアログウィンドウを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。

3. [選択]ボタンをクリックします。[ビットタグ/バイトタグ]ダイアログが開きます。



4. チェックボックスを使って、メモリエリアの特定のバイトを読み書きするために、アクセスを有効にするべきかどうかを定義します。
5. [選択]フィールドでは、AS メモリのアドレス指定タイプとして、"ワード"だけが表示されます。
6. [選択]フィールドで変更するバイト数を、選択します。
7. [選択]ボタンを使用して、AS のタグアドレスを定義する[アドレスプロパティ]ダイアログを開きます。

注記

S5 の場合、フラグ、入力、出力はバイト単位でアドレス指定し、データブロック(DB、DX)はワード単位でアドレス指定します。

[バイトにアクセス]チェックボックスを選択すると、[アドレスプロパティ]ダイアログのフィールドの表示に影響します。

ワード指向タグの場合、タグアドレスと PLC メモリへのアクセスがワード単位なので、上記の[ビットタグ/バイトタグ]は開きません。


ワード単位でアクセスするタグの設定方法

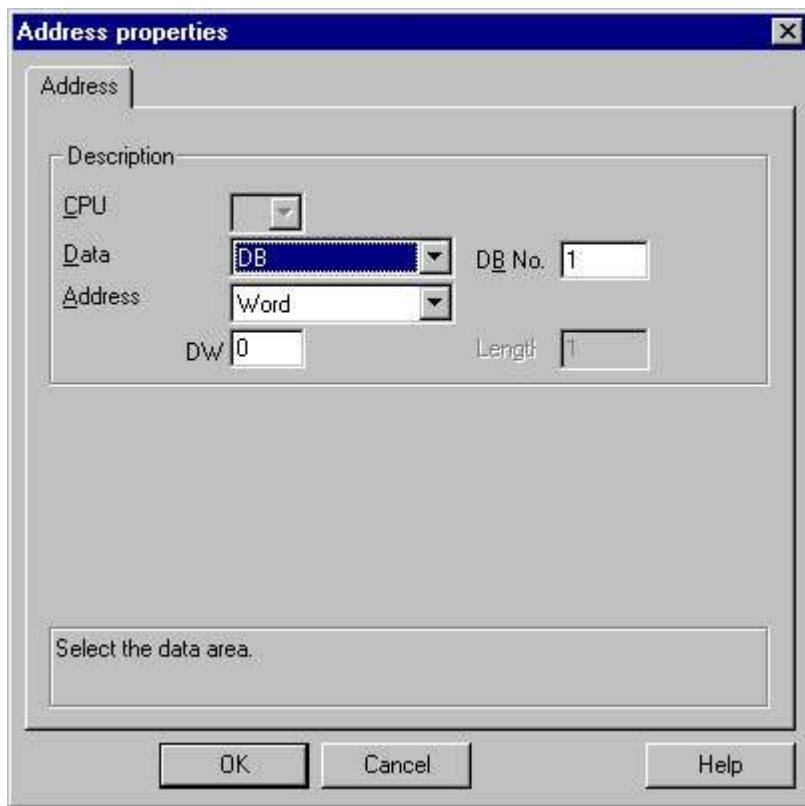
はじめに

AS のタグのアドレスは、ここで説明するダイアログで定義されます。

- [2 進]や[8 ビット値]のタグの場合、最初に[ビットタグ/バイトタグ]ダイアログが開き、そこで AS のメモリエリアへのアクセスがビット単位かバイト単位かを定義します。
- ワード指向タグの場合、タグアドレスと AS メモリへのアクセスがワード単位なので、[ビットタグ/バイトタグ]ダイアログは開きません。

手順

1. タグを選択してから、[データタイプ]フィールドで、タグに必要なデータタイプ(符号付き 16 ビット値など)を選択します。
2. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



3. タグが、[アドレス]タブの[データエリア]フィールドの、データブロック、フラグエリアエリア、入力エリアまたは出力エリアのどれに位置するかを、選びます。
4. タグがデータブロックにある場合、[DB 番号]フィールドも表示されます。ここで、データブロック番号を入力します。
5. [アドレス指定]フィールドで、アドレス指定のタイプを入力します。通常は、デフォルト定義を使えます。
6. 個別フィールド(例、[DW])にアドレスを入力します。

注記

"2 進"または"8 ビット値"タイプのタグの場合、このダイアログのフィールドの表示は、[ビットタグ/バイトタグ]ダイアログの[ビット/バイトへのアクセス]での選択によって異なります。

ワード指向データエリアのタグに書き込む場合、開始アドレスが残存バイトにあり、タグ長は偶数である必要があります。

未処理データタグの設定方法


はじめに

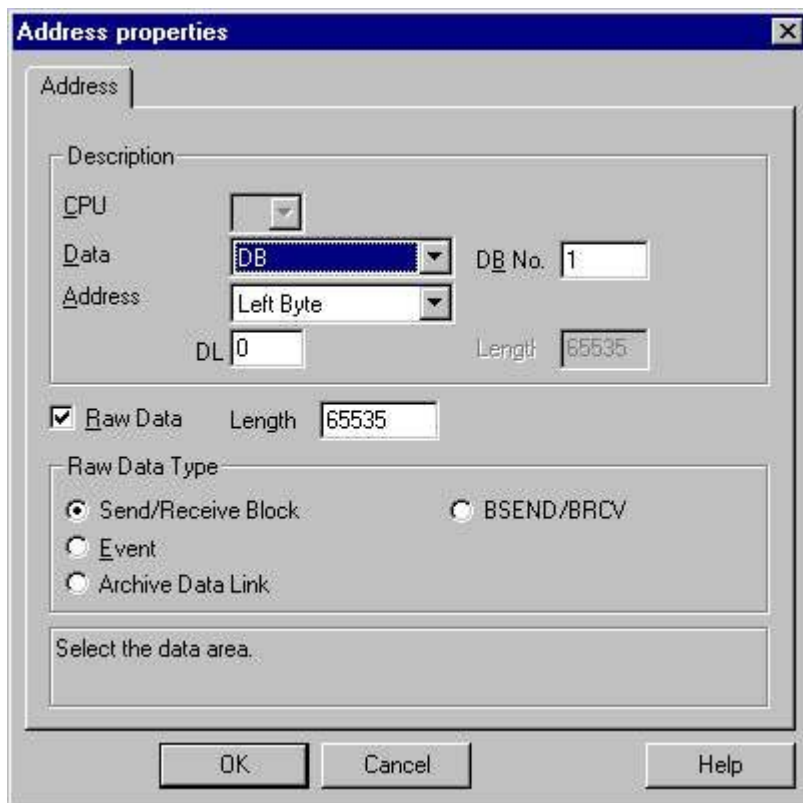
以下に、未処理データタグのアドレスの定義方法を説明します。

注記

ワード指向データエリアのタグに書き込む場合、開始アドレスが残存バイトにあり、タグ長は偶数である必要があります。

手順

1. タグを選択してから、[データタイプ]フィールドで、[未処理データタイプ]エントリを選択します。
2. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



3. タグが、[データエリア]フィールドの、データブロック、フラグ、入力エリアまたは出力エリアのどれに位置するかを、選びます。
4. タグがデータブロックにある場合、[DB 番号]フィールドも表示されます。ここで、データブロック番号を入力します。
5. [アドレス指定]フィールドで、アドレス指定のタイプを入力します。通常、デフォルトの定義を使用できます。
6. 下のフィールドにデータアドレスを入力します。このフィールドのラベルは、[アドレス指定]フィールドのエントリによって異なります(例: ワードアドレス指定の場合は[DBW])。
7. [長さ]フィールドに、未処理データブロックの長さをバイト単位で入力します。
8. どのタイプの未処理データタグが、[未処理データタイプ]エリアで対象となるかを定義します。

6.7.3.4 システムパラメータ

チャンネルユニットのシステムパラメータ

はじめに

WinCC の標準設定とは異なる設定が必要な場合は、チャンネルユニットの[システムパラメータ]ダイアログを使用して、必要な変更をすべて実行することができます。

システムパラメータは、使用するすべてのプロトコルでほとんど同一です。TCP/IP プロトコルの実装時に、インストール中のデバイス名が違うだけです。

以下の個別ポイントを変更できます。

- デバイス名
- トランスポートパラメータ

注記

システムパラメータは、AS の全ての CP に適用されます。

デバイス名

WinCC とオートメーションシステム間の通信は、論理デバイス名を介して実行されます。これらの名前は通信モジュールのインストール中に割り当てられ、ユニットに固有です。デバイス名は論理デバイス名を示します。論理デバイス名はデフォルト定義として、ISO トランスポートプロトコルで"/CP_H1_1:/SCP"の名前を、TCP/IP プロトコルで"/TCP_IP:/SCP"の名前を、与えられます。

注記

TCP/IP プロトコルを使用する場合、WinCC でのデバイス名が、[PG/PC インターフェースの設定]ダイアログの[アプリケーションのアクセスポイント]と一致することを、必ず確認してください。デバイス名は、[PG/PC インターフェースの設定]でも変更しなければなりません。

トランスポートパラメータ

PDU サイズ、設定試行などチャンネルユニットの固有設定は、トランスポートパラメータで実行します。

デバイス名の変更方法

はじめに

チャンネルユニットのパラメータは、例えば論理デバイス名またはトランスポートパラメータなどのシステムパラメータで、設定されます。

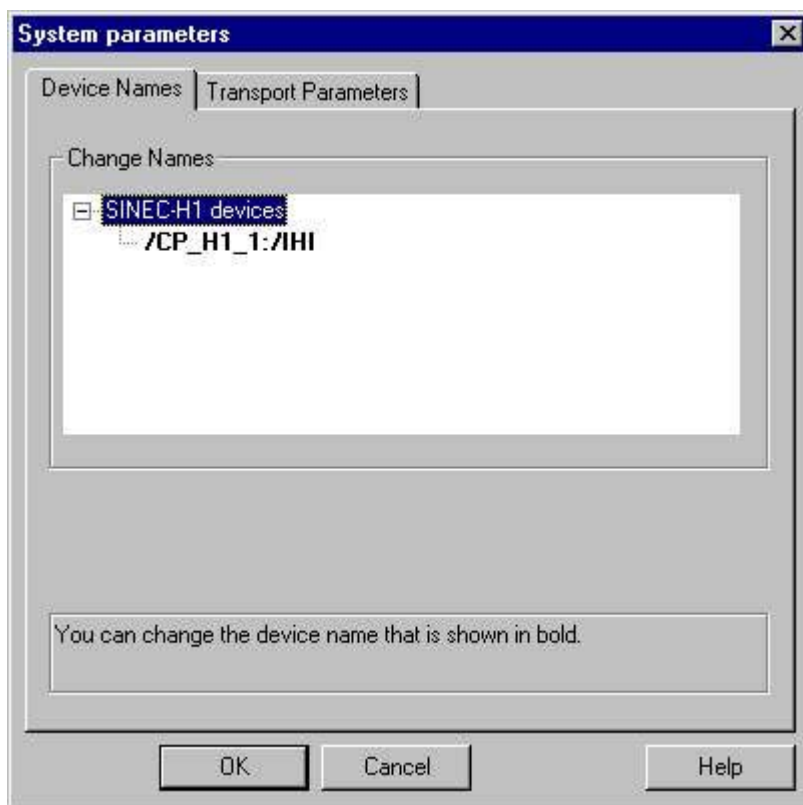
システムパラメータは、使用するすべてのプロトコルでほとんど同一です。

以下の例のように、チャンネルユニット"CP1413-x"と ISO トランスポートプロトコルを使用して、通信が説明されます。

TCP/IP プロトコルの実装時に、インストール中のデバイス名が違うだけです。

手順

1. コンテキストメニューからチャンネルユニットを選択して、[システムパラメータ]ダイアログウィンドウを開きます。
2. [デバイス名]タブを選択します。



3. ここで、太字で表示されているデバイス名をマウスで選択し、デバイス名の[名前]フィールドでマウスクリックすることによって、変更できます。

注記

デバイス名は、ハードウェアドライバのインストール中に定義されます。他の望ましくない名前を定義した場合に限り、デバイス名を変更してもかまいません。

トランスポートパラメータの変更方法

手順

1. チャンネルユニットを選択し、ショートカットメニューを使用して[システムパラメータ]ダイアログウィンドウを開きます。
2. [トランスポートパラメータ]タブを選択します。



3. [PDU サイズ]の値を、通信モジュール CP 1430 で設定された値に設定します。
4. [設定試行]フィールドで、接続確立の試行回数を定義します。
5. [送信繰り返し期間]領域で[無限]を選択します。
6. [確認時間]フィールドに、例えば値 30 を入力すると、通信パートナーがこの時間内に応答しなかった場合(例えば AS が[停止]状態)、最大 30 秒後にタグのステータスが通知されます。

6.7 S5 Ethernet 第4層

6.7.4 付録

6.7.4.1 付録

はじめに

チャンネル"SIMATIC S5 Ethernet 第4層"に関する追加情報を、付録で提供します。

6.7.4.2 内部エラーコードと定数

内部エラーコードと定数

はじめに

以下の表に、最も重要なエラーコードと定数を示します。この情報は"内部関係者"を対象としたものです。従って、エラーコードと定数の意味について、これ以上詳細に述べたことはありません。

- 接続障害中のエラーコード
- iNA960 メッセージ
- SCI メッセージ

接続障害中のエラーコード

はじめに

最も重要なエラーコードが、このセクションにリストされています。エラーコードがこの表に記載されていないエラーが発生した場合は、WinCC ホットラインにお電話ください。

Fehler_0002-INVALID_RQ

障害のあるリクエストブロック。

Fehler_0004-NO_RESOURCES

CP に空いているリソースがありません。

Fehler_0006-UNKNOWN_REFERENCE

間違ったオープン参照が定義されました。

Fehler_0008-BUFFER_TOO_SHORT

ユーザーバッファが短すぎます。

Fehler_0010-BUFFER_TOO_LONG

ユーザーバッファが長すぎます。

Fehler_0012-ILLEGAL_REQ

間違った"negot_options"が定義されました。

Fehler_0014-REM_ABORT

リモートステーションによって接続が中断されました。

Fehler_0016-LOC_TIMEOUT

タイムアウト。

Fehler_0018-UNKNOWN_CONN_CLASS

未知の接続クラス。

Fehler_0020-DUP_REQ

接続は既に確立されています。

Fehler_0022-CONN_REJECT

接続要求がリモートによって拒否されました。

Fehler_0024-NEGOT_FAILED

接続が、欠陥のある"negot-option"によって中断されました。

Fehler_0026-ILLEGAL_ADDRESS

欠陥のあるトランスポートアドレス。

Fehler_0028-NETWORK_ERROR

バスまたは CP が中断されました。

Fehler_0030-PROTOCOL_ERR

プロトコルエラー。

Fehler_0032-ILLEGAL_RB_LENGTH

リクエストブロック長が間違っています。

Fehler_0784-E_NO_HW

通信ハードウェアが見つかりません。

- 通信モジュールに欠陥があります。
- 通信モジュールが正しくインストールされていません。
- 間違ったポートアドレスが定義されました。

Fehler_0786-E_CNF

ドライバが間違って設定されているか、またはレジストリに無効なパラメータがあります。

Fehler_0787-E_BAUDRATE

ボーレートまたは割り込みベクトルの定義が間違っています。

Fehler_0788-E_HSA

HSA(最上位ステーションアドレス)の定義が間違っています。

Fehler_0789-E_TS

定義されたローカル参加者番号(TS_ADR)は既に割り当てられています。

Fehler_0791-E_INT_NOT_PROV

定義された割り込みベクトル(IRQ)は、通信モジュールでは使用できません。

Fehler_0792-E_INT_BUSY

定義された割り込みベクトル(IRQ)は、通信モジュールで既に占有されています。

Fehler_0800-E_NO_FILE

選択された通信ドライバをロードできません；ファイルが見つかりませんでした。

- 通信ドライバが正しくインストールされていません。

Fehler_0897-E_LOGDEV

論理デバイスが、レジストリで定義されていません。

- 通信ドライバが正しくインストールされていません。
- レジストリのエントリーが壊れているか、あるいは削除されています。
- [PG/PC インターフェースの設定]プログラムで、論理デバイス名の設定を確認してください。
- "システムパラメータ-デバイス"マスクの論理デバイス名の設定を、確認してください。

Fehler_0898-E_L2DRIVER

エントリ"L2DRIVER"がレジストリにありません。

- モジュールエラー、またはモジュールが間違ってインストールされています。

Fehler_0900-E_L4DRIVER

エントリ"L4DRIVER"がレジストリにありません。

- モジュールエラー、またはモジュールが間違ってインストールされています。

Fehler_30000-EC_WATCHDOG

ウォッチドッグエラー。

Fehler_30001-EC_PDUERROR

PDU は予定されていません。

Fehler_30005-EC_ONLERROR

S7-Online-DLL のローディング障害。

iNA960 メッセージ**一般的な iNA960 メッセージ**

OK_RESP	1	0x01	エラーなしで要求が実行されました
OK_EOM_RESP	3	0x03	エラーなしでデータブロックが受信されました
OK_DECIDE_REQ_RESP	5	0x05	エラーなしで要求が実行されました
OK_CLOSED_RESP	7	0x07	ローカルユーザーによって接続が中断されました

iNA960 エラーメッセージ

INVALID_REQ	2	0x02	障害のあるリクエストブロック
NO_RESOURCES	4	0x04	CP に空いているリソースがありません
UNKNOWN_REFERENCE	6	0x06	間違ったオープン参照が定義されました
BUFFER_TOO_SHORT	8	0x08	ユーザーバッファが短すぎます
BUFFER_TOO_LONG	10	0x0A	ユーザーバッファが長すぎます
ILLEGAL_REQ	12	0x0C	間違った"negot_options"が定義されました
REM_ABORT	14	0x0E	リモートステーションによって接続が中断されました
LOC_TIMEOUT	16	0x10	タイムアウト
UNKNOWN_CONN_CLASSES	18	0x12	未知の接続クラス
DUP_REQ	20	0x14	接続は既に確立されています
CONN_REJECT	22	0x16	接続要求がリモートによって拒否されました

NEGOT_FAILED	24	0x18	接続が、欠陥のある"negot-option"を中断します
ILLEGAL_ADDRESS	26	0x1A	欠陥のあるトランスポートアドレス
NETWORK_ERROR	28	0x1C	バスまたは CP が中断されました
PROTOCOL_ERR	30	0x1E	プロトコルエラー
ILLEGAL_RB_LENGTH	32	0x20	リクエストブロック長が間違っています

SCI メッセージ

『SINEC 通信インターフェース SCI』 マニュアル(A/5～15)の説明を参照して下さい。

SCI メッセージ

SCP_OK	0	0x00	エラーなし
SCP_INCONS	201	0xC9	マイナーデバイス番号が 00 ではありません
SCP_RESOURCE	202	0xCA	DPRAM リクエストが無効です
SCP_CONFIG	203	0xCB	設定エラー(NUM_PROCS)
SCP_NOCONFIG	204	0xCC	SCP ドライバが設定されていません
SCP_PARAM	206	0xCE	間違ったモード
SCP_DEVOPEN	207	0xCF	既に関われています
SCP_BOARD	208	0xD0	ボードが挿入/認識されていません
SCP_SOFTWARE	209	0xD1	IRQ エラー、またはソフトウェアが見つかりません
SCP_MEM	210	0xD2	DPRAM の低メモリ
SCP_MODE	211	0xD3	ダウンロードプロセスが終了していません
SCP_LOADER	212	0xD4	ローダーから応答がありません
SCP_SIGNAL	213	0xD5	プロセスが非同期で開始しました
SCP_NOMESS	215	0xD7	プロセスのメッセージが到着していません
SCP_USERMEM	216	0xD8	length_of_buffer が小さすぎます。
SCP_WINDOW	217	0xD9	SEND コールが多すぎます
SCP_TIMEOUT	219	0xDB	SCP でタイムアウト
SCP_ATTACH	220	0xDC	リセットが実行されていません/チャンネルがまだアクティブです

6.7 S5 Ethernet 第4層

SCP_ILLEGAL_REQUEST	221	0xDD	無許可のリクエスト
SCP_ERECOVERF	223	0xDF	バッファが scp_receive で取得されていません
SCP_ECLOSED	224	0xE0	全てのバッファが接続に割り当てられています
EUSERMAX	225	0xE1	
SCP_EINTR	226	0xE2	
SCP_BOARD_OPEN	231	0xE7	
SCP_NO_WIN_SERV	233	0xE9	
EPROTECT	234	0xEA	ライセンスが見つかりません

SCI メッセージ

SCP_DB_FILE_DOES_NOT_EXIST	240	0xF0
SCP_DB_FILE_CLOSE_NOT_OK	241	0xF1
SCP_SEND_NOT_SUCCESSFUL	242	0xF2
SCP_RECEIVE_NOT_SUCCESSFUL	243	0xF3
SCP_NO_DEVICE_AVAILABLE	244	0xF4
SCP_ILLEGAL_SUBSYSTEM	245	0xF5
SCP_ILLEGAL_OPCODE	246	0xF6
SCP_BUFFER_TOO_SHORT	247	0xF7
SCP_BUFFER_1_TOO_SHORT	248	0xF8
SCP_ILLEGAL_PROTOCOL_SEQUENCE	249	0xF9
SCP_ILLEGAL_PDU_ARRIVED	250	0xFA
SCP_REQUEST_ERROR	251	0xFB
SCP_NO_LICENSE	252	0xFC

SCP インターフェースの追加オンライン DLL メッセージ

E_TIMER_INIT	768	0x0300	WIN タイマー設定リクエストが失敗しました
E_INIT_COM	769	0x0301	
E_NO_HW	784	0x0310	MPI モジュールが見つかりません
E_HW_DEFECT	785	0x0311	ハードウェアの問題

E_CNF	786	0x0312	間違った設定パラメータ
E_BAUDRATE	787	0x0313	間違ったボーレート/間違った IntVector
E_HSA	788	0x0314	間違って設定された HSA
E_TS	789	0x0315	設定されたアドレスは既に割り当てられています
E_OCC	790	0x0316	HW_Device は既に割り当てられています
E_INT_NOT_PROV	791	0x0317	割り込みは利用できません
E_INT_BUSY	792	0x0318	割り込みは占有されています
E_SAP	793	0x0319	SAP が無効です： SAP は占有されていません
E_UNPLUGGED	794	0x031a	リモートステーションが見つかりません
E_SYNI	795	0x031b	Syni エラーが発生しました。
E_AMPRO	796	0x031c	AMPRO 2 でシステムエラーがレポートされました
E_BUFFSIZE	797	0x031d	このサイズのバッファは作成されていません
E_NO_FILE	800	0x0320	DLL/VxD ファイルが見つからないか、またはレジストリのエントリが壊れています
E_NO_ENTRY	801	0x0321	DLL にアドレスが存在しません
E_VERSION	816	0x0330	SMC ドライバと SMC ファームウェアでバージョンが矛盾しています
E_COMCNF	817	0x0331	COM ポートの設定に問題があります
E_NO_SMC	818	0x0332	SMC が応答しません
E_COMMBADID	819	0x0333	COM ポートが設定されていません
E_COMMOPEN	820	0x0334	COM ポートを使用できません
E_SMCBUSY	821	0x0335	シリアルドライバは、現在他の設定に使われています
E_SMCMODEM	822	0x0336	PC/MPI ケーブルへの接続が存在しません。
E_SMCNOLEG	823	0x0337	PC/MPI ケーブルがリクエストを拒否し、必要なオーソリゼーションがありません
E_ONLINE	896	0x0380	IOCTL インターフェースで内部エラー
E_LOGDEV	897	0x0381	論理デバイスがレジストリにありません
E_L2DRIVER	898	0x0382	L2DRIVER エントリがレジストリにありません
E_L4DRIVER	900	0x0384	L4DRIVER エントリがレジストリにありません
E_SYSERROR	1023	0x03FF	システムエラー

チャンネル固有のエラーコード

EC_WATCHDOG	30000	0x7530	ウォッチドッグエラー
EC_PDUERROR	30001	0x7531	PDU は予定されていません
EC_ONLERROR	30005	0x7535	S7-Online-DLL のローディング失敗

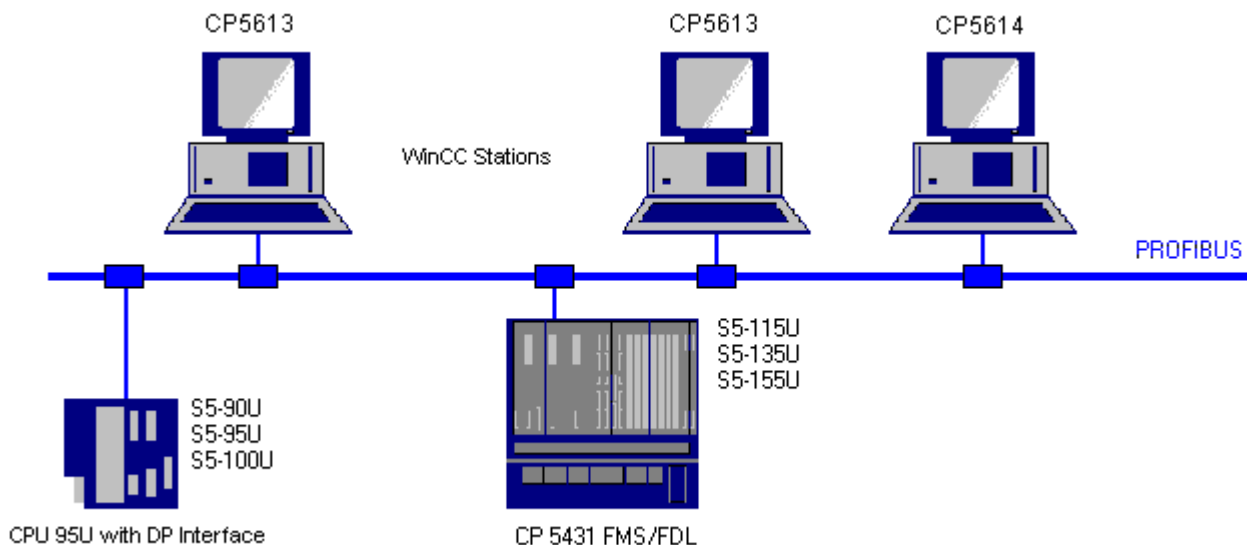
6.8 S5 PROFIBUS FDL

6.8.1 WinCC チャンネル"SIMATIC S5 Profibus FDL"

はじめに

「SIMATIC S5 Profibus FDL」チャンネルは、WinCC ステーションと SIMATIC S5 オートメーションシステムとの間の通信に使用されます。この場合、PROFIBUS (プロセスフィールドバス) ネットワークタイプと FDL (フィールドデータリンク) が使われます。

PROFIBUS は、中小規模のデータボリューム用のネットワークです。最大 127 の接続可能なノードを使用して、幅広いオートメーションタスクに対応できます。



タグは、要求フレームと応答フレームを使用した FDL プロトコルを使用して、PROFIBUS 経由で読み込み/書き込みされます。要求フレームは、WinCC からオートメーションデバイスへ送信されます。AS は応答フレームで応答します。

FDL 接続は、ローカルおよびリモート接続エンドポイント(サービスアクセスポイント)で指定されます。

ここでは以下を説明します。

- 「SIMATIC S5 Profibus FDL」チャンネルを使用したデータ転送の設定方法。
- サンプルプロジェクトの作成方法

チャンネルユニット FDL (CP5412/A2-1)

使用する通信プロセッサにかかわらず、「FDL (CP5412/A2-1)」チャンネルユニットを介して SIMATIC S5 に接続する可能性があります。

このチャンネルユニットは、最大 24 接続までをサポートします。チャンネルが機能するためには、チャンネルユニットと接続を作成する必要があります。

サービスアクセスポイント

SAP は、PROFIBUS ノード内のローカルなデータインターフェースです。SAP は、WinCC および AS で設定する必要があります。サービスアクセスポイントで、一意の ID が定義されます。この一意の ID は、WinCC と AS の間の通信に必要です。

アクティブ接続

アクティブ接続は、Fetch (取り出し) 接続とも呼ばれます。これは、アクティブなパートナーが通信パートナーからデータをフェッチする接続です。データをフェッチする通信パートナーをパッシブパートナーと呼びます。

パッシブ接続

アクティブな AS が要求フレームなしでパッシブな WinCC パートナに非同期にデータを送信する場合、パッシブ接続が存在します。

6.8.2 サポートされるデータタイプおよびデータ範囲

はじめに

特定のデータタイプおよびデータ範囲だけが、PROFIBUS FDL 経由の SIMATIC S5 からの通信でサポートされます。

サポートされるデータタイプ

WinCC データタイプ	SIMATIC S5 データタイプ
2 進タグ	ビット
符号付き 8 ビット値	SIMATIC S5 に存在せず
符号なし 8 ビット値	バイト

WinCC データタイプ	SIMATIC S5 データタイプ
符号付き 16 ビット値	WORD
符号なし 16 ビット値	WORD
符号付き 32 ビット値	DWORD
符号なし 32 ビット値	DWORD
浮動小数点数 32 ビット IEEE 754	DWORD
浮動小数点数 64 ビット IEEE 754	SIMATIC S5 に存在せず
テキストタグ、8 ビット文字セット	バイト配列
テキストタグ、16 ビット文字セット	SIMATIC S5 に存在せず
未処理データタイプ	バイト配列

SIMATIC S5 タグへのアクセス

SIMATIC S5 タグへのアクセスは、ワード単位でデータブロック DB あるいは拡張データブロック DX に対して行なわれます。これは読み取りおよび書き込みアクセスを許可します。

データタイプがビットの SIMATIC S5 タグへのアクセス

データタイプがビットの SIMATIC S5 タグは、読み取りアクセスのみを許可します。この制限は、アクティブあるいはパッシブ接続に適用されます。

データタイプがバイトの SIMATIC S5 タグへのアクセス

データタイプが BYTE の SIMATIC S5 タグは、読み取りアクセスのみを許可します。

バイトタグを設定するには、アドレス指定に、16 ビットデータワードの"左バイト"または"右バイト"を選択する必要があります。

データタイプがバイト配列の SIMATIC S5 タグへのアクセス

データタイプがバイト配列の SIMATIC S5 タグは、読み取りアクセスのみを許可します。

6.8.3 WinCC チャンネル"SIMATIC S5 Profibus FDL"の機能

はじめに

WinCC から PROFIBUS の通信ドライバを経由する通信の能力を、以下に示します。すべてのサポートされるデータタイプと、それぞれのタイプ変換能力もリストされています。

注記

タグを書き込むときの特徴

WinCC で設定する場合、オートメーションシステムのデータ領域に 2 つ以上のタグを書き込む際に、全てのタグが個別に転送されることを確認します。

この動作は、スクリプトなどに「SetTagMultiWait」機能で複数のタグを書き込む場合に特に重要です。この機能は、それに転送されたすべてのタグの送信が完了したときのみ実行されるため、タグの数が多いと著しく長い待ち時間が発生することがあります。

タグの数が多い場合に「待機」機能の使用が必要かどうかを判断するための確認が必要です。この場合、特にデータが AS データ領域で連続的である場合、未処理データタグの使用も良い考えです。

タイプ変換

特定の値の範囲や、例えば 10 進から BCD(「符号なし 8 ビット値」から「ByteToBCDWord」への変換)へのような変換の場合、タイプ変換が必要です。デフォルトではタイプ変換は実行されません。

以下の表に、サポートされている WinCC データタイプとタイプ変換のそれぞれの機能を示します。

WinCC データタイプ	タイプ変換
2 進タグ	いいえ
符号付き 8 ビット値	S5 では利用できません
符号なし 8 ビット値	はい
符号付き 16 ビット値	はい
符号なし 16 ビット値	はい
符号付き 32 ビット値	はい
符号なし 32 ビット値	はい
浮動小数点数 32 ビット IEEE 754	はい
浮動小数点数 64 ビット IEEE 754	S5 では利用できません

WinCC データタイプ	タイプ変換
テキストタグ、8 ビットフォント	いいえ
テキストタグ、16 ビットフォント	S5 では利用できません
未処理データタイプ	いいえ

WinCC 側

通信ドライバ SIMATIC S5 Profibus FDL は、次の通信プロセッサを使用した通信をサポートしています。

通信プロセッサ	バスタイプ
CP 5613 A3	PCI
CP 5612	PCI

AS 側

プログラマブルロジックコントローラは、一般的に 2 つの異なる方法で PROFIBUS ネットワークに接続できます。

接続は、中央モジュールの統合インターフェース経由または特別な通信モジュールを使用して行われます。

システム	モジュール
S5-90U、S5-95U、S5-100U	CPU95U
S5-115U、S5-135U、S5-155U	CP5431 FMS/DP

注記

L2-SS による S5-95U への接続では、WinCC はパッシブパートナーにしかできないので、フェッチ接続はできません。

6.8 S5 PROFIBUS FDL

6.8.4 チャンネルの設定

6.8.4.1 チャンネル"SIMATIC S5 Profibus FDL"の設定方法

はじめに

チャンネル"SIMATIC S5 Profibus FDL"を設定するステップを、このセクションおよび以降のセクションで説明します。

このセクションに、チャンネル"SIMATIC S5 Profibus FDL"をどのように設定するかを示します。

手順

1. タグ管理のナビゲーションエリアで、[タグ管理]ノードのショートカットメニューから、[新規ドライバの追加]エントリを選択します。
2. "SIMATIC S5 Profibus FDL"ドライバを選択します。チャンネルが作成され、[タグ管理]に通信ドライバが表示されます。

6.8.4.2 チャンネルユニット"FDL (CP5412/A2-1)"

はじめに

「SIMATIC S5 Profibus FDL」通信ドライバには、「FDL (CP5412/A2-1)」チャンネルユニットのみが含まれています。

WinCC と SIMATIC S5 プログラマブルロジックコントローラ間の通信は、「FDL (CP5412/A2-1)」チャンネルユニットを介して行われます。

1 つのチャンネルユニットに、最大 24 の接続を作成できます。設定された接続ごとに特別な接続パラメータを設定する必要があります。それぞれの設定済みタグを、タグパラメータで定義する必要があります。

注記

「FDL (CP5412/A2-1)」チャンネルユニットの名前は、通信ドライバ「SIMATIC S5 Profibus FDL.CHN」にバインドされ、使用される通信プロセッサとは無関係です。

例えば、CP5613 A3 を通信プロセッサとして使用できます。

タグパラメータ

それぞれの設定済みタグに対して、以下のタグパラメータを指定する必要があります。

- データ領域(DB など)
- データブロック番号
- アドレス指定(例、"左バイト")
- 開始アドレス(例えば、アドレス指定に"左バイト"が選択されている場合、DL 0)

接続パラメータ

それぞれの設定済み接続に対して、以下の接続パラメータを指定する必要があります。

- AS のステーションアドレス
- 優先度
- 自分および外部の SAP(サービスアクセスポイント)が、読み取りおよび書き込み機能に対して指定されなくてはなりません。

接続をアクティブまたはパッシブ接続にするかどうかは、読み取り機能用にも設定する必要があります。アクティブ読み取り接続の場合、値は WinCC ステーションによって要求されます。パッシブ接続の場合には、WinCC ステーションへの値の転送が AS によって起動されます。

6.8.4.3 接続の設定方法

必要条件

- チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。

手順

1. チャンネルユニット"FDL (CP5412/A2-1)"を選択します。
2. チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。

4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。

The screenshot shows the 'Connection Parameters' dialog box with the following settings:

- Connection** tab selected.
- Profibus** section: **PLC Station Address** is 2.
- Priority** section: **Low** is selected.
- READ - Funktion** section: **OS active, WinCC is the active partner** is selected. **Own SAP** is 4, **Foreign SA** is 10.
- WRITE - Function** section: **Own SAP** is 5, **Foreign SA** is 11.
- A text box at the bottom says: 'Enter the station address of the connection partner (1 to 126)'.
- Buttons at the bottom: **OK**, **Cancel**, **Help**.

5. [PLC ステーションアドレス]フィールドで、AS の一意のアドレスを入力します。
6. [優先度]オプションボタンは、FDL 接続に対して必ず"低"に設定する必要があります。
7. WinCC ステーションのファンクションは、[WinCC がアクティブパートナーなら OS アクティブ]または[WinCC がパッシブパートナーなら OS パッシブ]フィールドで定義されます。必要なオプションボタンを有効にします。
8. [自分の SAP]および[他人の SAP]フィールドで、読み込みおよび書き出しアクセスに設定されている SAP アドレスを、入力します。SAP の値範囲は 2～54 です。
9. [OK]をクリックして、開いているダイアログをすべて閉じます。
10. 接続のショートカットメニューから[新規タグ]オプションを選択します。[タグのプロパティ]ダイアログが開きます。タグを設定します。
11. [OK]をクリックして、開いているダイアログをすべて閉じます。

6.8.4.4 タグの設定

タグの設定

はじめに

WinCC と AS の間の、チャンネル"SIMATIC S5 Profibus FDL"を経由した通信用に、データタイプ 2 進、バイト、ワードを、WinCC 内で定義できます。以下に、これらのデータタイプのタグの設定方法を説明します。

ビット単位でアクセスするタグの設定方法

はじめに

このセクションでは、AS のアドレスエリアにビット単位でアクセスするためのタグの設定方法を説明します。

注記


タグへのビット単位のアクセスは、読み取りアクセスだけです。

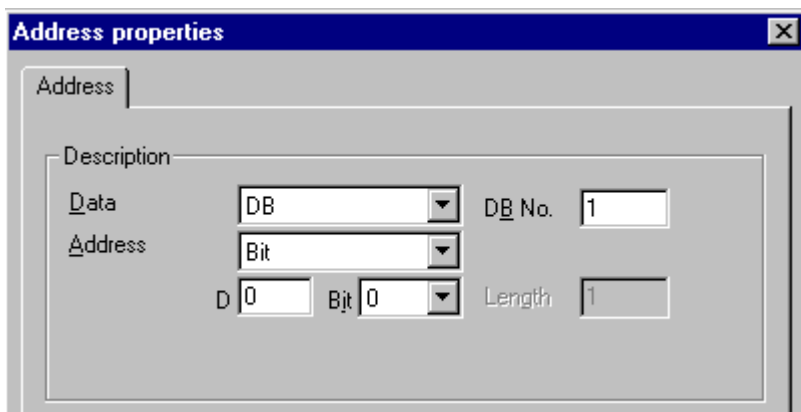
必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプとして[2 進タグ]を設定します。

5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. フィールド[D]にバイトアドレスを、フィールド[ビット]にビットアドレスを入力します。
この左側のフィールドのラベルは、[データエリア]フィールドのエントリによって異なります(例、データエリア[DB]およびタグのデータタイプ 2 進タグに対して、[D])。
7. [OK]をクリックして、開いているダイアログをすべて閉じます。

注記

[アドレス指定]フィールドの[ビット]エントリは、WinCC タグの 2 進タグデータタイプによって定義されているため、変更できません。

バイト単位でアクセスするタグの設定方法

はじめに

このセクションでは、AS のアドレスエリアにバイト単位でアクセスするためのタグの設定方法を説明します。


注記

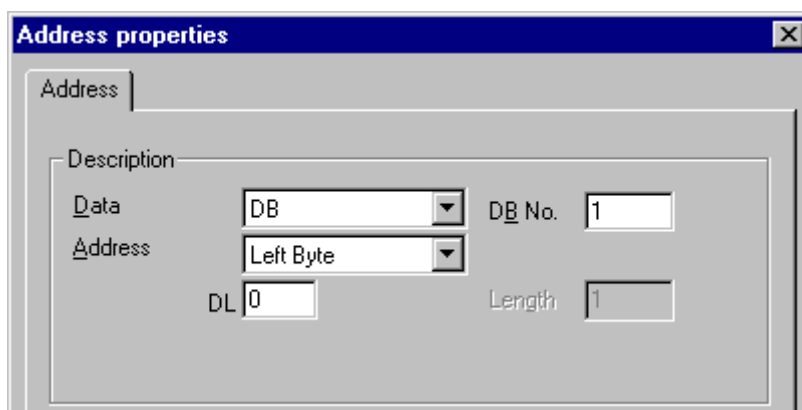
タグへのバイトでのアクセスは、読み取りアクセスだけです。

必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]に設定します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. [アドレス指定]フィールドで、[左バイト]と[右バイト]のいずれかを選択できます。
7. [DL]フィールドにバイトアドレスを入力します。このフィールドのラベルは、[アドレス指定]フィールドのエントリによって異なります(例、[左バイト]アドレス指定の場合は[DL])。
8. [OK]をクリックして、開いているダイアログをすべて閉じます。

ワード単位でアクセスするタグの設定方法

はじめに

このセクションでは、AS のアドレスエリアにワードでアクセスするためのタグの設定方法を説明します。


注記

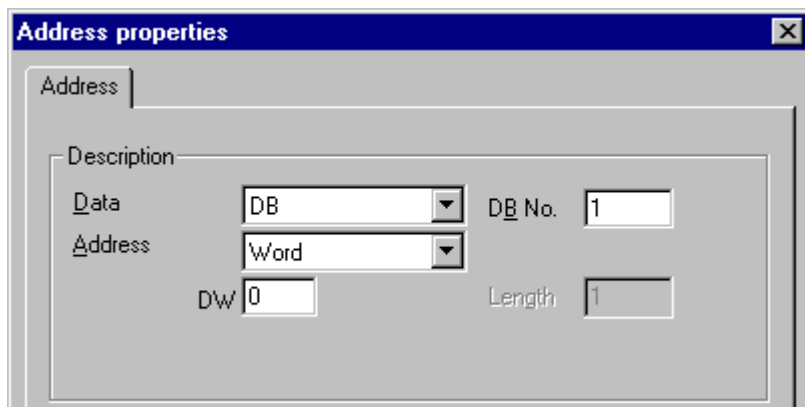
タグへのワード単位のアクセスは、読み取りおよび/または書き込みアクセスです。

必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプを[符号なし 16 ビット値]に設定します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. 下のフィールドにワードアドレスを入力します。このフィールドのラベルは、[アドレス指定]フィールドのエントリによって異なります(例:[ワード]アドレス指定の場合は[DW])。
7. [OK]をクリックして、開いているダイアログをすべて閉じます。

注記

[アドレス指定]フィールドの[ワード]エントリは、WinCC タグの[符号なし 16 ビット値]データタイプによって定義されているため、変更できません。

6.8.4.5 システムパラメータ

チャンネルユニットのシステムパラメータ

はじめに

WinCC の標準設定とは異なる設定が必要な場合は、チャンネルユニットの[システムパラメータ]ダイアログを使用して、必要な変更をすべて実行することができます。

以下の個別ポイントを変更できます。

- デバイス名
- 書き込み/読み取りモニタ時間

デバイス名

WinCC とオートメーションシステム間の通信は、論理デバイス名を介して実行されます。これらの名前は通信モジュールのインストール中に割り当てられ、ユニットに固有です。デバイス名は論理デバイス名を示します。このフィールドは、デフォルトではエントリ"/CP_L2_1:/SCP"で定義されます。

書き込み/読み取りモニタ時間

書き込み/読み取りモニタ時間は、AS の書き込み/読み取り応答に対する最長待ち時間を、秒単位で表したものです。定義された時間内に AS から応答が無い場合、接続が切断されます。このフィールドには、デフォルトでは待ち時間値 30 秒が割り当てられます。

注記

システムパラメータは、AS の全ての CP に適用されます。

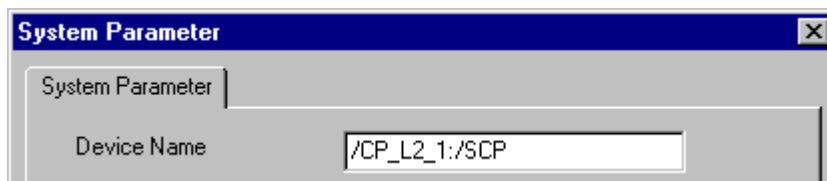
デバイス名の変更方法

必要条件

- チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。

手順

1. チャンネルユニットのショートカットメニューで、[システムパラメータ]を選択します。[システムパラメータ]ダイアログが開きます。



2. [デバイス名]フィールドに、アクセスポイントの名前を入力します。この名前は、Windows の[スタート] → [設定] → [コントロールパネル] → [PG/PC インターフェースの設定]で行なった設定と、一致しなければなりません。
3. [OK]ボタンをクリックして、ダイアログを閉じます。

注記

WinCC を再起動しないと変更は有効になりません。

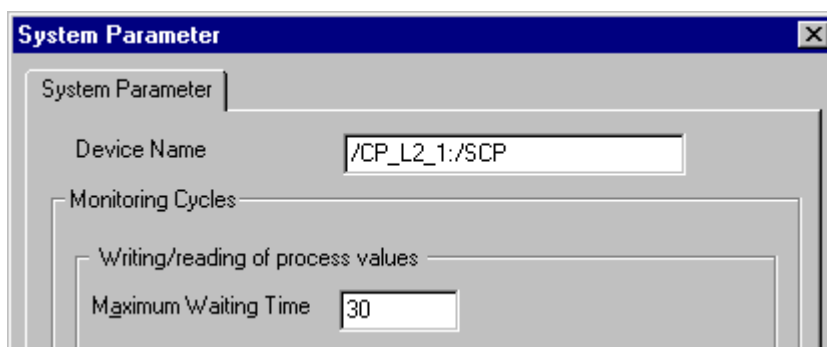
プロセス値の書き込み/読み取りモニタ時間の変更方法

必要条件

- チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。

手順

1. チャンネルユニットのショートカットメニューで、[システムパラメータ]を選択します。[システムパラメータ]ダイアログが開きます。



2. [最長待ち時間]フィールドに、必要な値を秒単位で入力します。1~3600 秒の範囲で値を定義できます。このフィールドには、デフォルト値 30 秒が割り当てられます。
3. [OK]ボタンをクリックして、ダイアログを閉じます。

注記

WinCC を再起動しないと変更は有効になりません。

6.8.5 特殊ファンクション

6.8.5.1 "SIMATIC S5 Profibus FDL"チャンネルの特殊ファンクション

はじめに

"SIMATIC S5 Profibus FDL"チャンネルには特殊ファンクションがあり、その機能について、この章で説明します。

6.8.5.2 [SIMATIC S5 Profibus FDL]チャンネルの未処理データタグ

[SIMATIC S5 Profibus FDL]チャンネルの未処理データタグ

はじめに

タイプ[未処理データタイプ]のタグは、データテレグラムです。

未処理データタグは、AS との間のユーザーデータブロックの転送に必要です。

SIMATIC S5 Profibus FDL が使用する未処理データタグは、最長で 220 バイトの長さです。

バイト配列としての未処理データタグ

バイト配列としての未処理データタグは、データブロックのアドレスおよび長さ(例、DB100、DW20、長さ 40 バイト)を使用してアドレス指定される普通のプロセスタグと同じように、扱われます。

スクリプトを使用した未処理データタグの書き込み

WinCC で設定されたタグ長さよりも長い未処理データタグが VB スクリプトを使用して S5 に書き込まれると、書き込みプロセスが中断します。


その代わりに、[SetTagRaw]ファンクションを使用して、C スクリプト経由で未処理データタグを書き込みます。このファンクションでは、書き込まれるタグの長さを指定します。

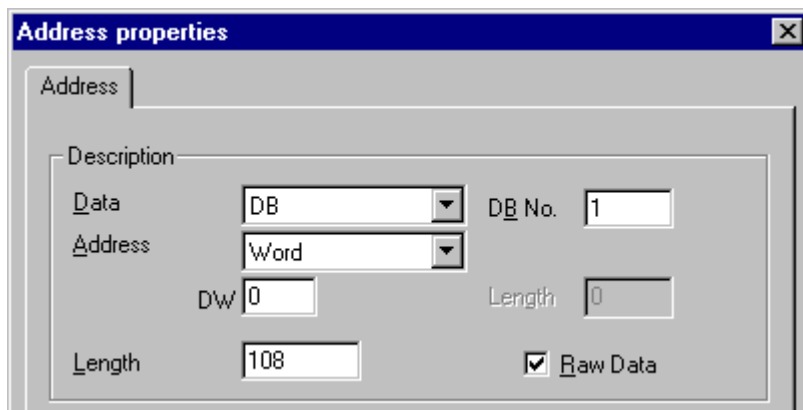
未処理データタグの設定方法

必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、[未処理データタイプ]を選択します。
5. [選択]ボタンをクリックして、[アドレスプロパティ]ダイアログを開きます。
[アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、ボタンをクリックしてください。



6. [未処理データ]チェックボックスをマークします。
7. [長さ]フィールドに、未処理データブロックの長さを入力します(単位はバイト)。
8. [データエリア]で、データが位置する PLC のデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。
9. [アドレス指定]フィールドで、アドレス指定のタイプを設定します。WinCC タグのデータタイプ[未処理データタイプ]では、[左バイト]、[右バイト]、[ワード]、[ダブルワード]の入力が可能です。

10. 下のフィールドには、開始アドレスの値を入力します。この左側のフィールドのラベルは、[データエリア]および[アドレス指定]フィールド内のエントリによって異なります(例、アドレス指定タイプのデータエリアが[DB]、[ワード]の場合は[DW])。
11. [OK]をクリックして、開いているダイアログをすべて閉じます。

6.8.5.3 通信タイプの設定

通信タイプの設定

はじめに

FDL 接続は、WinCC がアクティブパートナまたはパッシブパートナとして実行されるように、設定できます。

WinCC をアクティブパートナとして設定した場合、値は WinCC ステーションによって要求されます。

WinCC をパッシブパートナとして設定した場合、WinCC ステーションへの値の転送は、AS によって起動されます。

アクティブなデータ転送の設定方法

はじめに

このセクションでは、AS のアドレスエリアへのアクティブなデータ転送の設定方法を、示します。

注記

2 つ以上の接続を設定する場合、SAP は一度だけ割り付けできることに注意してください。

必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。
3. AS で定義された SAP アドレスを知っている必要があります。

手順

1. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。

The screenshot shows the 'Connection Parameters' dialog box with the following settings:

- Connection** tab selected.
- Profibus**: PLC Station Address is 2.
- Priority**: Low is selected.
- READ - Funktion**:
 - ☒ OS active, WinCC is the active partner
 - ☐ OS passive, WinCC is the passive partner
 - Own SAP: 4
 - Foreign SA: 10
- WRITE - Funktion**:
 - Own SAP: 5
 - Foreign SA: 11
- Text box: Enter the station address of the connection partner (1 to 126)
- Buttons: OK, Cancel, Help

2. [接続]タブの[PLC ステーションアドレス]フィールドに、AS のステーションアドレスを入力します。
3. 以下のように[読み取り - ファンクション]エリアに設定します。
4. [WinCC がアクティブパートナーなら OS アクティブ]オプションを有効にします。
5. [自分の SAP]フィールドに WinCC ステーションの SAP-ID を入力します。
6. [他人の SAP]フィールドに AS の SAP-ID を入力します。
7. 以下のように[書き込み - ファンクション]エリアに設定します。
8. [自分の SAP]フィールドに WinCC ステーションの SAP-ID を入力します。
9. [他人の SAP]フィールドに AS の SAP-ID を入力します。
10. [OK]をクリックして、開いているダイアログをすべて閉じます。

パッシブなデータ転送の設定方法

はじめに

このセクションでは、AS のアドレスエリアへのパッシブなデータ転送の設定方法を、示します。

注記

2 つ以上の接続を設定する場合、SAP は一度だけ割り付けできることに注意してください。

必要条件

1. チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。
2. チャンネルユニット"FDL (CP5412/A2-1)"で接続を定義する必要があります。
3. AS で定義された SAP アドレスを知っている必要があります。

手順

1. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。

The screenshot shows the 'Connection Parameters' dialog box with the following settings:

- Connection** tab selected.
- Profibus**:
 - PLC Station Address: 2
- Priority**:
 - Low (selected)
- READ - Funktion**:
 - OS passive, WinCC is the passive partner (selected)
 - Own SAP: 4
 - Foreign SA: 10
- WRITE - Funktion**:
 - Own SAP: 5
 - Foreign SA: 11
- Specify the SAP-ID of the connection partner (2...54)
- Buttons: OK, Cancel, Help

2. [接続]タブの[PLC ステーションアドレス]フィールドに、AS のステーションアドレスを入力します。
3. 以下のように[読み取り - ファンクション]エリアに設定します。
4. [WinCC がパッシブパートナーなら OS パッシブ]オプションを有効にします。
5. [自分の SAP]フィールドに WinCC ステーションの SAP-ID を入力します。
6. [他人の SAP]フィールドに AS の SAP-ID を入力します。
7. 以下のように[書き込み - ファンクション]エリアに設定します。
8. [自分の SAP]フィールドに WinCC ステーションの SAP-ID を入力します。
9. [他人の SAP]フィールドに AS の SAP-ID を入力します。
10. [OK]をクリックして、開いているダイアログをすべて閉じます。

6.8.6 "SIMATIC S5 Profibus FDL"チャンネルの設定例

6.8.6.1 "SIMATIC S5 Profibus FDL"チャンネルの設定例

はじめに

この例では、グラフィックデザイナーの I/O フィールドを設定し、AS のデータ処理ブロックに必要な値を割り当てます。

6.8.6.2 AS のデータ処理ブロックの設定方法

はじめに

このセクションでは、AS の標準ファンクションブロック OB 21 (L2ANLAUF)および OB 1 (L2SNDRCV)を、設定します。

デフォルトでは、PROFIBUS FDL による SIMATIC S5 接続のデータトラフィックは、以下のブロックを使用して処理されます。

この例の中で、以下の SAP 番号が使用されます。

	WinCC	プログラマブルコントローラ
読み取り機能の SAP 番号	12	6
書き込み機能の SAP 番号	11	4

ブロック

機能	ブロック
スタートアップ OB 20、21、22 用	FB-L2ANLAUF (FB 9)
サイクリック操作 OB 1 用	FB-L2SNDRCV (FB 10)
両方の FB の内部作業データブロックとして	DB-L2DBVC3 (DB 10) DB-L2DBVC4 (DB 11) DB-L2DBVC5 (DB 12)

スタートアップブロック

起動ブロックでは、通信パラメータが指定され、作業 DB が登録され、通信プロセッサが同期されます。

6.8 S5 PROFIBUS FDL

例えば、これらの作業ステップは、呼び出しファンクションブロック FB9 L2ANLAUF によって実行されます。

サイクリックブロック

フレームトラフィックは、サイクリック FB で処理されます。

受信フレームは、宛先データブロックに入力されます。この実行中にエラーが発生すると、フレームは拒否され、エラーメッセージが発行されます。

ユーザーは、標準データ処理ブロックと同様の方法で、送信されるフレームを指定します。

転送が完了した後にフィードバックが発生します。

必要条件

- データ処理ブロック SYNCHRON、CONTROL、SEND、RECEIVE が、AS で使用可能でなければなりません。

手順

1. スタートアップブロック(OB 20、21、22)は STEP5 ソフトウェアで、メニュー項目[エディタ] → [STEP5 ブロック] → [プログラムファイル中]を使用することによって、作成されます。
例の中で、プログラムブロックの名前は「L2ANLAUF」です。
2. 以下のパラメータが、事前に割り付けられている必要があります。
 - CP (例えば CP5431)のインターフェース番号(SSNR)
 - WinCC コンピュータの通信プロセッサ CP 5613 A3 の PROFIBUS アドレス(RADR)
この番号は、ネットワークで一意であることが必要です。
 - 例えば、WinCC ステーションの SAP を指定する RVC4(書き込み用)と RVC5(読み取り用)パラメータなど、要求タイプを使用する接続パラメータ。これらの SAP は、WinCC で接続が作成されたときに指定されます。
 - 通信プロセッサ用の FDL 接続の設定時に設定されるリクエスト番号(ANR4 と ANR5)
 - 作業データブロック DBX4(書き込み用)および DBX5(読み取り用)の番号

```

OB 21
NETZWERK 1 von 1          Synchronisieren CP 5431
:
:SPA FB 9
NAME :L2ANLAUF
SSNR : KF +0              SSNR of CP 5431
TIM3 : KT 000.0          nr
TIM7 : KT 000.0          nr
RADR : KF +8             PROFIBUS-Address WinCC Station
RUC3 : KF +0             nr
RUC4 : KF +4             SAP WRITE
RUC5 : KF +6             SAP READ
RUC6 : KF +0             nr
RUC7 : KF +0             nr
ANR3 : KF +0             nr
ANR4 : KF +134           ANR WRITE
ANR5 : KF +135           ANR READ
ANR6 : KF +0             nr
ANR7 : KF +0             nr
DBX3 : KY 000,000        nr
DBX4 : KY 000,011        Work-DB WRITE
DBX5 : KY 000,012        Work-DB READ
DBX6 : KY 000,000        nr
DBX7 : KY 000,000        nr
S/R3 : KF +0             nr
:
:BE

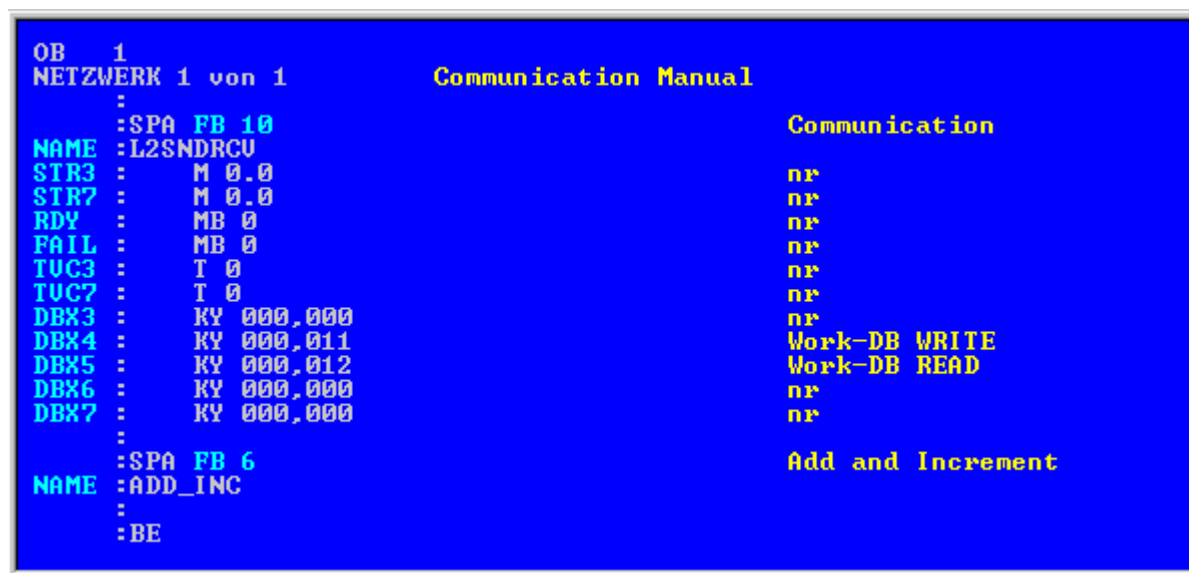
```

3. OB 1(サイクリック操作)は STEP5 ソフトウェアで、メニュー項目[エディタ] → [STEP5 ブロック] → [プログラムファイル中]を使用することによって、作成されます。
例の中で、プログラムブロックの名前は「L2SNDRCV」です。

6.8 S5 PROFIBUS FDL

4. WinCC との通信は、例えば通信プロセッサ CP5431 や機能ブロック FB10 L2SNDRCV を使用して実行されます。

WinCC がデータを送信および要求する場合は、この目的で 2 つの関連する入出力パラメータのみを指定する必要があります。これらは、使用される要求タイプの 2 つの作業データブロックの番号を指定するパラメータ DBX4(書き込み用)と DBX5(読み取り用)です。これらの SAP は、WinCC で接続が作成されたときに指定されます。



5. STEP 5 プログラムをプログラマブルロジックコントローラにダウンロードします。これは STEP 5 ソフトウェアで、メニュー項目[オブジェクト] → [ブロック] → [転送] → [PLC ファイル]を使用することによって、実行されます。以前作成されたブロックのすべてをオートメーションシステムにダウンロードするためには、[選択]フィールドで[すべてのブロック]オプションを選択します。

6.8.6.3 I/O フィールドの設定方法


はじめに

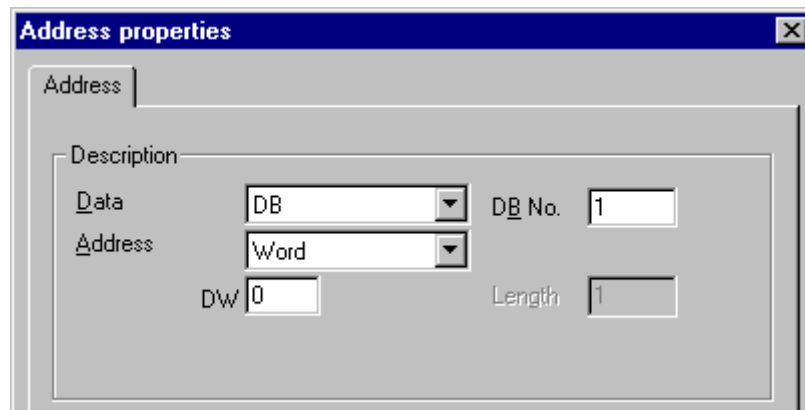
このセクションでは、I/O フィールドを設定します。

必要条件

- チャンネル"SIMATIC S5 Profibus FDL"をプロジェクトに統合する必要があります。

手順

1. チャンネルユニット"FDL (CP5412/A2-1)"のショートカットメニューから[新規接続]オプションを選択し、"TestFDL"と呼ばれる接続を設定します。
2. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。
AS のステーションアドレスを、[PLC ステーションアドレス]フィールドに入力します。
WinCC がアクティブパートナーあるいはパッシブパートナーになるように、FDL 接続を構成できます。WinCC をアクティブパートナーとして設定した場合、値は WinCC ステーションによって要求されます。WinCC をパッシブピアと設定した場合、WinCC ステーションへの値の転送は AS によって起動されます。
[OK]をクリックして、開いているダイアログをすべて閉じます。
3. テーブルエリアの下にある[タグ]タブをクリックします。
4. [名前]列の先頭の空きセルをクリックします。
"FDLWord1_Test"を名前として入力します。
5. [データタイプ]フィールドで、データタイプを[符号なし 16 ビット値]に設定します。
6. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



7. [データエリア]で、データが位置する PLC のデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。[DW]フィールドにバイトアドレスを入力します。
8. [OK]をクリックして、開いているダイアログをすべて閉じます。
9. スマートオブジェクト"I/O フィールド"を、グラフィックデザイナーの画像に統合する必要があります。
10. [I/O フィールド]を設定済タグに接続します。
11. ツールバーにある[ランタイムを有効にする]ボタンをクリックするか、または[ファイル]メニューから[ランタイムを有効にする]を選択して、プロジェクトを有効にします。設定済タグへの全ての変更が、ランタイムの[I/O フィールド]に表示されます。

6.9 S5 Programmers Port AS511

6.9.1 WinCC チャンネル"SIMATIC S5 Programmers Port AS5111"

はじめに

通信ドライバ"SIMATIC S5 Programmers Port AS511"は、SIMATIC S5 オートメーションシステムへの、TTY インターフェース経由のシリアル接続で活用されます。

この章の内容

- "SIMATIC S5 Programmers Port AS5111"チャンネルによるデータ転送の設定方法。
- 接続およびタグの設定方法。

チャンネルユニット

通信ドライバには 1 つのチャンネルユニットがあり、シリアル接続用に COM ポートを制御します。

以下の機能を使用できます。

- "Siemens 固有"プロトコル経由のシリアル通信用チャンネルユニット S5-AS511

6.9.2 タグのデータタイプ

はじめに

論理接続に必要なタグを定義します。WinCC の観点から、以下のデータタイプにアクセスできます。

- 2 進タグ
- 符号なし 8 ビット値
- 符号付き 8 ビット値
- 符号なし 16 ビット値
- 符号付き 16 ビット値
- 符号なし 32 ビット値

- 符号付き 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- テキストタグ、8 ビット文字セット
- 未処理データタイプ

6.9.3 チャンネルの設定

6.9.3.1 "SIMATIC S5 Programmers Port AS511"チャンネルの設定

はじめに

チャンネル"SIMATIC S5 Programmers Port AS511"の設定には、以下のステップが必要です。

- 接続の設定
- タグの設定

6.9.3.2 接続の設定方法

はじめに

シリアル接続を使用したプロセス接続は、SIMATIC S5 オートメーションシステムで使用可能です。AS 511 通信プロセッサは、オートメーションシステムで使用します。

WinCC では、追加の通信モジュールは不要です。通信は、PG 760 の TTY ポートあるいはシステムの標準装置の一部である COM ポート経由で、設定されます。後者の場合、追加のポートコンバータ V.24/V.28 <---> TTY が必要です。

このシリアルリンクは、最大伝送レート 19200 ボーまでをサポートします。

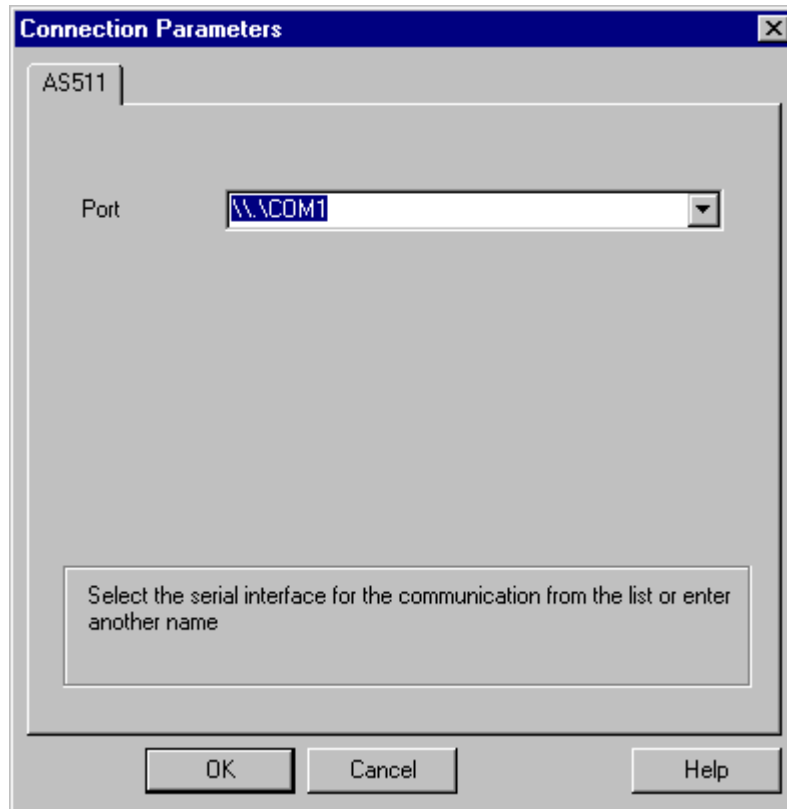
PC のシリアルポートの 1 つを AS511-NT ドライバに割り付けるには、以下の手順を使えます。

注記

"AS511"チャンネル経由の WinCC と S5 オートメーションシステムの間の通信中は、AS でデータブロックを転送、作成あるいは削除できません。S5 のメモリも圧縮できません。この最後の制限は、S5 におけるメモリの絶対アドレス指定の結果です。変更が必要な場合は、WinCC への接続を切断しなければなりません。

手順

1. チャンネルユニット"S5-AS511"の下で接続を選択します。
2. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。



3. [ポート]フィールドで、シリアルリンクに使用するポートを選択します。

6.9.3.3 タグの設定

タグの設定

はじめに

チャンネル"SIMATIC S5 Programmers Port AS511"経由での WinCC と AS の間の接続では、WinCC で異なるデータタイプのタグを作成できます。これは以下のセクションで説明されています。

注記

WinCC では、タグアドレスの正当性を確認しません。AS で使用できないアドレスを使用する場合、ステータス[アドレス指定エラー]が設定されます。

DB および DX データブロックでは、読み取りと書き込みができるのは、アドレス 255 までだけです。

回数は書き込めません。

タグのアドレスの設定方法


はじめに

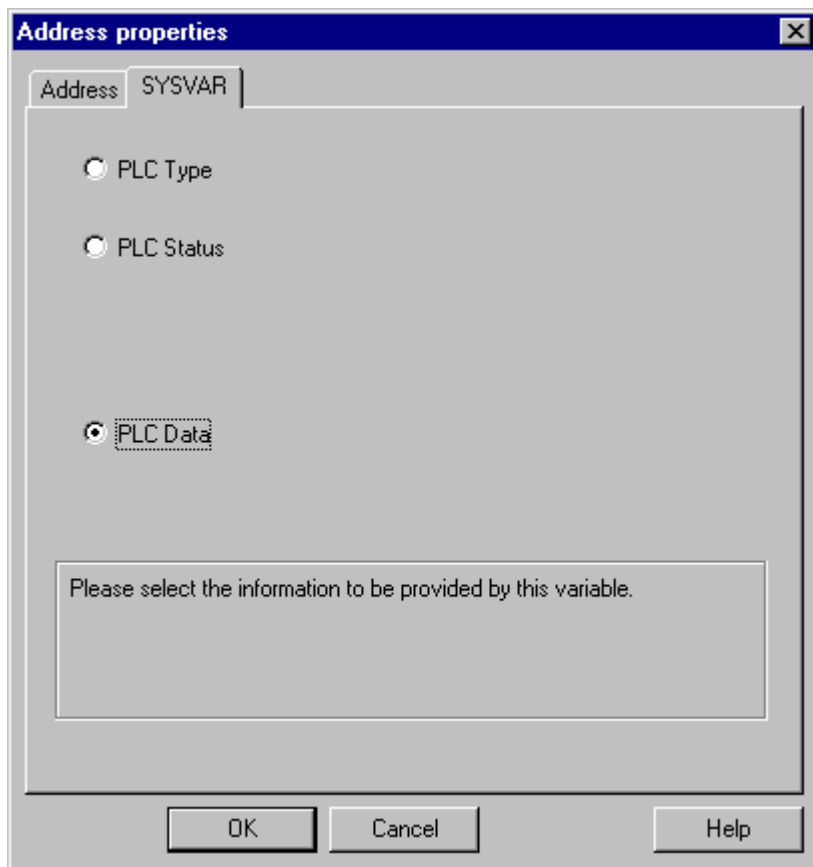
SIMATIC S5 のアドレス構造に準じてタグアドレスを入力します。

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックしてから、タグを選択します。
3. [全般]タブからタグに必要なデータタイプ(符号付き 8 ビット値など)を選択します。

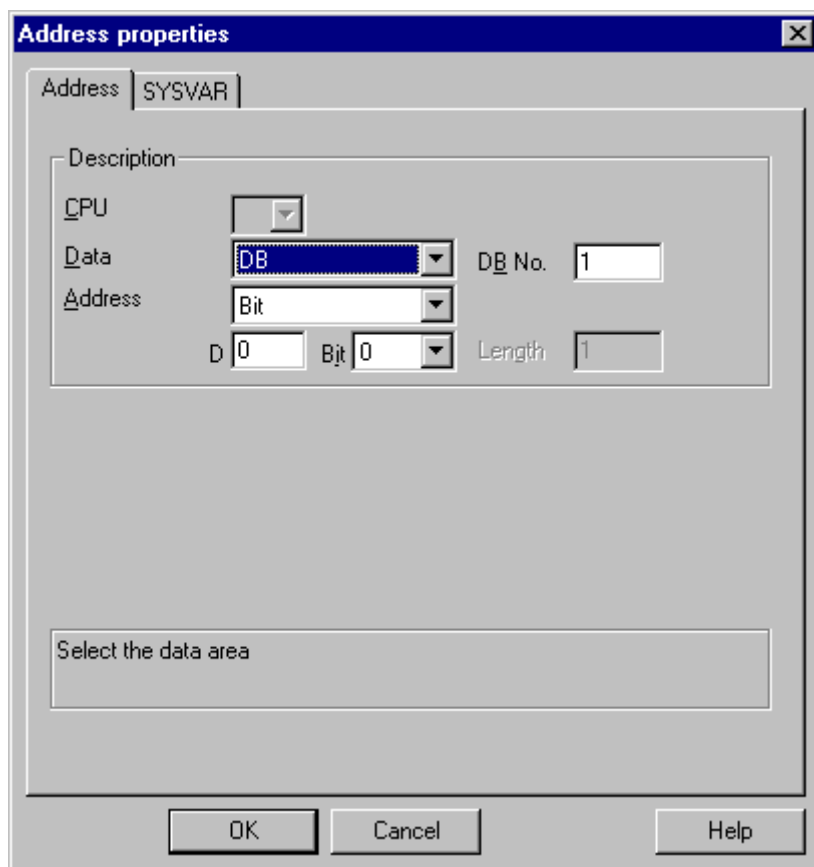
6.9 S5 Programmers Port AS511

4. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
[SYSVAR]タブを選択します。



5. [選択]フィールドをクリックして、タグが"PLC タイプ"、現在のステータス("PLC ステータス")あるいは他のデータ("PLC データ")を転送するかどうかを、選択します。

6. "PLC データ"を選択した場合のみ、[アドレス]タブをクリックしてタグの S5 アドレスを定義する必要があります。



7. タグが、[データエリア]フィールド内でデータブロック、拡張データブロック、フラグエリア、入力範囲または出力範囲に位置するかどうかを選択します。
8. タグがデータブロックにある場合、[DB 番号]フィールドも表示されます。ここで、データブロック番号を入力します。
9. [アドレス指定]フィールドで、アドレス指定のタイプを入力します。通常、デフォルトの定義を使用できます。
10. 個別フィールド(例、[DW])にアドレスを入力します。

多くの場合、PLC のメモリへのアクセスは、バイト単位またはワード単位でしか実行できません。2 進および[8 ビット]タグを使用すると、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、PLC のメモリ内の個別ビットの変更にも使用できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリ

アが PLC から読み取られ、対応するビットおよび/またはバイトが修正されます。その後、このデータが PLC のメモリに書き戻されます。

注記

読み取りデータエリアで PLC に行なわれた変更は、データエリアに書き戻された時点で上書きされます。

タグのタイプによって、オートメーションシステムのメモリにビット単位かバイト単位でアクセスできます。

WinCC では、タグアドレスの正当性を確認しません。AS で使用できないアドレスを使用する場合、ステータス[アドレス指定エラー]が設定されます。


DB および DX データブロックでは、読み取りと書き込みができるのは、アドレス 255 までだけです。

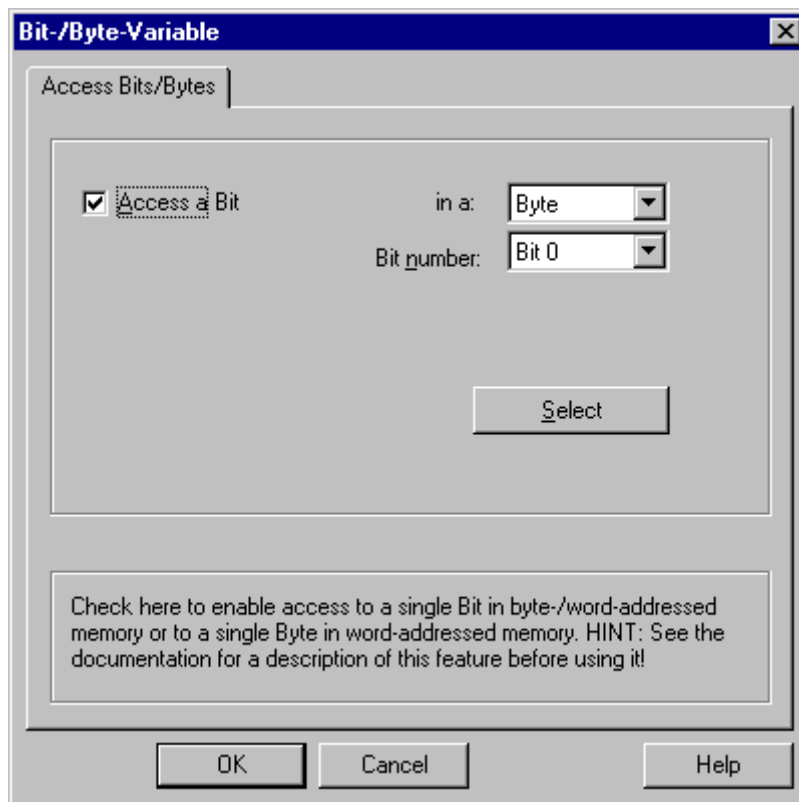
回数は書き込めません。

ビットでアクセスするタグの設定方法

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプとして[2 進タグ]を設定します。

5. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。




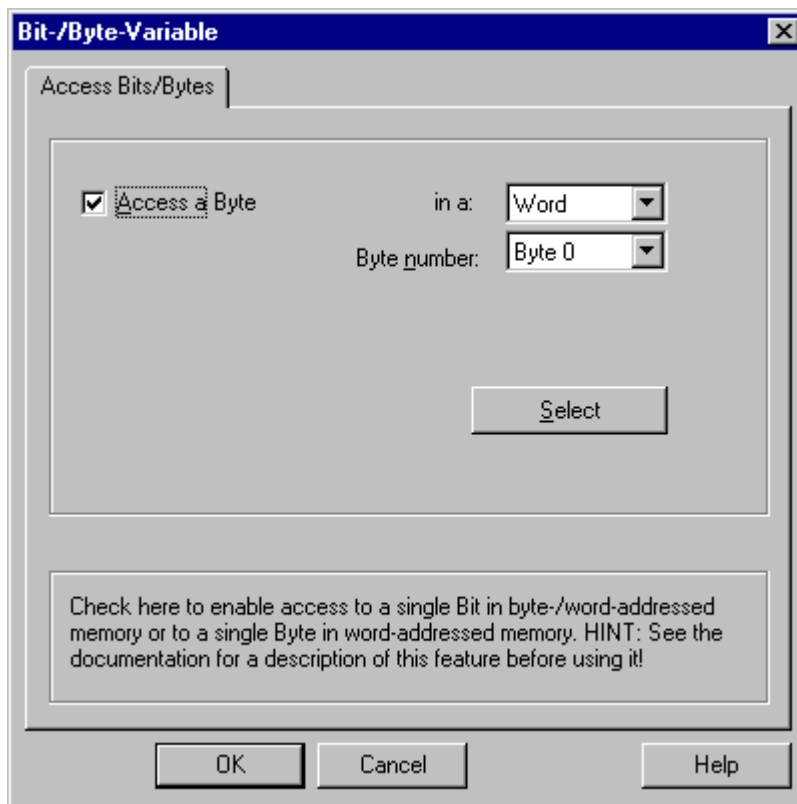
6. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
7. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します。
8. [選択]フィールドで変更するビット数を選択します。

バイト単位でアクセスするタグの設定方法

手順

1. タグを構成する接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]または[符号付き 8 ビット値]に設定します。

5. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



6. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
7. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します。
8. [選択]フィールドで変更するバイト数を選択します。

6.10 S5 Serial 3964R

6.10.1 WinCC チャンネル"SIMATIC S5 Serial 3964R"

はじめに

通信ドライバ"SIMATIC S5 Serial 3964R"は、WinCC ステーションと SIMATIC S5 オートメーションシステムのためのシリアルリンクのために、実装されます。

この章の内容

- "SIMATIC S5 Serial 3964R"チャンネルによるデータ転送の設定方法。
- 接続およびタグの設定方法。

チャンネルユニット

通信ドライバには 1 つのチャンネルユニットがあり、シリアルリンク用に COM ポートを制御します。

以下の機能を使用できます。

- 3964R または 3964 プロトコルを使用したシリアル通信用のチャンネルユニット S5-RK512 (3964R)。

6.10.2 タグのデータタイプ

はじめに

論理接続に必要なタグを定義します。WinCC の観点から、以下のデータタイプにアクセスできます。

- 2 進タグ
- 符号なし 8 ビット値
- 符号付き 8 ビット値
- 符号なし 16 ビット値
- 符号付き 16 ビット値
- 符号なし 32 ビット値

6.10 S5 Serial 3964R

- 符号付き 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- テキストタグ、8 ビット文字セット
- 未処理データタイプ

6.10.3 チャンネルの設定

6.10.3.1 "SIMATIC S5 Serial 3964R"チャンネルの設定

はじめに

チャンネル"SIMATIC S5 Serial 3964R"の設定には、以下のステップが必要です。

6.10.3.2 接続の設定方法

はじめに

シリアル接続を使用したプロセス接続は、SIMATIC S5 オートメーションシステムで使用可能です。オートメーションシステムでは、通信プロセッサ CP 544 または 2 番目のプラグインのシリアルポートが、CPU モジュール(モジュールレセプタクル SI2)で使用されます。

WinCC では、追加の通信モジュールは不要です。通信は、システムで利用できるデフォルトの COM ポートによって行なわれます。

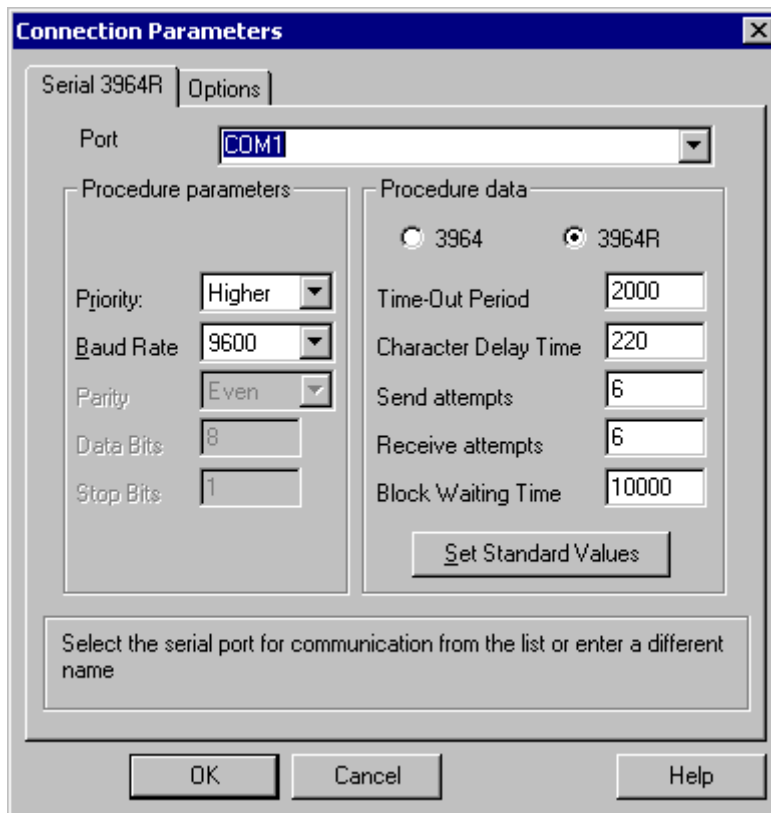
このシリアルリンクは、最大伝送レート 19200 ボーまでをサポートします。

注記

SIMATIC S5 がジョブタイプ"Pseudowrite"でアクティブに送信している場合、メッセージの長さが 64 ワードを超えてはいけません。

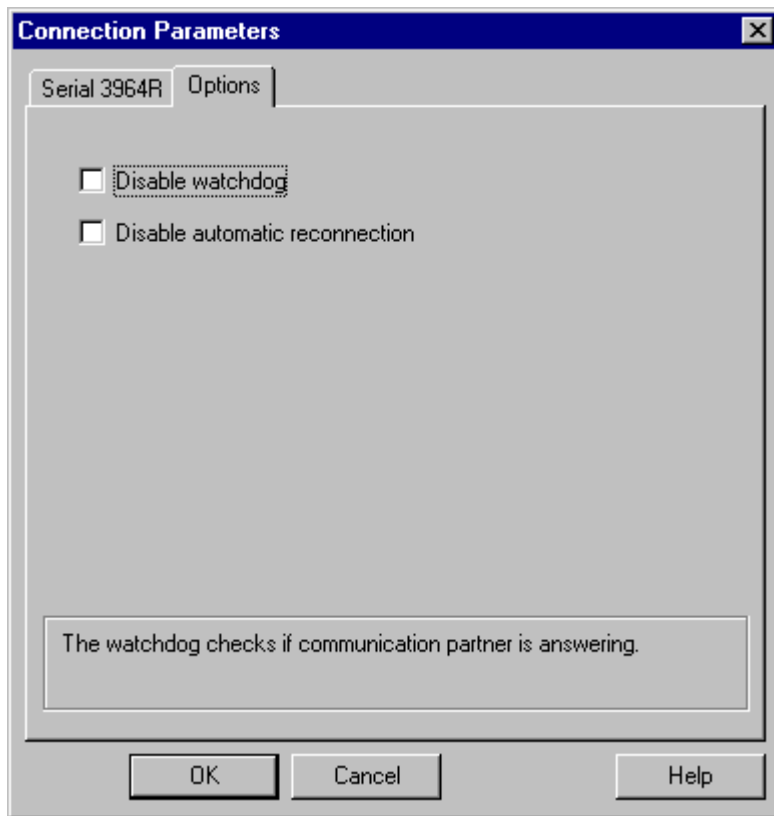
手順

1. 接続を選択し、ショートカットメニューから[接続パラメータ]を選択します。
2. [Serial 3964R]タブを選択します。



3. [ポート]フィールドで、接続用の通信ポート(COM1 または COM2)を選択します。
4. データ転送速度を、[手順パラメータ]エリアの[ボーレート]フィールドで使用される値に、設定します。起動時に矛盾(WinCC とオートメーションシステムによるラインの同時ビッド)がある場合の優先度を、[優先度]フィールドに設定します。
優先度の設定は、SIMATIC S5 で設定されている優先度と異なっていることが必要です。
5. [手順データ]エリアで、"3964"または"3964R"のどちらかのラインプロトコルを選択します。
例外的な場合に、手順データ(確認時間、文字遅延時間など)のデフォルト値だけを変更します。オートメーションシステムのパラメータと一致することを確認します。

6. ここで[オプション]タブを選択します。



7. サイクリックなライフビートのモニタを無効にし、[オプション]タブで自動再接続を無効にすることができます。

6.10.3.3 タグの設定

タグの設定

はじめに


チャンネル"SIMATIC S5 3964R"経由の WinCC と AS の間の接続では、データタイプ 2 進およびバイトを、WinCC 内で定義できます。以下に、これらのデータタイプのタグの設定方法を説明します。

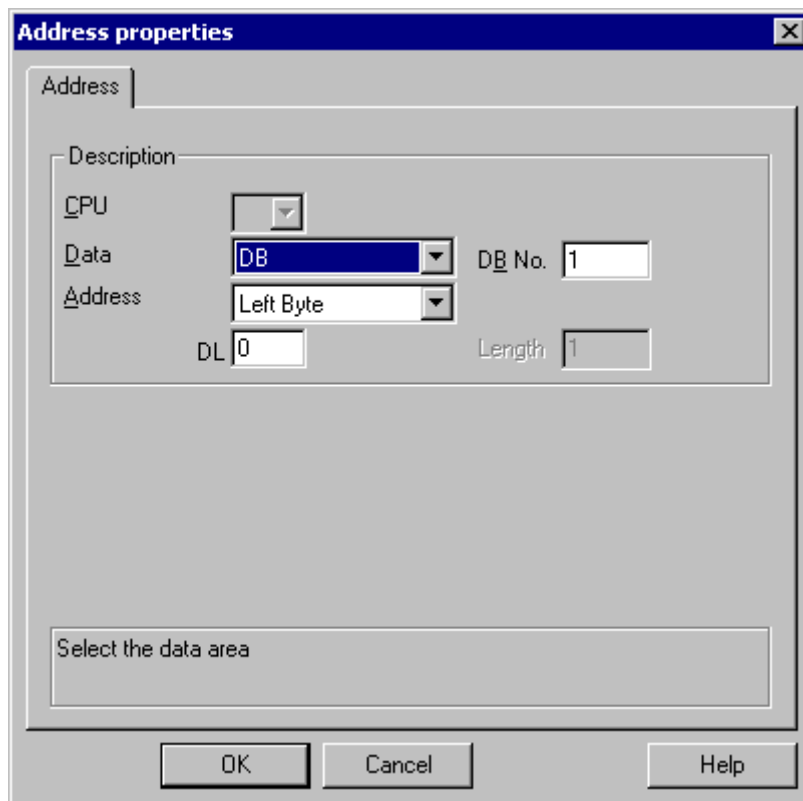
タグのアドレスの設定方法

はじめに

SIMATIC S5 のアドレス構造に準じてタグアドレスを入力します。

手順

1. タグを選択してから、[データタイプ]フィールドで、タグに必要なデータタイプ(符号付き 8 ビット値など)を設定します。
2. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
[アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



3. タグが、[データエリア]フィールド内でデータブロック、拡張データブロック、フラグエリア、入力範囲または出力範囲に位置するかどうかを選択します。
4. タグがデータブロックにある場合、[DB 番号]フィールドも表示されます。ここで、データブロック番号を入力します。
5. [アドレス指定]フィールドで、アドレス指定のタイプを入力します。通常、デフォルトの定義を使用できます。
6. 個別フィールド(例えば[DL])にアドレスを入力します。

注記

入力、出力、タイマ、カウンタのアドレスエリアへの、読み取りアクセスのみが可能です。データブロック(DB、DX)への読み取りおよび書き込みアクセスが可能です。

255 より大きなデータワードアドレスは使用しないでください。RK512 のシステム特性から、0～255 のデータワードアドレスだけが許可されます。

これより大きいアドレスを構成することは可能ですが、接続で構成された全てのタグでデータの破損が発生します。

多くの場合、PLC のメモリへのアクセスは、バイト単位またはワード単位でしか実行できません。2 進および[8 ビット]タグを使用すると、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、PLC のメモリ内の個別ビットの変更にも使用できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが PLC から読み取られ、対応するビットおよび/またはバイトが修正されます。その後、このデータが PLC のメモリに書き戻されます。


注記

読み取りデータエリアで PLC に行なわれた変更は、データエリアに書き戻された時点で上書きされます。

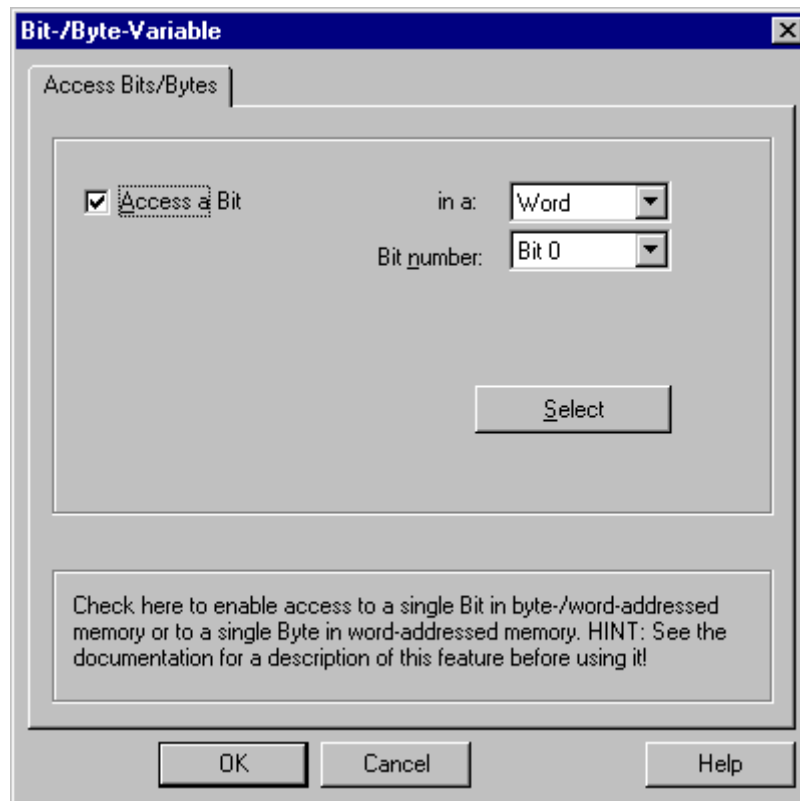
タグのタイプによって、オートメーションシステムのメモリにビット単位かバイト単位でアクセスできます。

ビットでアクセスするタグの設定方法

手順

1. タグを選択してから、[データタイプ]フィールドで、データタイプを[2 進タグ]に設定します。
2. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
3. [選択]ボタンをクリックします。[ビットタグ/バイトタグ]ダイアログが開きます。

4. [ビットのアクセス]チェックボックスを選択して、ビットアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します。
7. [選択]フィールドで変更するビット数を選択します。


注記

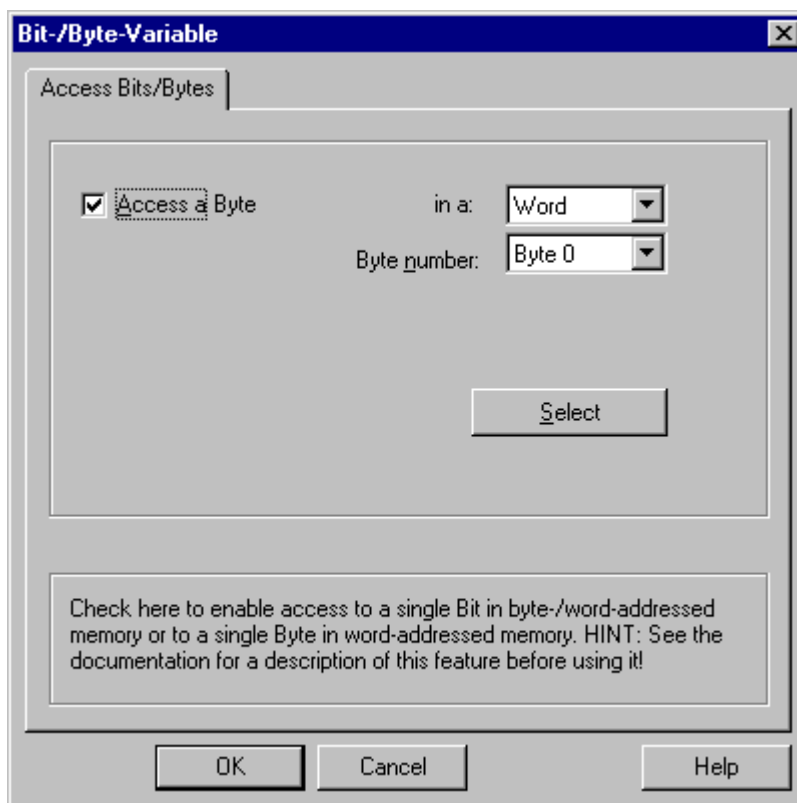
S5 では、フラグ、入力、出力はバイト単位でアドレス指定され、データブロック (DB、DX) はワード単位でアドレス指定されます。

入力、出力、タイマ、カウンタのアドレスエリアへは、読み取りアクセスのみが可能です。データブロック (DB、DX) への読み取りおよび書き込みアクセスが可能です。

バイト単位でアクセスするタグの設定方法

手順

1. タグを選択してから、[データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]または[符号付き 8 ビット値]に設定します。
2. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
3. [選択]ボタンをクリックします。[ビットタグ/バイトタグ]ダイアログが開きます。
4. [バイトのアクセス]チェックボックスを選択して、バイトアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します。
7. [選択]フィールドで変更するバイト数を選択します。

注記

S5 では、フラグ、入力、出力はバイト単位でアドレス指定され、データブロック (DB、DX) はワード単位でアドレス指定されます。

入力、出力、タイマ、カウンタのアドレスエリアへは、読み取りアクセスのみが可能です。データブロック (DB、DX) への読み取りおよび書き込みアクセスが可能です。

6.11 SIMATIC S7 Protocol Suite

6.11.1 WinCC チャンネル"SIMATIC S7 Protocol Suite"

はじめに

"SIMATIC S7 Protocol Suite"チャンネルは、WinCC ステーションと SIMATIC S7 オートメーションシステム間の通信をサポートします。このスイートは、各種のネットワークプロトコルおよびネットワークタイプをサポートします。

このセクションでは、以下の方法について説明します。

- チャンネルに関する各種の接続およびタグを設定する
- サンプルプロジェクトを作成する
- AR_SEND 機能、未処理データタグ、ソフトウェアの冗長化など、このチャンネル専用の機能を使用する

動作の原則

チャンネル"SIMATIC S7 Protocol Suite"は、SIMATIC S7-300 と SIMATIC S7-400 オートメーションシステムをリンクするために、使用されます。

使用する通信ハードウェアに応じて、システムでは以下のチャンネルユニット経由の通信がサポートされます。

- Industrial Ethernet および Industrial Ethernet (II):SIMATIC NET Industrial Ethernet を使用する通信プロセッサ(CP 1612 A2、CP 1613 A2 など)を経由する通信用。
- MPI(マルチポイントインターフェース) : MPI 通信プロセッサまたは通信モジュール(例: CP 5613 A3)を介してプログラミング装置(例: PG 760/PC RI45)の内部 MPI インターフェースを経由する通信用。
- 名前指定接続 : シンボル接続を経由する STEP 7 での通信用。これらのシンボル接続は、STEP 7 を使用して設定され、例えば、H/F システムを冗長化と組み合わせた AS S7-400 との高可用性通信のために必要になります。
- PROFIBUS および PROFIBUS (II) : SIMATIC NET PROFIBUS を使用する通信プロセッサ(CP 5613 A3 など)を経由する通信用。
- スロット PLC : WinCC コンピュータの PC カードとしてインストールされるスロット PLC (例: WinAC Pro)との通信用。

- ソフト PLC : WinCC コンピュータのアプリケーションとしてインストールされるソフトウェア PLC (例: WinAC Basis)との通信用。
- TCP/IP : TCP/IP プロトコルを使用するネットワークとの通信用。

チャンネルおよびタグの診断の詳細については、「通信診断」を参照してください。

詳細手順

追加情報およびチャンネル設定の詳細な例は、次の「WinCC V6 通信マニュアル」を参照してください。

- <http://support.automation.siemens.com/WW/view/en/21320307> (<http://support.automation.siemens.com/WW/view/en/21320307>)

チャンネルおよびタグの診断に関する追加情報は、「通信診断 (ページ 615)」を参照してください。

下記も参照

ソフトウェアの二重化 - 接続固有内部タグ (ページ 504)

"SIMATIC S7 Protocol Suite"チャンネル - コンフィグレーション (ページ 414)

サポートされているデータタイプの概要 (ページ 413)

チャンネルユニットの選択 (ページ 410)

チャンネルおよびタグの診断 (ページ 615)

<http://support.automation.siemens.com/WW/view/en/21320307> (<http://support.automation.siemens.com/WW/view/en/21320307>)

6.11.2 チャンネルユニットの選択

はじめに

既存または計画されたネットワークへの通信接続を作成するには、以下のものを選択する必要があります。

- チャンネルのいずれかのチャンネルユニット
- WinCC ステーションに適した通信プロセッサ
- 特定のオートメーションシステムに適した通信モジュール

このセクションには、各種バリエーションの概要を示します。

WinCC 用の通信プロセッサには 2 種類があります。

- いわゆるハードネット用の通信プロセッサ。これらの通信プロセッサには専用のマイクロプロセッサがあり、コンピュータ CPU に対する負荷を軽減します。同時に 2 種類のプロトコルを使用することが可能です(マルチプロトコルオペレーション)。
- いわゆるソフトネット用の通信プロセッサ。これらの通信プロセッサには、専用のマイクロプロセッサはありません。一度に 1 つのプロトコルしか使用できません(モノプロトコルオペレーション)。

チャンネルユニットの割り付け

以下の表は、ネットワークとオートメーションシステムに対する、「SIMATIC S7 Protocol Suite」チャンネルのチャンネルユニットの割り付けを示しています。

チャンネルのチャンネルユニット	通信ネットワーク	AS
MPI	MPI	S7-300 および S7-400
PROFIBUS + PROFIBUS (II)	PROFIBUS	S7-300 および S7-400
Industrial Ethernet + Industrial Ethernet (II)	Industrial Ethernet	S7-300 および S7-400
TCP/IP	TCP/IP 経由の Industrial Ethernet	S7-300 および S7-400
名前指定接続	Industrial Ethernet または PROFIBUS	S7-400 H/F システム
Slot PLC	"ソフト K バス" (内部)	PC 内部
Soft-PLC	"ソフト K バス" (内部)	PC 内部

MPI

MPI を介した S7-300 と S7-400 オートメーションシステムとの通信のために、「MPI」チャンネルユニットは「SIMATIC S7 Protocol Suite」チャンネルで利用できます。

MPI ネットワークは主に、指定されたパラメータおよびノード数と伝送速度の制限を持つ PROFIBUS ネットワークに対応しています。MPI 経由の通信には、PROFIBUS ネットワーク用と同じ通信プロセッサとモジュールが使用されます。また、同じ通信プロトコルも使用されます。

オートメーションシステムの通信接続

MPI ネットワークを介した S7-300 と S7-400 オートメーションシステムの通信は、AS 内部 MPI インターフェースを介して、または適切な通信モジュールを使用して行うことができます。以下の表に、推奨されるコンポーネントを示します。

システム	CPU または通信モジュール(推奨)
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

WinCC 用の通信プロセッサ

以下の表に、WinCC ステーションを MPI ネットワークに接続する場合に推奨される通信プロセッサを示します。MPI 通信の場合、WinCC コンピュータ 1 台あたり使用できる通信プロセッサは 1 つだけです。各カードには、それぞれの通信プロトコルに適したドライバソフトウェアもあります。

通信プロセッサ(WinCC)	デザイン/タイプ
CP 5613 A3	PCI カード/ハードネット
CP 5612	PCI カード/ソフトネット

PROFIBUS

PROFIBUS を介した S7-300 と S7-400 オートメーションシステムとの通信では、「PROFIBUS」と「PROFIBUS II」チャンネルユニットが「SIMATIC S7 Protocol Suite」チャンネルで使用できます。

6.11 SIMATIC S7 Protocol Suite

これらのチャンネルユニットは、ハードネットモジュールおよびソフトネットモジュールを使用した通信をサポートします。

オートメーションシステムの通信接続

PROFIBUS ネットワークを介した S7-300 と S7-400 オートメーションシステムの通信は、AS 内部インターフェースを介して、または通信モジュールを使用して行うことができます。以下の表に、推奨されるコンポーネントを示します。

システム	CPU または通信モジュール
S7 -300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

WinCC 用の通信プロセッサ

以下の表に、WinCC ステーションを PROFIBUS に接続する場合に推奨される通信プロセッサを示します。「PROFIBUS」チャンネルユニットは、ハードネットカードおよびソフトネットカードを使用した通信をサポートします。WinCC ステーションでは、これらのモジュールを最大 2 つ使用することができます。各通信プロセッサには、それぞれの通信プロトコルに適したドライバソフトウェアもあります。

通信プロセッサ(WinCC)	デザインタイプ
CP 5613 A3	PCI カード/ハードネット
CP 5612	PCI カード/ソフトネット

Industrial Ethernet および TCP/IP

WinCC では、Industrial Ethernet を介した通信用の複数のチャンネルユニットが「SIMATIC S7 Protocol Suite」チャンネルで利用できます。

- S7 機能を使用した「ISO」プロトコルの場合は「Industrial Ethernet」および「Industrial Ethernet (II)」チャンネルユニット
- S7 機能を使用した「ISO-on-TCP」プロトコルの場合の「TCP/IP」チャンネルユニット

これらのチャンネルユニットは、ハードネットモジュールおよびソフトネットモジュールを使用した通信をサポートします。

オートメーションシステム用通信モジュール

「ISO」または「ISO-on-TCP」プロトコルを備えた Industrial Ethernet を介した S7-300 または S7-400 オートメーションシステムの通信には、適切な通信モジュールが装備されています。以下の表に、推奨されるコンポーネントを示します。

システム	Industrial Ethernet 用通信モジュール	TCP/IP プロトコル用通信モジュール
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

WinCC 用の通信プロセッサ

Industrial Ethernet を介した WinCC ステーションの「ISO」または「ISO-on-TCP」プロトコルによる通信は、表に示す推奨通信プロセッサを使用して行われます。

各通信プロセッサには、それぞれの通信プロトコルに適したドライバソフトウェアもあります。

通信プロセッサ(WinCC)	デザインタイプ
CP 1612 A2	PCI カード/ソフトネット
CP 1613 A2	PCI カード/ハードネット

6.11.3 サポートされているデータタイプの概要

はじめに

タグをコンフィグレーションする場合、AS 内のデータフォーマットでデータタイプと変換を定義する必要があります。

以下の表に、このチャンネルでサポートされるデータタイプと、タイプ変換の使用法を示します。

サポートされるデータタイプ

データタイプ	タイプ変換
2 進タグ	いいえ
符号付き 8 ビット値	はい

6.11 SIMATIC S7 Protocol Suite

データタイプ	タイプ変換
符号なし 8 ビット値	はい
符号付き 16 ビット値	はい
符号なし 16 ビット値	はい
符号付き 32 ビット値	はい
符号なし 32 ビット値	はい
浮動小数点数 32 ビット IEEE 754	はい
テキストタグ、8 ビットフォント	いいえ
未処理データタイプ	いいえ

タイプ変換の詳細については、「通信」でも説明します。

6.11.4 チャンネルのコンフィグレーション

6.11.4.1 "SIMATIC S7 Protocol Suite"チャンネル - コンフィグレーション

はじめに

このセクションでは、「SIMATIC S7 Protocol Suite」チャンネルのコンフィグレーション方法について説明します。

1. チャンネルのインストール
2. チャンネルユニットの選択
3. 接続をコンフィグレーション
4. タグのコンフィグレーション

システムパラメータコンフィグレーション

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

下記も参照

チャンネルユニットのシステムパラメータ (ページ 451)

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルのチャンネルユニット (ページ 425)

チャンネルおよびタグの診断 (ページ 615)

6.11.4.2 "SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法

はじめに

このセクションでは、"SIMATIC S7 Protocol Suite"チャンネルのインストール方法について説明します。

1. チャンネルのインストール
2. チャンネルユニットの選択
3. 接続の作成
4. タグの挿入
5. カスタマイズされた WinCC インストールでのシステムパラメータの設定


必要条件:

- 通信モジュールが組み込まれていること。
- ハードウェアドライバがインストールされていること。
- AS とのケーブル接続が存在すること。

手順

1. タグ管理のナビゲーションエリアで、[タグ管理]ノードのショートカットメニューから、[新規ドライバの追加]エントリを選択します。
2. ドライバ"SIMATIC S7 Protocol Suite"を選択します。
チャンネルが作成されます。
通信ドライバと関連するチャンネルユニットがタグ管理に表示されます。
3. 必要なチャンネルユニットを選択し、ショートカットメニューから[新規接続]エントリを選択します。
4. 接続の名前を入力します。
5. 接続確立と接続ステータスのシステムタグを作成するには、接続のショートカットメニューで[有効化/無効化タグの作成]エントリを選択します。
内部タググループ「ConnectionStates」に以下のタグが作成されます。
 - @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx
6. テーブルエリアの下にある[タグ]タブをクリックします。
7. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
8. [データタイプ]フィールドで、希望するデータタイプを選択します。
[プロパティ]領域で、タグの開始値と代替値を定義するオプションがあります。
特定のチャンネルユニットの接続のタグを設定するための詳細な説明が必要な場合は、ダイアログを閉じて、関連するチャンネルユニット内のトピック「タグの設定」に進みます。

6.11 SIMATIC S7 Protocol Suite

9. [アドレスプロパティ]ダイアログを開きます。
この目的で、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。
- 10.[OK]ボタンをクリックして、両方のダイアログを閉じます。
- 11.WinCC システムおよび通信ハードウェアが標準のものではない場合、システムパラメータを非標準値に設定する必要があります。
詳細については、「システムパラメータ」を参照してください。

下記も参照

AS シンボルをオフラインでダウンロードする方法 (ページ 416)

6.11.4.3 AS シンボルをオフラインでダウンロードする方法

はじめに

以下の S7 チャンネルをオフラインで設定できます。

- SIMATIC S7 Protocol Suite
- SIMATIC S7-1200, S7-1500 Channel

この目的に対して、例えば、既存の TIA Portal プロジェクトからデータレコードをエクスポートし、エクスポートファイルを WinCC プロジェクトにロードします。

サポートされているエクスポートフォーマット

以下のファイルフォーマットがインポートでサポートされます。

フォーマット	内容	説明
*.bin	2 進データ	WinCC タグ管理からのエクスポート: <ul style="list-style-type: none"> [タグ管理]の表示 > 接続のショートカットメニュー: [AS シンボル] > [ファイルに保存] 「SIMATIC S7 Protocol Suite」チャンネルによりサポートされていません。
*.sdz	ストラクチャードエクスポート	WinCC タグ管理からのエクスポート: <ul style="list-style-type: none"> [シンボル]の表示 > [メニュー]: [編集] > [エクスポート] また、ナビゲーションエリアから構造情報をエクスポートします。
*.zip	TIA Portal エクスポートファイル	[SIEMENS SIMATIC SCADA Export]ツールを使用して TIA Portal からエクスポート

TIA Portal の「SIEMENS SIMATIC SCADA Export」

データレコードを TIA Portal プロジェクトからエクスポートするには、「SIEMENS SIMATIC SCADA Export」ツールを使用します。

TIA Portal プロジェクトにおいて、PLC のショートカットメニューで[SIEMATIC SCADA にエクスポート]エントリを選択します。

各種 TIA Portal バージョン用のツールを Industry Online Support でダウンロードして使用することができます。

- 「SIMATIC SCADA Export」ダウンロード (ID 109748955) (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)
- 「SIMATIC SCADA Export」ドキュメント (ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

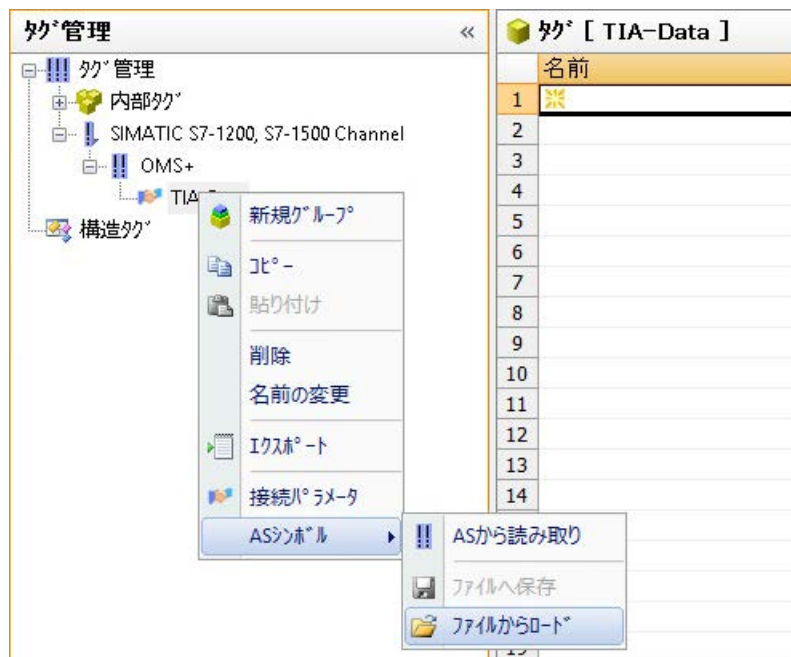
必要条件

- AS が TIA Portal でコンパイルされていること。
- PLC の対応する設定データがエクスポートされ、.zip ファイルなどで利用可能になります。

- 通信プロセッサと関連するハードウェアドライバが WinCC プロジェクトにインストールされていること。
- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。
- [タグ管理]エディタが開いていること。

手順

1. 接続のショートカットメニューで[AS シンボル]>[ファイルからロード]を選択します。



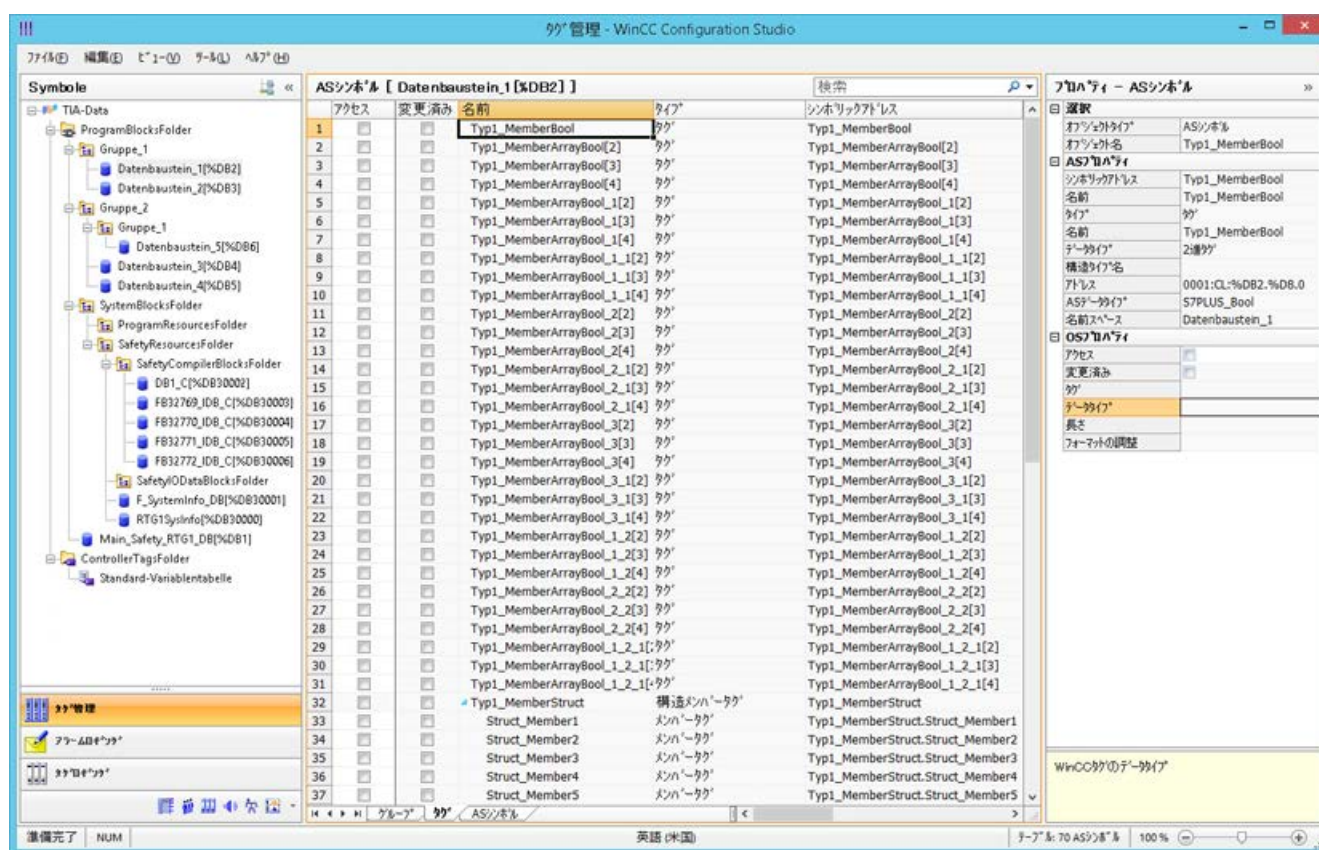
2. ロード対象の希望するデータレコードを選択します。
利用可能なコントローラデータがロードされます。

結果

設定がインポートされ、[シンボル]の表示が開かれます。

ロードされたデータがテーブルエリアの[AS シンボル]タブに表示され、後続の処理に対して利用できるようになります。

ロードされているデータが構造も含んでいる場合、[AS 構造]タブが追加で表示されます。
エディタが閉じられた後に、[AS シンボル]と[AS 構造]タブが再度非表示になります。



シンボルの表示

次のボタンを使用して、タグ管理をデフォルト表示と[シンボル]の表示の間で切り替えます。

ボタンは、データレコードがロードされた後にのみ使用可能になります。

ナビゲーションエリア

構造ツリーのデータの表示は、TIA Portal からの階層に対応しています。

テーブルエリア

検出された WinCC タグが AS タグに一致しないとき、[変更済み]列のチェックボックスは、自動的に選択されます。また、ユーザーがこれらを使用してフィルタリングできるようにします。

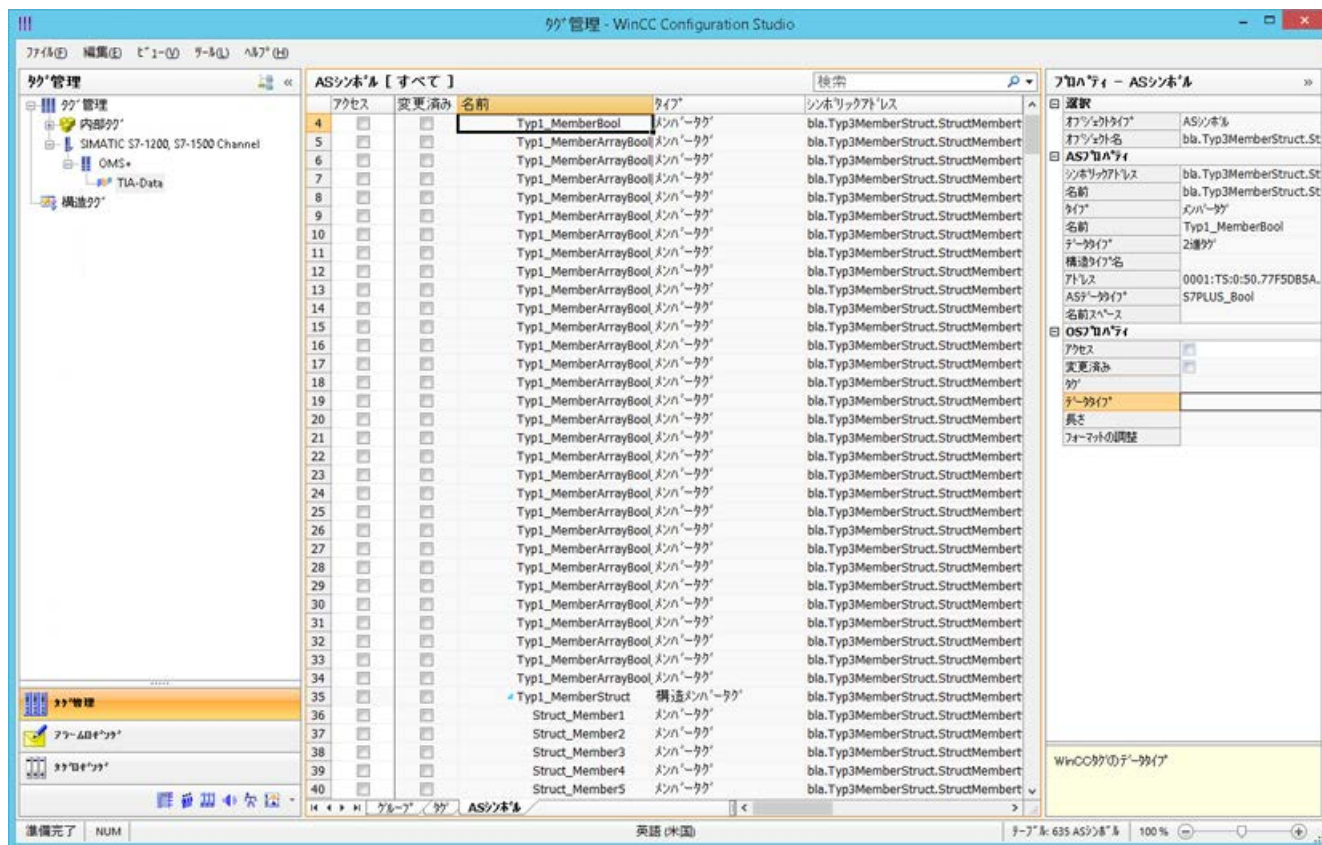
[アクセス]列のチェックボックスを選択することで、検出された AS タグから WinCC タグを作成します。

[タグ管理]の AS シンボル

[AS シンボル]タブを介した[タグ管理]の AS シンボルへのアクセス権限もあります。

データブロック特有の[シンボル]の表示とは対照的に、コントローラのすべての利用可能なタグがここに表示されます。

この表示はまた、AS にもはや存在していない以前に設定されていたタグも表示します。



下記も参照

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

AS プロジェクトデータのエクスポート方法 (ページ 423)

AS 構造の構成方法 (ページ 421)

「SIMATIC SCADA Export」ドキュメント (ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

「SIMATIC SCADA Export」ダウンロード (ID 109748955) (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)

6.11.4.4 AS 構造の構成方法

はじめに

AS シンボルをロードする場合、コントロールシステムの構造もインポートされます。

手順は通信チャンネルに依存します。

- SIMATIC S7 Protocol Suite:
 - ファイルからのロード
- SIMATIC S7-1200, S7-1500 Channel
 - ファイルからのロード
 - AS からのロード

タグ管理の AS 構造

AS 構造は、デフォルトビューおよび[AS 構造]タブの[シンボル]ビューで表示されます。

WinCC で AS 構造を使用するには以下の方法があります。

- AS 構造タグ用に WinCC 構造タイプを作成します。
構造が、WinCC タグ管理の[構造タグ]に、構造タイプとして作成されます。
また、構造タイプ要素は、含まれている[タグタイプメンバ]それぞれに対して作成されます。
- WinCC 構造タイプを AS 構造タグに割り付けます。
次いで、各[タグタイプメンバ]に対して選択された構造タイプの構造タイプ要素を選択します。

タグ管理で、WinCC 構造タイプの名前および構造タイプ要素の名前を変更します。AS 構造の割り付けは、自動的に調整されます。

必要条件

- 次の方法のいずれかによって、PLC の設定データにアクセスしていること。
 - PLC への接続がランタイム時に確立された。
 - エクスポートされた設定データが、例えば zip ファイルとして使用可能である。
- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。

手順

1. [AS から読み取り]または[ファイルからロード]を介して AS シンボルをロードします。
ロードされたメッセージは、[シンボル]タグ管理ビューに表示されます。
ロードされた構造は、[AS 構造]タブで表示されます。
AS からロードするとき、構造名は転送されません。ID が、[構造タグタイプ]の名前として表示されます。

Symbole		AS 構造		
CPU1516_V2	ProgramBlocksFolder	WinCC構造	名前	タイプ°
		1	SimData1UDT	構造タグタイプ°
		2	LEN_UDT1	構造タグタイプ°

2. [AS 構造]をクリックします。
構造の要素を表示するには、構造名の前の矢印をクリックします。
3. 構造の行全体を選択し、ショートカットメニューから[構造を作成]エントリを選択します。

AS 構造			
WinCC構造	名前	タイプ°	
1 SimData1UDT	SimData1UDT	構造タグタイプ°	
2 BitData	BitData	タグタイプメンバ°	
3 IntData	IntData	タグタイプメンバ°	
4 RealData	RealData	タグタイプメンバ°	
5 ByteData	ByteData	タグタイプメンバ°	
6	LEN_UDT1	構造タグタイプ°	
7		タグタイプメンバ°	
8		構造タグタイプ°	
9		構造タグタイプ°	
10			
11			
12			
13			
14			
15			

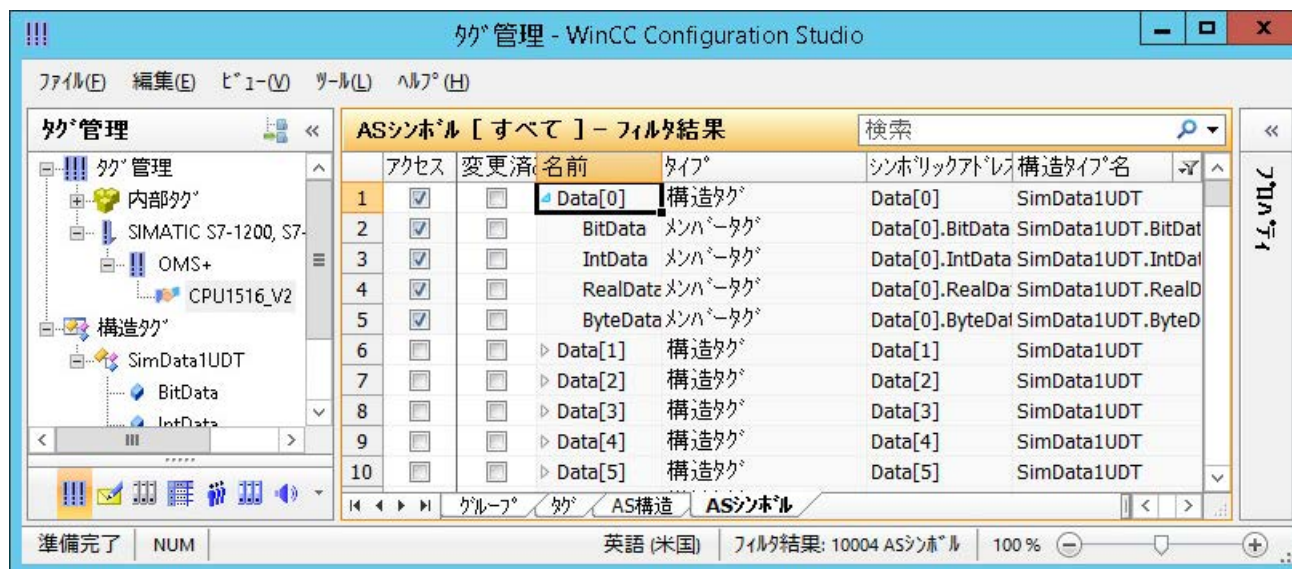
あるいは、WinCC タグ管理で作成済みの構造タイプを選択します。
次いで、構造タイプ要素を[タグタイプメンバ]に割り付けます。

AS 構造		
WinCC構造	名前	タイプ°
1	SimData1UDT	構造タグタイプ°
2	SimData1UDT	
3	LEN_UDT1	
4	WinCC_StrucType1	

構造タイプが、AS 構造の各[構造タグタイプ]に対して、WinCC タグ管理で作成されます。
構造タイプ要素が、各[タグタイプメンバ]に対して作成されます。

4. [タグ管理]ビューの[AS シンボル]タブを選択します。

5. 構造タグとメンバタグのみを表示させるには、[構造タイプ名]列で希望する AS 構造にフィルタを掛けます。



6. WinCC タグ管理で AS 構造タグにアクセスするには、[アクセス]フィールドを有効にします。含まれているメンバタグが自動的に有効になります。
AS 構造タグは、構造タグとして、WinCC タグ管理で作成されます。

結果

WinCC タグ管理の構造タイプおよび構造タグを介して、AS 構造タグにアクセスできます。

このようにして、例えば、WinCC フェイスプレートタイプの AS 構造にアクセスし、フェイスプレートインスタンスで表示することができます。

下記も参照

AS シンボルをオフラインでダウンロードする方法 (ページ 416)

AS プロジェクトデータのエクスポート方法 (ページ 423)

6.11.4.5 AS プロジェクトデータのエクスポート方法

AS シンボルのエクスポート

オフライン設定にエクスポートファイルを使用します。

6.11 SIMATIC S7 Protocol Suite

AS プロジェクトデータを次のフォーマットでエクスポートできます。

通信チャンネル	エクスポートされたデータ	エクスポートファイルのフォーマット
SIMATIC S7-1200, S7-1500 Channel	AS シンボルおよび AS 構造	2 進データ: *.bin 構造化されたエクスポート: *.sdz
SIMATIC S7 Protocol Suite	AS シンボルおよび AS 構造	構造化されたエクスポート: *.sdz

必要条件

- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。
- AS プロジェクトデータをロードし、WinCC で設定しています。

手順:2 進データのエクスポート

- タグ管理で接続を選択します。
- ショートカットメニューから、[AS シンボル]>[ファイルに保存]エントリを選択します。
[エクスポート]ダイアログが開きます。
- 保存パスを選択して、ファイル名を入力します。
[エクスポート]ボタンを押してダイアログを閉じます。
設定データが 2 進データセットとして.bin ファイルにエクスポートされます。

手順:構造化済みデータのエクスポート

- タグ管理で[シンボル]ビューを選択します。
- [編集]>[エクスポート]メニューコマンドを選択します。
- 保存パスを選択して、ファイル名を入力します。
[エクスポート]ボタンを押してダイアログを閉じます。
設定データは*.sdz ファイルにエクスポートされます。
構造化済みのエクスポートには、ナビゲーションエリアからの構造情報も含まれます。

下記も参照

AS シンボルをオフラインでダウンロードする方法 (ページ 416)

AS 構造の構成方法 (ページ 421)

6.11.4.6 チャンネルユニット

"SIMATIC S7 Protocol Suite"チャンネルのチャンネルユニット

はじめに

次の章では、チャンネルユニットと対応する接続のコンフィグレーション方法を説明しています。同一のチャンネルユニットに複数の接続をコンフィグレーションすることができます。

下記も参照

"TCP/IP"チャンネルユニット (ページ 441)

"ソフト PLC"チャンネルユニット (ページ 440)

"スロット PLC"チャンネルユニット (ページ 438)

チャンネルユニット"PROFIBUS(I + II)" (ページ 434)

"名前指定接続"チャンネルユニット (ページ 431)

"マルチポイントインターフェース"チャンネルユニット (ページ 428)

チャンネルユニット"工業用イーサネット"+"工業用イーサネット(II)" (ページ 425)

"工業用イーサネット(I+II)" チャンネルユニット

チャンネルユニット"工業用イーサネット"+"工業用イーサネット(II)"

動作の原則

チャンネルユニット「Industrial Ethernet」は、Industrial Ethernet 経由で WinCC と S7 オートメーションシステムを接続するのに使用されます。CP 343-1 経由のオートメーションシステム S7-300 や、CP 443-1 経由の S7-400 の場合などには、通信モジュール(CP)による通信が可能です。

WinCC では、異なる通信プロセッサ(例: CP 1613 A2)を使用できます。2 番目の通信プロセッサは、「Industrial Ethernet (II)」チャンネルユニットを介してアドレス指定することができます。通信は「ISO」トランスポートプロトコルによって行われるため、ローカルデータベースに論理接続を設定する必要はありません。

これらのチャンネルユニットに関するファンクションと設定はすべて同じです。

ユニット固有の用語

通信プロセッサ

通信プロセッサ(CP)とは、WinCC コンピュータと特定のネットワークとの通信が行われるモジュールです。

"ISO"トランスポートプロトコル

ISO トランスポートは ISO-OSI 参照モデルのレイヤーの 1 つで、接続によるデータ転送に関するサービスを提供します。トランスポートレイヤーでは、データフローコントロール、ブロッキング、承認などのタスクを扱います。

プロトコルは、物理回線上の内容に関するデータトラフィックの構造体を定義します。特に、オペレーションモード、接続確立時のプロシージャ、データのバックアップ、伝送速度を定義します。

Industrial Ethernet

Industrial Ethernet は、工業環境の中でも最も効率的なサブネットです。工業用イーサネットは工場や小室レベルに適しており、多数のユーザー間による長距離を介した大量のデータ交換を助長します。

Industrial Ethernet は、IEEE 802.3 標準に準拠したオープンな通信ネットワークとして標準化されています。工業用イーサネットの主要な利点は、そのスピード、シンプルな拡張可能性、開放性に加え、高い可用性とワールドワイドな実用性です。また設定プロセスにもあまり手間がかかりません。

下記も参照

タグのコンフィグレーション (ページ 445)

"工業用イーサネット"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 426)

"工業用イーサネット"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

WinCC が PLC と通信するには、このチャンネルユニットに加えて論理接続も必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

S7 オートメーションシステムでは、通信モジュールが通信に使用されます(例: S7-300 の場合は CP 343-1、S7-400 の場合は CP 443-1)。

WinCC では、CP 1613 A2 などの通信プロセッサが使用されます。2 番目の通信プロセッサは、チャンネルユニット「Industrial Ethernet (II)」経由でアドレス指定/追加できます。チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

注記

S7-300/S7-400:CPU のラック/スロット番号

外部通信モジュールで S7-300 または S7-400 を使用する際には、CPU のラック/スロット番号を入力する必要があります。

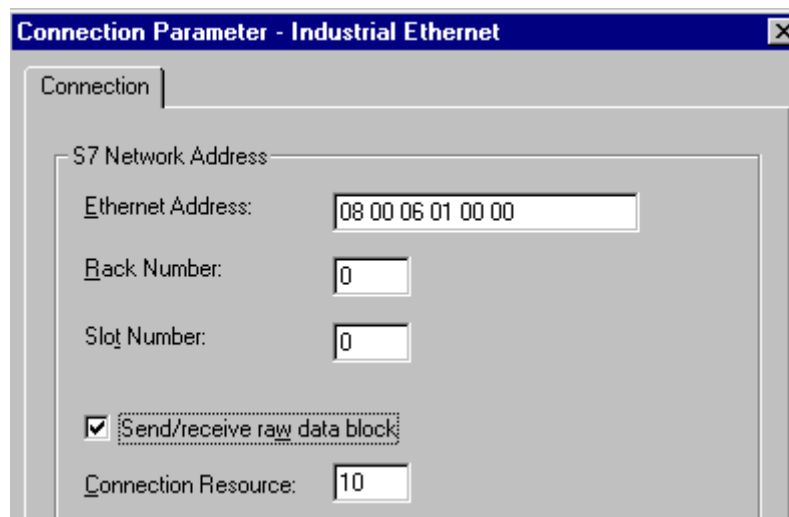
間違ったラック番号またはスロット番号を入力すると、通信リンクが確立されません。

必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。

手順

1. チャンネルユニット「Industrial Ethernet」のショートカットメニューで[新規接続]エントリを選択します。
新しい接続が作成されます。
2. 接続名を入力します(例: 「Test_Ind_Eth」)。
3. ショートカットメニューで[接続パラメータ]接続を選択します。
[接続パラメータ - Industrial Ethernet]ダイアログが開きます。



4. [Ethernet アドレス]フィールドに、バス上のオートメーションシステムのステーションアドレスを入力します。

6.11 SIMATIC S7 Protocol Suite

5. [ラック番号]フィールドに、アドレス指定される CPU が配置されているラックの番号を入力します。
6. 指定したラックの CPU のスロット番号を、対応する[スロット番号]フィールドに入力する必要があります。
7. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
このチェックボックスが有効な場合は、[接続リソース]フィールドが編集可能になります。
接続リソースに対しては、16 進数値を入力してください。
この接続リソースは、PLC で接続を設定する際に STEP7 によって割り付けられます。
8. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

チャンネルおよびタグの診断 (ページ 615)

"マルチポイントインターフェース"チャンネルユニット**"マルチポイントインターフェース"チャンネルユニット**

動作の原則

MPI チャンネルユニットは、WinCC を MPI 経由でオートメーションシステム SIMATIC S7-300 および S7-400 に接続するのに使用されます。

これを実行するには、WinCC で以下のものが使用されます。

- プログラミングデバイスの内部 MPI インターフェース(例: PG 760/PC RI45)
- 通信プロセッサ(例: CP 5613 A3) (PCI カード)

いわゆる MPI モジュール(ISA カード)も使用できます。ただし、もはや入手できません。この代わりに通信プロセッサが使用されるようになっています。

AS の場合は、接続が CPU の MPI インターフェースまたは対応する通信モジュール経由で行われます。

ユニット固有の用語

MPI

MPI とはマルチポイントインターフェースのことで、複数のノードが可能な通信接続です。通信ネットワークへの接続は、次のように行われます。

- AS の場合は、CPU の MPI インターフェースまたは通信モジュールを使用して行われます。
- WinCC の場合は、ビルトインの MPI インターフェース(例: プログラミングデバイス)または通信プロセッサ(ネットワークカード)を使用して行われます。

通信プロセッサ

通信プロセッサ(CP)とは、WinCC コンピュータと特定のネットワークとの通信が行われるモジュールです。

下記も参照

タグのコンフィグレーション (ページ 445)

"マルチポイントインターフェース"チャンネルユニット接続をコンフィグレーションする方法 (ページ 429)

"マルチポイントインターフェース"チャンネルユニット接続をコンフィグレーションする方法

はじめに

WinCC が PLC と通信するには、このチャンネルユニットに加えて論理接続も必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

S7-300 および S7-400 PLC は、いずれも内部 MPI インターフェースまたは通信モジュール(例: CP 342-5 (SIMATIC S7-300)または CP 443-5 (SIMATIC S7-400))を使用します。

WinCC が PG 760/PC RI45 にインストールされている場合は、内部 MPI インターフェースを使用できます。使用しない場合、ビルトインの MPI モジュールがある必要があります。また、通信モジュールを使用することもできます。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

注記

S7-300/S7-400:CPU のラック/スロット番号

外部通信プロセッサで S7-300 または S7-400 を使用する際には、CPU のラック/スロット番号を入力する必要があります。

正しくないラック番号またはスロット番号を入力すると、通信接続は確立されません。

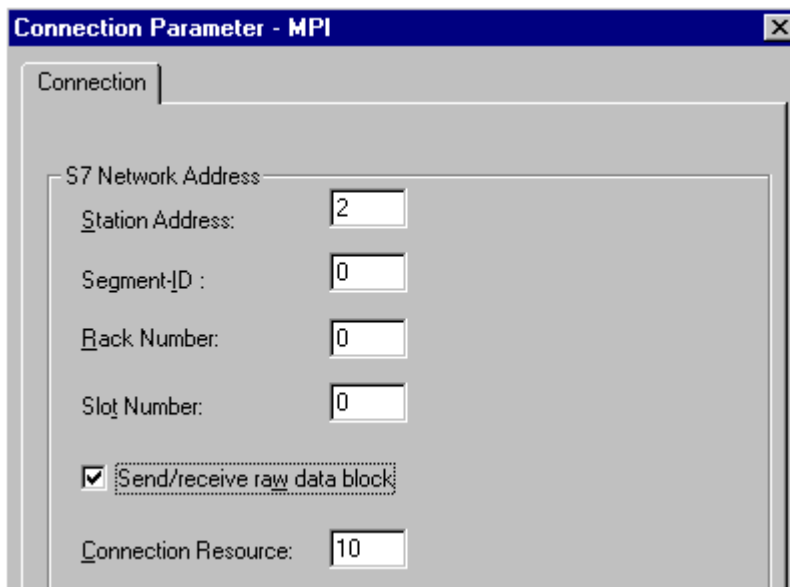
S7-300 を使用する場合は、CPU の内部 MPI インターフェース経由で接続するのであれば、ラック/スロット番号として 0 を入力する必要があります。

必要条件

- (通信)ドライバ"SIMATIC S7 Protocol Suite"をプロジェクトに統合して下さい。

手順

1. チャンネルユニット[MPI]のショートカットメニューで[新規接続]エントリを選択します。
新しい接続が作成されます。
2. 接続名として"Test_MPI"と入力します。
3. ショートカットメニューで[接続パラメータ]接続を選択します。
[接続パラメータ - MPI]ダイアログが開きます。



4. 適切なフィールドにあるバス上のオートメーションシステムの[ステーションアドレス]フィールドにステーションアドレスを入力します。

5. [セグメント ID]フィールドは現在サポートされていません。この値は"0"のままにしておく必要があります。
6. [ラック番号]フィールドに、アドレス指定される CPU が配置されているラックの番号を入力します。
7. 指定したラックに CPU の[スロット番号]を入力します。
8. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
このチェックボックスが有効な場合は、[接続リソース]フィールドも有効になります。
接続リソースに対しては、16 進数値を入力してください。
この接続リソースは、PLC で接続を設定する際に STEP7 によって割り付けられます。
9. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

"名前指定接続"チャンネルユニット

"名前指定接続"チャンネルユニット

動作

このチャンネルユニットは、STEP 7 を使用して設定されるシンボル接続をセットアップするために使用されます。これにより、シンボル接続名を使用して WinCC を二重化接続および非二重化接続にアドレス指定することができます。これらのシンボル接続は、H/F システムを二重化するとともに S7-400 PLC を使用して高可用性通信を実現する場合などに必要となります。

STEP 7 では、NETPRO アプリケーションを使用してシンボル接続名が設定されます。接続名、接続パラメータ、アプリケーション名はデータベース(*.XDB)に保存されます。このデータベースは PLC/OS エンジニアリングツールの「Mapper」によって自動的に対応する

WinCC プロジェクトディレクトリに保存されます。ただし、「Mapper」を使用していない場合などに、このディレクトリ以外の場所にコピーすることもできます。

注記

WinCC システムでは、通信参加者につき 1 つの XDB ファイルしか作成することができません。

そのため、XDB ファイルは複数の WinCC コンピュータ上でコピーすることも、使用することもできません。

WinCC でこのデータベースを有効にするには、次のようなオプションがあります。

- XDB ファイルがプロジェクトディレクトリに保存されていない場合は(「Mapper」ツールを使用しなかったためなど)、WinCC を起動する前に、STEP 7 の[設定]タブの[PG/PC インターフェースの設定] (コントロールパネル)で、XDB ファイルのパスと名前を入力する必要があります。

WinCC を起動したときに、プロジェクトディレクトリにファイルがない場合は、この外部ディレクトリから XDB ファイルが読み込まれます。この方法は、複数のプロジェクトで中央に保存された同一データベースを使用する場合に役立ちます。

- 「Mapper」ツールを使用した場合は、XDB ファイルは WinCC プロジェクトディレクトリに自動的にコピーされます。WinCC が起動してプロジェクトが開くと、S7 チャンネルからデータが読み取られ、Windows のレジストリデータベースに入力されます。

その後、選択したアプリケーション名にシンボル接続名を割り付けることで、WinCC での接続を設定できます。

注記

アプリケーション名および接続名をここに手動で入力することも可能です。

CS モードでは名前がチェックされないため、名前の綴りが STEP 7 で設定したとおりになっていることを確認する必要があります。

これは、たとえば、次の場合に必要です。

- シンボル接続名に使用可能な XDB ファイルはありません。この場合、設定を直接「コンポーネント設定ツール」に転送します。
 - プロジェクトは別のコンピュータに転送されます。
-

ユニットに関する一般用語

通信プロセッサ

通信プロセッサ(CP)とは、PLC と特定のネットワーク間の通信をサポートするモジュールを言います。

下記も参照

タグのコンフィグレーション (ページ 445)

"名前指定接続"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 433)

"名前指定接続"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

シンボル接続経由で WinCC が S7-400 PLC と通信するには、このチャンネルユニットに加えて論理接続も必要です。

論理接続のセットアップ時には、[接続名]フィールドにリストされたシンボル接続名のいずれかが選択したアプリケーション名に割り付けられます。

シンボル接続名とアプリケーション名は STEP 7 で構成されます。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

注記

アプリケーション名および接続名をここに手動で入力することも可能です。

CS モードでは名前がチェックされないため、名前の綴りが STEP 7 で設定したとおりになっていることを確認する必要があります。

これは、たとえば、次の場合に必要です。

- シンボル接続名に使用可能な XDB ファイルはありません。この場合、設定を直接「コンポーネント設定ツール」に転送します。
 - プロジェクトは別のコンピュータに転送されます。
-

必要条件

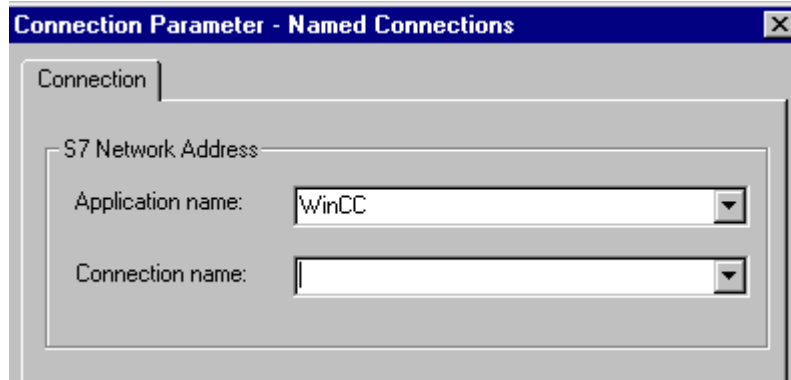
- 「SIMATIC S7 Protocol Suite」チャンネルがプロジェクトに統合されていること。

手順

1. チャンネルユニット[名前指定接続]のショートカットメニューで[新規接続]エントリを選択します。
新しい接続が作成されます。
2. 接続名を入力します(たとえば、"Test_NC")。

6.11 SIMATIC S7 Protocol Suite

3. 接続のショートカットメニューで[接続パラメータ]を選択します。
[接続パラメータ - 名前指定接続]ダイアログが開きます。



4. [アプリケーション名]フィールドに、STEP 7 で構成されたアプリケーションの名前を入力します。デフォルト値は WinCC です。
5. [接続名]フィールドに、STEP 7 で構成されたシンボル接続の名前を入力します。
6. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

"PROFIBUS (I+II)"チャンネルユニットチャンネルユニット**"PROFIBUS(I + II)"**

動作の原則

このチャンネルユニットは、PROFIBUS ネットワーク経由で WinCC を SIMATIC S7-300 および S7-400 オートメーションシステムに接続するのに使用されます。

S7 オートメーションシステムでは、通信モジュールが使用されます(例: S7-300 の場合は CP 342-5、S7-400 の場合は CP 443-5)。

WinCC では、CP 5613 A3 などの通信プロセッサが使用されます。

2 番目の通信プロセッサは、チャンネルユニット**"PROFIBUS II"**経由でアドレス指定できます。結果として、最大接続数を増やすことができます。

ユニット固有の用語

PROFIBUS

PROFIBUS は、セルとフィールドレベル用のオープンでベンダーニュートラルな通信システムで、最大 127 ノードまで接続できるように設計されています。PROFIBUS は、欧州規格 EN 50170、Volume 2、PROFIBUS に基づいています。PROFIBUS は、アクセス方法として低レベルのマスタスレーブを使用するトークンパッシングを使用します。

通信プロセッサ

通信プロセッサ(CP)とは、WinCC コンピュータと特定のネットワークとの通信が行われるモジュールです。

下記も参照

タグのコンフィグレーション (ページ 445)

"PROFIBUS"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 435)

"PROFIBUS"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

チャンネルユニットに加えて、PLC と通信するために WinCC にも論理接続が必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

S7 オートメーションシステムでは、通信モジュールが使用されます(例: S7-300 の場合は CP 342-5、S7-400 の場合は CP 443-5)。

WinCC では、CP 5613 A3 などの通信プロセッサが使用されます。2 番目の通信プロセッサは、チャンネルユニット"PROFIBUS II"経由でアドレス指定できます。

6.11 SIMATIC S7 Protocol Suite

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

注記

オフ状態での接続

PROFIBUS 通信のスタートアップ時に、WinCC コンピュータの電源がオンの状態で通信プロセッサを PROFIBUS に接続すると、PROFIBUS エラーが発生します。

このため、PROFIBUS を接続する前に、コンピュータの電源をオフにすることを推奨します。

この指示に従わなかった場合は、複数のトークンが (PROFIBUS 標準に従って) バスで発生する可能性があり、その結果、バスエラーが発生します。

S7-300/S7-400:CPU のラック/スロット番号

外部通信モジュールで S7-300 または S7-400 を使用する際には、CPU のラック/スロット番号を入力する必要があります。

間違ったラック番号またはスロット番号を入力すると、通信リンクが確立されません。

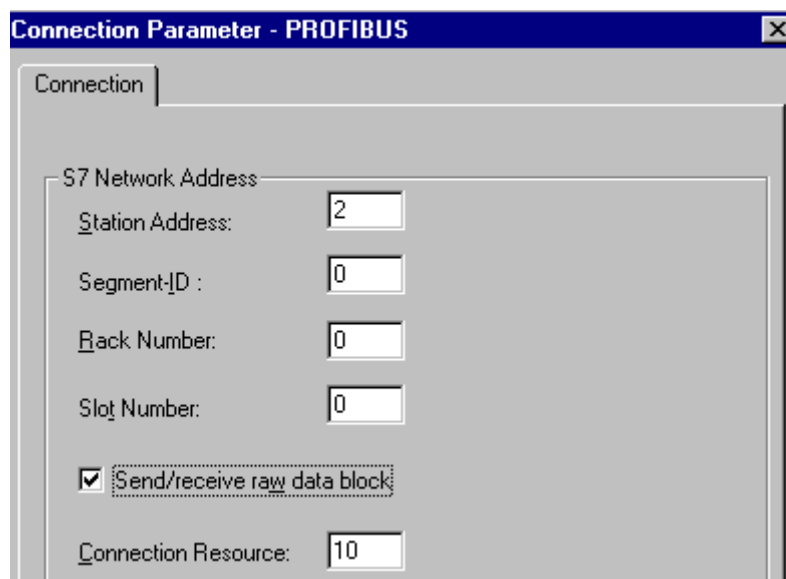
必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。

手順

1. チャンネルユニット [PROFIBUS] のショートカットメニューで [新規接続] エントリを選択します。
新しい接続が作成されます。
2. 接続名として "Test_PROFIBUS" と入力します。

3. ショートカットメニューで[接続パラメータ]接続を選択します。
[接続パラメータ - PROFIBUS]ダイアログが開きます。



4. 適切なフィールドに、バス上のオートメーションシステムの[ステーションアドレス]を入力します。
5. [セグメント ID]フィールドは現在サポートされていません。この値は"0"のままにしておく必要があります。
6. アドレス指定される CPU が配置される[ラック番号]を入力します。
7. 指定したラックに CPU の[スロット番号]を入力します。
8. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
このチェックボックスが有効な場合は、[接続リソース]フィールドも有効になります。
接続リソースに対しては、16 進数値を入力してください。
この接続リソースは、PLC で接続を設定する際に STEP7 によって割り付けられます。
9. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

"スロット PLC"チャンネルユニット

"スロット PLC"チャンネルユニット

操作原理

チャンネルユニット"スロット PLC"は、WinCC と WinCC コンピュータにインストールされた最大 4 つのスロット PLC (WinAC Pro)との通信を提供します。スロット PLC は統合インターフェースを備えているため、追加の通信ハードウェアを必要とせずに WinCC とスロット PLC 間を接続することができます。

下記も参照

タグのコンフィグレーション (ページ 445)

"スロット PLC"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 438)

"スロット PLC"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

WinCC がインストールされている SPS カードと通信するには、このチャンネルユニットに加えて論理接続も必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

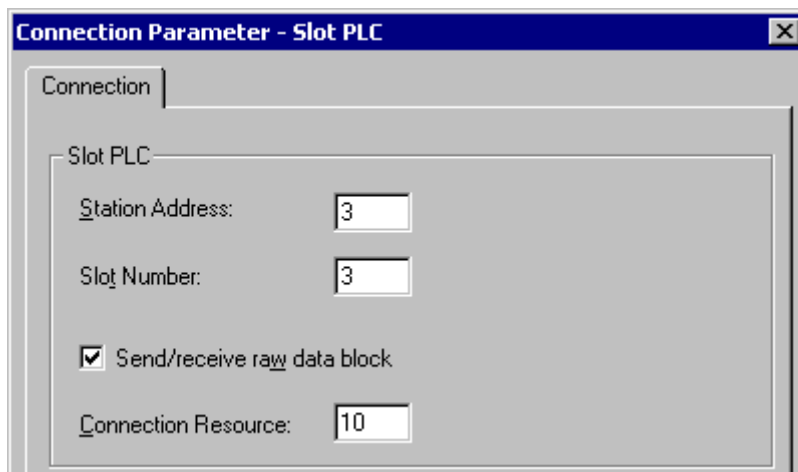
必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- 複数のスロット PLC を構成する場合は、スロット PLC バージョン 3.4 が必要です。

手順

1. チャンネルユニット[スロット PLC] のショートカットメニューで[新規接続]エントリを選択します。
新しい接続が作成されます。
2. 接続名を入力します(たとえば、"Test_SPLC")。

3. ショートカットメニューの[接続パラメータ]接続を選択します。
[接続パラメータ - スロット PLC]ダイアログが開きます。



4. [ステーションアドレス]フィールドに、ソフト K バス上のスロット PLC のステーションアドレスを入力します。
5. [スロット番号]フィールドに、スロット PLC がインストールされるスロットの番号を入力します。
6. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
7. このチェックボックスが有効な場合は、[接続リソース]フィールドも有効になります。接続リソースに対しては、16 進数値を入力してください。この接続リソースは、PLC で接続を設定する際に STEP 7 によって割り付けられます。
8. [OK]ボタンをクリックして、両方のダイアログを閉じます。

注記

接続パラメータの[ステーションアドレス]と[スロット番号]はインストールされている複数のスロット PLC と同一で、[スロット番号][3]から始まらなければなりません。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

"ソフト PLC"チャンネルユニット

"ソフト PLC"チャンネルユニット

操作原理

チャンネルユニット"スロット PLC"は、WinCC と WinCC コンピュータにインストールされたソフト PLC (WinAC Basic)との通信を提供します。WinCC とソフト PLC 間の接続には、その他の通信ハードウェアは必要ありません。

下記も参照

タグのコンフィグレーション (ページ 445)

"ソフト PLC"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 440)

"ソフト PLC"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

チャンネルユニットに加えて、ソフト PLC と通信するために WinCC にも論理接続が必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

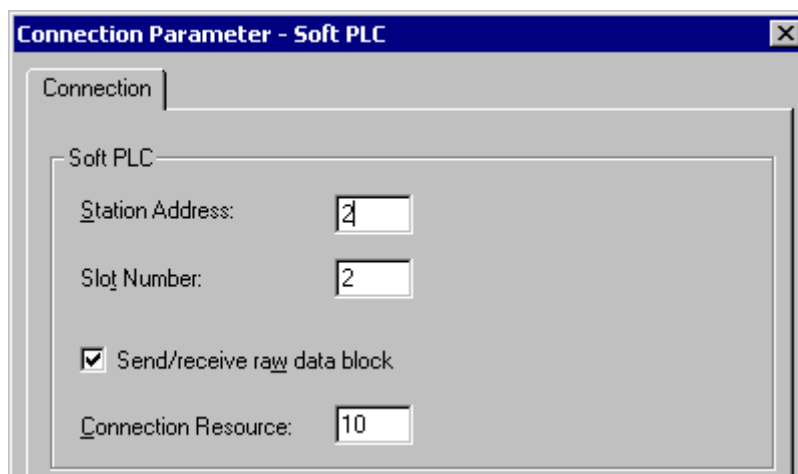
必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。

手順

1. チャンネルユニット[ソフト PLC] のショートカットメニューで[新規接続]エントリを選択します。
新しい接続が作成されます。
2. 接続名を入力します(たとえば、"Test_SOFTPLC")。

3. ショートカットメニューの[接続パラメータ]接続を選択します。
[接続パラメータ - スロット PLC]ダイアログが開きます。



4. [ステーションアドレス]フィールドに、ソフト K バス上のソフト PLC のステーションアドレスを入力します。
5. [スロット番号]フィールドに、スロットの番号を入力します。このスロット番号は、ソフト PLC のハードウェア構成中に構成され、同一 WinCC コンピュータで複数のソフト PLC を使用する際に必要です。
6. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
7. このチェックボックスが有効な場合は、[接続リソース]フィールドも有効になります。接続リソースに対しては、16 進数値を入力してください。この接続リソースは、PLC で接続を設定する際に STEP 7 によって割り付けられます。
8. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

"TCP/IP"チャンネルユニット

"TCP/IP"チャンネルユニット

動作の原則

チャンネルユニット「TCP/IP」は、「ISO-on-TCP トランSPORT」プロトコルを使用した Industrial Ethernet 経由で、WinCC とオートメーションシステム SIMATIC S7-300 および S7-400 を接続するのに使用されます。

プロトコルは、拡張 RFC 1006 を使用する標準 TCP/IP に対応しています。TCP/IP はデータのブロックがない場合に通信を使用するため、この拡張が必要です。

オートメーションシステム S7-300 の場合、通信は通信モジュール(CP 343-1 TCP など、S7-400 の場合は CP 443-1 TCP または CP 443-1 IT)経由で行われます。

WinCC では、CP 1613 A2 などの通信プロセッサが使用されます。

通信は ISO-on-TCP トランスポートプロトコルによって行われるため、ローカルデータベースに論理接続を設定する必要はありません。

ユニット固有の用語

通信プロセッサ

通信プロセッサ(CP)とは、WinCC コンピュータと特定のネットワークとの通信が行われるモジュールです。

ISO トランスポートプロトコル

ISO トランスポートは ISO-OSI 参照モデルのレイヤーの 1 つで、接続によるデータ転送に関するサービスを提供します。トランスポートレイヤーでは、データフローコントロール、ブロッキング、承認などのタスクを扱います。

プロトコルは、物理回線上的内容に関するデータトラフィックの構造体を定義します。特に、オペレーションモード、接続確立時のプロシージャ、データのバックアップ、伝送速度を定義します。

Industrial Ethernet

Industrial Ethernet は、工業環境の中でも最も効率的なサブネットです。工業用イーサネットは工場や小室レベルに適しており、多数のユーザー間による長距離を介した大量のデータ交換を助長します。

Industrial Ethernet は、IEEE 802.3 標準に準拠したオープンな通信ネットワークとして標準化されています。工業用イーサネットの主要な利点は、そのスピード、シンプルな拡張可能性、開放性に加え、高い可用性とワールドワイドな実用性です。また設定プロセスにもあまり手間がかかりません。

下記も参照

タグのコンフィグレーション (ページ 445)

"TCP/IP"チャンネルユニットの接続をコンフィグレーションする方法 (ページ 443)

"TCP/IP"チャンネルユニットの接続をコンフィグレーションする方法

はじめに

WinCC が PLC と通信するには、このチャンネルユニットに加えて論理接続も必要です。特定のパラメータはすべて、論理接続の確立時に定義されます。

オートメーションシステム S7-300 の場合、通信は通信モジュール(CP 343-1 TCP など、S7-400 の場合は CP 443-1 TCP または CP 443-1 IT)経由で行われます。

WinCC では、CP 1613 A2 などの通信プロセッサが使用されます。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

注記

S7-300/S7-400:CPU のラック/スロット番号

外部通信モジュールで S7-300 または S7-400 を使用する際には、CPU のラック/スロット番号を入力する必要があります。

間違ったラック番号またはスロット番号を入力すると、通信リンクが確立されません。

必要条件

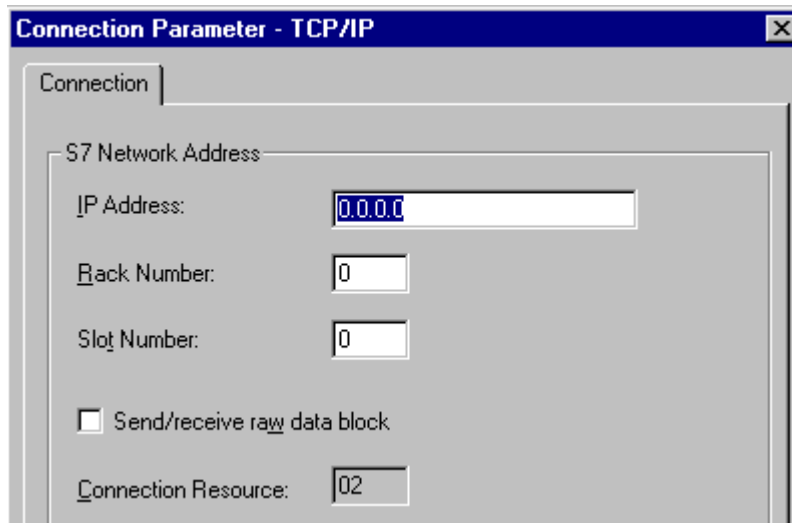
- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。

手順

1. [TCP/IP]チャンネルユニットのショートカットメニューで[新規接続]エントリを選択します。新しい接続が作成されます。
2. 接続名として"Test_TCP"と入力します。

6.11 SIMATIC S7 Protocol Suite

3. ショートカットメニューで[接続パラメータ]接続を選択します。
[接続パラメータ - TCP/IP]ダイアログが開きます。



4. [IP アドレス] フィールドに、バス上のオートメーションシステムのインターネットプロトコルアドレスを入力します。
5. [ラック番号]フィールドに、アドレス指定される CPU が配置されているラックの番号を入力します。
6. 指定したラックの CPU のスロット番号を、対応する[スロット番号]フィールドに入力する必要があります。
7. この接続を使用して BSEND/BRCV データブロックを転送したい場合は、[未処理データブロックの送信/受信]チェックボックスを有効にします。
このチェックボックスが有効な場合は、[接続リソース]フィールドも有効になります。
接続リソースに対しては、16 進数値を入力してください。
この接続リソースは、PLC で接続を設定する際に STEP7 によって割り付けられます。
8. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

タグのコンフィグレーション (ページ 445)

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

6.11.4.7 タグのコンフィグレーション

タグのコンフィグレーション

はじめに

以下のセクションでは、タグをコンフィグレーションする方法について説明します。これは PLC のデータエリアへアクセスする方法や WinCC タグのデータタイプとは異なります。

チャンネル、接続、タグの診断に関する詳細情報については、「通信診断」を参照してください。

下記も参照

テキストタグのコンフィグレーション方法 (ページ 449)

ワード単位のアクセスによるタグのコンフィグレーション方法 (ページ 448)

バイト単位のアクセスによるタグのコンフィグレーション方法 (ページ 446)

ビット単位でアクセスするためのタグのコンフィグレーション方法 (ページ 445)

ビット単位でアクセスするためのタグのコンフィグレーション方法

はじめに

このセクションでは、PLC のアドレスエリアにビット単位でアクセスするためのタグのコンフィグレーション方法を説明します。


必要条件

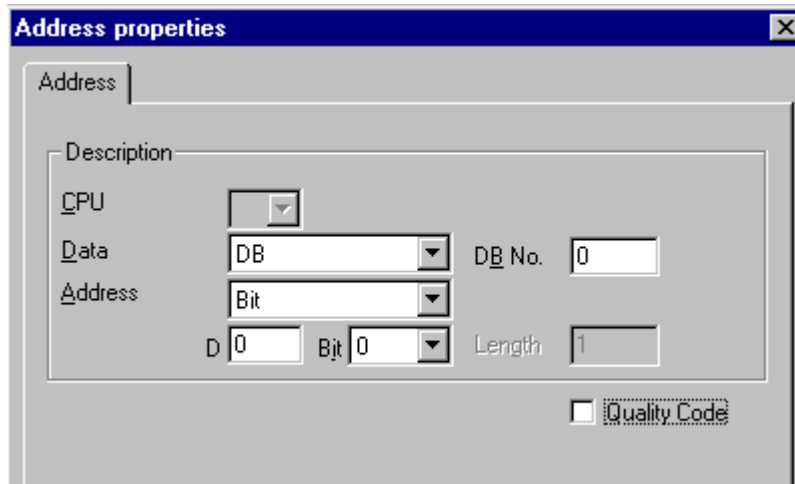
- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- "工業用 Ethernet"などのチャンネルユニットに"Test_Ind_Eth"などの接続を作成しておくこと。

手順

1. 接続[Test_Ind_Eth]を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
[名前]フィールドに、タグ用の名前として"ETH_Var1_bit"を入力します。

6.11 SIMATIC S7 Protocol Suite

4. [データタイプ]フィールドで、データタイプとして[2 進タグ]を設定します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
[データエリア]で、データ指定されるオートメーションシステムのデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。



6. [アドレス]フィールド内の[ビット]エントリは、WinCC タグの[2 進変数]データタイプによって定義されているため、修正できません。
7. 下の 2 つのフィールドにバイトアドレスとビットアドレスを入力します。この左側のフィールドのラベルは、[データエリア]フィールド内のエントリによって異なります(例: データエリア[DB]およびタイプ[2 進変数]の場合は[D])。
8. タグに品質コードがあり、WinCC で使用する場合は、[品質コード]チェックボックスを有効にします。この場合、コードは PLC にも存在しなければなりません。チェックボックスは、データエリアに[DB]が選択されている場合のみ有効にします。
9. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

バイト単位のアクセスによるタグのコンフィグレーション方法


はじめに

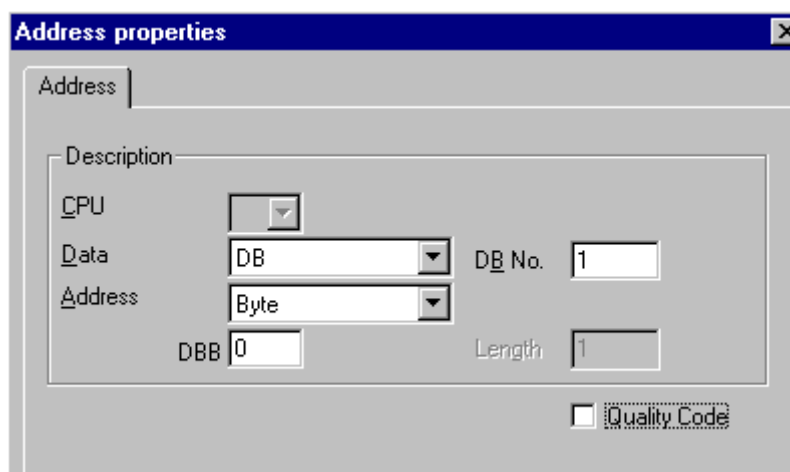
このセクションでは、PLC のアドレスエリアにバイト単位でアクセスするためのタグのコンフィグレーション方法を説明します。

必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- "工業用 Ethernet"などのチャンネルユニットに"Test_Ind_Eth"などの接続を作成しておくこと。

手順

1. 接続[Test_Ind_Eth]を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
[名前]フィールドに、タグ用の名前として"ETH_Var1_byte"を入力します。
4. [データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]に設定します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
[データエリア]で、データ指定されるオートメーションシステムのデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。



6. [アドレス]フィールド内の[ビット]エントリは、WinCC タグの[符号なし 8 ビット値]データタイプによって定義されているため、修正できません。
7. 下のフィールドにバイトアドレスを入力します。この左側のフィールドのラベルは、[データエリア]フィールド内のエントリによって異なります(例: データエリア[DB]およびタイプ[符号なし 8 ビット値]の場合は[D])。
8. タグに品質コードがあり、WinCC で使用する場合は、[品質コード]チェックボックスを有効にします。この場合、コードは PLC にも存在しなければなりません。チェックボックスは、データエリアに[DB]が選択されている場合のみ有効にします。
9. [OK]ボタンをクリックして、両方のダイアログを閉じます。

下記も参照

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

ワード単位のアクセスによるタグのコンフィグレーション方法

はじめに

このセクションでは、PLC のアドレスエリアにワード単位でアクセスするためのタグのコンフィグレーション方法を説明します。


この手順は、長さが 4 バイト("ダブルワード")以上のタグにも適用されます。

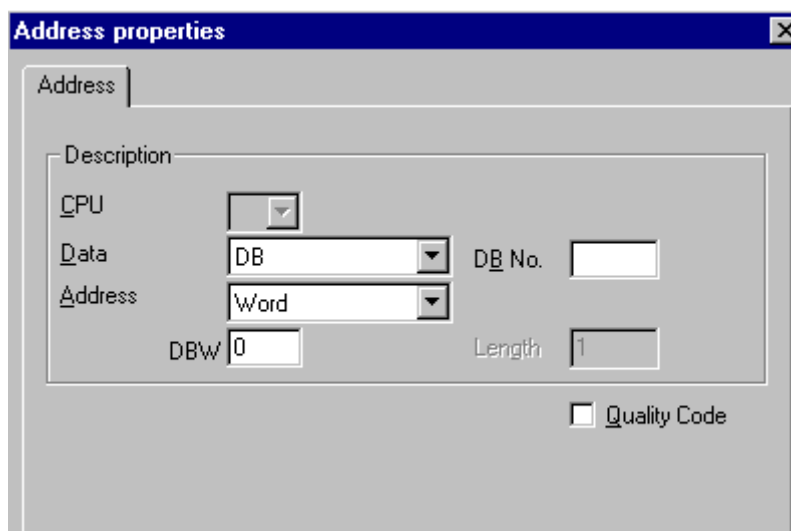
必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- "工業用 Ethernet"などのチャンネルユニットに"Test_Ind_Eth"などの接続を作成しておくこと。

手順

1. 接続[Test_Ind_Eth]を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
[名前]フィールドに、タグ用の名前として"ETH_Var3_word"を入力します。
4. [データタイプ]フィールドで、データタイプを[符号なし 16 ビット値]に設定します。

5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
[データエリア]で、データ指定されるオートメーションシステムのデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。



6. [アドレス]フィールド内の[ワード]エントリは、WinCC タグの[符号なし 16 ビット値]データタイプによって定義されているため、修正できません。
7. [アドレス指定]フィールドには、アドレスの数値を入力します。この左側のフィールドのラベルは、[データエリア]フィールド内のエントリによって異なります(例: タイプ[符号なし 16 ビット値]の場合は[DBW])。
8. タグに品質コードがあり、WinCC で使用する場合は、[品質コード]チェックボックスを有効にします。この場合、コードは PLC にも存在しなければなりません。チェックボックスは、データエリアに[DB]が選択されている場合のみ有効にします。
9. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

テキストタグのコンフィグレーション方法

はじめに

このセクションではテキストタグのコンフィグレーション方法について説明します。

SIMATIC S7 Protocol Suite チャンネルのテキストタグについては、WinCC ではコントロールワードと文字列の実際のユーザデータで構成される S7 文字列タイプのみをサポートしています。


- WinCC でテキストタグを構成するには、PLC メモリにあるコントロールワードのアドレスをユーザデータの前に入力します。コントロールワードの第 1 バイトにはカスタマイズされた最大文字列長が格納され、第 2 バイトには実際の長さが格納されます。
- PLC メモリにデータ構造体を入力する場合、WinCC でテキストタグ用に構成された長さは、2 バイトのコントロールワードが増加することに注意してください。テキストタグのデータ構造体がメモリ内で次々と直接入力されると、後続データが上書きされます。
- V5.0 より前のバージョンではテキストタグのコンフィグレーション時にユーザデータのアドレスが入力されるのに対し、バージョン V5.0 以降からはコントロールワードのアドレスを入力する必要があるため、PCS-7 のバージョンを V4.01 から V5.0 SP1 に切り替える際には、マッピングを再度実行する必要があります。
- 読取り時には、コントロールワードとユーザデータと一緒に読み取られ、第 2 バイトにある現在の長さが評価されます。第 2 コントロールバイトに格納された現在の長さに対応するユーザデータのみが、WinCC テキストタグに転送されます。
- 書込み時には、文字列の実際の長さが決定され("0"文字)、現在の長さが格納されているコントロールバイトがユーザデータと一緒に PLC に渡されます。

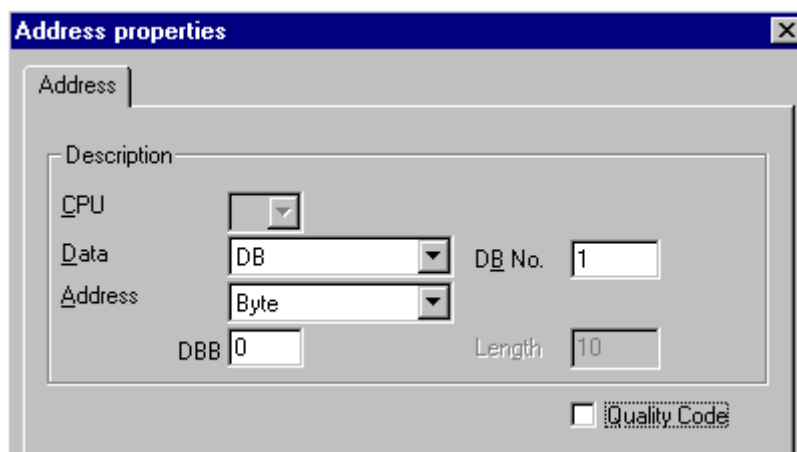
必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- "工業用 Ethernet"などのチャンネルユニットに"Test_Ind_Eth"などの接続を作成しておくこと。

手順

1. 接続[Test_Ind_Eth]を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
[名前]フィールドに、タグ用の名前として"ETH_Var3_Text"を入力します。
4. [データタイプ]フィールドで、データタイプとして[テキストタグ 8 ビットフォント]を設定します。

5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
[データエリア]で、データ指定される PCL のデータエリアを設定します。[DB]をデータエリアに選択した場合、有効[DB 番号]フィールドにデータブロック番号を入力します。



6. [アドレス指定]フィールドのエントリは、[バイト]または[ワード]にしか変更できません。これは、これらのエントリが WinCC タグのデータタイプ[テキストタグ 8 ビットフォント]によって決定されているためです。
7. [アドレス指定]フィールドには、アドレスの数値を入力します。コントロールワードのアドレスを入力します。この左側のフィールドのラベルは、[データエリア]フィールド内のエントリによって異なります(例: タイプ[ワード]の場合は[DBW])。
8. タグに品質コードがあり、WinCC で使用する場合は、[品質コード]チェックボックスを有効にします。この場合、コードは PLC にも存在しなければなりません。チェックボックスは、データエリアに[DB]が選択されている場合のみ有効にします。
9. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

"SIMATIC S7 Protocol Suite"チャンネルをコンフィグレーションする方法 (ページ 415)

6.11.4.8 システムパラメータ

チャンネルユニットのシステムパラメータ

はじめに

WinCC の標準設定とは異なるコンフィグレーションが必要な場合は、チャンネルユニットの[システムパラメータ]ダイアログを使用して、必要な変更をすべて実行することができます。

以下のものを修正できます。

- 論理デバイス名
- AS でのサイクリック読取りサービスの使用

論理デバイス名

WinCC とオートメーションシステム間の通信は、論理デバイス名を介して実行されます。これらの名前は通信カードのインストール中に割り当てられ、ユニット固有のものです。このフィールドには、デフォルトのユニット固有エン트리(例:チャンネルユニット"マルチポイントインターフェース"の場合は"マルチポイントインターフェース")が指定されます。

PLC でのサイクリック読取りサービスの使用

S7-PLC のサイクリック読取りサービス(サイクリックタグサービスとも呼ばれる)を使用するかどうかを指定できます。これらのサイクリック読取りサービスは、周期的に読み取られるタグを個々の要求にグループ化して、それらを PLC に転送します。PLC は、要求を受け取った直後と、サイクル時間が経過するたびに必要なデータを転送します。

サイクリック読取りサービスが有効になっている場合、修正転送も使用することができます。この場合、値を変更した時のみデータが転送されます。このファンクションは PLC によってサポートされている必要があります。

注記

SIMATIC S7 およびユニットのタブ上のシステムパラメータは、ユニット固有なので、チャンネルユニット内で各チャンネルごとに別々に設定できます。

下記も参照

論理デバイス名の変更方法 (ページ 456)

システムパラメータのコンフィグレーション方法 (ページ 454)

PLC のサイクリック読み取りサービス (ページ 453)

PLC のサイクリック読み取りサービス

はじめに

"SIMATIC S7 Protocol Suite"チャンネルのシステムパラメータでは、S7-AS のサイクリック読み取りサービス(AS サイクリックタグサービスとも呼ばれる)を使用すべきかどうかを指定できます。これらのサイクリック読み取りサービスは、周期的に読み取られるタグを個々の要求にグループ化して、それらを PLC に転送します。PLC は、要求を受け取った直後と、サイクル時間が経過するたびに必要なデータを転送します。要求されたデータが必要でなくなったときには(画面が変更された場合など)、WinCC により PLC 内のサイクリック読み取りサービスが削除されます。

通常は、PLC でサイクリック読み取りサービスが使用されます。このため、チャンネルユニットのシステムパラメータでは、対応するチェックボックスがすでに有効になっています(デフォルト設定)。この設定は、サイクリックサービスを使用したくない場合にのみ変更してください。

サイクリック読み取りサービスが有効になっている場合、修正転送のみ使用することができます。この場合、値を変更した場合のみ、AS サイクルごとに 1 度だけ AS からデータが転送されます。このファンクションは PLC によってサポートされている必要があります。

読み取り要求は継続して AS に送られて処理される必要がないため、サイクリック読み取りサービスおよび修正転送を使用すれば、AS と AS-OS ともに通信が緩和されます。

非周期読み取りサービスの場合は、読み取られるタグが個々の要求に結合され、PLC に転送されます。PLC は、必要なデータを 1 度だけ送信します。要求のサイクルは、WinCC によって形成されます。

CPU でのサイクリック読み取りサービスの数

サイクリック読み取りサービスの数は、S7-PLC で使用可能なリソースによって決まります。S7-300 max.では、最大 4 つのサイクリックサービスが使用可能で、S7-416 または S7-417 では、最大 32 のサイクリックサービスが可能です。この数字は、PLC と通信するすべての人に適用されます。すなわち、複数の WinCC システムが S7-PLC と通信している場合は、使用可能なリソースを共有する必要があります。リソースの最大数を超えると、それ以上のサイクリック読み取りサービスへのアクセスが拒否されます。この場合、WinCC は非周期読み取り要求を使用してこのデータを要求するとともに、サイクル形成を実行しなければなりません。

スクリプトでの外部タグの要求

選択しているピクチャに、"GetTagWord()"ファンクションを使用して外部タグを要求するスクリプトが含まれていない場合は、サイクリック読み取りサービスを利用しても、そのピクチャを開いた後の初期更新には影響しません。ピクチャが開くときに"GetTagWord()"を

6.11 SIMATIC S7 Protocol Suite

使用してスクリプトが実行される場合は、このスクリプトに誤ったコンフィグレーションがあると、ピクチャの変更に従って新しいタグ要求がこのチャンネルに繰り返し送信される可能性があります。スクリプトで外部タグが要求されている場合は、トリガイイベントとして"タグ"を入力する必要があります。

システムパラメータのコンフィグレーション方法

はじめに

このセクションでは、"SIMATIC S7 Protocol Suite"チャンネルのシステムパラメータをコンフィグレーションする方法について説明します。

[システムパラメータ]ダイアログは、以下の 2 つのタブで構成されます。

- SIMATIC S7 タブ
- 単位タブ

SIMATIC S7 およびユニットのタブ上のシステムパラメータは、ユニット固有なので、チャンネルユニット内で各チャンネルごとに別々に設定できます。

これらのタブは、S7 チャンネルのすべてのユニットで同一です。したがって、チャンネルユニット"マルチポイントインターフェース"用のダイアログをすべての例で使用します。

パラメータ値に対する変更は、WinCC の再起動後に有効になります。

注記

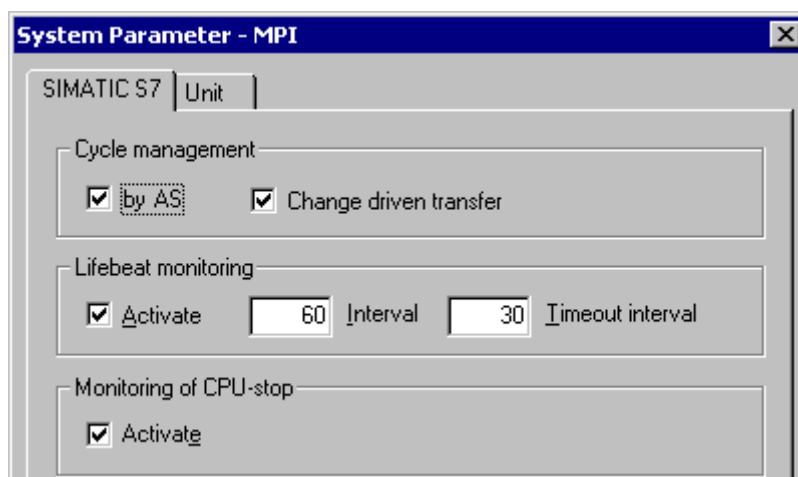
プロジェクトを別のコンピュータにコピーする際には、[ユニット]タブ上の設定は保持されますが、[SIMATIC S7]タブ上の設定は保持されません。

前提条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。

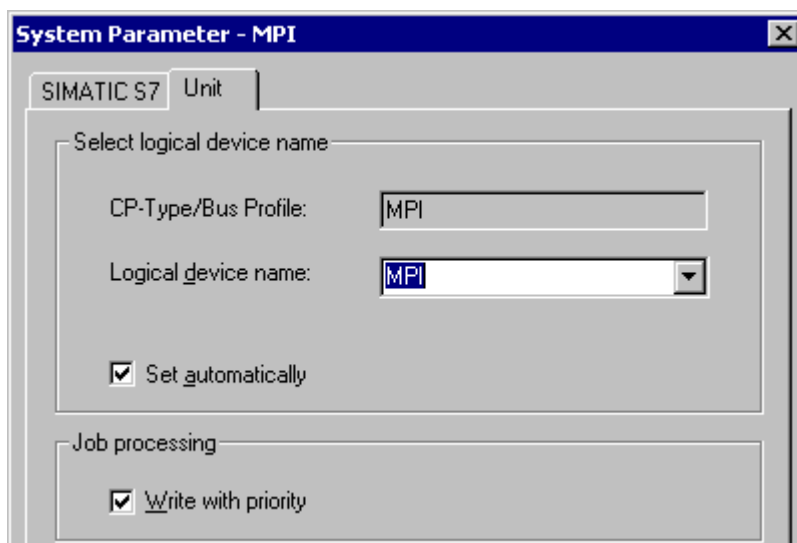
手順

1. [タグ管理]で[SIMATIC S7 Protocol Suite]チャンネルを選択します。必要なチャンネルユニットのポップアップメニューを使用して、[システムパラメータ]ダイアログを開きます。
2. [SIMATIC S7]タブを選択します。チャンネルによるタグのサイクリック読取りを有効にして、修正転送を使用する場合は、[AS による]および[操作中の転送を修正]チェックボックスにチェックを入れます。使用可能であれば、PLC のサイクリックサービスがここで使用されます。詳細については、「PLC のサイクリック読取りサービス」を参照してください。



3. ライフビートモニタリングのファンクションを使用する場合は、[ライフビートモニタリング]エリアの[有効化]チェックボックスを有効にしてください。[間隔]フィールドに、ライフビートテレグラムの転送時間間隔値(秒数)を入力します。
[モニタ時間]フィールドに、ライフビートテレグラムの応答をモニタする時間(秒数)を入力します。
4. WinCC が、S7-CPU が停止ステータスになっているときに通信エラーを示す必要がある場合は、[CPU 停止モニタ]エリアの[有効化]チェックボックスを有効にします。

5. [ユニット]タブを選択します。[論理デバイス名]フィールドに表示される名前は、インストールされている通信プロセッサによって異なります。この名前を変更するのは、通信プロセッサのインストール時に別の名前を選択した場合だけにしてください。詳細については、「論理デバイス名の変更」を参照してください。



6. この通信タイプに対して通信プロセッサが 1 つしかインストールされていない場合で、ランタイムの開始時に自動的にデバイス名を設定する必要がある場合は、[自動設定]チェックボックスを有効にします。
7. 書き込み要求処理を読取り要求処理よりも優先する必要がある場合は、[書き込み優先]チェックボックスを有効にします。
8. [OK]ボタンを押してダイアログを閉じます。

下記も参照

論理デバイス名の変更方法 (ページ 456)

PLC のサイクリック読み取りサービス (ページ 453)

論理デバイス名の変更方法

はじめに

S7 との通信は、論理デバイス名を介して実行されます。これらの名前は通信プロセッサのインストール中に割り当てられ、ユニット固有のものです。

インストール済みの通信プロセッサによって決まるデバイス名に対しては、いくつかの事前設定が確立されています。下記の「デフォルトのデバイス名」表に、これらの名前をリストします。

タブは S7 チャンネルのすべてのユニットで同一なので、この説明ではチャンネルユニット"マルチポイントインターフェース"用のダイアログを示します。

デフォルトのデバイス名

チャンネルユニット	デフォルトのデバイス名
工業用イーサネット	CP_H1_1:
工業用イーサネット(II)	CP_H1_2:
MPI	MPI
名前指定接続	VM/
PROFIBUS	CP_L2_1:
PROFIBUS(II)	CP_L2_2:
スロット PLC	SLOT_PLC
ソフト PLC	SOFT_PLC
TCP/IP	CP-TCPIP

前提条件

- "SIMATIC S7 P rotocol Suite"チャンネルをプロジェクトに統合してください。
- チャンネルユニット(例:"マルチポイントインターフェース")との接続を作成して下さい。

手順

1. タグ管理で、必要なチャンネルユニットを選択します。
2. ポップアップメニューを使用して、[システムパラメータ]ダイアログウィンドウを開きます。
3. [ユニット]タブを選択します。
4. [論理デバイス名]フィールドにデバイス名を指定します。選択リストからエントリを選択するか、新規の名前を手動で入力することができます。
使用可能な名前はすべて、"PG/PC インターフェースのコンフィグレーション"(コントロールパネル)ツールによって決定されます。このツールがインストールされていない場合は、現在設定されているデバイス名のみが表示されます。別の論理デバイス名を指定した場合は、メッセージが表示されます。
手動で入力を行うのは、コンフィグレーション側のステーションにインストールされていない通信カードをターゲットステーションが使用している場合だけにしてください。
5. [OK]ボタンをクリックして、ダイアログを閉じます。

注記

論理デバイス名は、デバイス設定と完全に一致しなければなりません。この場合、たとえば"工業用イーサネット"や"PROFIBUS"に対するデフォルトの論理デバイス名には、名前の末尾にコロンが付きます。

パラメータ値に対する変更は、WinCC の再起動後に有効になります。

6.11.5 特殊ファンクション

6.11.5.1 "SIMATIC S7 Protocol Suite"チャンネルの特殊ファンクション

はじめに

SIMATIC S7 Protocol Suite はいくつかの特殊ファンクションを持っています。この章ではこれらのファンクションについて説明します。

下記も参照

ソフトウェアの二重化 (ページ 502)

"SIMATIC S7 Protocol Suite"チャンネルの未処理データタグ (ページ 493)

S7 ファンクションブロック AR_SEND によるデータ交換 (ページ 458)

6.11.5.2 S7 ファンクションブロック AR_SEND によるデータ交換

S7 ファンクションブロック AR_SEND によるデータ交換

はじめに

S7-400 AS のファンクションブロック AR_SEND が、プロセス値をプロセス値アーカイブに転送するために使用されます。

動作原理

PLC プロセス値を WinCC のプロセス値アーカイブに転送するには、S7-400 PLC に SFB 37 "AR_SEND"と呼ばれる統合ファンクション構成要素がなければなりません。

AR_SEND 構成要素の基本ファンクションは、データをアーカイブタグに供給できます。AR_ID-Subnumber を使用すると、データを複数のタグに供給できます。AR_SEND 構成要素を使用する場合、プロセス値はアーカイブに個々に送信されるのではなく、PLC に蓄積されてからパケットで転送されます。これにより、使用されるネットワーク上の負荷を軽減します。

PLC では、AR_SEND 構成要素の CPI に従属した番号(たとえば CPU 416 max. 32 AR_SEND)を使用できます。AR_ID を、順に各 AR_SEND 構成要素に割り当てることができます。各 AR_ID には 4095 までのサブ番号を使用できるため、転送可能なプロセスデータの量を増やすには、サブ番号を使用します。

実際には、AR_SEND 構成要素ごとのアーカイブタグの数は、転送するデータ領域の最大の長さまでに制限されます。"データブロック構造の構造とパラメータ"についての詳細情報は、"プロセス値の数"パラメータの説明を参照してください。

AR_ID および AR_ID サブ番号は、PLC 内のデータとアーカイブタグへの割り当てを確立するために使用され、PLC 内のデータベースのデータ構造を設定する間に、その他のパラメータを使用して定義されます。

この割り当ては WinCC で設定されますが、その他のパラメータは自動的に評価されます。

WinCC における設定は PLC 内のこれらの値がベースとなるので、PLC で SFB 37 "AR_SEND"を最初に設定して、データブロックを構築しておく必要があります。AR_SEND ファンクション構成要素の設定の詳細は、S7-400 PLC のマニュアルを参照してください。

AR_SEND 変異形の概要

変異形： AR_SEND 用...	AR_SEND ごとのプロセスコントロールされたアーカイブタグの数	使用目的
1 つのアーカイブタグ	1	アーカイブタグのプロセス値を転送するため。プロセス値も非常に短い時間間隔で読み取ることができます。
複数のアーカイブタグ	AR_ID サブ番号の数に対応	複数のアーカイブタグのプロセス値を転送するため。プロセス値も非常に短い時間間隔で読取ることができます。
複数のアーカイブタグ(最適化)	AR_ID サブ番号の数に対応	最大数のアーカイブタグにサイクリックデータを、一度にそれぞれに 1 つずつのデータを供給するために、使用します。

下記も参照

アーカイブタグが複数の場合の AR_SEND 変数形 (ページ 475)

アーカイブタグが複数の場合に AR_SEND 変数形をコンフィグレーションする方法 (ページ 491)

アーカイブタグが 1 つの場合に AR_SEND 変数形をコンフィグレーションする方法 (ページ 489)

アーカイブタグが複数の場合の AR_SEND 変数形 (最適化) (ページ 487)

アーカイブタグが 1 つの場合の AR_SEND 変数形 (ページ 470)

データブロック - 構造体とパラメータ

はじめに

「AR_SEND」ファンクションブロックを使用してデータが転送される前に、AS で 1 つ以上のデータブロックとして転送対象のデータが提供されます。データブロックの構造は、各種のパラメータによって異なります(たとえば、使用する AR_SEND 変数形で、タイムスタンプまたはプロセス値のデータタイプを使用)。

以下に、データブロックで使用されるパラメータについて説明します。個々のパラメータ値は、AS 内のデータブロックおよび「AR_SEND」ファンクションブロックのパラメータ化で設定されます。

パラメータ化は、WinCC でデータブロックが評価されるときにチェックされます。WinCC がデータブロックの構造の中でエラーを検出するか、アーカイブタグの設定と受信データが一致しない場合は、WinCC 診断ログブックに、以下の構造を持つエントリが記録されます。

「日付、時間、1003080、4、ユーザー名、コンピュータ名、NRMS7PMC、PdeReceive:接続 connectionname から不明のパラメータ AR_SEND ...+エラーについて説明しているその他の情報」

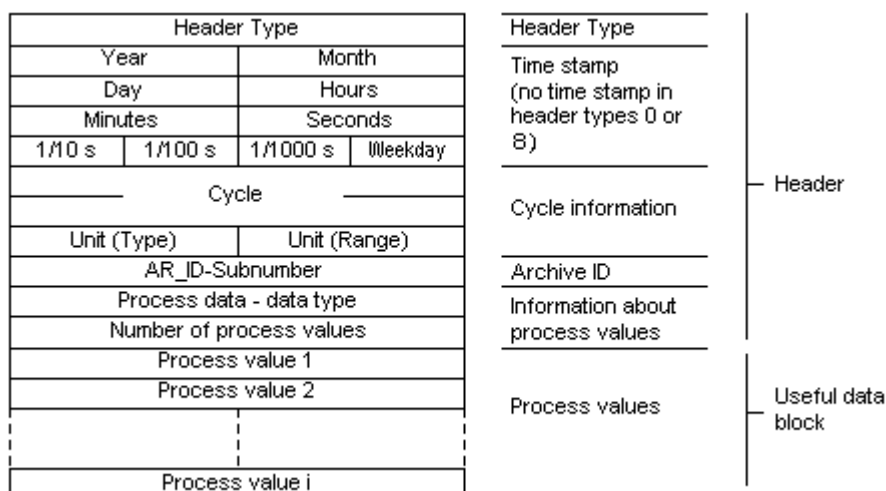
メッセージシステムが WinCC システムメッセージで設定されている場合、この診断エントリは 1003080 と番号の付いた OS プロセスコントロールメッセージもトリガします。ログブックエントリのテキストは、このメッセージのコメントで見つかります。

データブロックの構造

各データブロックは、ヘッダーとユーザーデータエリアで構成されています。

- ヘッダーには、プロセス値およびそのサイクル、場合によってはタイムスタンプに関する情報が含まれます。
- ユーザーデータエリアには、実際のプロセス値が含まれます。

転送されるデータエリアは、1 つ以上のデータブロックで形成されます。



注記

データブロックでは、各行が 2 バイトを表します。プロセス値の長さは、データタイプによって 1 バイトか複数バイトになります。詳細については、「プロセス値の数」パラメータの説明を参照してください。

パラメータの内容

ヘッダータイプ

ヘッダータイプは、ヘッダーに含まれる情報のタイプを定義します。

ヘッダー タイプ	タイムスタンプ	AR_ID サブ番号
0	タイムスタンプなしのヘッダー	AR_ID サブ番号なしのヘッダー
1	タイムスタンプのあるヘッダー	AR_ID サブ番号なしのヘッダー
8	タイムスタンプなしのヘッダー	AR_ID サブ番号のあるヘッダー
9	タイムスタンプのあるヘッダー	AR_ID サブ番号のあるヘッダー

注記

ヘッダータイプ 0 および 8 の場合、タイムスタンプのバイトはヘッダーに含まれません。これらのバイトはデータブロック内でも予約されないため、ヘッダーは 8 バイトだけ短くなります。

AR_ID サブ番号

AS ユーザーデータと WinCC アーカイブタグ間の割り付けを指定し、2 つの位置で設定されます。

- WinCC では、プロセスコントロールされたアーカイブタグの設定時
- PLC では、転送するユーザーデータエリアの設定時

サブ番号は、ヘッダータイプ 8 または 9 にのみ関係します。サブ番号に有効な値の範囲は 1 ～4095 です。パラメータは、16 進数値(1 - 0FFF)として WinCC で出力されます。

タイムスタンプ

タイムスタンプには、日付と時刻が SIMATIC S7 BCD フォーマットで格納されます。週日エントリは WinCC では評価されません。

注記

夏時間/標準時間:オートメーションシステムの時間設定

オートメーションシステム S7 は、夏時間/標準時間の切り替えを認識しません。タイムスタンプの夏時間または標準時間への修正は、標準 DLL によって WinCC で実行されます。こうして修正された時間および夏時間/標準時間の ID は、WinCC アプリケーションで使用可能になります。修正された時間および ID は、タグロギングなどのアーカイブに追加されます。

正確な時間設定を確認するため、AS で[WinCC エクスプローラ]>[コンピュータプロパティ]>[パラメータ]>[PLC 時間設定]の WinCC コンピュータプロパティと同じ時間設定を選択します。

サイクル

プロセス値が読み取られるサイクル。このパラメータは、単位(範囲)で指定された時間単位のファクタです。データの長さ:ダブルワード。

たとえば、

「サイクル」 = 10、「単位(範囲)」 = 4 には次の意味があります。プロセス値の読み取りサイクル = 10 秒

単位(タイプ)

時間情報のタイプを指定し、[プロセス値の数]パラメータに影響します。

番号	意味
1	プロセス値は等しい時間間隔で読み取られます。 開始時間はヘッダーのタイムスタンプに入力されており、必須です。プロセス値の間の時間間隔は、[単位(範囲)]およびファクタ[サイクル]の時間単位により定義されます。
2	各プロセス値には、タイムスタンプがあります。 ヘッダーで割り付けられている可能性のあるタイムスタンプは評価されません。 フォーマットは、8 バイトの長さのヘッダーのタイムスタンプに対応しています。

6.11 SIMATIC S7 Protocol Suite

番号	意味
3	各プロセス値には、2 ワードのデータ長の時間単位での相対時間差があります。絶対時間は、ヘッダー(=開始時間)のタイムスタンプと、[単位(範囲)]に設定される時間単位の相対時間差の、合計です。ヘッダーのタイムスタンプエントリは必須です。
4	各プロセス値には AR_ID サブ番号が含まれます。ヘッダーで与えられるタイムスタンプが、プロセス値に適用されます。ヘッダーのタイムスタンプエントリは必須です。

単位(範囲)

単位(タイプ)用に使われる時間の単位を指定する = 1 または 3。

番号	意味
1	予備
2	予備
3	ミリ秒
4	秒
5	分
6	時間
7	日

プロセスデータのデータタイプ

プロセス値は S7 フォーマットに直接格納されます。

番号	S7 データタイプ	WinCC データタイプ
0	BYTE	BYTE
1	WORD	WORD
2	INT	SWORD
3	DWORD	DWORD
4	DINT	SDWORD
5	REAL	FLOAT

プロセス値の数

転送されたデータエリアには特定の数のプロセス値が含まれ、これは[単位(タイプ)]のエントリによって異なります。この数は、転送されたデータエリアの最大の長さ(16 キロバイト)に制限されます。

S7 ファンクション「AR_SEND」および「BSEND/BRCV」を使用して S7-400 と通信する際のリソースの制限が考慮に入れられます。これは、AR_SEND ファンクションや BSEND/BRCV ファンクションを使用して AS から WinCC に一度に送信できるデータの最大量は 16 KB であることを意味しています。

注記

AR_SEND 変数形「複数のアーカイブタグ」の場合、このパラメータには以下の制限が適用されます。

さまざまなアーカイブタグのデータブロックは、必ずワードバウンダリで始まる必要があります。したがって、「データタイププロセス値」=0 (BYTE)と「単位(タイプ)」=1 (等しく空いている時間間隔のプロセス値)の組み合わせの場合、偶数のプロセス値(=バイト)が「プロセス値の数」パラメータに入力される必要があります。この制限は、この AR_SEND 変数形、およびデータタイプと「単位(タイプ)」のこの組み合わせの場合にのみ、適用されます。

例:

1x 最大 16 KB の BSEND

または 1x 8 KB の AR_SEND + 1x 8 KB の BSEND

または 1x 10 KB の AR_SEND + 1x 2 KB の AR_SEND + 1x 4 KB の BSEND

単位 (タイプ)	プロセス値の数の意味
1	等間隔でのプロセス値の読み取り: WORD または INT のデータタイプのプロセス値は最大で 8000、または DWORD、DINT または REAL のデータタイプでは 4000 の値を転送できます。
2	タイムスタンプ付きのプロセス値: ユーザーデータエリアの各エレメントは、タイムスタンプ(8 バイト)と値で構成されます。したがって、WORD または INT のデータタイプのプロセス値は最大で 1600、また DWORD、DINT または REAL のデータタイプでは 1333 の値を転送できます。

単位 (タイプ)	プロセス値の数の意味
3	時間差付きのプロセス値: ユーザーデータエリアの各エレメントは、タイムスタンプ(4 バイト)と値で構成されます。したがって、WORD または INT のデータタイプのプロセス値は最大で 2666、また DWORD、DINT または REAL のデータタイプでは 2000 の値を転送できます。
4	プロセス値は AR_ID サブ番号(複数変数の AR-SEND - 最適化)を含みます。 タイプ 4 では、プロセス値は AR_ID サブ番号付き 1 ワード(値範囲: 1-0x0FFF)と 1 つの値で構成されます。このように、ユーザーデータエリアは、AR_ID サブ番号が先行するプロセス値の配列で構成されます。したがって、WORD または INT のデータタイプのプロセス値は最大で 3992、また DWORD、DINT または REAL のデータタイプでは 2660 の値を転送できます。

注記

データブロックで与えられた AR_ID サブ番号は、すべて WinCC で設定される必要があります。設定されていないサブ番号が見つかった場合、WinCC はユーザーデータの解釈を停止します。

さまざまなアーカイブタグのデータブロックは、必ずワードバウンダリで始まる必要があります。したがって、データタイプ BYTE と「単位(タイプ)」= 1 (等しい間隔の時間間隔のプロセス値)では、偶数のプロセス値(=バイト)をパラメータ「プロセス値の数」に入力する必要があります。この制限は、この AR_SEND 変数形、およびデータタイプと「単位(タイプ)」のこの組み合わせの場合にのみ、適用されます。

下記も参照

アーカイブタグが複数の場合に AR_SEND 変数形をコンフィグレーションする方法 (ページ 491)

アーカイブタグが 1 つの場合に AR_SEND 変数形をコンフィグレーションする方法 (ページ 489)

アーカイブタグが複数の場合の AR_SEND 変数形 (最適化) (ページ 487)

アーカイブタグが複数の場合の AR_SEND 変数形 (ページ 475)

アーカイブタグが 1 つの場合の AR_SEND 変数形 (ページ 470)

AR_SEND 変数形のプロパティの概要

はじめに

例の基礎として、表には様々な AR_SEND 変数形のプロパティと可能なパラメータが示されています。

この表は、使用可能な組み合わせのすべてを表示しているわけではありません。

"ヘッダータイプ"列から"プロセス値データタイプ"列までは、ヘッダーに現れる順に表示してあります。

注記

AR_ID サブ番号および AR_ID サブ番号の値は、ファンクションブロック"AR_SEND"および AS 内のデータブロックのデータ構造体のコンフィグレーション時に、その他のパラメータの値と一緒に設定されます。

アーカイブタグの変数形

例/ プロパティ	例- 番号	ヘッダ ー タイプ	データ/時 間 (ヘッダー のタイム スタンプ)	サイク ル ファク タ	ユニ ット (タイ プ)	ユニッ ト (範囲)	AR_ID - サブ番 号	プロセ ス値の データ タイプ	プロセ ス値の 最大 数	例の プロセス値 構造体
1つのタイムスタンプ付き各プロセス値 (バイト)	1	0	存在しない	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 バイトタイムスタンプ +1 バイト プロセス値
等間隔のタイムスタンプ付きのプロセス値	2	1	関係あり	>=1	1	3~7	0	0 1; 2 3; 4; 5	16000 8000 4000	1 ワードの プロセス値

6.11 SIMATIC S7 Protocol Suite

例/ プロパティ	例- 番号	ヘッダ- タイプ	データ/時 間 (ヘッダ- のタイム スタンプ)	サイク ル ファク タ	ユニ ット (タイ プ)	ユニッ ト (範囲)	AR_ID - サブ番 号	プロセ ス値の データ タイプ	プロセ ス値の 最大 数	例の プロセス値 構造体
1つのタイムスタンプ付き各プロセス値 (ワード)	3	1	該当なし	0	2	0	0	0 1; 2 3; 4; 5	3200 1600 1333	8 バイトタイムスタンプ+1 ワードプロセス値
時間差付きの各プロセス値	4	1	関係あり	>=1	3	3~7	0	0 1; 2 3; 4; 5	5332 2666 2000	8 バイトタイムスタンプ+1 バイトプロセス値

複数のアーカイブタグの変数形

例/ プロパティ	例- 番号	ヘッダ- タイプ	データ/時 間 (ヘッダ- のタイム スタンプ)	サイク ル ファク タ	ユニ ット (タイ プ)	ユニッ ト (範囲)	AR_ID - サブ番 号	プロセ ス値の データ タイプ	プロセ ス値の 最大 数	例のプロセ ス値構造体
1つのタイムスタンプ付き各プロセス値 (バイト)	5	8	存在しない	0	2	0	1~ 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 バイトタイムスタンプ+1 バイトプロセス値
等間隔のタイムスタンプ付きのプロセス値	6	9	関係あり	>=1	1	3~7	1~ 4095	0 1; 2 3; 4; 5	16000 8000 4000	1 ワードのプロセス値

例/ プロパティ	例- 番号	ヘッダ- タイプ	データ/時 間 (ヘッダー のタイム スタンプ)	サイク ル ファク タ	ユニ ット (タイ プ)	ユニッ ト (範囲)	AR_ID - サブ番 号	プロセ ス値の データ タイプ	プロセ ス値の 最大 数	例のプロセ ス値構造体
1つのタイ ムスタン プ付き各 プロセス 値 (ワー ド)	7	9	該当 なし	0	2	0	1～ 4095	0 1; 2 3; 4; 5	3200 1600 1333	8 バイトタ イムスタン プ +1 ワード プロセス値
時間差付 きの各プ ロセス値	8	9	関係あり	0	3	3～7	1～ 4095	0 1; 2 3; 4; 5	5332 2666 2000	8 バイトタ イムスタン プ +1 バイト プロセス値

複数のアーカイブタグの変数形 - 最適化

例/ プロパティ	例- 番号	ヘッダ- タイプ	データ/時 間 (ヘッダー のタイム スタンプ)	サイク ル ファク タ	ユニ ット (タイ プ)	ユニッ ト (範囲)	AR_ID - サブ番 号	プロセス 値の データタ イプ	プロセス 値の最大 数	例の プロセス値 構造体
AR_ID サ ブ番号付 き各プロ セス値	9	1	関係あり	0	4	0	0	1; 2 3; 4; 5	3992 2660	1 ワードの サブ番号 + 1 ワード のプロセス 値

下記も参照

アーカイブタグが複数の場合の AR_SEND 変数形 (最適化) (ページ 487)

アーカイブタグが複数の場合の AR_SEND 変数形 (ページ 475)

アーカイブタグが 1 つの場合の AR_SEND 変数形 (ページ 470)

アーカイブタグが 1 つの場合の AR_SEND 変数形

はじめに

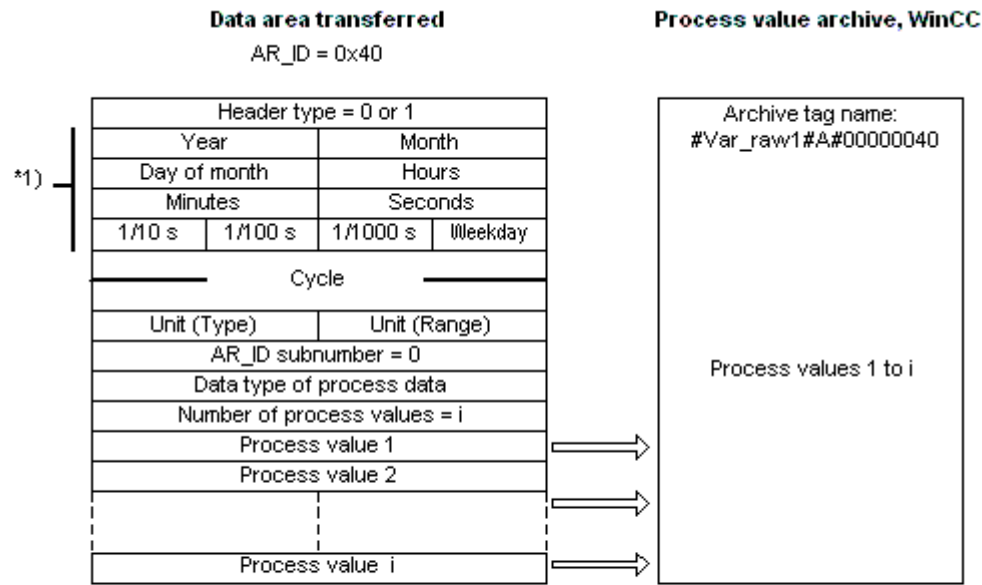
この変数形を使用して、プロセス値とともにアーカイブタグを指定します。旧バージョンの WinCC (V5.0 以前)でも使用できます。

この変数形のプロパティは以下のとおりです。

- ヘッダータイプは、0 または 1 でなければならない(AD_ID Subnumber なしおよびタイムスタンプ付き/なし)。
- ヘッダーの AR_ID Subnumber の値は求められない。
- WinCC の場合、アーカイブタグのプロセス値が転送されるため、アーカイブタグ名には AR_ID Subnumber は含まれない。

データ領域の構造体の例

転送されるデータ領域は、1 つのデータブロックで構成されています。



*1) = time stamp omitted with header type 0

下記も参照

データブロック構造体の例 4:1 つのアーカイブタグ; 相対タイムスタンプ(時間差)付きの各プロセス値 (ページ 474)

データブロック構造体の例 3:1 つのアーカイブタグ; 専用タイムスタンプ付き各プロセス値 (ページ 473)

データブロック構造体の例 2:1 つのアーカイブタグ; 等間隔のタイムスタンプ (ページ 472)

データブロック構造体の例 1:1 つのアーカイブタグ; タイムスタンプ付き各プロセス値 (ページ 471)

AR_SEND 変数形のプロパティの概要 (ページ 467)

データブロック構造体の例 1:1 つのアーカイブタグ; タイムスタンプ付き各プロセス値

はじめに

この例では、プロセス値が 1 つのアーカイブタグのみに対して転送されます。ヘッダーにはタイムスタンプがないので、対応するバイト数は確保されません。従って、各プロセス値 (1 バイト) はタイムスタンプ (8 バイト) が先行します。

プロセス値のデータタイプはバイトです。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 0
2.0	
4.0	Cycle = 0
6.0	Unit (Type) = 2 Unit (Range) = 0
8.0	AR_ID-Subnumber = 0
10.0	Process data - data type = 0
12.0	Number of process values = 3 (max. 3200)
14.0	Year=2001 Month=10
16.0	Day=05 Hours=13
18.0	Minutes=40 Seconds=00
20.0	1/10 s 1/100 s 1/1000 s Weekday
22.0	Process value 1 -
24.0	Year=2001 Month=10
26.0	Day=05 Hours=14
28.0	Minutes=40 Seconds=00
30.0	1/10 s 1/100 s 1/1000 s Weekday
32.0	Process value 2 -
34.0	Year=2001 Month=10
36.0	Day=05 Hours=15
38.0	Minutes=40 Seconds=00
40.0	1/10 s 1/100 s 1/1000 s Weekday
42.0	Process value 3 -

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 2:1 つのアーカイブタグ; 等間隔のタイムスタンプ

はじめに

この例では、プロセス値が 1 つのアーカイブタグに対して転送されます。

1 秒の等間隔のタイムスタンプは、パラメータ"サイクル"=1 および"ユニット(範囲)"=4 (=秒)を使用して作成されます。

プロセス値のデータタイプは WORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 1
2.0	Year=2001
4.0	Month=10
6.0	Day=05
8.0	Hours=13
10.0	Minutes=40
12.0	Seconds=00
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle =1
24.0	Unit (Type) = 1
26.0	Unit (Range) = 4
28.0	AR_ID-Subnumber = 0
30.0	Process data - data type = 1
32.0	Number of process values = 8 (max. 8000)
34.0	Process value 1
36.0	Process value 2
	Process value 3
	Process value 4
	Process value 5
	Process value 6
	Process value 7
	Process value 8

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 3:1 つのアーカイブタグ; 専用タイムスタンプ付き各プロセス値

はじめに

この例では、プロセス値が 1 つのアーカイブタグのみに対して転送されます。ヘッダー内のタイムスタンプは重要ではありません。従って、各プロセス値 (1 ワード) はタイムスタンプ (8 バイト) が先行します。

プロセス値のデータタイプは SWORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 1
2.0	Year=0
4.0	Day=0
6.0	Minutes=0
8.0	1/10 s
10.0	1/100 s
12.0	1/1000 s
14.0	Weekday
16.0	Cycle =0
18.0	Unit (Type) = 2
20.0	Unit (Range) = 0
22.0	AR_ID-Subnumber = 0
24.0	Process data - data type = 2
26.0	Number of process values = 2 (max. 1600)
28.0	Year=2001
30.0	Month=10
32.0	Day=05
34.0	Hours=13
36.0	Minutes=40
38.0	Seconds=00
40.0	1/10 s
	1/100 s
	1/1000 s
	Weekday
	Process value 1
	Year=2001
	Month=10
	Day=05
	Hours=14
	Minutes=40
	Seconds=00
	1/10 s
	1/100 s
	1/1000 s
	Weekday
	Process value 2

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 4:1 つのアーカイブタグ; 相対タイムスタンプ(時間差)付きの各プロセス値

はじめに

この例では、アーカイブタグに対してタイムスタンプ付きでプロセス値が転送されます。パラメータ"ユニット(タイプ)" = 3 を使用して、各プロセス値にはヘッダー内のタイムスタンプと相対する時間差(4 バイト)が付きます。時間差のユニットは、パラメータ"ユニット(範囲)" = 4 を使用して秒に設定されます。

プロセス値のデータタイプは DWORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 1
2.0	Year=2001
4.0	Month=10
6.0	Day=05
8.0	Hours=13
10.0	Minutes=40
12.0	Seconds=00
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle = 0
24.0	Unit (Type) = 3
26.0	Unit (Range) = 4
28.0	AR_ID-Subnumber = 0
30.0	Process data - data type = 3
32.0	Number of process values = 3 (max. 2000)
34.0	Relative time difference in seconds
36.0	Process value 1
38.0	Relative time difference in seconds
40.0	Process value 2
42.0	Relative time difference in seconds
44.0	Process value 3

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

アーカイブタグが複数の場合の AR_SEND 変数形

はじめに

この変数を使用して、1 つ以上のアーカイブタグにプロセス値を指定します。各アーカイブタグでは、AR_ID Subnumber を割り付け、転送されるデータエリア内でデータブロックを作成します。

"x"プロセス値は、各 AR_ID Subnumber 用に転送できます。"データブロック構造の構造体とパラメータ"についての詳細情報は、"プロセス値の数"パラメータの説明を参照してください。

任意の"装置(タイプ)"および"装置(範囲)"に従って、アーカイブタグの値のタイムスタンプを受け取る、あるいは転送されるデータエリアから導き出します。そして WinCC プロセス値アーカイブに送られます。

この変数形のプロパティは以下のとおりです。

- ヘッダータイプは、8 または 9 でなければならない(タイムスタンプ付き/なし、および AR_ID Subnumber 付き)。
- すべての AR_ID Subnumber で、転送されるデータエリア内でデータブロックを作成する必要がある。
- 各データブロックの AR_ID サブ番号は、1 以上でなければならない。
- WinCC では、アーカイブタグ名には AR_ID サブ番号が付いている。

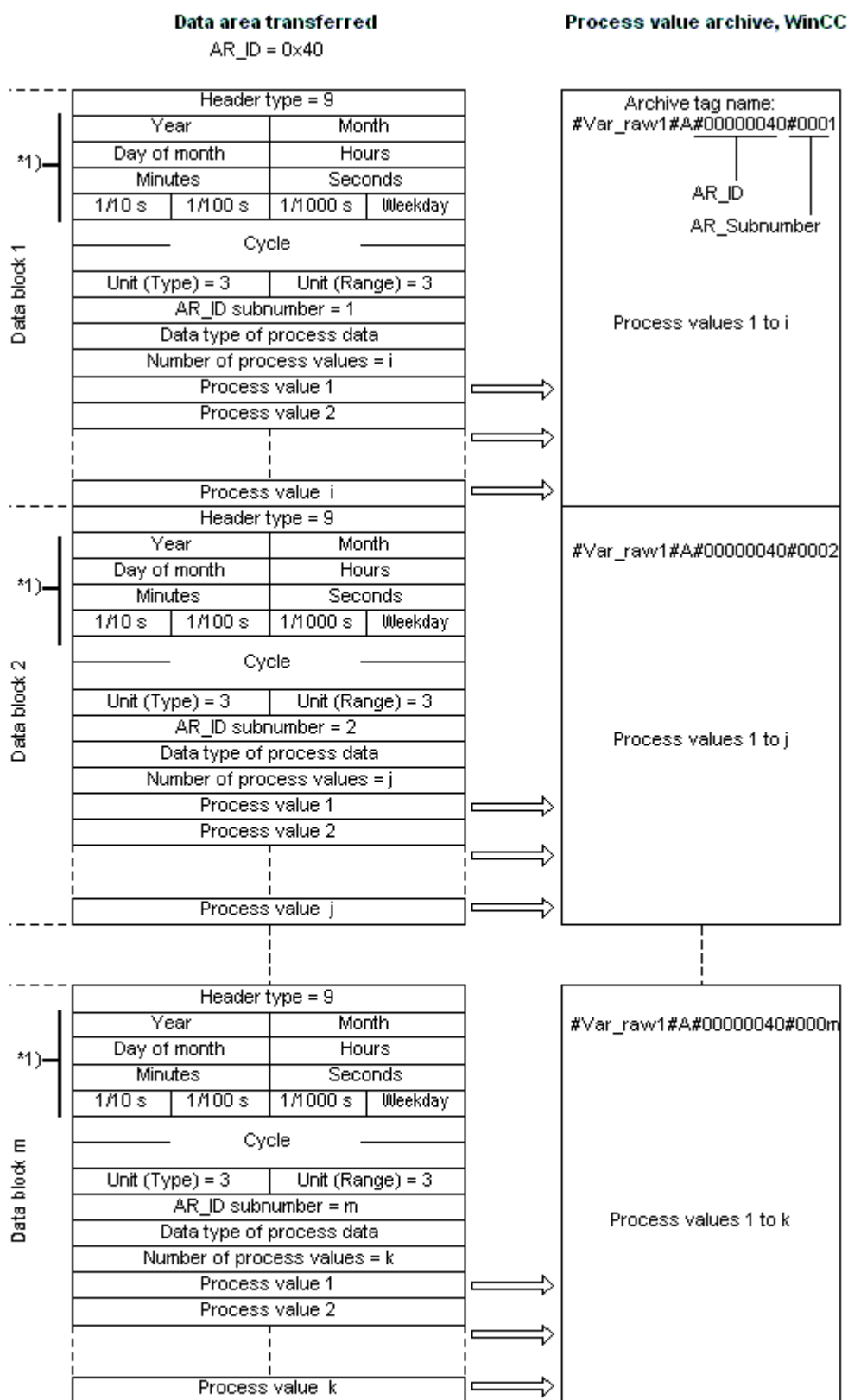
注記

データブロックに与えられた AR_ID サブ番号は、すべて WinCC でコンフィグレーションされる必要があります。コンフィグレーションされていないサブ番号が見つかり、WinCC はユーザーデータのインタープリタを停止します。

さまざまなアーカイブタグのデータブロックは、必ずワードバウンダリで始まる必要があります。したがって、"データタイププロセス値" = 0 (バイト) と "ユニット(タイプ)" = 1 (等しく空いている時間間隔のプロセス値) の組み合わせの場合、偶数のプロセス値(=バイト)が "プロセス値の数" パラメータに入力される必要があります。この制限は、この AR_SEND 変数、およびデータタイプと "装置(タイプ)" のこの組み合わせの場合にのみ適用されます。

データ領域の構造体の例

転送されるデータエリアは、指定されるアーカイブタグの数に対応して、1 つ以上のデータブロックで構成されます。



*1) = omitted with header type 0 or 8

下記も参照

データブロック構造体の例 8:複数のアーカイブタグ; 相対タイムスタンプ(時間差)付きの各プロセス値 (ページ 485)

データブロック構造体の例 7:複数のアーカイブタグ; 専用タイムスタンプ付き各プロセス値 (ページ 483)

データブロック構造体の例 6:複数のアーカイブタグ; 等間隔のタイムスタンプ (ページ 481)

データブロック構造体の例 5:複数のアーカイブタグ; 専用タイムスタンプ付き各プロセス値 (ページ 479)

AR_SEND 変数形のプロパティの概要 (ページ 467)

データブロック構造体の例 5:複数のアーカイブタグ; 専用タイムスタンプ付き各プロセス値

はじめに

この例では、複数のアーカイブタグに対するプロセス値が転送されます。

さまざまなアーカイブタグのデータブロックは、順々に配置されます。データブロックごとに異なる AR_ID-サブ番号が入力されます。

ヘッダーにはタイムスタンプがないので、対応するバイト数は確保されません。従って、各プロセス値 (1 バイト) はタイムスタンプ (8 バイト) が先行します。

プロセス値のデータタイプはバイトです。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 8
2.0	Cycle = 0
4.0	
6.0	Unit (Type) = 2 Unit (Range) = 0
8.0	AR_ID-Subnumber = 1
10.0	Process data - data type = 0
12.0	Number of process values = 3
14.0	Year=2001 Month=10
16.0	Day=05 Hours=13
18.0	Minutes=40 Seconds=00
20.0	1/10 s 1/100 s 1/1000 s Weekday
22.0	Process value 1 -
24.0	Year=2001 Monat=10
26.0	Day=05 Stunden=14
28.0	Minutes=40 Sekunden=00
30.0	1/10 s 1/100 s 1/1000 s Weekday
32.0	Process value 2 -
34.0	Year=2001 Month=10
36.0	Day=05 Hours=15
38.0	Minutes=40 Seconds=00
40.0	1/10 s 1/100 s 1/1000 s Weekday
42.0	Process value 3 -
44.0	Header Type = 8
46.0	Cycle = 0
48.0	
50.0	Unit (Type) = 2 Unit (Range) = 0
52.0	AR_ID-Subnumber = 2
54.0	Process data - data type = 0
56.0	Number of process values = 2
58.0	Year=2001 Month=10
60.0	Day=05 Hours=12
62.0	Minutes=40 Seconds=00
64.0	1/10 s 1/100 s 1/1000 s Weekday
66.0	Process value 1 -
68.0	Year=2001 Month=10
70.0	Day=05 Hours=13
72.0	Minutes=40 Seconds=00
74.0	1/10 s 1/100 s 1/1000 s Weekday
76.0	Process value 2 -

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 6:複数のアーカイブタグ; 等間隔のタイムスタンプ

はじめに

この例では、複数のアーカイブタグに対するプロセス値が転送されます。さまざまなアーカイブタグのデータブロックは、順々に配置されます。データブロックごとに、ヘッダー内に異なる AR_ID-サブ番号が入力されます。

1 秒の等間隔のタイムスタンプは、パラメータ"サイクル"=1 および"ユニット(範囲)"=4 (=秒)を使用して作成されます。

プロセス値のデータタイプは WORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 9
2.0	Year=2001
4.0	Month=10
6.0	Day=05
8.0	Hours=13
10.0	Minutes=40
12.0	Seconds=00
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle =1
24.0	Unit (Type) = 1
26.0	Unit (Range) = 4
28.0	AR_ID-Subnumber = 1
30.0	Process data - data type = 1
32.0	Number of process values = 8
34.0	Process value 1
36.0	Process value 2
38.0	Process value 3
40.0	Process value 4
42.0	Process value 5
44.0	Process value 6
46.0	Process value 7
48.0	Process value 8
50.0	Header Type = 9
52.0	Year=2001
54.0	Month=10
56.0	Day=05
58.0	Hours=12
60.0	Minutes=40
62.0	Seconds=00
64.0	1/10 s
66.0	1/100 s
68.0	1/1000 s
	Weekday
	Cycle =1
	Unit (Type) = 1
	Unit (Range) = 4
	AR_ID-Subnumber = 2
	Process data - data type = 1
	Number of process values = 5
	Process value 1
	Process value 2
	Process value 3
	Process value 4
	Process value 5

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 7:複数のアーカイブタグ; 専用タイムスタンプ付き各プロセス値

はじめに

この例では、複数のアーカイブタグに対するプロセス値が転送されます。さまざまなアーカイブタグのデータブロックは、順々に配置されます。データブロックごとに異なる AR_ID-サブ番号が入力されます。

ヘッダー内のタイムスタンプは重要ではありません。従って、各プロセス値（1 ワード）はタイムスタンプ（8 バイト）が先行します。

プロセス値のデータタイプは SWORD です。

6.11 SIMATIC S7 Protocol Suite

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 9
2.0	Year=0
4.0	Month=0
6.0	Day=0
8.0	Hours=0
10.0	Minutes=0
12.0	Seconds=0
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle =0
24.0	Unit (Type) = 2
26.0	Unit (Range) = 0
28.0	AR_ID-Subnumber = 1
30.0	Process data - data type = 2
32.0	Number of process values = 3
34.0	Year=2001
36.0	Month=10
38.0	Day=05
40.0	Hours=13
42.0	Minutes=40
44.0	Seconds=00
46.0	1/10 s
48.0	1/100 s
50.0	1/1000 s
52.0	Weekday
54.0	Process value 1
56.0	Year=2001
58.0	Month=10
60.0	Day=05
62.0	Hours=14
64.0	Minutes=40
66.0	Seconds=00
68.0	1/10 s
70.0	1/100 s
72.0	1/1000 s
74.0	Weekday
76.0	Process value 2
78.0	Year=2001
80.0	Month=10
82.0	Day=05
84.0	Hours=15
86.0	Minutes=40
88.0	Seconds=00
90.0	1/10 s
92.0	1/100 s
94.0	1/1000 s
96.0	Weekday
98.0	Process value 3
100.0	Header Type = 9
102.0	Year=0
104.0	Month=0
106.0	Day=0
108.0	Hours=0
110.0	Minutes=0
112.0	Seconds=0
114.0	1/10 s
116.0	1/100 s
118.0	1/1000 s
120.0	Weekday
122.0	Cycle =0
124.0	Unit (Type) = 2
126.0	Unit (Range) = 0
128.0	AR_ID-Subnumber = 2
130.0	Process data - data type = 2
132.0	Number of process values = 2
134.0	Year=2001
136.0	Month=10
138.0	Day=05
140.0	Hours=12
142.0	Minutes=40
144.0	Seconds=00
146.0	1/10 s
148.0	1/100 s
150.0	1/1000 s
152.0	Weekday
154.0	Process value 1
156.0	Year=2001
158.0	Month=10
160.0	Day=05
162.0	Hours=13
164.0	Minutes=40
166.0	Seconds=00
168.0	1/10 s
170.0	1/100 s
172.0	1/1000 s
174.0	Weekday
176.0	Process value 2

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

データブロック構造体の例 8:複数のアーカイブタグ; 相対タイムスタンプ(時間差)付きの各プロセス値

はじめに

この例では、アーカイブタグに対してタイムスタンプ付きでプロセス値が転送されます。

さまざまなアーカイブタグのデータブロックは、順々に配置されます。データブロックごとに異なる AR_ID-サブ番号が入力されます。

パラメータ"ユニット(タイプ)"=3 を使用して、各プロセス値にはヘッダー内のタイムスタンプと相対する時間差(4 バイト)が付きます。各アーカイブタグとそのデータブロックには、"ユニット(範囲)"パラメータによって時間差単位が個々に定義されます。

プロセス値のデータタイプは DWORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 9
2.0	Year=2001
4.0	Month=10
6.0	Day=05
8.0	Hours=13
10.0	Minutes=40
12.0	Seconds=00
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle =0
24.0	Unit (Type) = 3
26.0	Unit (Range) = 5
28.0	AR_ID-Subnumber = 1
30.0	Process data - data type = 3
32.0	Number of process values = 3
34.0	Relative time difference in minutes
36.0	Process value 1
38.0	Relative time difference in minutes
40.0	Process value 2
42.0	Relative time difference in minutes
44.0	Process value 3
46.0	Header Type = 9
48.0	Year=2001
50.0	Month=10
52.0	Day=05
54.0	Hours=12
56.0	Minutes=40
58.0	Seconds=00
60.0	1/10 s
62.0	1/100 s
64.0	1/1000 s
66.0	Weekday
68.0	Cycle =0
70.0	Unit (Type) = 3
72.0	Unit (Range) = 6
74.0	AR_ID-Subnumber = 2
76.0	Process data - data type = 3
78.0	Number of process values = 2
80.0	Relative time difference in hours
82.0	Process value 1
	Relative time difference in hours
	Process value 2

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

アーカイブタグが複数の場合の AR_SEND 変数形（最適化）

はじめに

この変数形は、最大数のアーカイブタグにそれぞれのプロセス値を 1 度に指定する必要がある時に使用します。この場合、転送されるデータエリアは 1 つのデータブロックだけで構成され、各プロセス値には AR_ID サブ番号とそれに関連する値があるだけです。

このデータブロックでは、データタイプはすべてのアーカイブタグのプロセス値と同じです。

この変数形のプロパティは以下のとおりです。

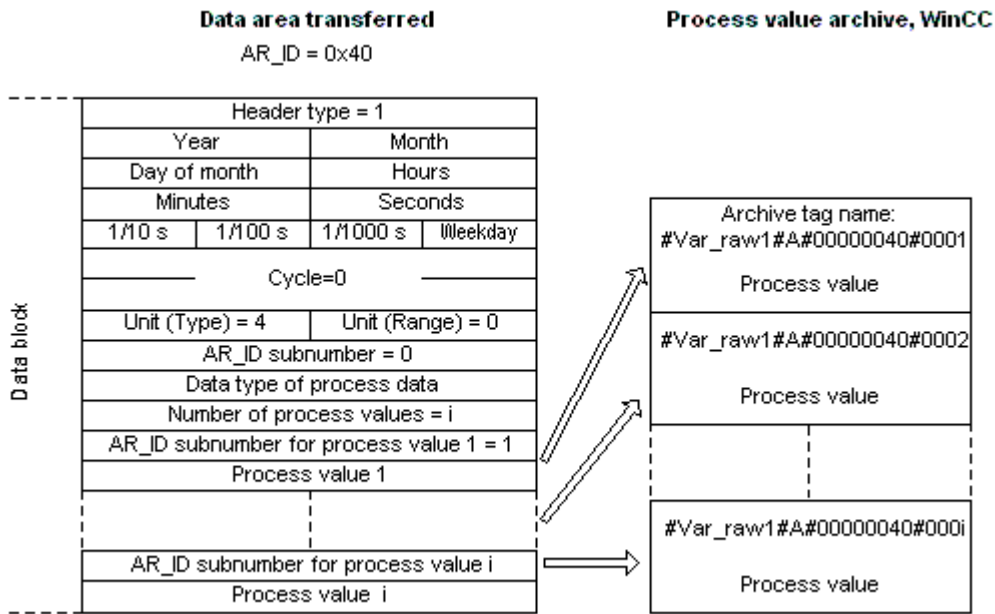
- ヘッダータイプは、1 でなければならない(タイムスタンプ付き、および AR_ID サブ番号なし)。
- データブロックの関連するプロセス値の AR_ID サブ番号は、1 以上でなければならない。ヘッダーの AR_ID Subnumber の値は求められない。
- "装置(タイプ)"パラメータは 4 でなければならない。すなわち、プロセス値には AR_ID サブ番号がある。
- "装置(範囲)"パラメータは 0 でなければならない。すなわち、ヘッダーのタイムスタンプはすべてのプロセス値に対して有効で、相対時間はない。
- WinCC では、アーカイブタグ名には AR_ID サブ番号が付いている。

注記

プロセス値に、WinCC アーカイブタグが見つからない AR_ID サブ番号があると、WinCC 診断ログへエントリされます。残りのプロセス値は、継続して処理されます。

データ領域の構造体の例

転送されるデータエリアは、1つのデータブロックだけで構成されています。



下記も参照

AR_SEND 変数形のプロパティの概要 (ページ 467)

データブロック構造体の例 9:複数のアーカイブタグ ; 最適化 (ページ 488)

データブロック構造体の例 9:複数のアーカイブタグ ; 最適化

はじめに

この例では、アーカイブタグに対してタイムスタンプ付きでプロセス値が転送されます。
このタイムスタンプはすべてのアーカイブタグに適用されます。

各プロセス値の前に対応する AR_ID-サブ番号が入力されます。

プロセス値のデータタイプは WORD です。

データ構成要素のデータブロック構造体

Adress in the DB	Data block sent
0.0	Header Type = 1
2.0	Year=2001
4.0	Month=10
6.0	Day=05
8.0	Hours=13
10.0	Minutes=40
12.0	Seconds=00
14.0	1/10 s
16.0	1/100 s
18.0	1/1000 s
20.0	Weekday
22.0	Cycle = 0
24.0	Unit (Type) = 4
26.0	Unit (Range) = 0
28.0	AR_ID-Subnumber = 0
30.0	Process data - data type = 1
32.0	Number of process values = 5 (max. 3992)
34.0	AR_ID-Subnumber
36.0	Process value 1
38.0	AR_ID-Subnumber
40.0	Process value 2
42.0	AR_ID-Subnumber
44.0	Process value 3
46.0	AR_ID-Subnumber
48.0	Process value 4
50.0	AR_ID-Subnumber
52.0	Process value 5

下記も参照

データブロック - 構造体とパラメータ (ページ 460)

アーカイブタグが 1 つの場合に AR_SEND 変数形をコンフィグレーションする方法

はじめに

AR_SEND ファンクションを使用したデータ交換には、多数の変数形があります。"1 つのアーカイブタグ"変数形の場合、AR_ID だけが使用されます。AR_ID サブ番号は使用されません。

AR_ID は、AS 内のデータのアーカイブタグへの割り付けを確立するために使用され、AS 内のデータブロック SFB 37 "AR_SEND"ファンクションブロックを構成する間に、その他のパラメータと一緒に構成されます。

WinCC では、プロセスコントロールされるタグのプロパティの中で、この割り付けが行われます。この割り付けは、WinCC で必要な唯一の設定です。このセクションでは、この割

り付けについて説明します。

WinCC のその他のパラメータは、自動的に評価されるため、構成する必要はありません。

注記

AR_ID サブ番号だけが付いたこの変数形を使用するには、ヘッダータイプを 0 または 1 として構成する必要があります。AR_ID サブ番号は 0 に設定する必要があります。

これには AR_ID サブ番号が付いていないため、WinCC バージョン V5.0 以前に構成されたすべてのアーカイブタグはこの変数形を使用できます。

WinCC バージョン 5.1 Hotfix 4 以降は、プロセスコントロールされたタグでアーカイブタグ名のエイリアスを指定する、あるいはシステムによって生成された内部名を使用することができます。バージョン V5.1 HF4 以降では、システムによって生成された名前には、未処理データ ID の代わりに割り付けられた未処理データタグの名前が含まれます。V5.1 HF4 からバージョンを移行したプロジェクトは、アーカイブタグ名をそのオリジナルフォーマットに使用する、あるいは変換することができます。この名前は、プロセスコントロールされたアーカイブタグのプロパティダイアログボックスを 1 度開いて閉じることにより変換されます。エイリアスは割り付ける必要はありません。


プロジェクト内で"AS-OS-Transfer"ファンクションを使用して外部タグをすべて再マッピングした場合、アーカイブタグ名をすぐに新規構造体に変換する必要があります!この後、新規構造体が保持されます。

必要条件

- 「AR_SEND」ファンクションブロックおよびデータブロック構造体を、まず AS に構成する必要があります、この設定情報は以下の手順の間は使用可能にすること。
- 「SIMATIC S7 Protocol Suite」チャンネルをプロジェクトに統合してください。
- 「工業用 Ethernet」などのチャンネルユニットに「Test_Ind_Eth」などの接続を作成しておくこと。
- [タグロギング]エディタでプロセス値アーカイブを構成しておくこと。

手順

1. [SIMATIC S7 Protocol Suite]チャンネルで、データ転送に使用すべき接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグに名前「Var_raw_arsend」を入力します。この名前は 24 文字を超えないでください。
未処理データタグ名は、プロセスコントロールタグの設定中にアーカイブタグ名の一部となるため、この長さに制限されます。
4. [データタイプ]フィールドで、[未処理データタイプ]を選択します。

5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
6. [未処理データ]チェックボックスを選択します。[未処理データタイプ]エリアで、タイプとして[アーカイブデータリンク]を選択します。[OK]をクリックして、開いているダイアログをすべて閉じます。
7. [タグロギング]エディタを開きます。[タグロギング]エディタのナビゲーションエリアで、[プロセス値アーカイブ]フォルダを選択します。テーブルエリアで[プロセスコントロールタグ]タブを選択し、未処理データタグ「Var_raw_arsend」を追加します。
8. [プロパティ]エリアでプロパティを編集します。
9. [変換 DLL]フィールドで、「nrms7p mc.nll」 エントリを選択します。
10. [ブロック ID]に 16 進数値で AR_ID を入力します。この値は、AS の設定で定義されています。
この AR_SEND 変数形ではサブ番号が使用されないため、[サブ番号]には何も入力しないでください。
11. [タグ名]フィールドに、システムによって生成された内部アーカイブタグ名が表示されます。割り付けられた未処理データタグおよび AR_ID の名前が含まれています。[アーカイブタグ名]で、必要に応じてこのアーカイブタグのエイリアスを定義できます。エイリアスが入力されない場合は、プロセス値アーカイブの管理および WinCC へのアーカイブタグのアドレス指定には内部アーカイブタグ名が使用されます。
12. タグロギングを閉じます。

アーカイブタグが複数の場合に AR_SEND 変数形をコンフィグレーションする方法

はじめに

アーカイブタグが複数の場合に AR_SEND ファンクションを使用したデータ交換には、多数の変数形があります。

- 「複数のアーカイブタグ」変数形を使用して、複数の値をそれぞれ異なる時間に複数のアーカイブタグに指定します。
- 「複数のアーカイブタグ - 最適化」変数形を使用して、1 つの値を 1 度に最大数のアーカイブタグにそれぞれ指定します。

どちらの変数形も、AR_ID と AR_ID サブ番号を使用します。

AR_ID と AR_ID サブ番号は、AS 内のデータとアーカイブタグ間の割り付けを確立します。これらは、データブロックおよびファンクションモジュール SFB 37「AR_SEND」を構成する時に AS 内でその他のパラメータと定義されます。

WinCC では、プロセスコントロールされるタグのプロパティの中で、この割り付けが行われます。この割り付けは、WinCC で必要な唯一の設定です。このセクションでは、この割

6.11 SIMATIC S7 Protocol Suite

り付けについて説明します。

WinCC のその他のパラメータは、自動的に評価されるため、構成する必要はありません。

注記

AR_ID サブ番号を使用するには、ヘッダタイプを 8 または 9 として構成する必要があります。

WinCC バージョン V5.0 で構成されたアーカイブタグには AR_ID サブ番号がないため、「1 つのタグ」変数形でのみ使用可能です。

WinCC バージョン 5.1 Hotfix 4 以降は、プロセスコントロールされたタグでアーカイブタグ名のエイリアスを指定する、あるいはシステムによって生成された内部名を使用することができます。バージョン V5.1 HF4 以降では、システムによって生成された名前には、未処理データ ID の代わりに割り付けられた未処理データタグの名前が含まれます。V5.1 HF4 からバージョンを移行したプロジェクトは、アーカイブタグ名をそのオリジナルフォーマットに使用する、あるいは変換することができます。この名前は、プロセスコントロールされたアーカイブタグのプロパティダイアログボックスを 1 度開いて閉じることにより変換されます。エイリアスは割り付ける必要はありません。


プロジェクト内で"AS-OS-Transfer"ファンクションを使用して外部タグをすべて再マッピングした場合、アーカイブタグ名をすぐに新規構造体に変換する必要があります!この後、新規構造体が保持されます。

必要条件

- 「AR_SEND」ファンクションブロックおよびデータブロック構造体を、まず AS に構成する必要があります、この設定情報は以下の手順の間は使用可能にすること。
- 「SIMATIC S7 Protocol Suite」チャンネルをプロジェクトに統合してください。
- 「工業用 Ethernet」などのチャンネルユニットに「Test_Ind_Eth」などの接続を作成しておくこと。
- [タグロギング]エディタでプロセス値アーカイブを構成しておくこと。

手順

1. [SIMATIC S7 Protocol Suite]チャンネルで、データ転送に使用すべき接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグに名前「Var_raw_arsend」を入力します。この名前は 24 文字を超えないでください。
未処理データタグ名は、プロセスコントロールタグの設定中にアーカイブタグ名の一部となるため、この長さに制限されます。
4. [データタイプ]フィールドで、[未処理データタイプ]を選択します。

5. [SIMATIC S7 Protocol Suite]チャンネルで、データ転送に使用すべき接続を選択します。接続のショートカットメニューから[新規タグ]オプションを選択します。[タグのプロパティ]ダイアログが開きます。
6. [名前]フィールドにタグの名前として「Var_raw_arsend」を入力します。この名前は 24 文字を超えないでください。未処理データタグ名は、プロセスコントロールタグの設定中にアーカイブタグ名の一部となるため、この長さに制限されます。
7. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
8. [未処理データ]チェックボックスを選択します。[未処理データタイプ]エリアで、タイプとして[アーカイブデータリンク]を選択します。[OK]をクリックして、開いているダイアログをすべて閉じます。
9. [タグロギング]エディタを開きます。[タグロギング]エディタのナビゲーションエリアで、[プロセス値アーカイブ]フォルダを選択します。テーブルエリアで[プロセスコントロールタグ]タブを選択し、未処理データタグ「Var_raw_arsend」を追加します。
10. [プロパティ]エリアでプロパティを編集します。
11. [変換 DLL]フィールドで、「nrms7p mc.nll」 エントリを選択します。[ブロック ID]に 16 進数値で AR_ID を入力します。この値は、AS の設定で定義されています。[サブ番号]に 16 進数値で AR_ID サブ番号を入力します。この値は、AS の設定で指定されています。
12. [タグ名]フィールドに、システムによって生成された内部アーカイブタグ名が表示されます。割り付けられた未処理データタグおよび AR_ID の名前が含まれています。[アーカイブタグ名]で、必要に応じてこのアーカイブタグのエイリアスを定義できます。エイリアスが入力されない場合は、プロセス値アーカイブの管理および WinCC へのアーカイブタグのアドレス指定には内部アーカイブタグ名が使用されます。
13. タグロギングを閉じます。

6.11.5.3 "SIMATIC S7 Protocol Suite"チャンネルの未処理データタグ

"SIMATIC S7 Protocol Suite"チャンネルの未処理データタグ

はじめに

- 未処理データタイプのタグはトランスポートレベルのデータテレグラムです。未処理データタグの内容は固定されていないため、送信者と受信者のみが転送データを解読できます。このデータタイプでは WinCC のフォーマット変更はありません。データの最大長は 65535 バイトです。
- WinCC は 2 種類の未処理データタグを区別しています。自由にアプリケーションで利用できる未処理データタグと S 7 ファンクションを処理するための未処理データタグです。

自由にアプリケーションで利用できる未処理データタグ

自由にアプリケーションで利用できる未処理データタグは、WinCC と PLC 間でユーザーデータブロックを転送するのに使用されます。この未処理データタグはユーザーデータしか処理しません。これは以下を区別します。

バイト配列としての未処理データタグ

BSEND/BRCV ファンクション用の未処理データタグ

S7 ファンクションを処理するための未処理データタグ

これらの未処理データタグにはチャンネル固有のヘッダーがなく、一般に WinCC 内のメッセージ/プロセスデータ記録システムで使用されます。

これらはチャンネル内部のタグ/ファンクションであるため、これ以上の説明はここで必要ありません。

下記も参照

S7 通信の BSEND/BRCV ファンクション用の未処理データタグ (ページ 498)

バイト配列としての未処理データタグ (ページ 494)

バイト配列としての未処理データタグ

はじめに

バイト配列としての未処理データタグは、WinCC と PLC 間でユーザーデータブロックを転送するのに使用されます。この未処理データタグはユーザーデータしか処理しません。

バイト配列としての未処理データタグは、データ領域のアドレスおよび長さ(例: DB100、DW20、長さ 40 バイト)によってアドレス指定される普通のプロセスタグと同じようにチャンネル内で扱われます。

未処理データの長さは転送可能なデータブロック 1 つに限定されており、PDU (プロトコルデータユニット) を使用して完全に転送可能でなければなりません。通信ドライバが送信できるデータブロックの最大長は、接続ステップ時に決められた PDU の長さから、ヘッダーおよび追加情報を差し引いた長さによって決まります。従って SIMATIC S7 で通常使用される PDU の長さは、次の最大長になります。

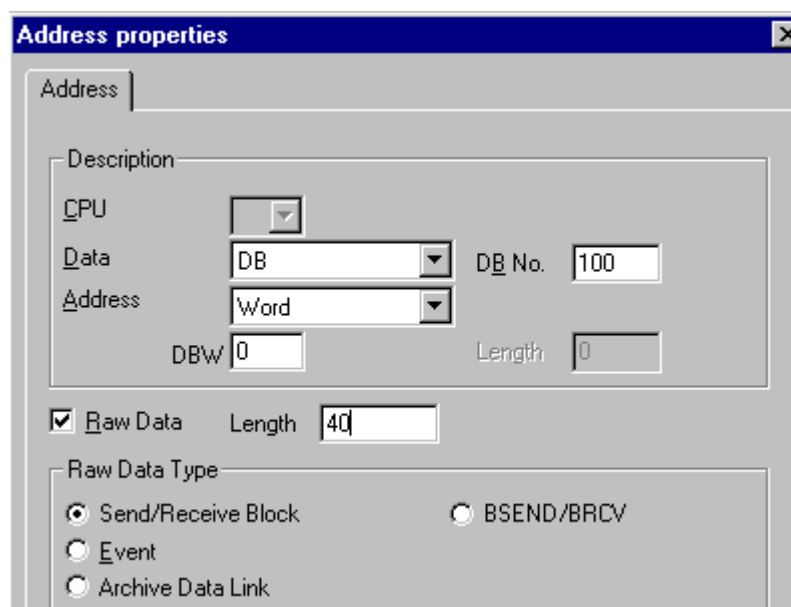
- S7-300 システム : PDU の長さ 240 バイト、データブロックの最大長 208 バイト
- S7-400 システム : PDU の長さ 480 バイト、データブロック最大長さ 448 バイト

これより長いデータブロックが転送された場合は、データがブロックされます。PLC では、S7 ソフトウェアによりブロックされます。WinCC ではスクリプトによりブロックされます。

バイト配列としての未処理データタグを設定する方法

データブロックの転送に対応した未処理データタグは、アドレスと長さを指定した[送信/受信ブロック]タイプの未処理データとして設定されます。

以下の図に、データワード 20 で始まるデータ構成要素 100 に、長さ 40 バイトのデータ領域を設定する場合の例を記載します。



バイト配列としての未処理データタグの読取り

未処理データタグは、"通常"のプロセスタグと同じ方法で読み取られます。データを受信すると、対応するデータブロックが AS から要求され、ユーザーに転送されます。

転送は必ず、WinCC が自発的に実行します。この身処理データタグを使用しても、AS が自発的にデータの突発的受信またはイベントコントロール型受信を実行

バイト配列としての未処理データタグの書き込み

未処理データタグは、"通常"のプロセスタグと同じ方法で書き込まれます。データブロックを送信し、AS から肯定的な確認応答を受信すると、データマネージャのイメージ内にデータブロックが転送されます。

下記も参照

バイト配列としての未処理データタグをコンフィグレーションする方法 (ページ 496)

バイト配列としての未処理データタグをコンフィグレーションする方法

はじめに


このセクションでは、"SIMATIC S7 Protocol Suite"チャンネルの未処理データをバイト配列として構成する方法について説明します。

コンフィグレーションは、チャンネルのすべてのユニットで同一です。この例では MPI チャンネルユニットとその接続を使用します。

必要条件

- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- チャンネルユニット(例: "マルチポイントインターフェース")との接続を作成して下さい。

手順

1. [SIMATIC S7 Protocol Suite]チャンネルで、データ転送に使用すべき接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグに名前"Var1_raw_byte"を入力します。
4. [データタイプ]フィールドで、[未処理データタイプ]を選択します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
6. [未処理データ]チェックボックスをマークします。[未処理データタイプ]エリアで、タイプとして[送信/受信ブロック]を選択します。この設定により、[アドレス記述]エリア内のフィールドと、[未処理データ]チェックボックスの横のフィールドの表示が決まります。
7. [長さ]フィールドに、未処理データブロックの長さを入力します(単位はバイト)。

8. [データエリア]で、データ指定される PCL のデータエリアを設定します。データエリアが [DB]として選択されている場合、有効[DB 番号]フィールドにデータブロック番号を入力します。
- "マルチポイントインターフェース"チャンネルユニットに接続する場合は、[CPU]フィールドは無効になります。

The screenshot shows the 'Address properties' dialog box. The 'Address' tab is selected. In the 'Description' section, the 'CPU' field is a dropdown menu. The 'Data' field is set to 'DB', and the 'DB No.' field contains the value '1'. The 'Address' field is set to 'Byte', and the 'DBB' field contains the value '0'. The 'Length' field contains the value '0'. Below this, the 'Raw Data' checkbox is checked, and the 'Length' field next to it contains the value '208'. In the 'Raw Data Type' section, the 'Send/Receive Block' radio button is selected, and the 'BSEND/BRCV', 'Event', and 'Archive Data Link' radio buttons are unselected.

9. [アドレス指定]フィールドで、アドレス指定のタイプを設定します。WinCC タグのデータタイプ[未処理データタイプ]では、[バイト]、[ワード]、[ダブルワード]の入力が可能です。
10. 下のフィールドには、開始アドレスの値を入力します。この左側のフィールドのラベルは、[データエリア]および[アドレス指定]フィールド内のエントリによって異なります(例: データエリア[DB]およびアドレス指定タイプ[バイト]の場合は[DBB])。
11. [OK]をクリックして、開いているダイアログをすべて閉じます。

注記

未処理データの長さは転送可能なデータブロック 1 つに限定されており、PDU（プロトコルデータユニット）を使用して完全に転送可能でなければなりません。通信ドライバが送信できるデータブロックの最大長は、接続ステップ時に決められた PDU の長さから、ヘッダーおよび追加情報を差し引いた長さによって決まります。従って SIMATIC S7 で通常使用される PDU の長さは、次の最大長になります。

- S7-300 システム: PDU の長さ 240 バイト、データブロック最大長さ 208 バイト
 - S7-400 システム: PDU の長さ 480 バイト、データブロック最大長さ 448 バイト
- 間違った長さでは読み取り/書き込みが拒絶され、標示されます。

S7 通信の BSEND/BRCV ファンクション用の未処理データタグ

はじめに

"BSEND/BRCV"ファンクションの未処理データタグは、WinCC と AS 間でユーザーデータブロックを転送するのに使用されます。この未処理データタグはユーザーデータしか処理しません。

この身処理データタイプは S7 通信の"BSEND/BRCV"ファンクションへのアクセスに使用できます。

指定接続を使用した「BSEND/BRCV」未処理データ通信は、以下のオートメーションシステムに対してサポートされています。

- S7-400
- S7-300
 - CPU319-3 PN/DP V2.5 以降
 - CPU317-2 PN/DP V2.6 以降
 - CPU315-2 PN/DP V3.1 以降

S7-300 コントローラでは、ファームウェアバージョン V3.x 以降を推奨します。

未加工データ通信は、通信プロセッサを介して確立することはできません。

- WinAC RTX 2010

データ転送は常に送信パートナーの責任で自発的に実行されます。従って"BSEND/BRCV"ファンクションはイベントコントロール型または突発的なデータブロック送信の実行にも使用できます。

リソース上の理由から、BSEND/BRCV 未処理データタグの数を小さく抑えることが推奨されます。

S7 ファンクション"AR_SEND"および"BSEND/BRCV"の使用のリソース制限

AR_SEND ファンクションや BSEND/BRCV ファンクションを使用して AS から WinCC に一度に送信できるデータの最大量は以下ようになります。

- S7-400 は 16 KB まで
- S7-300 は 8 KB まで

S7-400 での例:

- 1x 最大 16 KB の BSEND
- 1x 8 KB の AR_SEND + 1x 8 KB の BSEND
- 1x 10 KB の AR_SEND + 1x 2 KB の AR_SEND + 1x 4 KB の BSEND

注記

書き込みジョブの調整

書き込み用のデータブロックを AS に転送しても、そのデータブロックの全部または一部が AS の受信バッファから削除されない場合は、エラーが表示されるとともに、それ以降の書き込みが拒否されます。

このようなエラーが標示される間は、R_ID > 0x8000 0000 の書き込みジョブは接続固有のキューに書き込まれ、システムは書き込みジョブを 6 秒間繰り返そうとします。

転送の時間調整を行うのはユーザーの責任です。短い間隔で書き込みを送信する際には、この点を考慮する必要があります。

"BSEND/BRCV"ファンクションを使用する PBK 接続の設定

"BSEND/BRCV"ファンクションは、"ハード設定接続"、いわゆる PBK 接続(プログラムコンポーネント通信)でしか使用できません。

ハード設定接続を設定するには、接続パラメータで接続リソースを指定する必要があります(16 進数: 10~DF)。

この接続リソースは、PLC で接続を設定する際に STEP 7 によって割り付けられます。

接続はオートメーションシステムで受動接続エンドポイントとして設定する必要があります。

ジョブの読み取り/書き込み

ハード設定接続では、"通常"の読み取り/書き込みを実行できます。

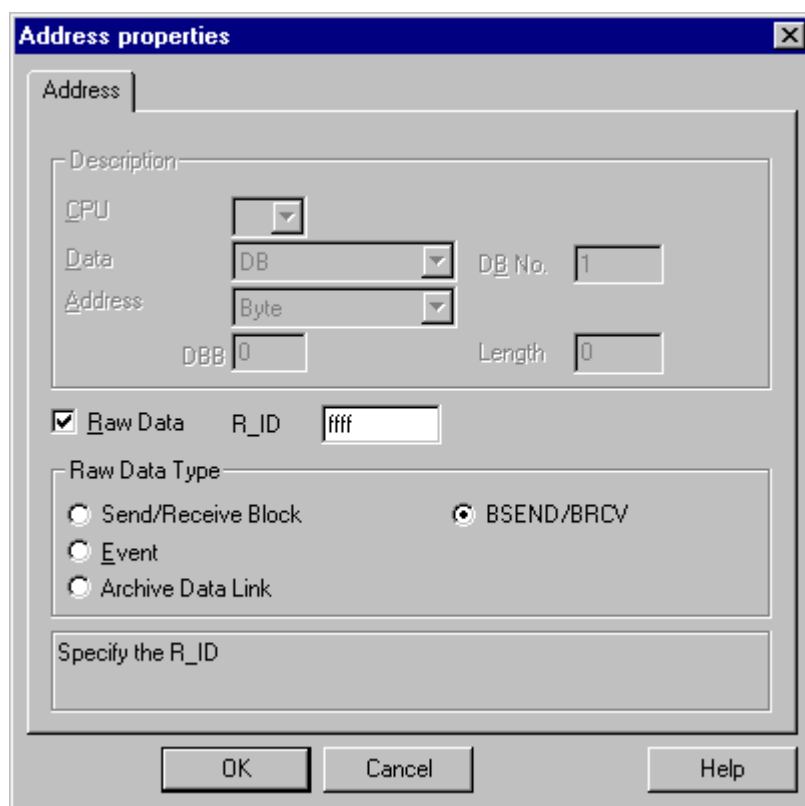
この接続で非常に大きいデータエリアを転送する必要がある場合は、いくつかの PDU でデータブロックが送信されます。

パフォーマンスの理由から、"BSEND/BRCV"ファンクション専用の接続を作成することをお勧めします。

BSEND/BRCV ファンクション用の未処理データタグの設定

"BSEND/BRCV"データブロックを転送するための未処理データタグは、"R_ID"を持つ"BSEND/BRCV"タイプの未処理データとして設定されます。

データ長は、送信/受信されたデータ量から黙示的に引き出されます。



"R_ID"パラメータ

"BSEND/BRCV"ファンクションでは、32 ビット長の R_ID を 16 進数として指定する必要があります。

R_ID はオートメーションシステムでの設定時に割り付けられ、複数のデータブロック転送を 1 つの転送で区別する際に使用されます。

送受信の呼び出しは常に、基本的な通信サブシステム（SIMATIC デバイスドライバ）ではこの R_ID を指す参照が通知されます。

したがって、未処理データタグには固有の R_ID が割り付けられます。

"BSEND/BRCV"未処理データタグの送信

"BSEND/BRCV"未処理データタグは、"通常"のプロセスタグが書き込まれるのと同じ方法で送信されます。

データブロックを送信し、AS から肯定的な確認応答を受信すると、データマネージャのイメージ内にデータブロックが転送されます。

"BSEND/BRCV"未処理データタグの受信

伝送は AS によって開始されるため、"BSEND/BRCV"の未処理データはチャンネルを介して散発的に送信されます。

したがって、S7 の未処理データタグを明示的に読み取ることは不可能です。

同期

BSEND/BRCV 機能には、同期化ファンクションは含まれていません。

スタートアップ段階で、データを受信するユーザーが誰もログオンしていない場合は、AS によって送信されたデータブロックが受信側で破棄されます。

したがって、ユーザーは、自分で同期化(データワードにフラグを設定することで、AS への送信指示を解除するなど)を実行しなければなりません。

下記も参照

BSEND/BRCV ファンクション用の未処理データタグをコンフィグレーションする方法 (ページ 501)

BSEND/BRCV ファンクション用の未処理データタグをコンフィグレーションする方法

はじめに

このセクションでは、"BSEND/BRCV"ファンクション用の"SIMATIC S7 Protocol Suite"チャンネルの未処理データを構成する方法について説明します。

コンフィグレーションは、チャンネルのすべてのユニットで同一です。この例では MPI チャンネルユニットとその接続を使用します。


必要条件

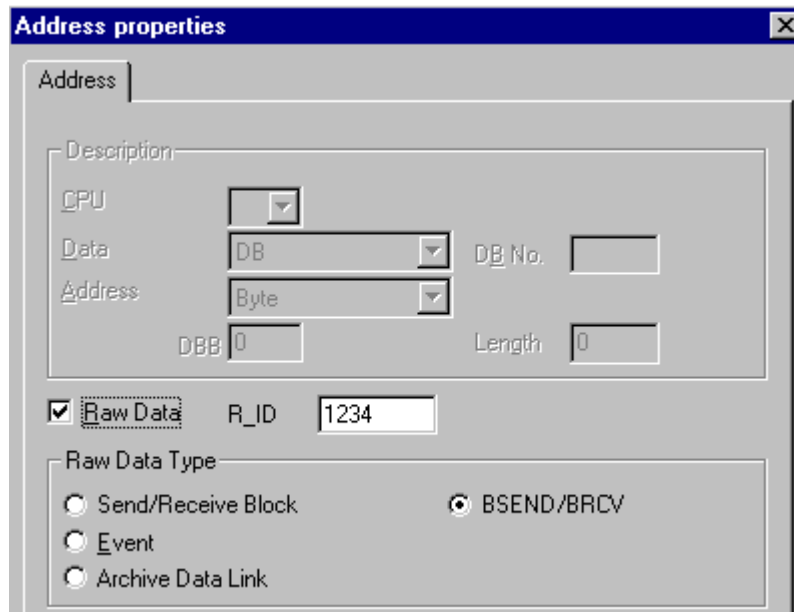
- "SIMATIC S7 Protocol Suite"チャンネルをプロジェクトに統合してください。
- チャンネルユニット(例:"マルチポイントインターフェース")との接続を作成して下さい。

手順

1. [SIMATIC S7 Protocol Suite]チャンネルで、データ転送に使用すべき接続を選択します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグに名前"Var2_raw_bsend"を入力します。

6.11 SIMATIC S7 Protocol Suite

4. [データタイプ]フィールドで、[未処理データタイプ]を選択します。
5. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。
6. [未処理データ]チェックボックスをマークします。[未処理データタイプ]エリアで、タイプとして[BSEND/BRCV]を選択します。[アドレス記述]エリアのフィールド表示はこれで無効化されます。
7. [R_ID]フィールドに ID の 16 進数値を入力します。R_ID は、コンフィグレーション時に AS に割り付けられています。



8. [OK]ボタンをクリックして、両方のダイアログを閉じます。

6.11.5.4 ソフトウェアの二重化

ソフトウェアの二重化

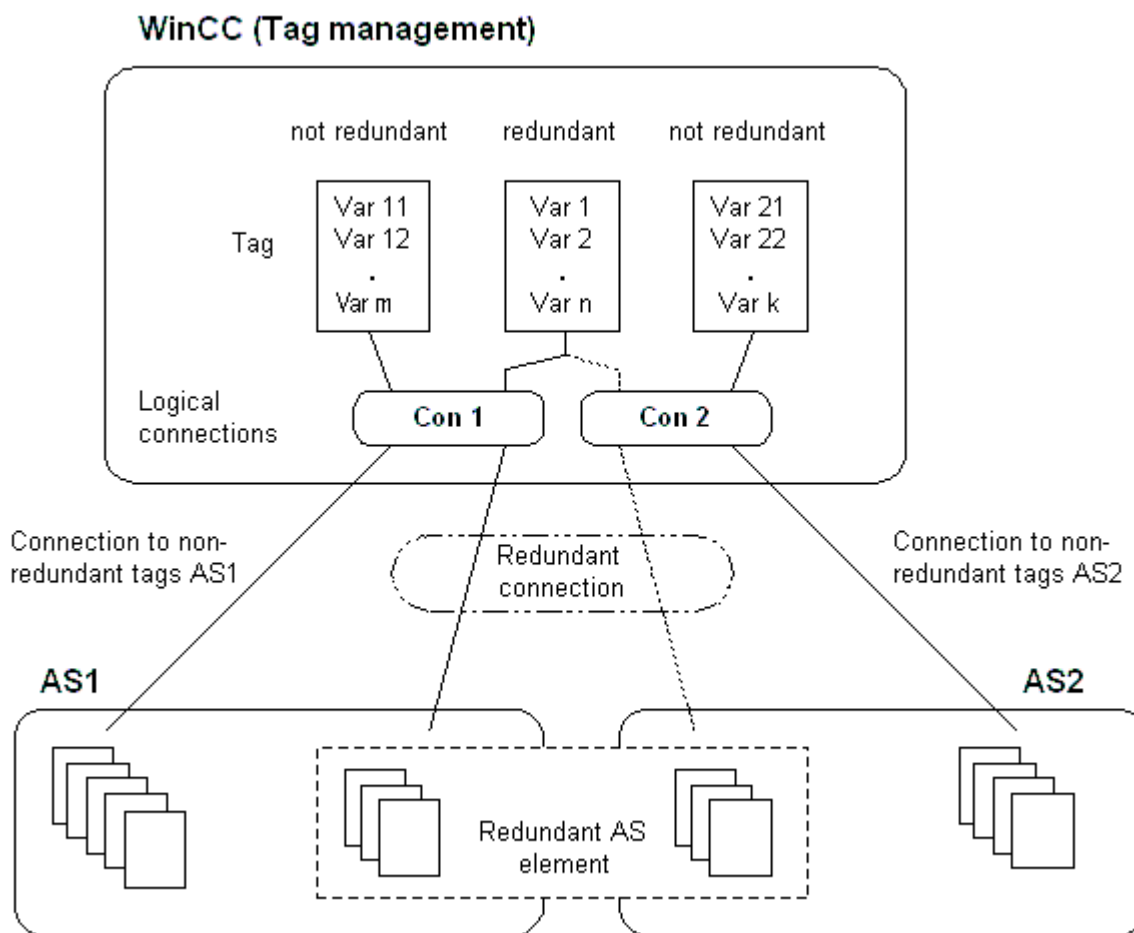
はじめに

ソフトウェアの二重化は、システムの安全関連部分をモニタするコスト効率のよいオプションを提供します。この場合のシステムには、2 つのオートメーションシステム S7-300 と S7-400 の二重接続を通して時間の限界がありません。

注記

このチャンネルにおけるソフトウェアの二重化では、SIMATIC S7-400 H の H レイヤー二重化と同じ機能は提供されません。

ファンクションには AS および WinCC でのコンフィグレーションが要求されます。



AS

2つのオートメーションシステムの間には二重接続が存在する場合は、ASの失敗時にもう一方がモニタを行います。モニタでは、プロセス全体またはプロセスの一部のみを扱うことができます。

アプリケーションプログラムのほかに、ソフトウェアの二重パッケージが両方のオートメーションシステムにインストールされています。このプログラムパッケージは、WinCCパッケージの範囲には含まれていません。一致するデータでは、2つのオートメーションシステムの間にはマルチポイントインターフェース、PROFIBUS-DP、または工業用イーサネットを使用した、既存の通信リンクも使用できる二重接続が必要になります。

WinCC

二重化接続のコンフィグレーションは、同じチャンネルユニット上で接続を使用する場合にも実行できます。コンフィグレーション時には、1つの接続、いわゆるメイン接続のみ

がコンフィグレーションされます。予備接続は、ダイナミックウィザード"二重化接続の設定"を使用してのみ挿入されます。

このウィザードは、接続固有内部タグおよびスクリプトも挿入します。これはランタイム中の接続と対応するメッセージの切り替えをコントロールします。

ランタイム中は、エラーが発生したときにスクリプトを使用して自動的に接続を切り替えることができます。ただし、接続固有内部タグ"@<connectionname>@ForceConnectionAddress"に記述することで、スクリプトを使用せずに手動で切り替えることも可能です。

AS2 では、AS2 の非二重部分からのデータを WinCC で使用する場合、2 番目の固定接続が必要です。

ソフトウェアの二重化を使用しても、ハードコンフィグレーション接続を二重化用にしか使用できないわけではありません。単一接続はそれぞれ二重化なしで使用できます。

下記も参照

WinCC でソフトウェアの二重化を削除する方法 (ページ 511)

ソフトウェアの二重化をコンフィグレーションする方法 (ページ 509)

ソフトウェアの二重化 - 接続固有内部タグ (ページ 504)

ソフトウェアの二重化 - 接続固有内部タグ

はじめに

接続固有内部タグは、二重化接続のコントロールに使用されます。これらのタグはダイナミックウィザード"二重化接続の設定"を使用してセットアップされ、関連する接続の"@<connectionname>"というタググループ内に収集されます。

これらのタグを使用して、接続ステータスを判断したり、接続の確立をコントロールすることができます。したがって、単一接続を介して複数の PLC をアドレス指定するなど、他のアプリケーションの実行に使用することも可能です。ただし、これらのタグをソフトウェア二重化パッケージなしで使用する場合は、タグを手動で作成する必要があります。

名前のフォーマット

接続固有の内部タグの名前は、関連接続の名前と識別子で構成されます。

"@<接続名>@<識別子>"

接続名の前には、システムタグとして識別するための"@記号が付きます。この識別子は、接続名の前に区切り文字として置かれます。

例:"@CPU_3@ConnectionState"

<接続名> = CPU_3

<識別子> = ConnectionState

注記

接続固有内部タグは外部タグとして数えられます（接続ごとに 8 つの外部タグ）。

WinCC のデータマネージャは、関連する接続が可能な場合にのみ、外部タグへのアクセスを許可します。ただし、接続ステータスに関係なく、接続固有内部タグの書き込みや読み取りを行うことは可能です。

ランタイムで、一部の接続固有内部タグの現在値が"WinCC チャンネル診断"ツールを使用して呼び出される場合があります。メインの接続が選択されると、このタグが[カウンタ]列に表示されます。さらに、WinCC Explorers のタグ管理で、タグの現在値をツールヒントとして表示することも可能です。

タグの概要

ソフトウェアの二重化の内部タグには、以下の識別子を使用できます。

ConnectionState

意味	接続ステータス このタグは現在の接続ステータスを検出するために使用できます。
タイプ	ダブルワード
アクセス	読み取り
デフォルト値	0 = "不良"
値	値 0 = 接続不良 1 = 動作用に接続準備済み 2 = 接続二重化(H システムでの二重化の場合のみ)

ConnectionError

意味	不良の原因 タグには不良の原因が含まれます。デフォルト = 0、接続がまだ確立されていないかエラーがないことを示します。接続を確立すると、このタグには再度 0(エラーなし)がロードされます。このエラーコードは、チャンネル固有の方法で解釈されます。S7 チャンネルは、このタグに SIMATIC デバイスドライバのエラーコードを渡します。
タイプ	ダブルワード
アクセス	読み取り
デフォルト値	0 = "エラーなし"
値	0 = エラーなし <> 0 = S7 エラーコード

ConnectionErrorString

意味	文字列としての不良の原因 タグには文字列としての接続エラーの原因が含まれます。この文字列は、現在選択されている言語で出力されます。デフォルト = "、接続がまだ確立されていないかエラーがないことを示します。S7 チャンネルでは、選択された言語に関係なく以下のテキストが"英語"で出力されます。
タイプ	TEXT8 [128]
アクセス	読み取り
デフォルト値	"" = "エラーなし"
値	"エラーなし" "エラー hhhh" = エラー hhhh が発生しました (ここで hhhh = 16 進数の S7 エラーコード)

ConnectionErrorCount

意味	接続エラーのカウンタ このタグの値は接続エラーごとに 1 ずつ増えます。カウンタはカウンタオーバーフロー時に 0 に戻って開始します。
タイプ	ダブルワード

アクセス	読み取り
デフォルト値	0

ConnectionEstablishMode

意味	接続確立モード このタグは接続を確立するための自動モードを設定するために使用できます。これが有効な場合、S7 チャンネルは約 4 秒間隔で失敗した接続の再確立を試みます。このタグの値が 0 の場合は、エラー後に 4 秒間隔の接続の再確立を試みずに、切断されたままになります。
タイプ	ダブルワード
アクセス	書き込み
デフォルト値	1
値	タグへの書き込みは次のアクションを引き起こします: 0 = 手動接続確立モード アクション:自動接続確立の無効化 <> 0 =自動接続確立モード アクション:自動接続確立モードの有効化

ForceConnectionState

意味	優先接続ステータス このタグは、優先接続ステータスをチャンネルに通知する際に使用できます。通常は、このタグの値 1(チャンネルが(適用可能な場合は約 4 秒の定期間隔で)接続の確立を試みる)になります。このタグに値 0 を書き込むと、チャンネルは接続に割り込みます。
タイプ	ダブルワード
アクセス	書き込み
デフォルト値	1
値	タグへの書き込みは次のアクションを引き起こします: 0 = 優先接続ステータス:接続切断 アクション:接続が確立されている場合に、切断を開始します。 1 = 優先接続ステータス:接続切断 アクション:接続が切断されている場合に、接続の確立を開始します。

ForceConnectionAddress

意味	<p>接続アドレスの選択</p> <p>このタグはどの接続アドレスを接続の確立に使用するかを定義します。</p>
タイプ	ダブルワード
アクセス	書き込み
デフォルト値	0
値	<p>ConnectionEstablishMode が"自動"に設定されている場合は、対応するアドレスへの接続が自動的にセットアップされます。</p> <p>タグへの書き込みは次のアクションを引き起こします:</p> <p>0 = コンフィグレーションされた接続パラメータによる接続 アクション:@ForceConnectionAddress が以前に 1 に設定されている場合、切断を開始します。</p> <p>1 = 代替接続パラメータを使用した接続。 アクション:@ForceConnectionAddress が以前に 0 に設定されている場合、切断を開始します。</p>

AlternateConnectionAddress

意味	<p>代替接続アドレス</p> <p>このタグには、代替接続アドレス文字列を入力できます。この文字列は、WinCC Explorer で接続パラメータとして表示される文字列と同じです。文字列はチャンネル固有です。システム起動(ランタイム)時には、コンフィグレーション済みのアドレスが S7 チャンネルのデフォルトとしてここに入力されます。アドレスがまだ設定されていない場合は、S7 チャンネルに対して"不正なアドレス"というテキストが入力されます。MPI を使用したステーションアドレス 3 を持つ S7-AS のアドレス詳細:"MPI,3 0,,0,0,02"</p>
タイプ	TEXT8 [255]
アクセス	書き込み
デフォルト値	"..." = "コンフィグレーション済みアドレス"
値	<p>このタグへの書き込みは以下のアクションを引き起こします:</p> <ul style="list-style-type: none"> - アドレスが書き込みプロセスにより変更された場合、接続は"代替接続パラメータによる接続"設定で切断されます。 - 接続モードが[自動]に設定されている場合は、接続は自動的に書き込まれたばかりのアドレスと共に確立されます。

ソフトウェアの二重化をコンフィグレーションする方法

はじめに

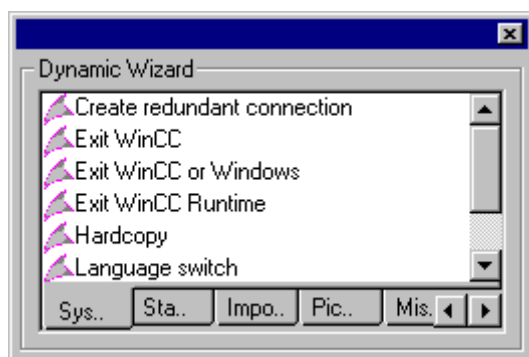
このセクションでは、WinCC で"SIMATIC S7 Protocol Suite"チャンネルの接続に対するソフトウェアの冗長化を設定する方法について、説明します。PLC もこのファンクションを使用するように設定する必要がありますが、これについてはこのマニュアルでは扱いません。

必要条件

1. "SIMATIC S7 Protocol Suite"チャンネルを、プロジェクトに統合する必要があります。
2. このチャンネルの、冗長化接続を設定するチャンネルユニットの 1 つで、接続を作成しておくこと。

手順

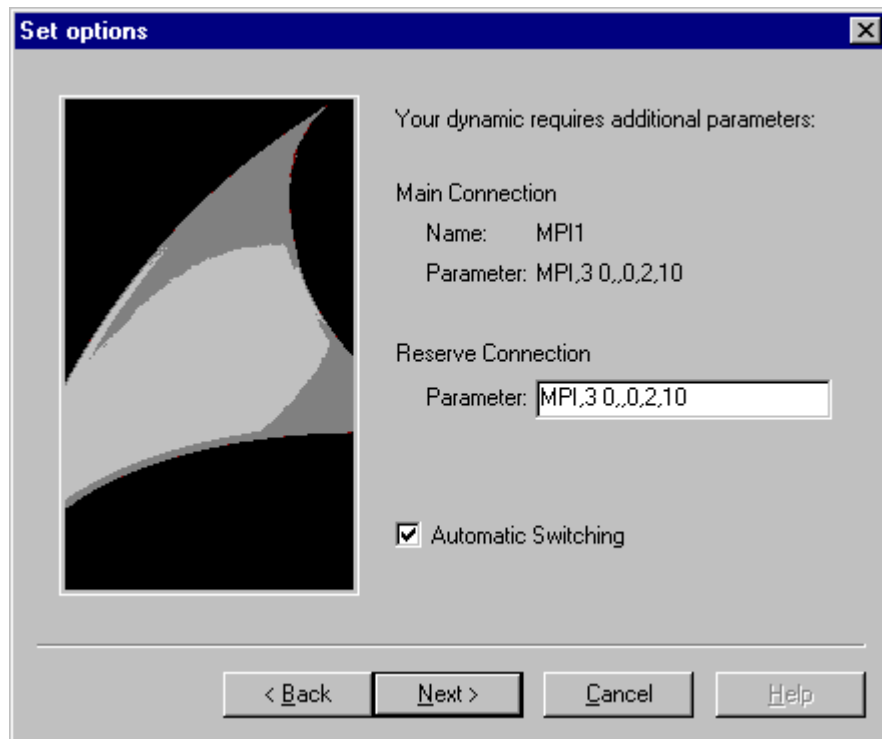
1. コンピュータのスタートアップパラメータで、[グローバルスクリプトランタイム]、[アラームロギングランタイム]、[グラフィックランタイム]モジュールを、有効にします。
詳細な情報は、「スタートアップパラメータの確認」を参照してください。
2. WinCC システムメッセージをアラームロギングにロードします。これらのシステムメッセージは、ソフトウェアの冗長化に関するメッセージを含んでいます。
このトピックの詳細については、「アラームロギングで WinCC システムメッセージを読む」を参照してください。
3. グラフィックデザイナの画面を開きます。[ダイナミックウィザード]ウィンドウで、[システムファンクション]タブを選択します。ダブルクリックして、ダイナミックウィザード"冗長化接続の設定"を起動します。



4. ウィザードの使い方の手順は、「ようこそ」で簡単に説明しています。[次へ]をクリックして、[オプションの設定]ダイアログを開きます。
5. メイン接続として使用される接続を選択し、[次へ]をクリックします。
これで、ウィザードが接続固有の内部タグを作成し、メイン接続のタググループ"@ "に保存します。

6.11 SIMATIC S7 Protocol Suite

6. [パラメータ]フィールドで、予備接続を確立する先の PLC のアドレスを入力します。
[自動スイッチング]チェックボックスをマークして、自動的に接続を切り替えるためのスクリプトを、ウィザードで生成します。
[次へ]をクリックします。
2 つの PLC への MPI 接続を示す冗長化のグラフィックが、画面に表示されます：



7. 行われた設定すべてが、[終了!]ダイアログに再度表示されます。訂正する場合は、単に[戻る]をクリックします。[終了]をクリックします。
ウィザードがスクリプトを生成し、グローバルスクリプトエディタのディレクトリ[C エディタ|アクション|アクション : <computername>]に、"@<connectionname>.pas"で保存します。

注記

以下の手順では、"冗長化接続の設定"ウィザードが使用されます。このウィザードは、ステップ 6 の完了時に接続固有内部変数を生成します。この時点でウィザードをキャンセルしているか、[終了]をクリックして手順を完了していない場合は、これらのタグは変わりません。

下記も参照

WinCC システムメッセージのアラームロギングへのロード方法 (ページ 512)

WinCC スタートアップパラメータをチェックする方法 (ページ 511)

WinCC でソフトウェアの二重化を削除する方法

はじめに

このセクションでは、WinCC で"SIMATIC S7 Protocol Suite"チャンネルの接続に対してソフトウェアの二重化を削除する方法について説明します。この機能を使用して非二重化接続に戻るよう PLC をコンフィグレーションする必要もありますが、これについてはこのマニュアルでは扱いません。

前提条件

- WinCC プロジェクトが無効になっていること。

手順

ソフトウェアの二重化を削除するステップは以下の 2 つです。

- "タグ管理"で、タグを含むタググループ"@<コンピュータ名>"を削除する。
- "グローバルスクリプト"で、スクリプト"@<接続名>.pas"を削除する。

手順

1. タグ管理で、ソフトウェアの二重化に対するメイン接続としてコンフィグレーションする接続を選択します。これには、ソフトウェア二重化の接続固有内部タグを持つタググループ"@<接続名>"が含まれています。このタググループを削除します。
2. アクション"@<接続名>.pas"に対するスクリプトを削除します。削除するには、"グローバルスクリプト"の C エディタを開きます。いくつかのサブディレクトリが表示されます。
3. [アクション|アクション:]ディレクトリを選択します。<コンピュータ名>".データウィンドウで、[アクション]タイプのスクリプト"@<接続名>.pas"を削除します。
4. "グローバルスクリプト"エディタを閉じます。

WinCC スタートアップパラメータをチェックする方法

手順

1. WinCC Explorer で、[コンピュータ]を選択します。
2. ショートカットメニューを開き、[プロパティ]を選択します。
3. [コンピュータリストプロパティ]ダイアログが開きます。[プロパティ]ボタンをクリックします。
4. [コンピュータプロパティ]ダイアログが開きます。

6.11 SIMATIC S7 Protocol Suite

5. [スタートアップ]タブをクリックし、エントリをチェックします。必要に応じ、ランタイムモジュールを有効/無効化するか、または他のアプリケーションをスタートアップリストに追加します。
6. 開いているダイアログを閉じます。

WinCC システムメッセージのアラームロギングへのロード方法

はじめに

このセクションでは、WinCC システムメッセージのプロジェクトへのロード方法を説明します。

手順

1. [アラームロギング]を開きます。
2. ナビゲーションエリアで、[システムメッセージ]ノードを選択します。
3. この表エリア、または[プロパティ]エリアのいずれかで使用するシステムメッセージの「使用済み」オプションを有効にします。
4. 「システムメッセージ」ノードのショートカットメニューからコマンド[使用される更新]を選択します。

接続障害中のエラーコード

「S7CHNdeu.chm」ファイルには、エラーコードの一覧が含まれます。

このファイルは、\SIEMENS\WinCC\bin のインストールパスにあります。

エラーコードのマニュアル

6.12 SIMATIC S7-1200, S7-1500 Channel

6.12.1 「SIMATIC S7-1200, S7-1500 Channel」 チャンネル

"SIMATIC S7-1200, S7-1500 Channel"は、WinCC ステーションと S7-1200 および S7-1500 オートメーションシステムとの間の通信に使用されます。

通信は TCP/IP プロトコルを使用して行われます。

このチャンネルは、冗長化 S7-1500R/H システムもサポートします。詳細情報は、「チャンネルの設定」 > 「冗長システム S7-1500R/H (ページ 534)」を参照してください。

チャンネルユニット

"SIMATIC S7-1200, S7-1500 Channel"には "OMS+" チャンネルユニットが付属しています。

チャンネルの診断

ランタイム時にコントローラの障害とエラーを表示するには、WinCC SysDiagControl を使用します。

詳細については、「通信診断 > 診断」チャンネル SIMATIC S7-1200/S7-1500 (ページ 638)」を参照してください。

推奨される通信プロセッサ

WinCC ステーションとオートメーションシステム S7-1200 または S7-1500 の間の通信には、以下の通信プロセッサが推奨されます:

- CP 1612 A2
- CP 1613 A2
- CP 1623
- CP 1628

下記も参照

SysDiagControl によるシステム診断 (ページ 638)

冗長システム S7-1500R/H (ページ 534)

6.12.2 サポートされているデータタイプの概要

はじめに

タグが構成されると、オートメーションシステム(AS)のデータタイプとデータフォーマットに対するフォーマットの適用が指定されます。

以下の表に、このチャンネルでサポートされるデータタイプと、タイプ変換の使用法を示します。

サポートされるデータタイプ/フォーマットの適用

データタイプ	タイプ変換
2 進タグ	いいえ
符号付き 8 ビット値	はい
符号なし 8 ビット値	はい
符号付き 16 ビット値	はい
符号なし 16 ビット値	はい
符号付き 32 ビット値	はい
符号なし 32 ビット値	はい
浮動小数点数 32 ビット IEEE 754	はい
浮動小数点数 64 ビット IEEE 754	はい
テキストタグ、8 ビットフォント	いいえ
テキストタグ、16 ビット文字セット	いいえ
未処理データタグ	いいえ
日付/時刻	はい

6.12.3 チャンネルのコンフィグレーション

6.12.3.1 "SIMATIC S7-1200, S7-1500 Channel"チャンネルのコンフィグレーション

概要

WinCC では、オートメーションシステムとの WinCC の通信に、論理接続が必要です。

このセクションでは、"SIMATIC S7-1200, S7-1500 Channel"の設定方法を説明します。

通信チャンネルを設定するには、タグ管理のショートカットメニューから、[新規ドライバの追加] > [SIMATIC S7-1200, S7-1500 Channel]を選択します。

接続パラメータ

S7 ネットワークアドレス

選択された製品ファミリーに依存するアドレスです。

- S7-1200 または S7-1500 製品ファミリーの場合、TCP/IP 接続の IP アドレスを入力します。
- 製品ファミリー WinAC S7-1500 の場合、S7-1507S に対してステーションアドレスを入力します。

アクセスポイントは、選択した製品ファミリーに適しているインターフェースを参照する必要がありますことに注意してください。

TCP/IP 接続

TCP/IP プロトコルの使用時に、論理接続用にオートメーションシステムの IP アドレスを定義する必要があります。

IP アドレスは、ドットで区切られた 4 つの数値で構成されます。数値は「0～255」の範囲でなければなりません。

注記

タイムアウト動作

TCP/IP プロトコルの使用時には、中断された接続を即座には検出できません。メッセージのチェックバックに 1 分間かかります。

ステーションアドレス

ステーションアドレスは、Profibus アドレスの数値範囲内にあります。

S7-1507S の設定中に、[インデックス]の[プロパティ]ダイアログでステーションアドレスを検索できます。

アクセスポイントとして、[PC 内部(ローカル)]インターフェースを選択します。

パスワードを使用した接続アクセスの保護

接続に「SIMATIC S7-1200, S7-1500 Channel」チャンネルを使用している場合、オートメーションシステムへのアクセスをパスワードで保護できます。

6.12 SIMATIC S7-1200, S7-1500 Channel

このアクセス保護には、オートメーションシステムでレベル 1、2 および 3 が定義されています。

WinCC での設定中に、必要なレベルに対して設定されたパスワードを適用します。

パスワードを設定しない場合、AS で設定されたレベルが自動的に使用されます。

接続ステータスの指定および決定

各接続について、内部タググループ「ConnectionStates」に以下のシステムタグを作成できます。

- 接続の確立/終了:
@<Connectionname>@ForceConnectionStateEx
- 接続ステータスの問い合わせ:
@<Connectionname>@ConnectionStateEx

接続ステータスに関する詳細は、WinCC 情報システムの「WinCC プロセス通信」 > 「ランタイム時の接続ステータスタグの設定 (ページ 170)」を参照してください。

ソフトウェアの冗長性

S7-1500R/H のソフトウェア冗長性を使用して作業する場合、次のタグを作成します。

- 冗長化接続の確立/終了:
@<Connectionname>@ForceConnectionState
- 冗長化接続ステータスの問い合わせ:
@<Connectionname>@ConnectionState

ソフトウェア冗長性のためのシステムタグは、「S7-1500R/H のソフトウェアの冗長化 (ページ 540)」で説明されています。

タグの設定

WinCC とオートメーションシステム間での"SIMATIC S7-1200, S7-1500 Channel"チャンネルを使用した接続の場合、WinCC で異なるデータタイプのタグが作成されます。対応する接続に対してプロセスタグを設定するか、オートメーションシステムの AS シンボルを WinCC タグ管理にロードします。

タグの設定は、オートメーションシステムのデータエリアでのアドレス指定によって異なります。

注記

TIA Portal の AS 設定のみ

オートメーションシステムの設定は TIA Portal でのみ変更できます。

HMI システムまたは Web サーバーのデータの上書き

操作しているとき、HMI システムまたは Web サーバーのデータが S7-1500 で上書きされることがあります。操作中、2 つのプロセス(PLC システムおよび HMI システム)が互いに独立して並行して実行されています。

両方のシステムが同じタグへの書き込みアクセスを行う場合、システム側のデータが上書きされることがあります。

詳細については、インターネットを参照してください:

- エントリ ID 109478253 の FAQ: 「HMI システムまたは Web サーバーのデータが S7-1500 でときどき上書きされるのはなぜですか? (<https://support.industry.siemens.com/cs/ww/en/view/109478253>)」

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

未処理データタグの設定 (ページ 517)

S7-1500R/H のソフトウェアの冗長化 (ページ 540)

<https://support.industry.siemens.com/cs/ww/en/view/109478253> (<https://support.industry.siemens.com/cs/ww/en/view/109478253>)

6.12.3.2 未処理データタグの設定

はじめに

"SIMATIC S7-1200, S7-1500 Channel"チャンネルは「未処理データタグ」データタイプをサポートしています。

「SIMATIC S7-1200, S7-1500 Channel」チャンネルの未処理データタグ

バイト配列としての未処理データタグは、WinCC と PLC 間でユーザデータブロックを転送するのに使用されます。この未処理データタグはユーザデータしか処理しません。

未処理データタグでは、例えば C スクリプト経由のタグ要求などコントローラの非循環読み取りサービスのみがサポートされます。

「SIMATIC S7-1200, S7-1500 Channel」チャンネルは、未処理データタグの循環読み取りサービスをサポートしません。

未処理データタグのアドレッシング

バイト配列としての未処理データタグは、データ領域のアドレスおよび長さ(例:DB1、DBB10、長さ 100 バイト)によってアドレス指定される普通のプロセスタグと同じようにチャンネル内で扱われます。

WinCC タグの「未処理データタグ」データタイプでは、「バイト」だけが可能です。未処理データ範囲の長さを除いて、パラメータは事前設定されており変更できません。

データブロックの長さ

通信ドライバによって送信できるデータブロックの最大長さを遵守してください。

- S7-1200 / S7-1500:データブロックの長さ最大 8000 バイト

大型のデータボリュームの転送

未処理データを使用してコントローラから WinCC に大容量データを転送できるようにする場合、37873547 がアプリケーション例に書き込まれます。

- S7-300/400/1500 と WinCC 間での大容量データの送受信 (<https://support.industry.siemens.com/cs/ww/en/view/37873547>)

下記も参照

"SIMATIC S7-1200, S7-1500 Channel"チャンネルのコンフィグレーション (ページ 514)

S7-300/400/1500 と WinCC 間での大容量データの送受信 (<https://support.industry.siemens.com/cs/ww/en/view/37873547>)

6.12.3.3 接続の設定方法

はじめに

"SIMATIC S7-1200, S7-1500 Channel"の設定には、以下のステップが必要です。

1. 接続の設定
2. タグの設定

必要条件

- "SIMATIC S7-1200, S7-1500 Channel"の通信ドライバーがインストールされ、プロジェクトに統合されています。
- SIMATIC プロジェクトは、オートメーションシステムで設定し、使用できます。

手順

1. WinCC エクスプローラの[タグ管理]エディタで"SIMATIC S7-1200, S7-1500 Channel"通信ドライバのメニュー構造を開きます。
2. チャンネルユニット"OMS+"のショートカットメニューで、[新規接続]エントリを選択します。
3. 接続の名前を入力します。
4. 接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続]ダイアログが開きます。
5. [製品ファミリー]から、オートメーションシステムを選択します。
6. オートメーションシステムの IP アドレスまたはステーションアドレスを入力します。
7. [アクセスポイント]を選択します。
アクセスポイントは、選択した製品ファミリーに属しているインターフェースを参照する必要があります。
8. オートメーションシステムのアクセス保護に使用するパスワードを入力します。
 - [変更]をクリックします。
 - 必要レベルに使用するパスワードを入力します。
 - [パスワードを再入力]フィールドで入力内容を繰り返します。
9. [OK]をクリックしてダイアログを閉じます。
10. 接続確立と接続ステータスのシステムタグを作成するには、接続のショートカットメニューで[有効化/無効化タグの作成]エントリを選択します。
内部タググループ「ConnectionStates」に以下のタグが作成されます。
 - @<Connectionname>@ForceConnectionStateEx
 - @<Connectionname>@ConnectionStateEx

6.12 SIMATIC S7-1200, S7-1500 Channel

S7-1500 ソフトウェアコントローラの制約

Simatic-Net を使用する S7-1507S の非互換性のため、ソフト PLC への Simatic-Net のインストールは現在不可能です。

これは、Simatic-Net が欠落しており、追加チャンネルを介した外部オートメーションシステムへの接続が不可能であることを意味しています。

OPC UA などの他のチャンネルを使用可能です。

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

6.12.3.4 最適化されたブロックアクセスなしでのタグの構成方法

はじめに

このセクションでは、オートメーションシステムのアドレス領域に最適化されたブロック単位でアクセスしない、WinCC のタグの設定方法を説明します。

必要条件


- TIA Portal のデータブロックでは、[ブロックアクセスの最適化]プロパティを無効にしています。
- "SIMATIC S7-1200, S7-1500 Channel"をプログラムに統合する必要があります。
- 接続は"OMS+"チャンネルユニットで作成する必要があります。

8 ビットテキストタグの設定に関する注意事項

"SIMATIC S7-1200, S7-1500 Channel"の 8 ビットのテキストタグについては、WinCC ではコントロールワードと文字列のユーザーデータで構成される S7 文字列タイプのみをサポートしています。

- WinCC の 8 ビットのテキストタグを設定するには、オートメーションシステム(AS)メモリ内のユーザーデータの前に存在するコントロールワードのアドレスを入力します。コントロールワードの第 1 バイトにはカスタマイズされた最大文字列長が格納され、第 2 バイトには実際の長さが格納されます。
- オートメーションシステムメモリでのデータ構造の作成に関して、WinCC で 8 ビットのテキストタグ用に設定された長さは、2 バイトのコントロールワードが増加することに注意してください。
8 ビットのテキストタグのデータ構造体がメモリ内で次々と直接入力されると、後続データが上書きされます。
- 読取り時には、コントロールワードとユーザーデータと一緒に読み取られ、第 2 バイトにある現在の長さが評価されます。
第 2 コントロールバイトに含まれている現在の長さに対応するユーザーデータのみが、WinCC の 8 ビットのテキストタグに転送されます。
- 書込み時には、文字列の実際の長さが決定され(「0」文字)、現在の長さが格納されているコントロールバイトがユーザーデータと一緒にオートメーションシステムに渡されます。

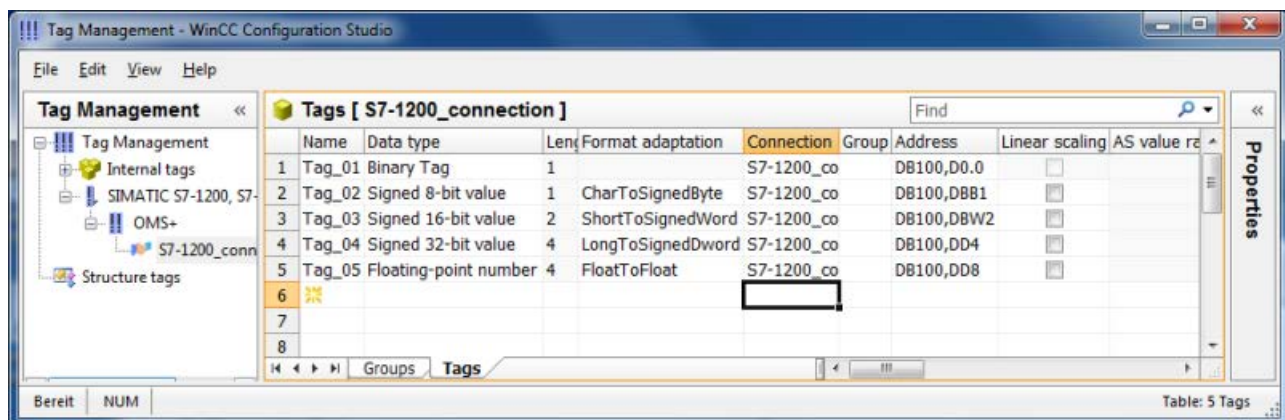
手順

- 必要な接続を選択します。
- テーブルエリアの下にある[タグ]タブをクリックします。
- 「名前」列の一番上の空きセルにタグの名前を入力します。
テーブルエリアで、または[プロパティ-タグ]データエリアの右側で、以下の設定を設定します。
- サポートされているデータタイプの 1 つを選択します。
- [アドレス]列で  をクリックします。
- タグアドレスを入力します。
- タグに品質コードがあり、それを WinCC で使用したい場合、[品質コード]チェックボックスを選択します。
オートメーションシステムにもコードが存在する必要があります。
チェックボックスは、データ領域で[DB]が選択されている場合のみ有効になります。
- [OK]をクリックしてダイアログを閉じます。

6.12 SIMATIC S7-1200, S7-1500 Channel

結果

ブロックアクセスが最適化されていないタグは、タグ管理で設定されます。



6.12.3.5 最適化されたブロックアクセスありでのタグの構成方法

はじめに

このセクションでは、オートメーションシステムのアドレス領域に最適化されたブロック単位でアクセスする、WinCC のタグの構成方法を説明します。

コントローラから WinCC プロジェクトにタグをインポートします。

オンライン変更のロードはできません

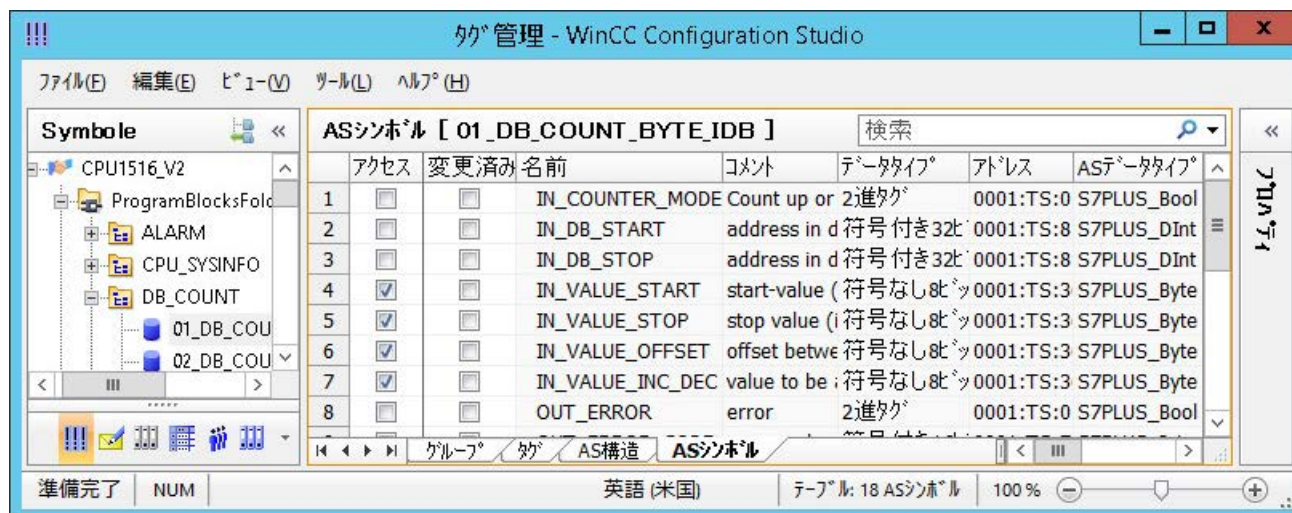
オンライン変更のロードを使用して、ランタイム時の[AS へのロード]を介して AS シンボルとして作成したタグを転送できません。

必要条件

- TIA Portal のデータブロックでは、[ブロックアクセスの最適化]プロパティが有効にされていること。
- "SIMATIC S7-1200, S7-1500 Channel"がプロジェクトに統合されていること。
- 接続は「OMS+」チャンネルユニットで作成する必要があること。
- この接続をランタイムで確立させる必要があること。

手順

1. 必要な接続を選択します。
2. 接続のショートカットメニューで[AS シンボル] > [AS から読み取り]を選択します。
使用可能なコントローラデータがロードされ、[シンボル]ビューが開きます。
ロードされたデータは、[AS シンボル]タブのテーブルエリアで表示されます。
ロードされているデータが構造も含んでいる場合、[AS 構造]タブが追加で表示されます。



3. AS シンボルは、自動的にタグ管理に含まれません。
必要な AS シンボルを[タグ]タブへ転送するには、[アクセス]列でそれぞれのチェックボックスを有効にします。
これで、選択したタグが、タグ管理で使用可能になります。



コントローラに接続しないで AS シンボルを編集

コントローラへの接続とは独立して、AS シンボルをオフラインで構成することができます。

これを行うには、ロードされた AS シンボルをファイルに保存します。

1. 必要な接続を選択します。
2. 接続のショートカットメニューで[AS シンボル] > [ファイルに保存]を選択します。

次に、AS シンボルをオフラインプロジェクトのタグ管理にロードすることができます。

1. 必要な接続を選択します。
2. 接続のショートカットメニューで[AS シンボル] > [ファイルからロード]を選択します。

オフライン設定に関する詳細情報は、「AS シンボルをオフラインでダウンロードする方法 (ページ 525)」を参照してください。

WinCC タグとコントローラとの同期化

コントローラまたはファイルをロードすると、タグ管理は AS シンボルのプロパティを確認します。

アドレス、データタイプおよびタグ名が WinCC プロジェクトの AS シンボルのプロパティと比較されます。

- シンボルのプロパティが一致しないと、[AS シンボル]タブの[変更済み]フィールドが有効化されます。
それぞれのプロパティフィールドが赤で強調表示されます。フィールドのヒントには追加情報が記載されています。
- WinCC タグがコントローラに見つからない場合は、関連付けられた AS シンボルの行全体が赤でハイライト表示されます。

これはたとえば次の場合に発生します。

- この WinCC プロジェクトは WinCC V7.3 で作成されました。移行されたプロジェクトにはまだすべての同期情報が含まれていません。
- AS シンボルのアドレスがコントローラで変更されました。これは、TIA Portal での設定の変更などによるものです。
- データタイプまたは AS シンボルの名前が変更されました。
- AS シンボルがコントローラで削除されました。

タグの再インポート

プロパティを同期するには、WinCC プロジェクトで使用している AS シンボルを更新します。

以下の手順を実行します。

1. [AS シンボル]タブで変更済みの AS シンボルを選択します。
移行されたプロジェクトを更新するには、すべての行を選択します。
2. [変更済み]フィールドの選択を解除します。

AS 設定のパラメータが再び読み取られ、タグ管理で使用されます。

注記

TIA Portal プロジェクトを移行する前に:AS シンボルの更新

TIA Portal バージョンをアップグレードするには、次の順序に従ってください。

1. WinCC タグとして使用しているすべての AS シンボルを更新します。
2. TIA Portal プロジェクトを移行します。
3. コントローラを TIA Portal にロードします。
4. WinCC タグとして使用しているすべての AS シンボルをもう一度更新します。

ロードした後、WinCC タグの AS シンボルへの割り付けが WinCC プロジェクトで維持されていることを確認します。

そうでない場合、割り付けがもはや最新でないため、タグを読み取ることができません。

下記も参照

AS プロジェクトデータのエクスポート方法 (ページ 532)

AS 構造の構成方法 (ページ 530)

6.12.3.6 AS シンボルをオフラインでダウンロードする方法

はじめに

以下の S7 チャンネルをオフラインで設定できます。

- SIMATIC S7 Protocol Suite
- SIMATIC S7-1200, S7-1500 Channel

この目的に対して、例えば、既存の TIA Portal プロジェクトからデータレコードをエクスポートし、エクスポートファイルを WinCC プロジェクトにロードします。

6.12 SIMATIC S7-1200, S7-1500 Channel

サポートされているエクスポートフォーマット

以下のファイルフォーマットがインポートでサポートされます。

フォーマット	内容	説明
*.bin	2 進データ	WinCC タグ管理からのエクスポート: <ul style="list-style-type: none"> • [タグ管理]の表示 > 接続のショートカットメニュー: [AS シンボル] > [ファイルに保存] 「SIMATIC S7 Protocol Suite」チャンネルによりサポートされていません。
*.sdz	ストラクチャードエクスポート	WinCC タグ管理からのエクスポート: <ul style="list-style-type: none"> • [シンボル]の表示 > [メニュー]: [編集] > [エクスポート] また、ナビゲーションエリアから構造情報をエクスポートします。
*.zip	TIA Portal エクスポートファイル	[SIEMENS SIMATIC SCADA Export]ツールを使用して TIA Portal からエクスポート

TIA Portal の「SIEMENS SIMATIC SCADA Export」

データレコードを TIA Portal プロジェクトからエクスポートするには、「SIEMENS SIMATIC SCADA Export」ツールを使用します。

TIA Portal プロジェクトにおいて、PLC のショートカットメニューで[SIEMATIC SCADA にエクスポート]エントリを選択します。

各種 TIA Portal バージョン用のツールを Industry Online Support でダウンロードして使用することができます。

- 「SIMATIC SCADA Export」ダウンロード (ID 109748955) (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)
- 「SIMATIC SCADA Export」ドキュメント (ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

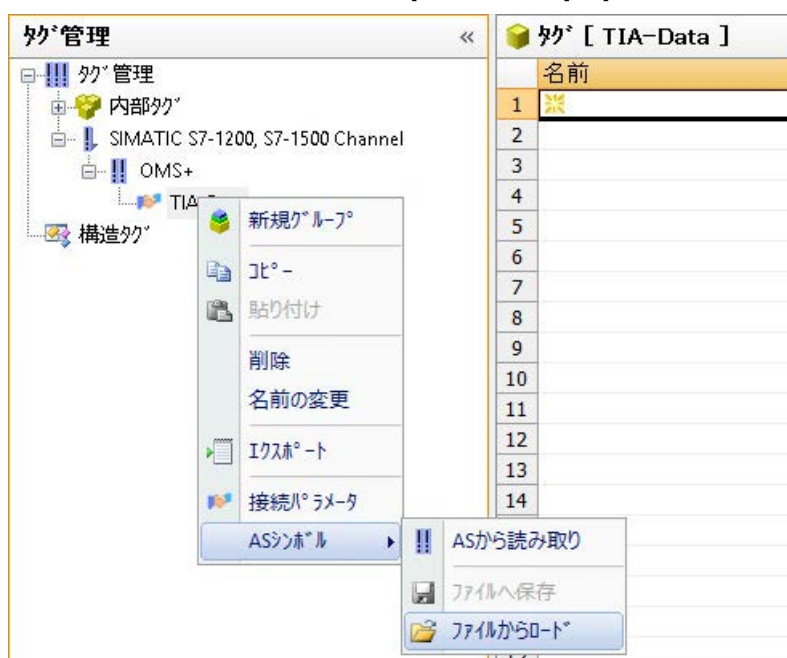
必要条件

- AS が TIA Portal でコンパイルされていること。
- PLC の対応する設定データがエクスポートされ、.zip ファイルなどで利用可能になります。

- 通信プロセッサと関連するハードウェアドライバが WinCC プロジェクトにインストールされていること。
- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。
- [タグ管理]エディタが開いていること。

手順

1. 接続のショートカットメニューで[AS シンボル]>[ファイルからロード]を選択します。



2. ロード対象の希望するデータレコードを選択します。
利用可能なコントローラデータがロードされます。

結果

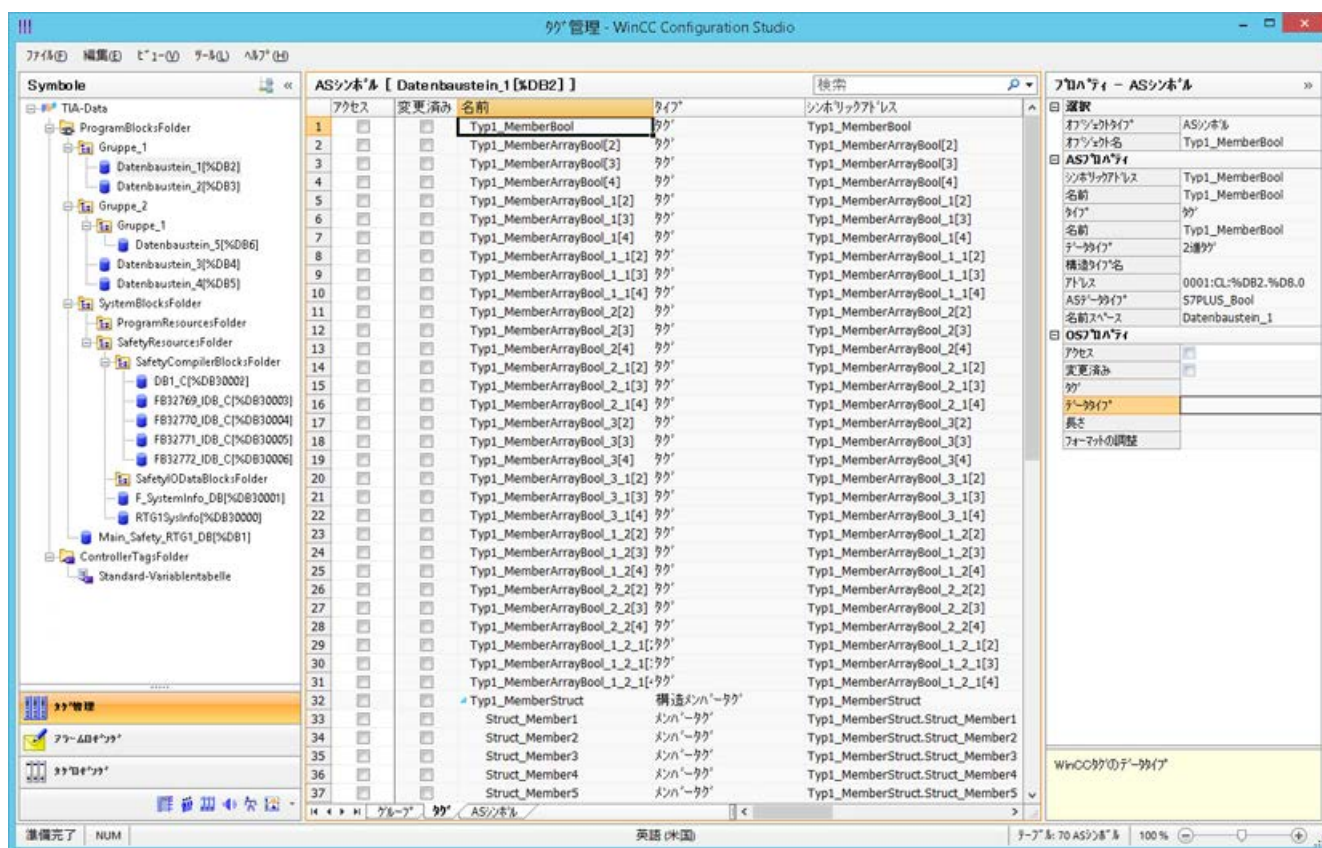
設定がインポートされ、[シンボル]の表示が開かれます。

ロードされたデータがテーブルエリアの[AS シンボル]タブに表示され、後続の処理に対して利用できるようになります。

ロードされているデータが構造も含んでいる場合、[AS 構造]タブが追加で表示されます。

エディタが閉じられた後に、[AS シンボル]と[AS 構造]タブが再度非表示になります。

6.12 SIMATIC S7-1200, S7-1500 Channel



シンボルの表示

次のボタンを使用して、タグ管理をデフォルト表示と[シンボル]の表示の間に切り替えます。

ボタンは、データレコードがロードされた後にのみ使用可能になります。

ナビゲーションエリア

構造ツリーのデータの表示は、TIA Portal からの階層に対応しています。

テーブルエリア

検出された WinCC タグが AS タグに一致しないとき、[変更済み]列のチェックボックスは、自動的に選択されます。また、ユーザーがこれらを使用してフィルタリングできるようにします。

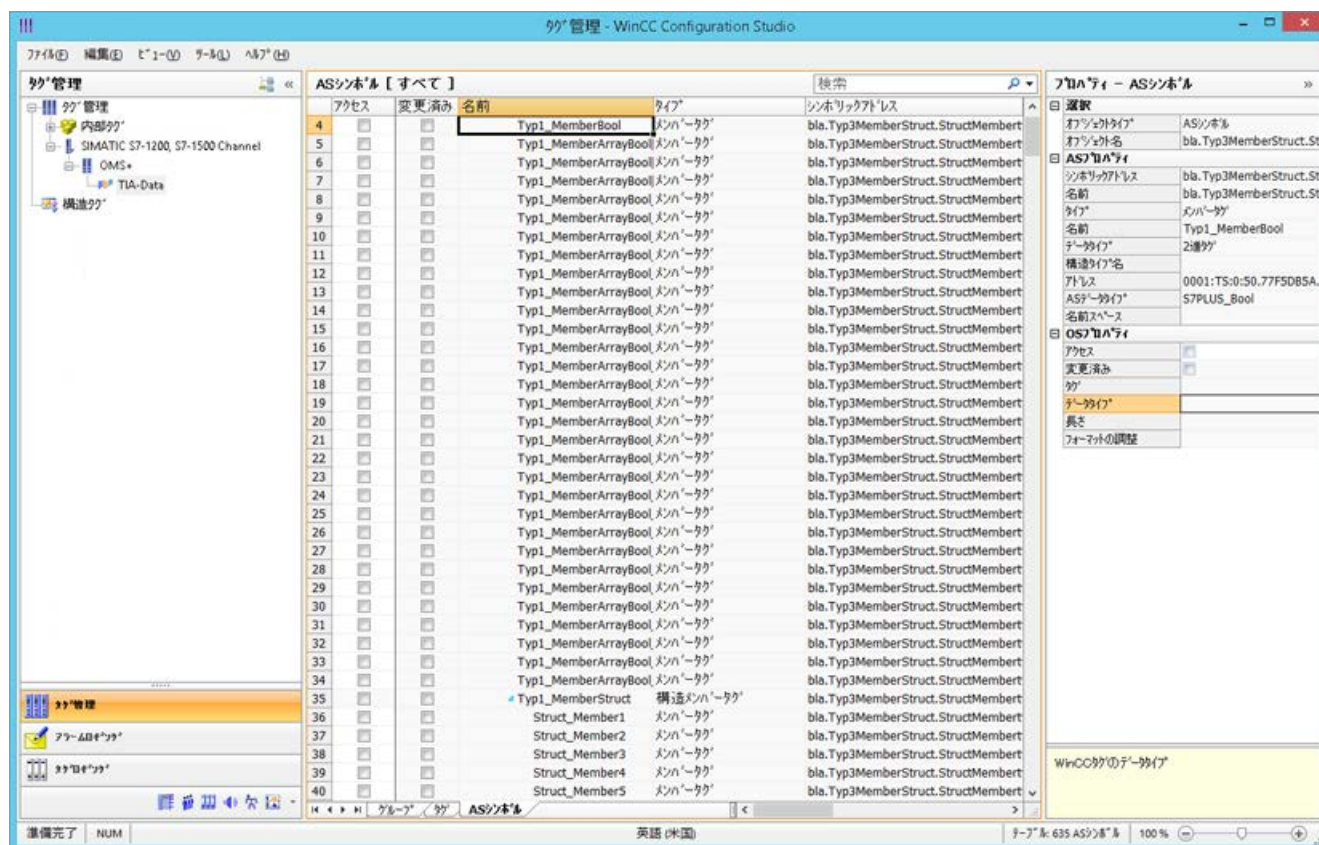
[アクセス]列のチェックボックスを選択することで、検出された AS タグから WinCC タグを作成します。

[タグ管理]の AS シンボル

[AS シンボル]タブを介した[タグ管理]の AS シンボルへのアクセス権限もあります。

データブロック特有の[シンボル]の表示とは対照的に、コントローラのすべての利用可能なタグがここに表示されます。

この表示はまた、AS にもはや存在していない以前に設定されていたタグ也表示します。



下記も参照

AS プロジェクトデータのエクスポート方法 (ページ 532)

AS 構造の構成方法 (ページ 530)

「SIMATIC SCADA Export」ドキュメント (ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

「SIMATIC SCADA Export」ダウンロード (ID 109748955) (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)

6.12.3.7 AS 構造の構成方法

はじめに

AS シンボルをロードする場合、コントロールシステムの構造もインポートされます。
手順は通信チャンネルに依存します。

- SIMATIC S7 Protocol Suite:
 - ファイルからのロード
- SIMATIC S7-1200, S7-1500 Channel
 - ファイルからのロード
 - AS からのロード

タグ管理の AS 構造

AS 構造は、デフォルトビューおよび[AS 構造]タブの[シンボル]ビューで表示されます。
WinCC で AS 構造を使用するには以下の方法があります。

- AS 構造タグ用に WinCC 構造タイプを作成します。
構造が、WinCC タグ管理の[構造タグ]に、構造タイプとして作成されます。
また、構造タイプ要素は、含まれている[タグタイプメンバ]それぞれに対して作成されます。
- WinCC 構造タイプを AS 構造タグに割り付けます。
次いで、各[タグタイプメンバ]に対して選択された構造タイプの構造タイプ要素を選択します。

タグ管理で、WinCC 構造タイプの名前および構造タイプ要素の名前を変更します。AS 構造の割り付けは、自動的に調整されます。

必要条件

- 次の方法のいずれかによって、PLC の設定データにアクセスしていること。
 - PLC への接続がランタイム時に確立された。
 - エクスポートされた設定データが、例えば zip ファイルとして使用可能である。
- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。

手順

1. [AS から読み取り]または[ファイルからロード]を介して AS シンボルをロードします。
ロードされたメッセージは、[シンボル]タグ管理ビューに表示されます。
ロードされた構造は、[AS 構造]タブで表示されます。
AS からロードするとき、構造名は転送されません。ID が、[構造タグタイプ]の名前として表示されます。

Symbol			AS 構造		
CPU1516_V2			WinCC構造	名前	タイプ
ProgramBlocksFolder			1	SimData1UDT	構造タグタイプ
01_DBs			2	LEN_UDT1	構造タグタイプ

2. [AS 構造]をクリックします。
構造の要素を表示するには、構造名の前の矢印をクリックします。
3. 構造の行全体を選択し、ショートカットメニューから[構造を作成]エントリを選択します。

AS 構造			
WinCC構造	名前	タイプ	
1 SimData1UDT	SimData1UDT	構造タグタイプ	
2 BitData	BitData	タグタイプメンバー	
3 IntData	IntData	タグタイプメンバー	
4 RealData	RealData	タグタイプメンバー	
5 ByteData	ByteData	タグタイプメンバー	
6 LEN_UDT1	LEN_UDT1	構造タグタイプ	
7		タグタイプメンバー	
8		構造タグタイプ	
9		構造タグタイプ	
10			
11			
12			
13			
14			
15			

あるいは、WinCC タグ管理で作成済みの構造タイプを選択します。
次いで、構造タイプ要素を[タグタイプメンバ]に割り付けます。

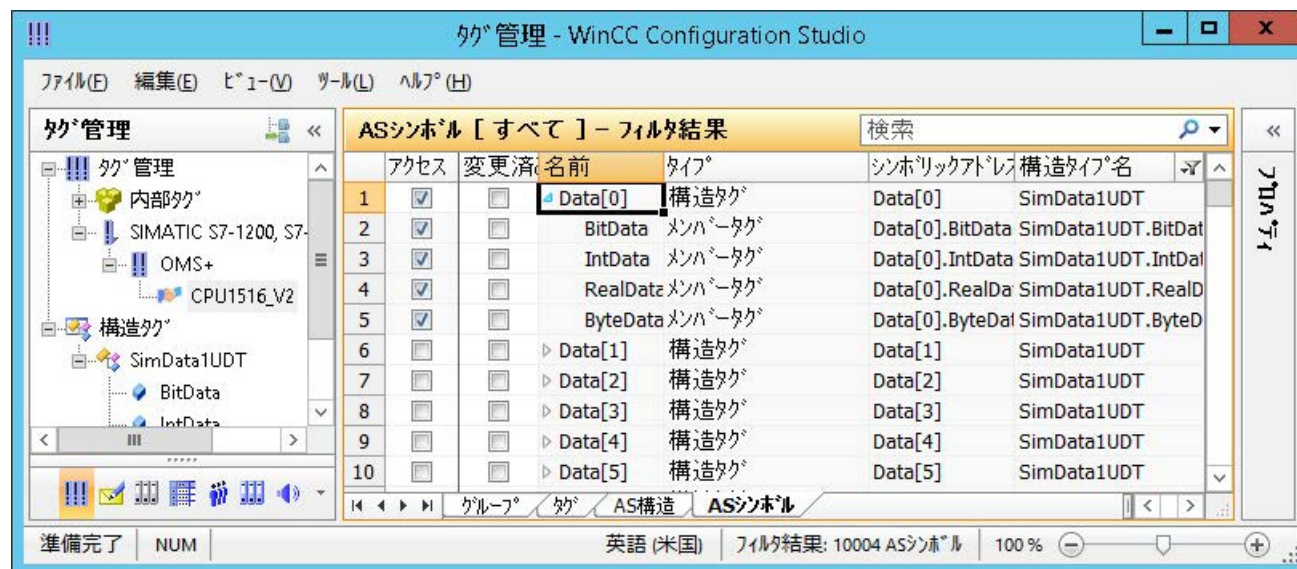
AS 構造		
WinCC構造	名前	タイプ
1 SimData1UDT	SimData1UDT	構造タグタイプ
2 LEN_UDT1		
3 WinCC_StrucType1		

構造タイプが、AS 構造の各[構造タグタイプ]に対して、WinCC タグ管理で作成されます。
構造タイプ要素が、各[タグタイプメンバ]に対して作成されます。

4. [タグ管理]ビューの[AS シンボル]タブを選択します。

6.12 SIMATIC S7-1200, S7-1500 Channel

5. 構造タグとメンバタグのみを表示させるには、[構造タイプ名]列で希望する AS 構造にフィルタを掛けます。



6. WinCC タグ管理で AS 構造タグにアクセスするには、[アクセス]フィールドを有効にします。含まれているメンバタグが自動的に有効になります。
AS 構造タグは、構造タグとして、WinCC タグ管理で作成されます。

結果

WinCC タグ管理の構造タイプおよび構造タグを介して、AS 構造タグにアクセスできます。
このようにして、例えば、WinCC フェイスプレートタイプの AS 構造にアクセスし、フェイスプレートインスタンスで表示することができます。

下記も参照

- AS シンボルをオフラインでダウンロードする方法 (ページ 525)
- AS プロジェクトデータのエクスポート方法 (ページ 532)
- 最適化されたブロックアクセスありでのタグの構成方法 (ページ 522)

6.12.3.8 AS プロジェクトデータのエクスポート方法

AS シンボルのエクスポート

オフライン設定にエクスポートファイルを使用します。

AS プロジェクトデータを次のフォーマットでエクスポートできます。

通信チャンネル	エクスポートされたデータ	エクスポートファイルのフォーマット
SIMATIC S7-1200, S7-1500 Channel	AS シンボルおよび AS 構造	2 進データ: *.bin 構造化されたエクスポート: *.sdz
SIMATIC S7 Protocol Suite	AS シンボルおよび AS 構造	構造化されたエクスポート: *.sdz

必要条件

- 接続が「SIMATIC S7-1200, S7-1500 Channel」または「SIMATIC S7 Protocol Suite」で作成されていること。
- AS プロジェクトデータをロードし、WinCC で設定しています。

手順:2 進データのエクスポート

1. タグ管理で接続を選択します。
2. ショートカットメニューから、[AS シンボル]>[ファイルに保存]エントリを選択します。
[エクスポート]ダイアログが開きます。
3. 保存パスを選択して、ファイル名を入力します。
[エクスポート]ボタンを押してダイアログを閉じます。
設定データが 2 進データセットとして.bin ファイルにエクスポートされます。

手順:構造化済みデータのエクスポート

1. タグ管理で[シンボル]ビューを選択します。
2. [編集]>[エクスポート]メニューコマンドを選択します。
3. 保存パスを選択して、ファイル名を入力します。
[エクスポート]ボタンを押してダイアログを閉じます。
設定データは*.sdz ファイルにエクスポートされます。
構造化済みのエクスポートには、ナビゲーションエリアからの構造情報も含まれます。

下記も参照

最適化されたブロックアクセスありでのタグの構成方法 (ページ 522)

AS シンボルをオフラインでダウンロードする方法 (ページ 525)

AS 構造の構成方法 (ページ 530)

6.12.3.9 冗長システム S7-1500R/H

CPU 冗長性

CPU 冗長性は、CPU の利用可能性が 2 倍になることを意味しています。SIMATIC メモリカードを、それぞれの CPU に挿入する必要があります。

冗長操作中に、両方の CPU が並行してユーザープログラムを実行します。

一方の CPU が故障すると、他方の CPU がプロセスに対する管理を維持します。

プライマリ CPU	CPU がプロセスを管理する R/H システムが RUN 冗長化システム状態になっている場合、プライマリ CPU がこのプロセスを実行します。
バックアップ CPU	次の CPU バックアップ CPU は、同時にユーザープログラムを実行し、プロセス制御がプライマリ CPU の障害イベントになっていると見なし ます。

追加情報は、製品サポートにある S7-1500R/H のマニュアルで参照してください。

- 「S7-1500」 > 「冗長化 CPU」 (<https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mfn=ps&pnid=25152&lc=en-WW>)
- 「冗長化システム S7-1500R/H マニュアル」 (<https://support.industry.siemens.com/cs/ww/en/view/109754833>) (2018 年 10 月)

オペレーティング目的

実際には、冗長化オートメーションシステムを使用して、高可用性やフェールセーフ機能を実現します。

- フォールトトレラントシステム:
2 つのシステムの並行操作によって、生産ダウンタイムの可能性を低減します。
- フェールセーフシステム:
安全シャットダウンから安全アイドリング位置によって、生命、環境および資本を保護
します。

**警告****フォールトトレラントシステムとフェールセーフシステムの違い**

S7-1500R/H は、フォールトトレラントオートメーションシステムですが、フェールセーフシステムではありません。

S7-1500R/H システムを使用して、安全関連プロセスを制御してはなりません。

使用方法

冗長化操作システムでは、個別のオートメーションコンポーネントの故障や誤作動はシステムの操作に影響してはなりません。

冗長化システム S7-1500R/H は、例えば、次のような分野で使用されています。

- トンネル
- 空港(例、荷物ハンドリングシステム)
- 地下鉄
- 造船
- 廃水処理プラント
- 高天井倉庫

S7-1500 R/H CPU のファンクション

S7-1500R/H 冗長化システムは、PROFINET リングのメディア冗長性(MRP)をベースにしています。

CPU の表示は、異なるメニューでコントロールおよびステータス情報を表示します。診断メッセージへの高速アクセスは、整備中のシステムのダウンタイムを最小化します。

CPU は、ドライブおよび制御ループの有効なコミッショニングおよび最適化のための、すべての CPU 変数のトレース機能をサポートしています。

メディアの冗長性(MRP)

「メディアの冗長性プロトコル」(MRP)は、ネットワークおよびシステムの使用可能性を保証するための機能です。

6.12 SIMATIC S7-1200, S7-1500 Channel

冗長化システムの2つのCPUは、MRPメディア冗長化プロトコルを使用するPROFINETリングに存在している必要があります。

- S7-1500Rは、PROFINETリングを使用して、2つのCPUを同期します。
- S7-1500Hは、光ファイバーケーブルを介した冗長化接続を使用して、2つのCPUを同期します。
H-CPUのPROFINET X1 インターフェースを介したPROFINETリングも、S7-1500Hに必須です。

メディア冗長性およびシステム冗長性S2に対する追加情報:

- 「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>): 「システム概要」 > 「S7-1500 R/H-CPU」 セクション
- 「STEP 7 V15 を使用する PROFINET」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/49948856>)

下記も参照

冗長化制御システムの設定 (ページ 536)

S7-1500R/H のソフトウェアの冗長化 (ページ 540)

検索: 「S7-1500」 > 「冗長化 CPU」 (<https://support.industry.siemens.com/cs/products?dtp=Manual&mt=2001&mfn=ps&pnid=25152&lc=en-WW>)

「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>)

「STEP 7 V15 を使用する PROFINET」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/49948856>)

6.12.3.10 冗長化制御システムの設定

STEP 7 および TIA Portal で CPU 冗長性を完全に設定します。

WinCC では、S7-1500 R/H-CPU の IP アドレスへの対応する接続のみを作成し、ソフトウェア冗長性のためにスクリプトを作成します。

IP アドレス

デバイスの IP アドレス

アクセス可能にする CPU および IO デバイスのインターフェースのため、ネットワーク内のインターフェースには、一意な IP アドレス、デバイス IP アドレスが必要です。

[ソフトウェア冗長性]ファンクションのためにデバイスの IP アドレスを使用します。

MAC アドレス

CPU には、各インターフェースおよびそのポートに対する、一意な MAC アドレスがあります。

PROFINET ポートの MAC アドレスは、LLDP プロトコルに必要です(例、「近接検出」機能用)。

MAC アドレスの番号範囲は連続しています。最初および最後の MAC アドレスは、CPU の右側のタイプラベルに印字されています。

システム IP アドレス

冗長化システム S7 1500R/H は、付加的にシステム IP アドレスもサポートしています:

システム IP アドレス X1	2 つの CPU の PROFINET X1 インターフェースに対するシステム IP アドレス
システム IP アドレス X2	2 つの CPU の PROFINET X2 インターフェースに対するシステム IP アドレス

他のデバイス(例、HMI デバイス、CPU、PG/PC)との通信に、システム IP アドレスを使用します。

デバイスは、常に、システム IP アドレスを介して冗長化システムのプライマリ CPU と通信します。これは、通信ピアがプライマリ CPU の障害後に、システム状態 RUN-Solo で新しいプライマリ CPU (以前のバックアップ CPU)と通信できることを確証します。

システム IP アドレスは、STEP 7 で有効化されます。

仮想 MAC アドレス

各システム IP アドレスには、仮想 MAC アドレスが含まれます。

2 つの PROFINET インタフェースの仮想 MAC アドレスは、互いに、および CPU の MAC アドレスとは異なる必要があります。

システム IP アドレスおよび仮想 MAC アドレスの設定に関する情報:

- 「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>): 「管理」 > 「設定手順」 セクション

システム IP アドレス/デバイス IP アドレスの利点

システム IP アドレスには、デバイス IP アドレスを上回る、次の利点があります。

- 通信ピアは、特にプライマリ CPU と通信します。
- システム IP アドレスを介した冗長化システム S7-1500R/H の通信は、プライマリ CPU が故障した場合でも、機能を継続します。

デバイス IP アドレスを使用したソフトウェア冗長性には、次の利点があります。

- 冗長性のある物理的接続(ネットワークケーブル)

IP アドレスに関する詳細情報:

- 「S7-1500R/H リダンダントシステム」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>): 「システム概要」 > 「システムとデバイス IP アドレス」 セクション
- 「STEP 7 V15 を使用する PROFINET」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/49948856>)

HMI デバイス

HMI デバイスは、プロセスのビジュアル化および機械レベル制御に対して使用されます。

冗長化システム S7-1500R/H 用に、標準 S7-1500 システム用と同じ HMI デバイスを使用します。

1 つまたは複数の HMI デバイスは、HMI 通信を介して CPU とデータをやり取りします(例、HMI Basic/Comfort/Mobile Panel)。

HMI デバイスの設定

HMI デバイスの冗長化システムへの接続は、対応するアプリケーションによって異なります。

STEP 7 で次の通信オプションを設定できます。

- HMI デバイスは、システム IP アドレスを介して冗長化システムと通信します。
- HMI デバイスは、デバイスの IP アドレスを介して、対応する R/H CPU と通信します(例えば、診断目的用)。

HMI デバイスの使用に関する追加情報:

- 「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>): 「HMI デバイスの使用」 セクション
- 「SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200AL、ET 200pro 通信」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/59192925>)

すべての使用可能な HMI デバイスの概要は、Industry Mall を参照してください。

設定

WinCC タグ管理では、[SIMATIC S7-1200, S7-1500 Channel]で、S7-1500 R/H コントローラへの単一接続を設定します。

CPU 冗長性

システム IP アドレスは、プライマリ CPU への接続を確立するために使用されます。

STEP 7 および TIA Portal で CPU 冗長性を完全に設定します。

これには以下のステップが含まれます。

1. プロジェクトおよび R/H CPU を作成します。
2. IP アドレスを割り付けます(デバイス IP アドレス)。
CPU の冗長化 ID は、STEP 7 のプロジェクトツリーで作成および割り付けられます。
ツリーの上側の CPU には、常に、冗長性 ID 1 があります。
3. システム IP アドレスを割り付けます(任意)。
4. スキャンサイクルモニタ時間を設定するか、デフォルト値を使用します。
5. IO デバイスを作成します。
6. IO デバイスを 2 つの冗長化 CPU を接続します。
7. S7-1500R/H 冗長化システムの CPU の MRP 役割を想定します。
STEP 7 は、「マネージャ(自動)」の MRP 役割を 2 つの CPU の PROFINET X1 インターフェースに自動的に割り付けます。
8. STEP 7 の他のリングノードに対して MRP 役割を定義します。
[デバイス]テーブルで、MRP 「クライアント」役割を、リングの他のすべてのノードに割り付けます。
9. STEP 7 プロジェクト外のノードに対してパラメータを割り付けます。
STEP 7 内にはないリングのノードに対して、MRP 「クライアント」役割を設定します。

設定およびコミショニングに関する追加情報:

- 「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>)

[ソフトウェアの冗長性]ファンクション

デバイス IP アドレスを使用したソフトウェア冗長性を使用する場合は、WinCC の冗長性コントローラに対してスクリプトも設定する必要があります。

詳細については、「S7-1500R/H のソフトウェアの冗長化 (ページ 540)」を参照してください。

下記も参照

冗長システム S7-1500R/H (ページ 534)

S7-1500R/H のソフトウェアの冗長化 (ページ 540)

「冗長化システム S7-1500R/H」 マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/109754833>)

「STEP 7 V15 を使用する PROFINET」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/49948856>)

「SIMATIC S7-1500、ET 200MP、ET 200SP、ET 200AL、ET 200pro 通信」 ファンクションマニュアル (<https://support.industry.siemens.com/cs/ww/en/view/59192925>)

6.12.3.11 S7-1500R/H のソフトウェアの冗長化

[ソフトウェアの冗長化]ファンクションを使用して、CPU の物理接続を使用して CPU の冗長性を操作します。冗長性設定に応じて、1 つまたは複数のネットワークカードまたは Siemens CP が必要です。

STEP 7 および TIA Portal で CPU 冗長性を完全に設定します。詳細情報は、「冗長化制御システムの設定 (ページ 536)」を参照してください。

ソフトウェア冗長性の原則**アドレス指定**

システム IP アドレスの代わりに、2 つの冗長化コントローラのデバイス IP アドレスを使用します。

システムタグ

冗長性制御およびモニタリングのために、タグ管理で必要なシステムタグを作成します。各通信接続に対して、対応する接続名を含むシステムタグを作成します。

- @<接続名>@<ソフトウェア冗長化用のシステムタグ>

システムタグのアドレスは、PLC の空きエリアを指定する必要があります。

ソフトウェアの冗長化用スクリプト

ソフトウェア冗長化をモニタリングするには、たとえば ANSI-C または VBScript でスクリプトを作成します。スクリプトは、周期的に呼び出されます(例、10 秒ごと)。

このスクリプトでは、プライマリ CPU とバックアップ CPU 間での切り換えを定義します。

ランタイムでの周期アクションとしてスクリプトを実行するには、コンピュータのスタートアップリストで[グローバルスクリプトランタイム]アプリケーションを有効にします。

シナリオ例

接続が中断された場合、WinCC は使用されたシステムタグを読み取りまたは設定します。

開始条件

- WinCC プロジェクトがランタイムであること。
- [PLC1]のプライマリ CPU への接続がアクティブであること。
- システムタグ「@<PLC1>@AlternativeAddress」にバックアップ CPU の有効なアドレスが含まれていること。例:
 - AccessPoint=abc;IPAddress=111.111.111.111;

接続エラー

1. 接続の損失は、[@<PLC1>@ConnectionState]タグによって決定されます。
2. 接続が無効化されます。
 - @<PLC1>@ForceConnectionState = 0
3. 接続パラメータが変更されます。
 - @<PLC1>@UseAlternativeAddress = 1
4. 接続が再確立されます。
 - @<PLC1>@ForceConnectionState = 1
5. WinCC は、バックアップ CPU への代替接続を確立します。

診断

次のシステムタグを使用して、2 つのコントローラの操作モードを評価します。

- @<PLC1>@OpStateConfiguredAddress
- @<PLC1>@OpStateAlternativeAddress

6.12 SIMATIC S7-1200, S7-1500 Channel

ソフトウェアの冗長化用システムタグ

S7-1500R/H コントローラへの接続用に、WinCC タグ管理で次のシステムタグを作成します。

タグ	用途	値	説明
@<...>@ForceConnection State	冗長接続の確立/終了	1 / 0	ランタイムが有効な場合の動作: <ul style="list-style-type: none"> 開始値 = 1:接続が確立されます。 開始値 = 0:接続は、無効なままになります。 データタイプ:符号なし 32 ビット値 アクセス:読み取り/書き込み
@<...>@ConnectionState	冗長接続ステータス	1 / 0	接続ステータスの問い合わせ: <ul style="list-style-type: none"> 1:接続の準備が整いました。 0:断線、または終了されました。 接続のタグはアーカイブされません。 データタイプ:符号なし 32 ビット値 アクセス:読み取り
@<...>@ConfiguredAddress	プライマリ CPU への接続	文字列	タグ管理で設定された接続のプロパティ データタイプ:テキストタグ 8 ビット文字セット。長さ=255 アクセス:読み取り
@<...>@AlternativeAddress	バックアップ CPU への接続	文字列	バックアップ CPU への代替接続のプロパティ タグは、初期値が必要です。例: <ul style="list-style-type: none"> AccessPoint=abc;IPAddress=111.111.111.11; 値は後で変更できます。 データタイプ:テキストタグ 8 ビット文字セット。長さ=255 アクセス:読み取り/書き込み
@<...>@CurrentAddress	現在のアドレス	文字列	現在使用されている接続のプロパティ データタイプ:テキストタグ 8 ビット文字セット。長さ=255 アクセス:読み取り

タグ	用途	値	説明
@<...>@UseAlternativeAddress	代替接続の使用	1 / 0	現在使用されている接続を決定します: <ul style="list-style-type: none"> 1:バックアップ CPU への代替接続 0:プライマリ CPU への接続 データタイプ:符号なし 32 ビット値 アクセス:読み取り/書き込み
@<...>@RedundantCPUs	冗長性の有効化	1 / 0	接続されたコントローラに冗長性があるかどうかを示します。 <ul style="list-style-type: none"> 1:はい 0:いいえ。CPU に冗長性はありません タグは、初期値が必要です。 値は後で変更できます。 データタイプ:符号なし 32 ビット値 アクセス:読み取り/書き込み
@<...>@OpStateConfiguredAddress	プライマリ CPU の動作モード	0 / 4 / 6 / 8 / 22	コントローラの動作モード: <ul style="list-style-type: none"> 0: not connected 4: STOP 6: STARTUP 8: RUN 22: SYNCUP データタイプ:符号なし 32 ビット値 アクセス:読み取り
@<...>@OpStateAlternativeAddress	バックアップ CPU の動作モード		

スクリプト例

```
'VBS381
Option Explicit
Function action

    Dim C1ConnectionState
    Set C1ConnectionState = HMIRuntime.Tags("@Connection1@ConnectionState")
    C1ConnectionState.Read

    Dim C1OpStateConfiguredAddress
    Set C1OpStateConfiguredAddress =
HMIRuntime.Tags("@Connection1@OpStateConfiguredAddress")
    C1OpStateConfiguredAddress.Read

    Dim C1OpStateAlternativeAddress
    Set C1OpStateAlternativeAddress =
HMIRuntime.Tags("@Connection1@OpStateAlternativeAddress")
    C1OpStateAlternativeAddress.Read

    Dim C1UseAlternativeAddress
    Set C1UseAlternativeAddress =
HMIRuntime.Tags("@Connection1@UseAlternativeAddress")
    C1UseAlternativeAddress.Read

    Dim C1ForceConnectionState
    Set C1ForceConnectionState =
HMIRuntime.Tags("@Connection1@ForceConnectionState")
    C1ForceConnectionState.Read

    'check if connection got disconnected
    If C1ConnectionState.Value = 0 Then
    'set force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write

    'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                C1UseAlternativeAddress.Value = 1
                C1UseAlternativeAddress.Write

            Case 1
                C1UseAlternativeAddress.Value = 0
                C1UseAlternativeAddress.Write

        End Select

    'reconnect
```



```
C1ForceConnectionState.Value = 1
C1ForceConnectionState.Write

'example for handling operating states:stop mode
'check if plc is in stop mode; stop mode does not deliver connectionstate
= 0
    Elseif (C1OpStateConfiguredAddress.Value = 4 And
C1UseAlternativeAddress.Value = 0) Or (C1OpStateAlternativeAddress.Value =
4 And C1UseAlternativeAddress.Value = 1) Then

        'set force connection state = 0
        C1ForceConnectionState.Value = 0
        C1ForceConnectionState.Write

        'switch between plc1 and plc2
        Select Case C1UseAlternativeAddress.Value

            Case 0
                C1UseAlternativeAddress.Value = 1
                C1UseAlternativeAddress.Write

            Case 1
                C1UseAlternativeAddress.Value = 0
                C1UseAlternativeAddress.Write

        End Select

        'reconnect
        C1ForceConnectionState.Value = 1
        C1ForceConnectionState.Write

    Else

    End If

Set C1ConnectionState = Nothing
Set C1OpStateConfiguredAddress = Nothing
Set C1OpStateAlternativeAddress = Nothing
Set C1UseAlternativeAddress = Nothing
Set C1ForceConnectionState = Nothing

End Function
```

下記も参照

冗長システム S7-1500R/H (ページ 534)

冗長化制御システムの設定 (ページ 536)

6.13 SIMATIC TI Ethernet 第4層

6.13.1 WinCC チャンネル"SIMATIC TI Ethernet 第4層"

はじめに

通信ドライバ「SIMATIC TI Ethernet Layer 4」は、Industrial Ethernet を使用して、WinCC ステーションと SIMATIC TI505 オートメーションシステムとの間のリンクを、取り扱います。通信は、ISO トランスポートプロトコルで処理されます。

このセクションでは、以下の方法について説明します。

- 「SIMATIC TI Ethernet Layer 4」チャンネルを使用してデータ転送を設定します。
- 接続とタグを設定します。

チャンネルユニット

最大2つの CP 1613 A2 を実行する2つのチャンネルユニットがあります。チャンネルユニットの機能は同じです。2つの CP 1613 A2 の論理デバイス名が異なるだけです。

論理デバイス名は、チャンネルユニットのシステムパラメータで変更できます。ここで、ISO トランスポートプロトコルのパラメータを設定することもできます。

以下のアプリケーション機能があります。

- SIMATIC Industrial Ethernet (例: CP 1613 A2)の通信モジュール用チャンネルユニット 505 Ethernet (CP 1413-1)。
- SIMATIC Industrial Ethernet (例: CP 1613 A2)の通信モジュール用チャンネルユニット 505 Ethernet(CP 1413-2)。

6.13.2 タグのデータタイプ

はじめに

論理接続に必要なタグを定義します。WinCC の観点から、以下のデータタイプにアクセスできます。

- 2 進タグ
- 符号なし 8 ビット値(VMS アドレス指定でのみサポート)
- 符号付き 8 ビット値(VMS アドレス指定でのみサポート)
- 符号なし 16 ビット値
- 符号付き 16 ビット値
- 符号なし 32 ビット値
- 符号付き 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- 未処理データタイプ

6.13.3 チャンネルの設定

6.13.3.1 チャンネル"SIMATIC TI Ethernet 第 4 層"の設定

はじめに

チャンネル"SIMATIC TI Ethernet 第 4 層"の設定には、以下のステップが必要です。

6.13.3.2 接続の設定方法

はじめに

SIMATIC TI505 オートメーションシステムでは、Industrial Ethernet 経由でプロセス接続が可能です。

オートメーションシステムで、通信モジュール CP 1434 TF を使用します。通信は、ISO トランスポートプロトコルで処理されます。

WinCC システムで、通信モジュール CP 1613 A2 を使用します。

6.13 SIMATIC TI Ethernet 第4層

通信は ISO トランスポートプロトコルによって行われるため、ローカルデータベースに論理接続を設定する必要はありません。

論理接続用に、WinCC は 1 つの接続をトランスポートレイヤに読み取り用(READ)に、1 つの接続を書き込み用(WRITE)に、確立します。両方の接続が確立された場合のみ、論理接続も確立済みと表示されます。

READ ファンクションのパラメータ

接続を設定するとき、WinCC の READ ファンクションにパラメータが定義されます。これらは SIMATIC TI で使用する要求と区別されます。

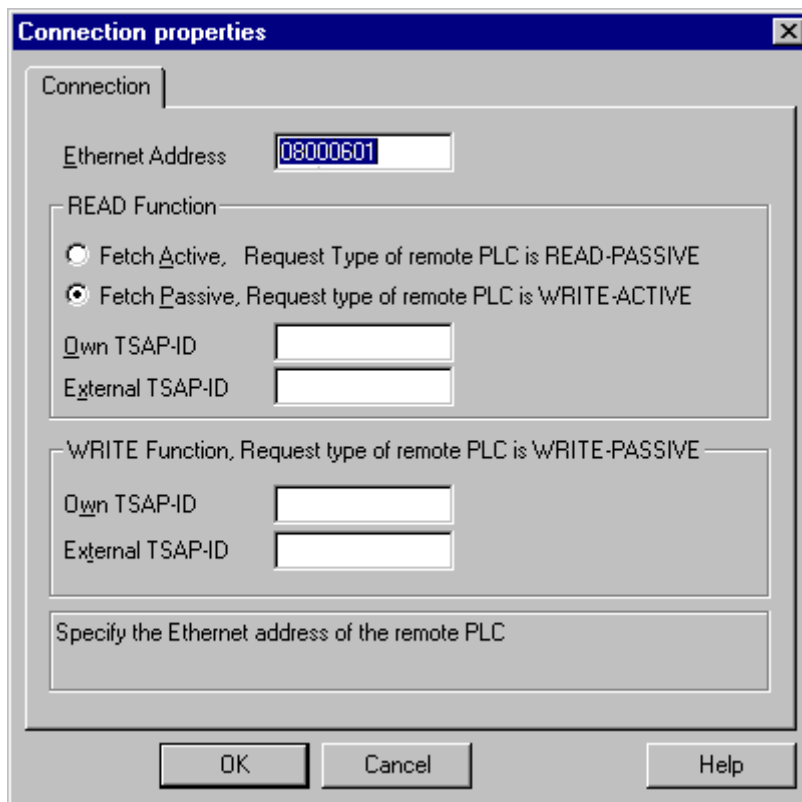
WinCC 側	SIMATIC TI 側
FETCH-アクティブ ([READ-アクティブ]要求)	READ-パッシブ ([READ-パッシブ]要求)
FETCH-パッシブ ([WRITE-パッシブ]要求)	WRITE-アクティブ ([WRITE-アクティブ]要求)

WRITE ファンクション用パラメータ"自分の TSAP-ID"および"外部 TSP-ID"

WinCC 側	SIMATIC TI 側
[WRITE アクティブ]要求	[WRITE パッシブ]要求

手順

1. 必要な接続を選択し、ショートカットメニューから「接続パラメータ」を選択します。
[接続プロパティ]ダイアログが開きます。



2. [Ethernet アドレス]フィールドで、Industrial Ethernet バスの SIMATIC TI のステーションアドレスを入力します。
3. ここで、WinCC システムの READ ファンクションのパラメータを定義します。
これらは SIMATIC TI で使用する要求と区別されます。
4. 次に、CP 1434 TF の設定中に[リモートパラメータ]エリアで"TSAP"として設定された割り当てられているフィールド[自分の TSAP-ID]に、値を入力します。
5. そこで、CP1434 TF の設定中に[ローカルパラメータ]で"TSAP"として設定された割り当てられているフィールド[外部 TSAP-ID]に、値を入力します。
6. パラメータ"自分の TSAP-ID"および"外部 TSP-ID"を、それぞれ WRITE ファンクション用に定義します。

6.13 SIMATIC TI Ethernet 第4層

6.13.3.3 タグの設定

タグの設定

はじめに

チャンネル"SIMATIC TI Ethernet 第4層"経由での WinCC と AS の間の接続用に、異なるデータタイプのタグを WinCC で作成できます。以下に、これらのデータタイプのタグの設定方法を説明します。


タグのアドレスの設定方法

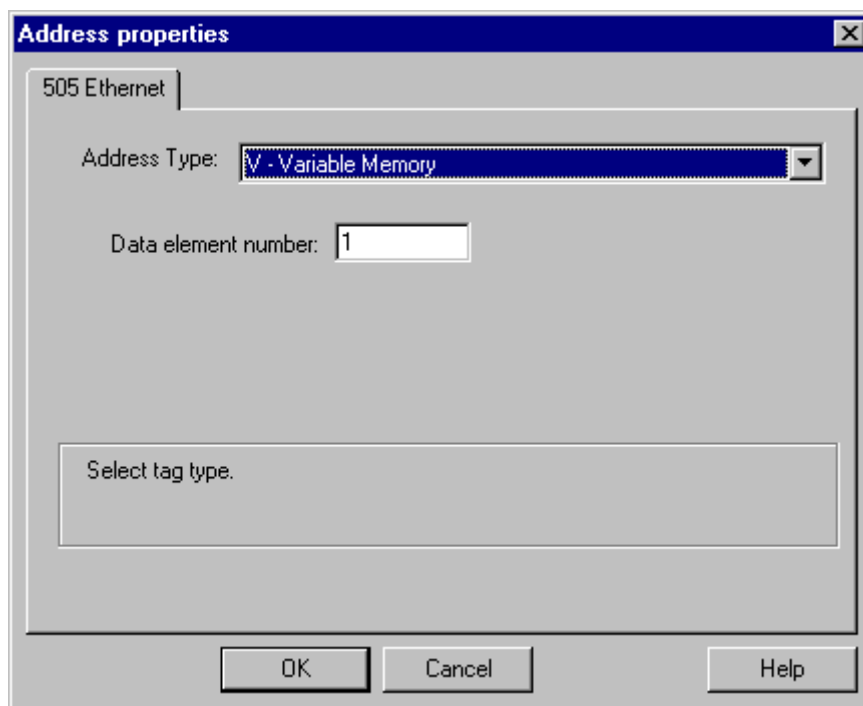
はじめに

SIMATIC TI505 のアドレス構造に準じてタグアドレスを入力します。

手順

1. タグの選択
2. [データタイプ]フィールドで、希望するデータタイプを選択します。
3. [2進]タグか[8ビット]タグの場合、[ビットタグ/バイトタグ]オプションが[プロパティ]エリアで利用できます。データを AS メモリに書き込む場合、対応するチェックボックス[ビットアクセス/バイトアクセス]をオンにします。

4. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



5. オートメーションシステムの[アドレスタイプ]フィールドで、タグの位置を選択します。選択したアドレスのタイプによっては、さらに多くの定義(例、[データ要素]フィールドのアドレスタイプに"V タグメモリ"など)をする必要があります。
6. [読み取り専用タグ]フィールドで、WinCC で書き込めないタグを指定できます。

注記

構造体タグはサポートされません。

アドレスタイプの説明は SIMATIC TI505 テクニカル文書に記載されています。

AS のメモリエリアへの書き込みアクセスは、チャンネル"TI Ethernet 第4層"でビットかバイトでのみ、実行できます。2進タグおよび[8ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

注記

読み取りデータエリアで AS によって行なわれた変更は、データエリアに書き戻された時点で上書きされます。

タグのタイプによって、AS のメモリに、ビットまたはバイトでアクセスできます。

ビットでアクセスするタグの設定方法


はじめに

AS のメモリエリアへの書き込みアクセスは、チャンネル"TI Ethernet 第4層"でビットかバイトでのみ、実行できます。2進タグおよび[8ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

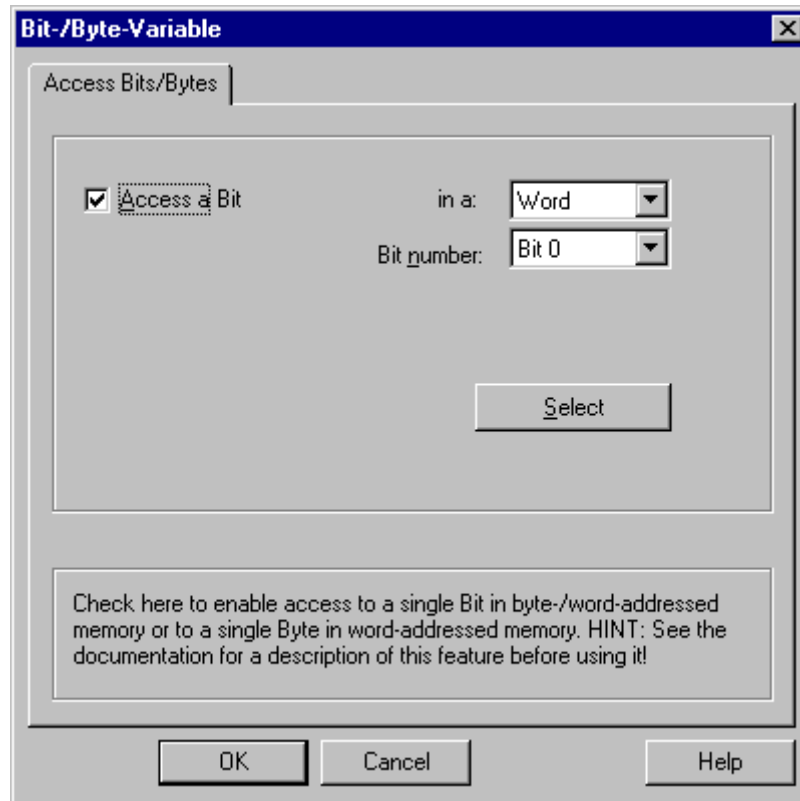
注記

読み取りデータエリアで AS によって行なわれた変更は、データエリアに書き戻された時点で上書きされます。

手順

1. タグを選択します。
2. [データタイプ]フィールドで、データタイプとして[2進タグ]を設定します。
3. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。

4. [ビットのアクセス]チェックボックスを選択して、ビットアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [アドレスタイプ]選択フィールドの PLC メモリのアドレス指定タイプを選択します。
7. 以下のリストから変更する要素数を選択します。

注記

アドレスタイプの説明は SIMATIC TI505 テクニカル文書に記載されています。

バイト単位でアクセスするタグの設定方法


はじめに

AS のメモリエリアへの書き込みアクセスは、チャンネル"TI Ethernet 第4層"でビットかバイトでのみ、実行できます。2進タグおよび[8ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

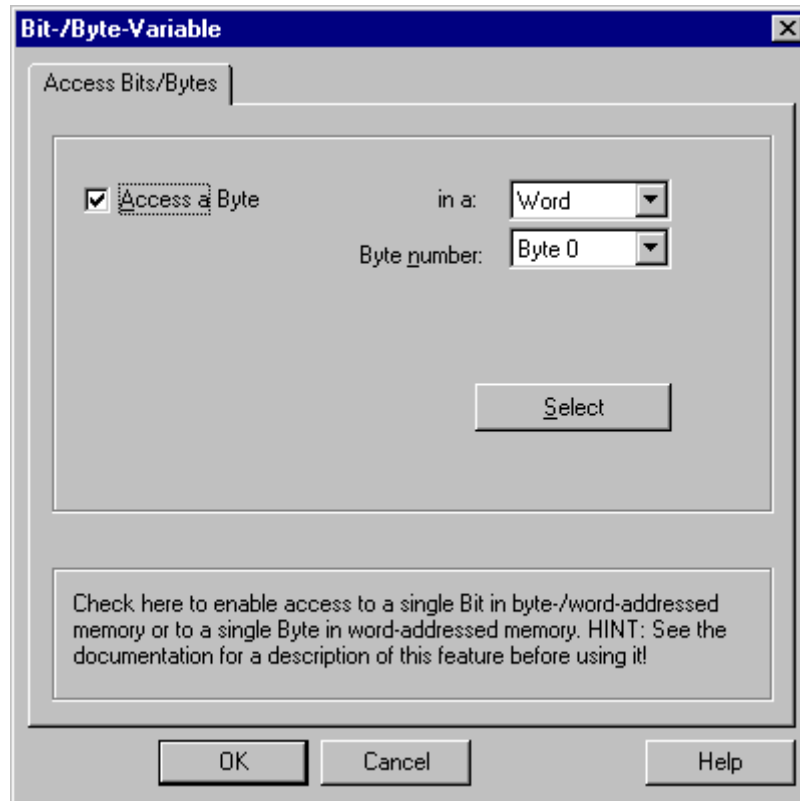
注記

読み取りデータエリアで AS によって行なわれた変更は、データエリアに書き戻された時点で上書きされます。

手順

1. タグを選択します。
2. [データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]または[符号付き 8 ビット値]に設定します。
3. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。

4. [バイトのアクセス]チェックボックスを選択して、バイトアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [アドレスタイプ]選択フィールドの PLC メモリのアドレス指定タイプを選択します。
7. 以下のリストから変更する要素数を選択します。

注記

アドレスタイプの説明は SIMATIC TI505 テクニカル文書に記載されています。

6.13.3.4 システムパラメータ

チャンネルユニットのシステムパラメータ

はじめに

WinCC の標準設定とは異なる設定が必要な場合は、チャンネルユニットの[システムパラメータ]ダイアログを使用して、必要な変更をすべて実行できます。

6.13 SIMATIC TI Ethernet 第4 層

以下の個別ポイントを変更できます。

- デバイス名
- トランスポートパラメータ

デバイス名

WinCC とオートメーションシステム間の通信は、論理デバイス名を介して実行されます。これらの名前は通信モジュールのインストール中に割り当てられ、ユニットに固有です。デバイス名は論理デバイス名を示します。論理デバイス名は、デフォルトとして"`CP_H1_1:/SCP`"と初期設定されます。

トランスポートパラメータ

PDU サイズ、設定試行などチャンネルユニットの固有設定は、トランスポートパラメータで実行します。

注記

システムパラメータは、AS の全ての CP に適用されます。

デバイス名の変更方法

はじめに

SIMATIC TI505 オートメーションシステムでは、Industrial Ethernet 経由でプロセス接続が可能です。

オートメーションシステムで、通信モジュール CP 1434 TF を使用します。通信は、ISO トランスポートプロトコルで処理されます。

WinCC システムで、通信モジュール CP 1613 A2 を使用します。

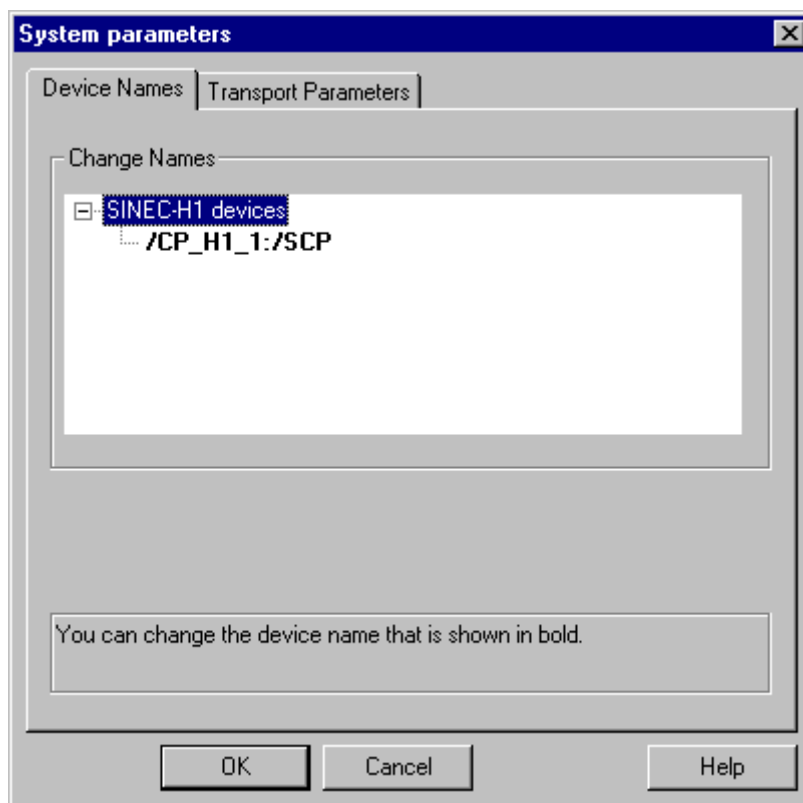
通信は ISO トランスポートプロトコルによって行われるため、ローカルデータベースに論理接続を設定する必要はありません。

必要条件

- チャンネル「SIMATIC TI Ethernet Layer 4」がこのプロジェクトで統合されていることが必要です。

手順

1. コンテキストメニューからチャンネルユニットを選択して、[システムパラメータ]ダイアログウィンドウを開きます。
2. [デバイス名]タブを選択します。



3. ここで、太字で表示されている論理デバイス名をマウスを使って選択し、[名前]フィールドでマウスクリックすることによって、変更できます。
ハードウェアドライバのインストール中に、論理デバイス名は、デフォルトとして「/CP_H1_1:/SCP」と初期設定されます。
他の望ましくない名前を定義した場合に限り、デバイス名を変更してもかまいません。

トランスポートパラメータの変更方法

はじめに

SIMATIC TI505 オートメーションシステムでは、Industrial Ethernet 経由でプロセス接続が可能です。

オートメーションシステムで、通信モジュール CP 1434 TF を使用します。通信は、ISO トランスポートプロトコルで処理されます。

WinCC システムで、通信モジュール CP 1613 A2 を使用します。

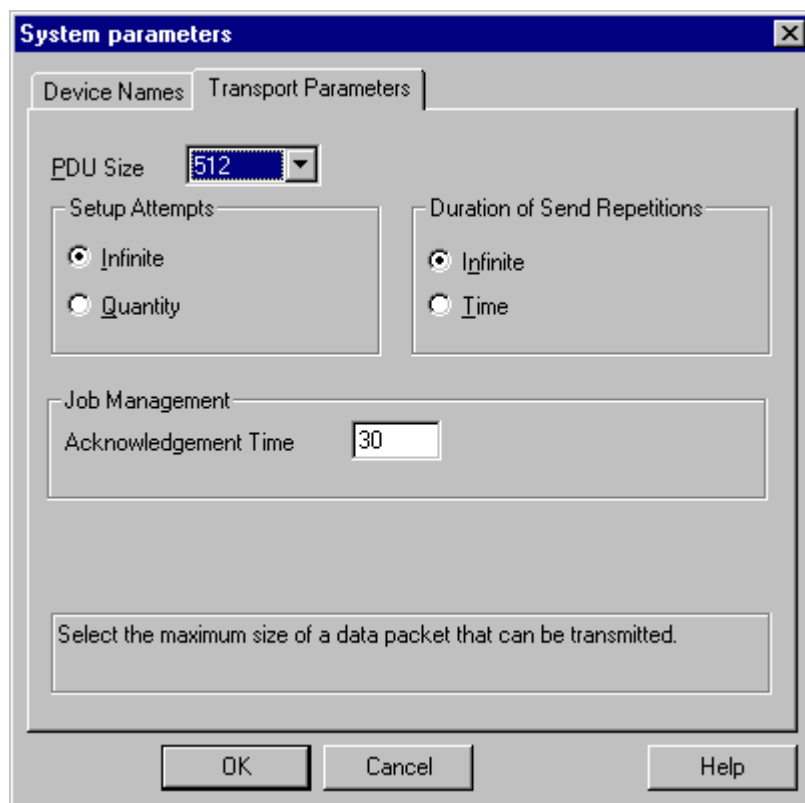
通信は ISO トランスポートプロトコルによって行われるため、ローカルデータベースに論理接続を設定する必要はありません。

必要条件

- チャンネル「SIMATIC TI Ethernet Layer 4」がこのプロジェクトで統合されていることが必要です。

手順

1. コンテキストメニューからチャンネルユニットを選択して、[システムパラメータ]ダイアログウィンドウを開きます。
2. [トランスポートパラメータ]タブを選択します。



3. [PDU サイズ]の値を、CP 1434 TF で設定された値に設定します。
4. [設定試行]フィールドで、接続確立の試行回数を定義します。
5. [送信繰り返し期間]領域で[無限]を選択します。
6. [確認時間]フィールドに値 30 を入力すると、通信パートナーがこの時間内に応答しなかった場合(例えば AS が「停止」状態)、最大 30 秒後にタグのステータスが通知されます。

6.14 SIMATIC TI Serial

6.14.1 WinCC チャンネル"SIMATIC TI シリアル"

はじめに

通信ドライバ"SIMATIC TI シリアル"は、WinCC ステーションと SIMATIC TI505 オートメーションデバイスの間のシリアルリンクを確立するために、使用されます。

この章の内容

- "SIMATIC TI シリアル"チャンネルによるデータ転送の設定方法。
- 接続およびタグの設定方法。

チャンネルユニット

通信ドライバには 1 つのチャンネルユニットがあり、シリアル接続用に COM ポートを制御します。

以下の機能を使用できます。

- TBP プロトコルまたは NITP プロトコルによるシリアル通信のための、チャンネルユニット"505 シリアルユニット#1"。

注記

1 つのチャンネルユニットを通して、2 つ以上の論理接続(異なる COM ポートで)を実行できます。

6.14.2 タグのデータタイプ

はじめに

論理接続に必要なタグを定義します。WinCC の観点から、以下のデータタイプにアクセスできます。

- 2 進タグ
- 符号なし 8 ビット値(VMS アドレス指定でのみサポート)
- 符号付き 8 ビット値(VMS アドレス指定でのみサポート)
- 符号なし 16 ビット値

6.14 SIMATIC TI Serial

- 符号付き 16 ビット値
- 符号なし 32 ビット値
- 符号付き 32 ビット値
- 浮動小数点数 32 ビット IEEE 754
- 未処理データタイプ

6.14.3 チャンネルの設定

6.14.3.1 "SIMATIC TI シリアル"チャンネルの設定

はじめに

チャンネル"SIMATIC TI シリアル"の設定には、以下のステップが必要です。

6.14.3.2 接続の設定方法

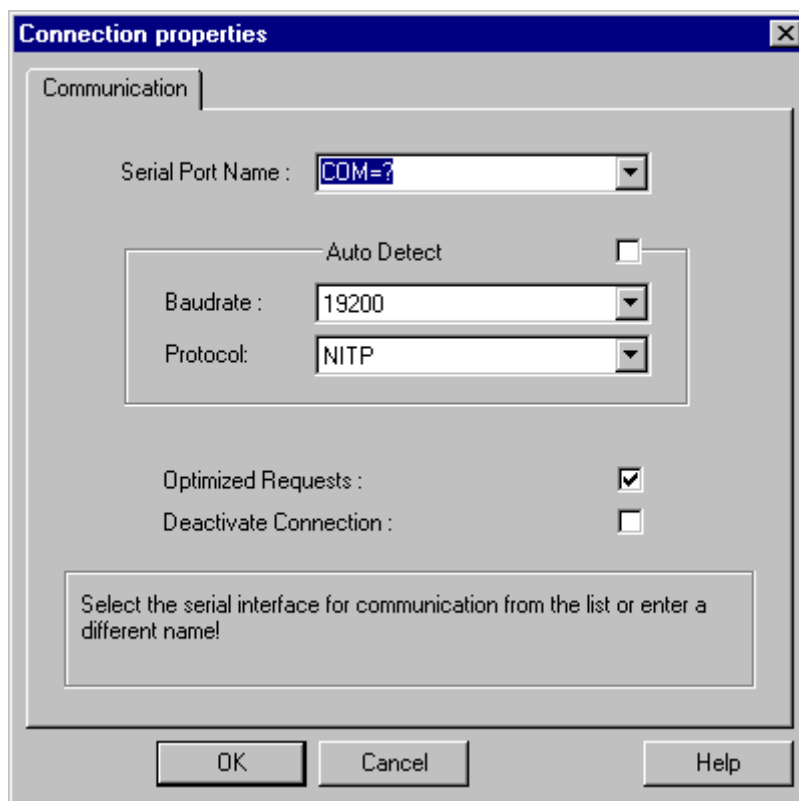
はじめに

シリアル接続を使用したプロセス接続は、SIMATIC TI505 オートメーションシステムで使用可能です。CPU モジュールのシリアルインターフェースは、オートメーションシステムで使用されます。

WinCC では、追加の通信モジュールは不要です。通信は、システムで利用できるデフォルトの COM ポートによって行なうことができます。

手順

1. 必要な接続を選択し、ショートカットメニューの[接続パラメータ]を選択します。
[接続プロパティ]ダイアログが開きます。



2. [シリアルポート]フィールドのシリアルリンクの通信インターフェース(例えば COM1、COM2 または設定済ポート)を選択します。
3. PLC が使用するデータ転送速度とプロトコルを、チャンネルユニットが自動的に検出する必要がある場合は、[自動的に検出]フィールドを選択します。
4. [ボーレート]フィールドおよび[プロトコル]フィールドで、使用するデータ転送速度とプロトコルを設定します。
5. [最適化された要求]フィールドを選択することによって、1 つの要求で複数のタグを転送するデータ転送を、最適化できます。
6. [接続の無効化]フィールドを選択すると、論理接続が無効になります。コミッシュョニング中に一時的に接続を無効にすることは、良い考えです。

6.14 SIMATIC TI Serial

6.14.3.3 タグの設定

タグの設定

はじめに

チャンネル"SIMATIC TI シリアル"経由での WinCC と PLC の間の接続では、WinCC で異なるデータタイプのタグを作成できます。これは以下のセクションで説明されています。


タグのアドレスの設定方法

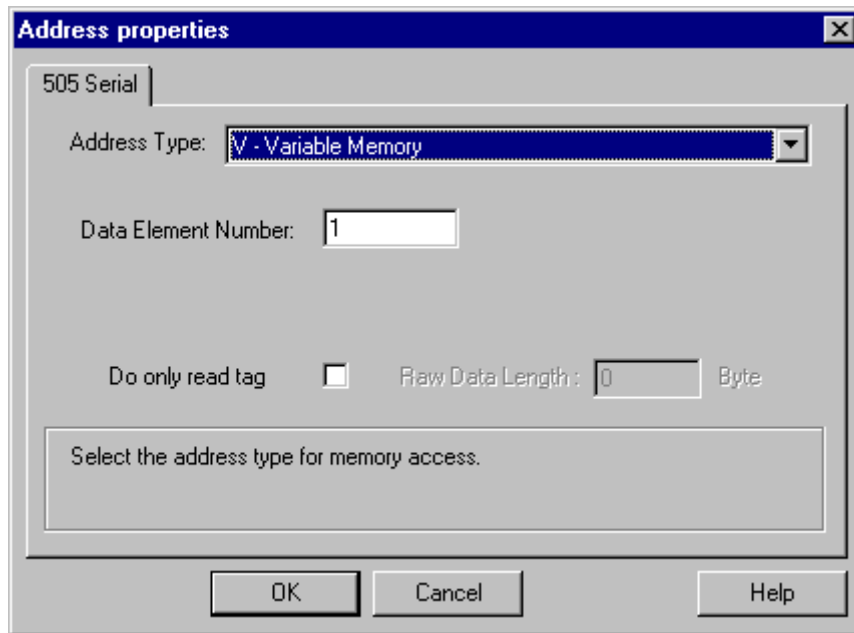
はじめに

SIMATIC TI505 のアドレス構造に準じてタグアドレスを入力します。

手順

1. タグを選択します。
2. [データタイプ]フィールドで、必要なデータタイプを設定します(符号付き 16 ビット値など)。
3. [2 進]タグか[8 ビット]タグの場合、[ビットタグ/バイトタグ]オプションが[プロパティ]エリアで利用できます。データを AS メモリに書き込む場合、対応するチェックボックス[ビットアクセス/バイトアクセス]をオンにします。

4. [アドレスプロパティ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



5. オートメーションシステムの[アドレスタイプ]フィールドで、タグの位置を選択します。選択したアドレスのタイプによっては、さらに多くの定義(例、[データ要素番号]フィールドのアドレスタイプに"V タグメモリ"など)をする必要があります。
6. [読み取り専用タグ]フィールドで、WinCC で書き込めないタグを指定できます。

注記

構造体タグは、アドレスエリア V、K、X、Y、C でサポートされます。
アドレスタイプの説明は SIMATIC TI505 テクニカル文書に記載されています。

AS のメモリエリアへの書き込みアクセスは、チャンネル"TI シリアル"でビットまたはバイトでのみ、実行できます。2 進タグおよび[8 ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

タグのタイプによって、AS のメモリに、ビットまたはバイトでアクセスできます。

ビットでアクセスするタグの設定方法


はじめに

AS のメモリエリアへの書き込みアクセスは、チャンネル"SIMATIC TI シリアル"でビットかバイトでのみ実行できます。2 進タグおよび[8 ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

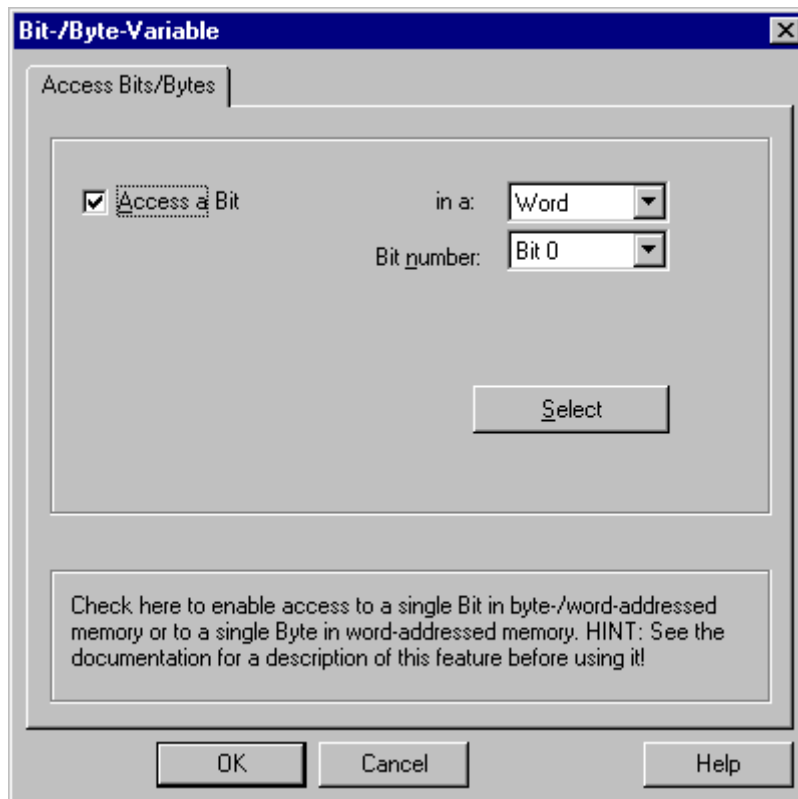
注記

読み取りデータエリアで AS によって行なわれた変更は、データエリアに書き戻された時点で上書きされます。

手順

1. タグを選択します。
2. [データタイプ]フィールドで、データタイプとして[2 進タグ]を設定します。
3. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。

4. [ビットのアクセス]チェックボックスを選択して、ビットアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します。
7. [選択]フィールドで変更するビット数を選択します。

注記

S5 の場合、フラグ、入力、出力はバイト単位でアドレス指定し、データブロック (DB、DX) はワード単位でアドレス指定します。

バイト単位でアクセスするタグの設定方法


はじめに

AS のメモリエリアへの書き込みアクセスは、チャンネル"SIMATIC TI シリアル"でビットかバイトでのみ実行できます。2 進タグおよび[8 ビット]タグを使用する場合、[アドレスプロパティ]ダイアログに加えて[ビットタグ/バイトタグ]ダイアログが開き、個別のビットまたはバイトの書き込みアクセスを定義できます。この目的で、書き込み要求が出されるたびにアドレス指定されたメモリエリアが AS から読み取られ、対応するビットやバイトが修正されます。その後、このデータが AS のメモリに書き戻されます。

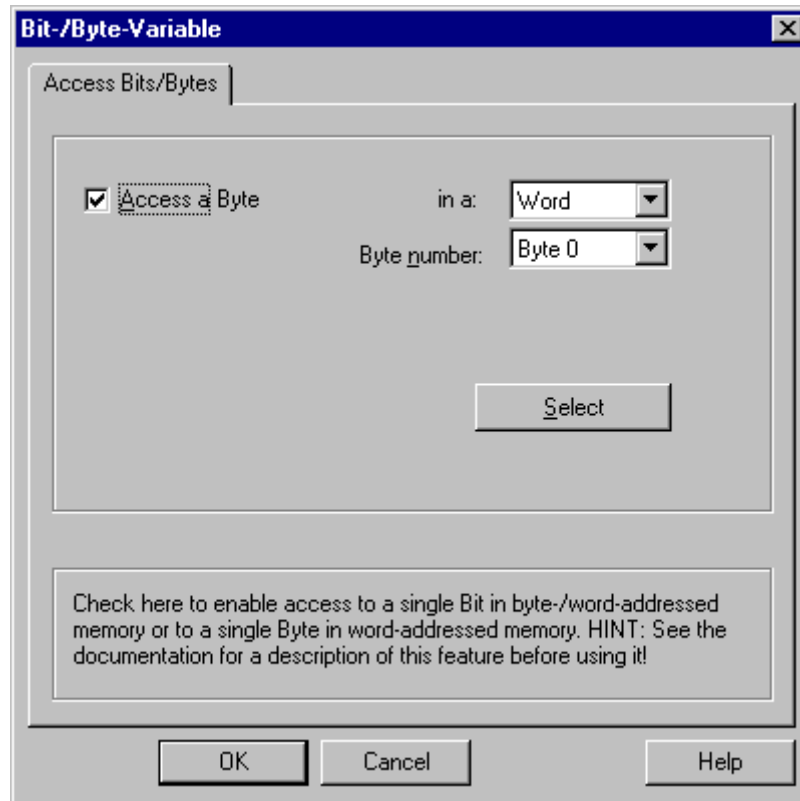
注記

読み取りデータエリアで AS によって行なわれた変更は、データエリアに書き戻された時点で上書きされます。

手順

1. タグを選択します。
2. [データタイプ]フィールドで、データタイプを[符号なし 8 ビット値]または[符号付き 8 ビット値]に設定します。
3. [ビットタグ/バイトタグ]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。

4. [バイトのアクセス]チェックボックスを選択して、バイトアドレス指定を定義します。



5. [選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。
6. [選択]フィールドで PLC メモリのアドレス指定タイプを選択します
7. [選択]フィールドで変更するバイト数を選択します。

6.15 SIMOTION

6.15.1 WinCC チャンネル"SIMOTION"

はじめに

"SIMOTION"チャンネルは、WinCC ステーションを SIMOTION オートメーションシステムに接続します。接続は TCP/IP プロトコルを使って、工業用 Ethernet 経由で確立されます。

SIMOTION は、動作制御アプリケーションと技術タスクに重点を置いた、オートメーションおよび駆動ソリューション用のシステムプラットフォームです。

SIMOTION モジュラーシステムは、様々なハードウェアプラットフォームに対応する、SIMOTION SCOUT エンジニアリングシステムおよび共通ランタイムシステムで構成されています。

SIMOTION SCOUT からプロジェクトをエクスポートして、SIMOTION SCOUT プロジェクトのデータにアクセスします。その後、エクスポートされたファイルから、Simotion Mapper を使って WinCC プロジェクトを作成します。

WinCC で後刻、SIMOTION オートメーションシステムの設定に対して、特定の変更（IP アドレスの変更など）を設定することができます。その他の変更は SIMOTION SCOUT で設定し、Simotion Mapper を使って再度エクスポートして、転送する必要があります。

注記

WinCC で"SIMOTION"チャンネルを設定するには、SIMOTION SCOUT エンジニアリングシステムおよび WinCC の設定に精通している必要があります。

6.15.2 サポートされているデータタイプの概要

はじめに

オートメーションシステムのデータフォーマットへのデータタイプとフォーマットの適用は、タグの設定で決定されます。

以下の表に、このチャンネルがサポートするデータタイプと、フォーマットに適応するアプリケーションを示します。

サポートされるデータタイプ

データタイプ	タイプ変換
2 進タグ	いいえ
符号付き 8 ビット値	はい
符号なし 8 ビット値	はい
符号付き 16 ビット値	はい
符号なし 16 ビット値	はい
符号付き 32 ビット値	はい
符号なし 32 ビット値	はい
浮動小数点数 32 ビット IEEE 754	はい
テキストタグ、8 ビットフォント	いいえ
未処理データタイプ	いいえ

6.15.3 チャンネルの設定

6.15.3.1 "SIMOTION"チャンネルの設定

はじめに

この章では、"SIMOTION"チャンネルの設定方法を説明します。

注記

"SIMOTION"チャンネルを設定するには、SIMOTION SCOUT エンジニアリングシステムおよび WinCC の設定に精通している必要があります。

"SIMOTION"チャンネルを設定するには、以下の手順に従います。

1. SIMOTION SCOUT プロジェクトを SIMOTION SCOUT からエクスポートします。
2. Simotion Mapper を使って WinCC プロジェクトを作成します。
3. WinCC プロジェクトを開きます。
4. システムパラメータを設定します。

チャンネルの診断、接続およびタグに関する詳細は、「診断'SIMOTION'チャンネル」の章を参照してください。

6.15 SIMOTION

6.15.3.2 SIMOTION SCOUT プロジェクトのエクスポート方法

はじめに

このセクションでは、タグとメッセージの定義を SIMOTION SCOUT からエクスポートする方法を説明します。

必要条件：

- SIMOTION SCOUT エンジニアリングシステムに精通していること。
- エクスポートする SIMOTION SCOUT プロジェクトへのアクセス権があること。

手順

1. SIMOTION SCOUT でエクスポートする SIMOTION SCOUT プロジェクトを開きます。
2. [ツール]で[OPC データのエクスポート]を選択します。
3. "SIMATIC NET V6.4"バージョン、希望するスコープ、およびエクスポート用に少なくとも「OPC アラーム/イベント」オプションを選択します。

注記

Simotion Mapper は、"SIMATIC NET V6.4"以外のエクスポートバージョンを処理しません。

4. エクスポート先のディレクトリを選択します。
5. 通信インターフェースを選択します。
プロジェクトがエクスポートされます。
6. 経路指定を使用している場合は、経路指定情報を入力します。

SIMOTION SCOUT プロジェクトがエクスポートされます。"OPC_Data.sti"および"OPC_AE.xml"ファイルが、エクスポート先のディレクトリに保存されます。

6.15.3.3 Simotion Mapper を使った WinCC プロジェクトの作成方法

はじめに

このセクションでは、Simotion Mapper を使って、エクスポートされた SIMOTION SCOUT プロジェクトから WinCC プロジェクトを作成する方法を説明します。

注記

古いバージョンの SIMOTION SCOUT プロジェクト用に WinCC プロジェクトが既に作成されている場合は、送信で SIMOTION パラメータのみを変更します。WinCC プロジェクトのこれ以外の全ての設定（アーカイブなど）は変更されません。

前提条件：

- SIMOTION SCOUT プロジェクトのエクスポートファイル"OPC_Data.sti"および"OPC_AE.xml"へのアクセス権があること。
- WinCC のインストールディレクトリへのアクセス権があること。

注記

SIMOTION SCOUT のプログラミング環境で、ウォッチテーブルを使って、エクスポートされるタグをフィルタ処理できます。ウォッチテーブルを使って、WinCC プロジェクト内のタグを少数に維持します。ウォッチテーブルの詳細については、SIMOTION SCOUT のオンラインヘルプを参照してください。

手順

1. WinCC のインストールディレクトリにある"SimotionMapper.exe"プログラムを起動します。
2. [開く]をクリックします。ファイル"OPC_Data.sti"および"OPC_AE.xml"があるディレクトリに移動します。
データが読み取られ、Simotion Mapper に表示されます。
3. Simotion Mapper Explorer で、WinCC プロジェクトに必要なグループとタグを選択します。
4. [新規 WinCC プロジェクトの作成]を選択します。
5. 「WinCC 接続名」を変更する場合は、表示された名前をクリックして、新しい名前を入力します。
6. 技術アラーム用の「最初の TA メッセージ数」を指定します。他の通信チャンネルのメッセージと衝突しない値を選択する必要があります。デフォルト値は 100 です。
Simotion Mapper は、1 つのメッセージと、Simotion の各接続用の指定した数で始まる合計 6 つのテンプレートメッセージを作成します。

6.15 SIMOTION

7. [マッピングの開始]をクリックします。WinCC プロジェクト用の転送先フォルダを選択します。
WinCC プロジェクトが作成されます。進捗バーが、手順の進捗状況を示します。

8. Simotion Mapper を閉じます。

WinCC プロジェクトが作成され、WinCC で開いて編集することができます。

注記

作成された WinCC プロジェクトを使用するために、WinCC の"SIMOTION"チャンネルのシステムパラメータを設定する必要があります。

6.15.3.4 Simotion Mapper を使った WinCC プロジェクトの変更方法

はじめに

このセクションでは、エクスポートされた SIMOTION SCOUT プロジェクトを、Simotion Mapper を使って既存の WinCC プロジェクトに追加する方法を説明します。たとえば、このようにして、同じ Simotion プロジェクトを 1 つの WinCC プロジェクトで複数回使用することができます。

注記

古いバージョンの SIMOTION SCOUT プロジェクト用に WinCC プロジェクトが既に作成されている場合は、送信で SIMOTION パラメータのみを変更します。WinCC プロジェクトのこれ以外の全ての設定（アーカイブなど）は変更されません。

前提条件：

- SIMOTION SCOUT プロジェクトのエクスポートファイル"OPC_Data.sti"および"OPC_AE.xml"へのアクセス権があること。
- WinCC のインストールディレクトリへのアクセス権があること。

注記

SIMOTION SCOUT のプログラミング環境で、ウォッチテーブルを使って、エクスポートされるタグをフィルタ処理できます。ウォッチテーブルを使って、WinCC プロジェクト内のタグの数を制限します。ウォッチテーブルの詳細については、SIMOTION SCOUT のオンラインヘルプを参照してください。

手順

1. 編集する WinCC プロジェクトを開きます。
2. WinCC のインストールディレクトリにある "SimotionMapper.exe" プログラムを起動します。
3. [開く] をクリックします。ファイル "OPC_Data.sti" および "OPC_AE.xml" があるディレクトリに移動します。
データが読み取られ、Simotion Mapper に表示されます。
4. Simotion Mapper Explorer で、WinCC プロジェクトに必要なグループとタグを選択します。
5. [開いたプロジェクトに追加] を選択します。
6. 既に作成済みのグループまたはタグを再度追加する場合は、表示されている名前をクリックして、「WinCC 接続名」を変更する必要があります。
7. 接続用のメッセージ、グループ、タグを転送しない場合、「WinCC 接続名」の選択を解除します。
8. タグを上書きするかどうかを指定します。
9. 技術アラーム用の「最初の TA メッセージ数」を指定します。他の通信チャンネルのメッセージと衝突しない値を選択する必要があります。デフォルト値は 100 です。
Simotion Mapper は、1 つのメッセージと、Simotion の各接続用の指定した数で始まる合計 6 つのテンプレートメッセージを作成します。

注記

既にマップされている「最初の TA メッセージ数」は変更しないでください。変更すると、メッセージが予測できない動作をすることがあります。

10. [マッピングの開始] をクリックします。WinCC プロジェクト用の転送先フォルダを選択します。
SIMOTION SCOUT プロジェクトが、開いている WinCC プロジェクトに追加されます。進捗バーが、手順の進捗状況を示します。
11. Simotion Mapper を閉じます。

WinCC プロジェクトが SIMOTION SCOUT プロジェクトによって拡張され、変更した設定と共に保存されました。

6.15.3.5 接続パラメータの変更方法

はじめに

このセクションでは、SIMOTION ネットワークアドレスの接続パラメータの変更方法を学習していただきます。

注記

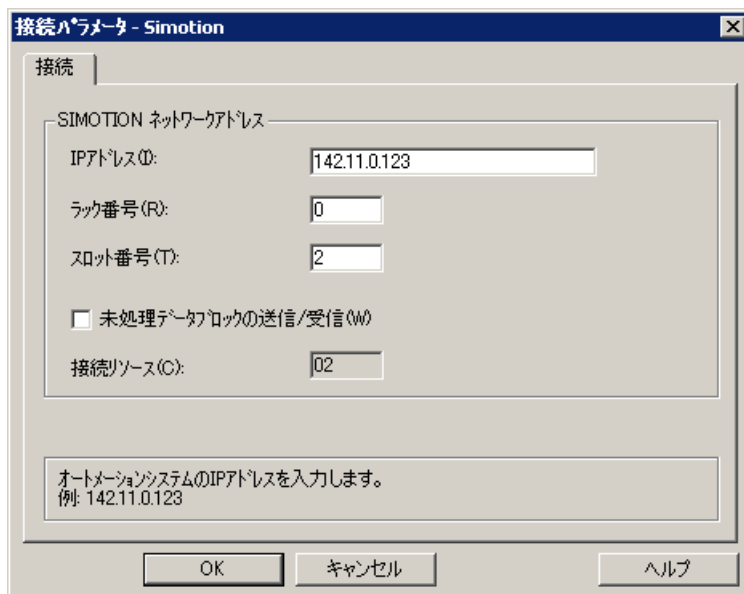
ここに示す接続パラメータのみを変更します。"SIMOTION"チャンネル用に新しい接続を作成しないでください。接続が正しく設定されていないと、PLC で制御エラーが発生することがあります。セクション「"SIMOTION"チャンネルの設定 (ページ 569)」の説明に従い、新規接続を構成します。

必要条件

- SIMOTION 通信ドライバが WinCC プロジェクトに統合されていること。
- 接続は"SIMOTION"チャンネルユニットで作成する必要があります。

手順

1. [タグ管理]エディタの"SIMOTION"通信ドライバのディレクトリ構造を開きます。
2. [SIMOTION]チャンネルユニット接続のショートカットメニューから[接続パラメータ]エントリを選択します。
[接続パラメータ - SIMOTION]ダイアログが開きます。



3. 各フィールドの SIMOTION ネットワークアドレスの接続パラメータを変更します。
4. [OK]をクリックして、それぞれの開いているダイアログボックスを閉じます。

6.15.3.6 タグアドレスの変更方法

はじめに

このセクションでは、"SIMOTION"チャンネルでのタグアドレスの変更方法を説明します。


注記

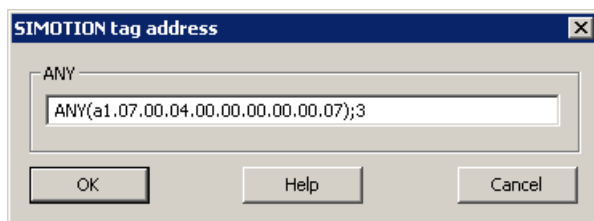
"SIMOTION"チャンネルのタグアドレスを変更するには、あらゆるポインタの使用方法を非常によく理解している必要があります。タグアドレスが正しく入力されないと、通信接続が確立されないことがあります。

必要条件

- "SIMOTION"チャンネルが WinCC プロジェクトに統合されていること。
- "SIMOTION"チャンネルユニットでタグとの接続が作成されていること。

手順

1. [SIMOTION タグアドレス]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックしてください。



2. タグアドレスを変更します。

6.15.3.7 システムパラメータ設定

チャンネルユニットのシステムパラメータ

はじめに

WinCC のデフォルト設定とは異なる設定が必要な場合は、[システムパラメータ]ダイアログボックスでこれを設定します。

以下のシステムパラメータを変更できます。

- 論理デバイス名
- AS のサイクリック読み取りサービスを使用するチャンネル

論理デバイス名

WinCC および PLC は、通信プロセッサが PLC にインストールされたときに割り当てられる論理デバイス名を使って通信します。

AS のサイクリック読み取りサービスを使用するチャンネル

PLC サイクリック読み取りサービスは、周期的に読み取られるタグを個々の要求にグループ化して、それらを PLC に転送します。PLC は、要求を受け取った時に初めて要求されたデータを送信し、以後はサイクル時間が経過するたびに再びデータを送信します。

サイクリック読み取りサービスが有効な場合、変更駆動型の転送機能を使うことができます。PLC が変更駆動型の転送をサポートしている場合、データは、値が変更されたときのみ転送されます。

システムパラメータのコンフィグレーション方法

はじめに

このセクションでは、"SIMOTION"チャンネルのシステムパラメータの設定方法を説明します。

[システムパラメータ]ダイアログは、以下の2つのタブで構成されます。

- [SIMOTION]タブ
- [ユニット]タブ

注記

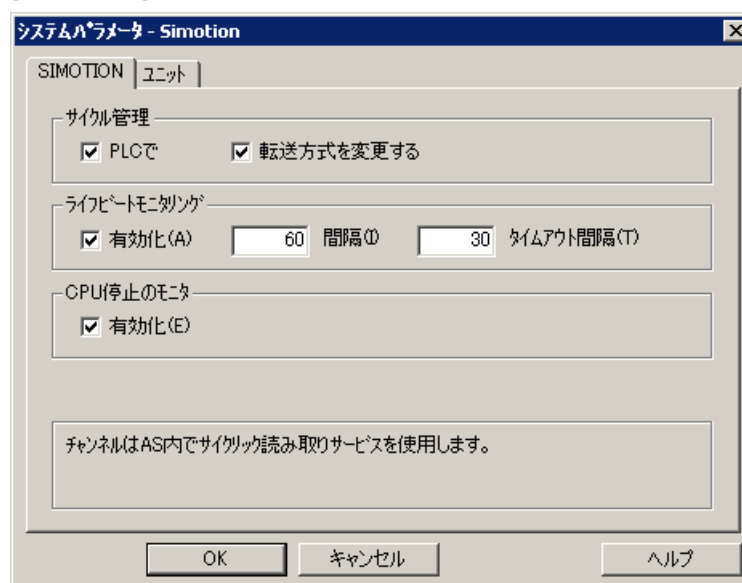
プロジェクトが他のコンピュータにコピーされるとき、[ユニット]タブの設定は保持されます。一方、[SIMOTION]タブの設定は削除されます。

前提条件

- "SIMOTION"チャンネルが WinCC プロジェクトに統合されていること。

手順

1. 変数管理で"SIMOTION"チャンネルを選択します。"Simotion"チャンネルユニットのショートカットメニューの、[システムパラメータ]ダイアログボックスを開きます。
2. [SIMOTION]タブを選択します。



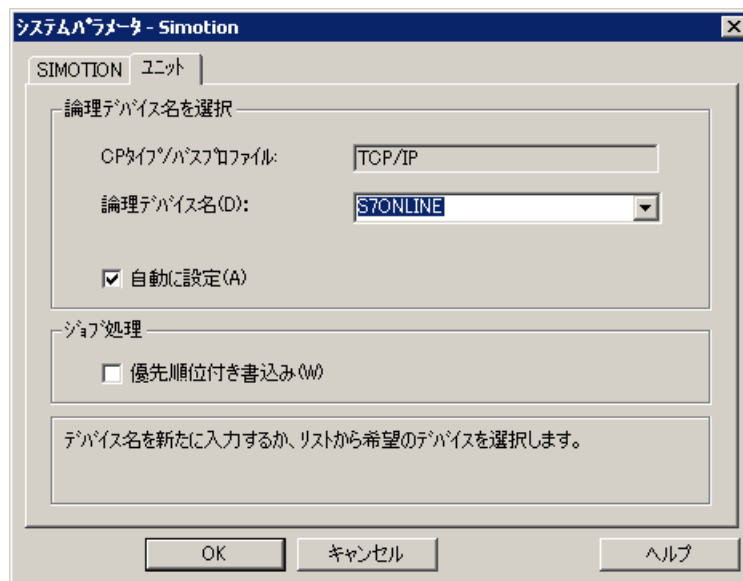
3. タグのサイクリック読み取りと変更駆動型の転送を有効にするには、[PLC で]および[変更駆動型の転送]を選択します。

注記

[サイクル管理]、[ライフビートモニタリング]および[停止モニタ]機能は、統合 SINAMICS サーボ制御によってサポートされていません。そのため"SIMOTION"チャンネルは、SINAMICS サーボ制御への接続に対応する設定を無視します。このチャンネルは、接続を確立するときに、AS がそれぞれの機能をサポートしているかどうかを判断します。

6.15 SIMOTION

4. 必要に応じて、[ライフビートモニタリング]機能を選択します。
ライフビートモニタリングメッセージを送信する間隔を、秒単位で決定します。
ライフビートモニタリングメッセージへの応答を監視するモニタ時間を、秒単位で決定します。
5. SIMOTION CPU が停止状態のときに、WinCC が通信の障害の信号を送るようにする場合は、[CPU 停止モニタ]を有効にします。
6. [ユニット]タブを選択します。
「論理デバイス名」のデフォルトとして、"S7ONLINE"が表示されます。使用している通信プロセッサのインストール時に異なる名前が選択されている場合は、デバイス名を変更する必要があります。



7. ランタイムの開始時にデバイス名を自動的に設定するには、[自動的に設定]を選択します。
8. 処理中の書き込みジョブに読み取りジョブより高い優先度を与えるには、[書き込み優先]を選択します。
9. [OK]をクリックしてダイアログを閉じます。

注記

WinCC を再起動しないと設定の変更は有効になりません。

論理デバイス名の変更方法

はじめに

WinCC および SIMOTION は、論理デバイス名を使用して通信します。これらの論理デバイス名は、通信プロセッサがインストールされるときに割り当てられています。

前提条件

- "SIMOTION"チャンネルがプロジェクトに統合されていること。
- "SIMOTION"チャンネルユニットで接続が作成されていること。

手順

1. タグ管理で、SIMOTION チャンネルを選択します。
2. ショートカットメニューで[システムパラメータ]ダイアログボックスを開きます。
3. [ユニット]タブを選択します。
4. [論理デバイス名]フィールドに、デバイス名を入力します。リストからエントリを選択するか、新しい名前を入力することができます。
デバイス名は、「PG/PC インターフェースの設定」ツールによって決定されます。ツールはシステム制御で呼び出します。そのツールがインストールされていない場合、現在設定されているデバイス名のみが表示されます。
異なる論理デバイス名を指定した場合は、メッセージが表示されます。
ターゲットステーションで使用されている通信プロセッサが設定システムにインストールされていない場合、名前のみを入力します。
5. [OK]をクリックしてダイアログを閉じます。

注記

WinCC を再起動しないと設定の変更は有効になりません。

6.15.4 診断"SIMOTION"チャンネル

6.15.4.1 "SIMOTION"チャンネルの診断の可能性

"SIMOTION"チャンネルとそのタグの診断とエラー検出には、以下の可能性があります。

通信プロセッサの設定のチェック

アクセスポイントのチェック後、通信プロセッサについても「PG/PC インターフェースの設定」アプリケーションでテストできます。通信プロセッサは、SIMATIC NET で同じ方法でチェックできます。

接続およびタグの設定のチェック

システムおよび接続パラメータの設定にエラーがあります。間違ったタグのアドレス指定が、間違ったタグ値の原因である場合もあります。

「チャンネル診断」によるチャンネルの診断

「チャンネル診断」を使って、ランタイム時のチャンネルおよび接続のステータスを問い合わせることができます。エラーは「エラーコード」で表示されます。

チャンネルのタグの診断

タグ管理で、ランタイム時に、現在の値、現在の品質コードおよびタグの最終変更時間を問い合わせることができます。

6.15.4.2 ログファイルのエントリの説明

はじめに

このチャンネルは、ステータスの重要な変更とエラーをログファイルに入力します。エントリは通信障害の分析をサポートします。

ファイルのすべてのエントリには、日付とタイムスタンプが、以下のフラグ名と説明と共に含まれています。

ログブックエントリの例：

```
2009-10-28 12:10:11,467 INFO Log starting ...
2009-10-28 12:10:11,483 INFO | LogFileName : D:\SIEMENS\WINCC\Diagnosis
\Simotion_01.LOG
2009-10-28 12:10:11,483 INFO | LogFileCount : 3
2009-10-28 12:10:11,483 INFO | LogFileSize : 1400000
2009-10-28 12:10:11,483 INFO | TraceFlags : fa000000
2009-10-28 12:10:11,498 INFO SIMOTION channel DLL started!
2009-10-28 12:10:11,498 INFO SIMOTION channel with own cycle creation!
2009-10-28 12:10:11,967 INFO Connection "D445": StartRegisterEvVariable for
dwVariableCount = 89
2009-10-28 12:10:11,967 INFO Connection "D445": RegisterEvVariable for Variable
"@D445@Checksum"!
...
```

2009-10-28 12:10:11,983 INFO Connection "D445": EndRegisterEvVariable

2009-10-28 12:10:12,436 INFO S7DOS release: @(#)TIS-Block Library DLL Version R8.0.0.0-REL-BASIS

2009-10-28 12:10:12,436 INFO S7DOS version: V8.0 / 0

2009-10-28 12:10:12,436 INFO SIMOTION version: V6.0 / Sep 15 2009 / 08:06:43

2009-10-28 12:10:12,436 INFO SIMOTION channel unit "Simotion" activated!

2009-10-28 12:10:12,451 ERROR Cannot connect to "SINAMICS_Integrated":Errorcode 0xFFDF 42C2!

2009-10-28 12:10:12,451 ERROR Cannot connect to "D445":Errorcode 0xFFDF 42C2!

"INFO"フラグの最も重要なエントリの記述

メッセージテキスト	意味
LogFileName : C:\Siemens\WinCC\Diagnose\ "channel_name".LOG	パスを付けたログファイル名
LogFileCount : "n"	チャンネルのログファイル数
LogFileSize : "x"	個々のログファイルのサイズ(バイト単位)
TraceFlags : c4000000	トレースファンクションにより使用されるフラグを16進数で表示します。
SIMOTION チャンネル DLL が起動しました。	開始メッセージ
SIMOTION チャンネル DLL が終了しました。	終了メッセージ

"ERROR"フラグの最も重要なエントリの記述

メッセージテキスト	意味
<接続名>に接続できません： エラーコード 0x0000 7<xxx>!	通信エラー WinCC の起動直後に、SIMOTION への通信を確立できませんでした。 <接続名> = 接続の名前 <xxx> 1...fff S7DOS、下層または AS からのファンクション呼び出しの結果として、チャンネルはその他のエラーコードをすべて受け取りました。
Connectionerror <nnn> <接続名>: エラーコード 0x0000 7xxx!	通信エラー WinCC の起動後、SIMOTION への通信を確立できませんでした。接続が切断されました。 <nnn> = この接続の接続終了数 <接続名> = 接続の名前 <xxx> 1...fff S7DOS、下層または AS からのファンクション呼び出しの結果として、チャンネルはその他のエラーコードをすべて受け取りました。
チャンネル API エラー: errorstring	チャンネル API エラー チャンネルにより、エラー文字列'errorstring'が WinCC Explorer に渡されました。エラーの関連性に従って、エラー文字列が情報ボックスに表示されます。エラー文字列の詳細については、API エラーテキストを参照してください。
API エラーの最大数になりました - API ログが無効になりました	チャンネル API エラー エラーおよびファンクションにより、API にエラーが周期的に発生する場合があります。これらのエラーメッセージによりログファイルがいっぱいになるのを避けるために、API エラーに対して出力されるのは最大 32 のメッセージです。

メッセージテキスト	意味
ストレージデータを書き込めません! ストレージデータを読み取れません/デフォルトデータを使用します ストレージデータが違法または破壊されています/デフォルトデータを使用します! ストレージデータがありません/デフォルトデータを使用します!	一般のチャンネルエラーメッセージです。
"unitname"ユニット内のデバイス名が"old devicename" から "new devicename" に変更されました。	初期化メッセージ

注記

"SIMOTION"チャンネルのエラーコードは、"SIMATIC S7 Protocol Suite"チャンネルのエラーコードに対応しています。エラーコードの詳細については、"SIMATIC S7 Protocol Suite"チャンネルのヘルプの「接続障害のエラーコード」の章を参照してください。

さらに、一貫性チェックが不合格の場合、SIMOTION はエラーコード 0x000 7301 を報告します。エラーメッセージの原因は、AS によって入力されたチェックサムに対する SIMOTION SCOUT からエクスポートされたデータが、接続されているデバイスのチェックサムに一致しないためです。

6.16 システム情報

6.16.1 "システム情報"チャンネル

内容

"システム情報"チャンネルは、時間、日付、ディスク容量などのシステム情報を評価するのに使用され、タイマやカウンタなどのファンクションを提供します。

この章では、以下について説明します。

- チャンネル、接続、およびタグをコンフィグレーションする
- システム情報をプロセスピクチャへに表示する
- システム情報を使用してメッセージをトリガし表示する
- システム情報をグラフに表示する
- 複数サーバからのシステム情報をマルチユーザーシステムで表示する

6.16.2 WinCC システム情報チャンネル

原理

システム情報チャンネルは時間、日付、ディスク容量などのシステム情報を評価するために使用され、タイマやカウンタなどのファンクションを提供します。

可能なアプリケーションは以下のとおりです。

- プロセス画像での時間、日付、曜日の表示
- スクリプトでのシステム情報の評価によるイベントのトリガ
- CPU 負荷のトレンドでの表示
- クライアントシステムの異なるサーバーの、使用可能なドライブ容量の表示とモニタ
- 使用可能なディスク容量のモニタおよびメッセージのトリガ

チャンネルは、チャンネルがインストールされているコンピュータのシステム情報に直接アクセスするため、ハードウェアを必要としません。チャンネルが機能するには、接続を

設定する必要があります。接続を追加することもできますが、正常な動作には必要ありません。

チャンネルおよびタグの診断の詳細については、「通信診断」を参照してください。

注記

ライセンシング

システム情報チャンネルに必要なプロセスタグに、ライセンスは必要ありません。このため、このタグはライセンスカウントに入力されません。

ユーザー権限

管理者権限を所持していない場合、システム情報チャンネルを使用するには、パワーユーザーおよび[パフォーマンスモニタユーザー]グループのメンバーになる必要があります。

接続ステータスのタグ

タグ@<connection_name>@ForceConnectionStateEx と

@<connection_name>@ConnectionStateEx は、「システム情報」チャンネルではサポートされていません。

通信マニュアル

通信マニュアルには、チャンネル設定の追加情報と各種の例が記載されています。このマニュアルは、下記のインターネットからダウンロードできます。

- <http://support.automation.siemens.com/>

アイテム番号による検索:

- A5E00391327

下記も参照

マルチユーザーおよびクライアントシステムにおける使用 (ページ 606)

システム情報の呼び出しと評価方法 (ページ 593)

システム情報チャンネルをコンフィグレーションする方法 (ページ 592)

他のソフトウェアの構成要素 との違い (ページ 591)

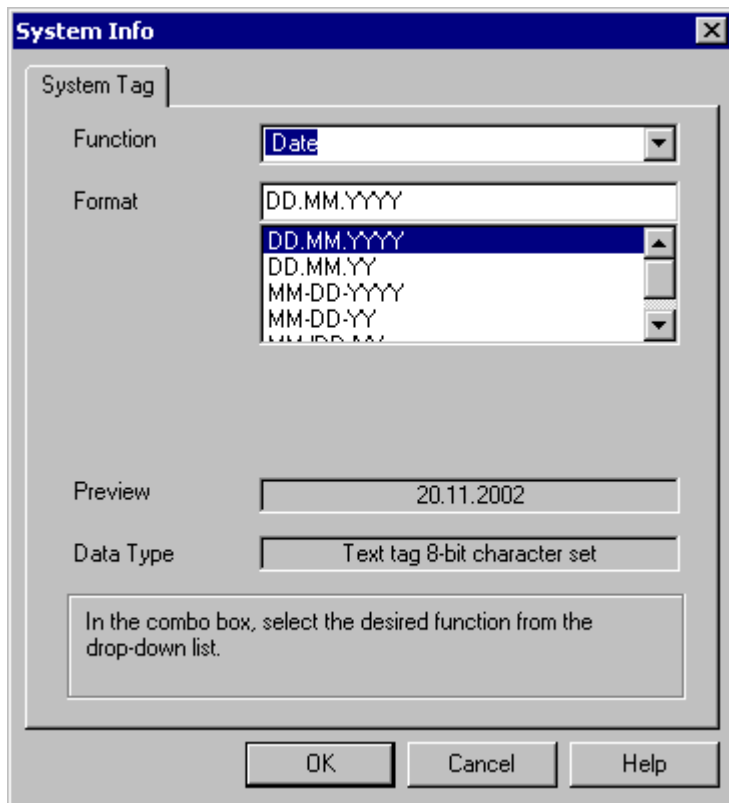
サポートされるシステム情報の概要 (ページ 586)

チャンネルおよびタグの診断 (ページ 615)

6.16.3 サポートされるシステム情報の概要

はじめに

[システム情報]ダイアログの[ファンクション]フィールドでは、WinCC タグにどのシステム情報を割り付けるのかを指定できます。[フォーマット]フィールドでは、表示フォーマットを設定します。



システム情報をサポートする"システム情報"チャンネル - 概要

機能	データタイプ	フォーマット	プレビュー
日付	テキストタグ 8 ビット文字セット	DD.MM.YYYY DD.MM.YY MM-DD-YYYY MM-DD-YY MM/DD/YY	21.10.1999 21.10.99 10-21-1999 10-21-99 10/21/99
日	符号なし 16 ビット値	DD	1...31
月	符号なし 16 ビット値	MM	1...12
年	符号なし 16 ビット値	YYYY	2000
平日	符号なし 16 ビット値 テキストタグ 8 ビット文字セット	テキスト:1 (月曜) ~ 7 (日曜) テキスト:月、火、水、 木、金、土、日	1...7 月...日
時間	テキストタグ 8 ビット文字セット 長さ=10 バイト 長さ = 12 バイト	HH:MM:SS HH:MM HH:MM AM,PM HH:MM:SS AM,PM	23:45:37 23:45 23:45 PM 23:45:37 PM
時間	符号なし 16 ビット値	HH	0...23
分	符号なし 16 ビット値	MM	0...59
秒	符号なし 16 ビット値	SS	0...59
ミリ秒	符号なし 16 ビット値	MSC	0...999

6.16 システム情報

機能	データタイプ	フォーマット	プレビュー
カウンタ	符号付き 32 ビット値	ZZZZ	0...9999
CPU 負荷	浮動小数点数 32 ビット IEEE 754	総負荷 (%) アイドル負荷 (%) プロセス負荷 (%)	0...100%
タイマ	符号付き 32 ビット値	TTTT	0...9999
空きメインメモリ	浮動小数点数 32 ビット IEEE 754	空き容量 (KB) 空き (%) 空き (バイト)	0...n KB 0...100% 0...n B
空き ディスク容量 (ローカルディスク)	浮動小数点数 32 ビット IEEE 754	空き (MB) 空き (%)	0...n MB 0...100%
プリンタモニタ	符号なし 32 ビット値	スプーラディスクの 使用容量 プリンタステータス ジョブステータス スプーラディスクの空き 領域 (KB) 空き PRT_OUT- ディスク領域 (KB) スプーラディレクトリの サイズ (KB) PRT_OUT ディレクトリのサイズ (KB)	0...n % 0...n 0...n 0...n kB 0...n kB 0...n kB 0...n kB
スワップファイル の ステータス	浮動小数点数 32 ビット IEEE 754	使用済 (KB) 使用済 (%) 使用可能 (KB)	0...n kB 0...100% 0...n kB

カウンタ

このファンクションは、スクリプトのテスト目的に使用できます。

タイマ

このファンクションを選択すると、[システム情報]ダイアログにフィールド[制限]および[終わり]が追加されます。

タイマは秒ごとに加算または減算されます。変更が行われる指示は、フィールド[制限]および[終わり]の初期値および終了値によって決定されます。初期値が終了値より小さい場合、タイマは加算されます。初期値が終了値より大きい場合、タイマは減算されます。

ランタイム時にタイマとリンクしている I/O フィールドに値が入力されると、初期および現在のタイマ値がこの値に設定されます。例:初期値が 0 で終了値が 60 に設定されているタイマ。ランタイムに 0 が書き込まれると、タイマはリセットされます。

無効にすると、再び元の初期値が適用されます。

CPU 負荷

マルチプロセッサ PC に関連する"総負荷 (%) および"アイドル負荷 (%) フォーマットの場合は、"0"から始まる CPU 番号を入力します。

"プロセス負荷 (%) "については、1 つのプロセスにインスタンスが複数ある場合はインスタンス数を入力します。

ディスクの空き容量

システムは、ローカルハードディスクまたはディスク上での使用可能な容量を決定するだけです。

プリンタモニタ

"プリンタステータス"および"ジョブステータス"フォーマットで、サーバー名を[プリンタ]フィールドに入力する必要があります。使用しているプリンタは、このシステム情報を使用可能にするためこのステータス情報をサポートする必要があります。

プリンタステータスを分析する場合は、以下に注意してください。

- ポートモニタにより、プリンタステータスがスプーラに送信されます。選択したプリンタポートにより、異なる Port Monitor DLL がインストールされます。Windows で提供されるポートモニタのうち"TCPMON.DLL"のみが、TCP/IP ポートを使用してプリンタステータスを送信できます。LPT ポートを使用した"LOCALMON.DLL"では、プリンタステータスを通信しません。
- プリンタステータスは、印刷ジョブが送信されたあとに限って評価されますが、ポートでのステータスのポーリング中は評価されません。

6.16 システム情報

"PRT_OUT ドライバ空き容量"および"PRT_OUT ディレクトリサイズ"フォーマットで、チャンネルは自動的に[ディレクトリ]フィールドのパスを決定します。

"プリンタステータス"フォーマットのエラーコード

ステータス	エラーコード
PRINTER_STATUS_PAUSED	0x00000001
PRINTER_STATUS_ERROR	0x00000002
PRINTER_STATUS_PENDING_DELETION	0x00000004
PRINTER_STATUS_PAPER_JAM	0x00000008
PRINTER_STATUS_PAPER_OUT	0x00000010
PRINTER_STATUS_MANUAL_FEED	0x00000020
PRINTER_STATUS_PAPER_PROBLEM	0x00000040
PRINTER_STATUS_OFFLINE	0x00000080
PRINTER_STATUS_IO_ACTIVE	0x00000100
PRINTER_STATUS_BUSY	0x00000200
PRINTER_STATUS_PRINTING	0x00000400
PRINTER_STATUS_OUTPUT_BIN_FULL	0x00000800
PRINTER_STATUS_NOT_AVAILABLE	0x00001000
PRINTER_STATUS_WAITING	0x00002000
PRINTER_STATUS_PROCESSING	0x00004000
PRINTER_STATUS_INITIALIZING	0x00008000
PRINTER_STATUS_WARMING_UP	0x00010000
PRINTER_STATUS_TONER_LOW	0x00020000
PRINTER_STATUS_NO_TONER	0x00040000
PRINTER_STATUS_PAGE_PUNT	0x00080000
PRINTER_STATUS_USER_INTERVENTION	0x00100000
PRINTER_STATUS_OUT_OF_MEMORY	0x00200000
PRINTER_STATUS_DOOR_OPEN	0x00400000
PRINTER_STATUS_SERVER_UNKNOWN	0x00800000
PRINTER_STATUS_POWER_SAVE	0x01000000

"ジョブステータス"フォーマットのエラーコード

ステータス	エラーコード
JOB_STATUS_PAUSED	0x00000001
JOB_STATUS_ERROR	0x00000002
JOB_STATUS_DELETING	0x00000004
JOB_STATUS_SPOOLING	0x00000008
JOB_STATUS_PRINTING	0x00000010
JOB_STATUS_OFFLINE	0x00000020
JOB_STATUS_PAPEROUT	0x00000040
JOB_STATUS_PRINTED	0x00000080
JOB_STATUS_DELETED	0x00000100
JOB_STATUS_BLOCKED_DEVQ	0x00000200
JOB_STATUS_USER_INTERVENTION	0x00000400
JOB_STATUS_RESTART	0x00000800

注記

"プリンタステータス"および"ジョブステータス"フォーマットのエラーコードは、Visual C 参照ファイル"Winspool.h"の値に対応しています。

6.16.4 他のソフトウェアの構成要素 との違い

はじめに

[システム情報]チャンネルのシステム情報の一部は、ActiveX コントロールを使用している WinCC で評価または表示することもできます。

ActiveX コントロールとの基本的な違いは、"システム情報"チャンネルのシステム情報は、WinCC タグに割り付けられることです。継続される評価(例: メッセージ、限界値などの評価)は、繰り返し実行でき、個別にコンフィグレーションされます。ActiveX コントロールは特定のアプリケーション用ですが、複数ユーザまたはクライアントシステムでも使用できます。

以下のシステム情報の場合、ActiveX コントロールと'システム情報'チャンネルの間には違いがあります。

時間

ActiveX コントロール"WinCC デジタル/アナログクロックコントロール"は、WinCC で時間を表示するために使用されます。このコントロールは、時間のアナログ表示もサポートします。このコントロールは、プロセスコントロールシステムオプションで使用される場合でも、システム情報チャンネルを必要としません。コントロールを使用して、WinCC クライアントの時間をそのプロセスピクチャ内に表示することができます。これは常にサーバのシステム時間を表示するため、システム情報チャンネルでは不可能です。

ディスクの空き容量

ActiveX コントロール"IX ディスク容量"は、WinCC で使用可能なディスク容量を表示するために提供されます。この Active コントロールはネットワークドライブの使用可能容量も表示でき、複数の限界値設定などその他の設定オプションをコントロールで直接サポートします。

6.16.5 チャンネルのコンフィグレーション

6.16.5.1 システム情報チャンネルをコンフィグレーションする方法

はじめに

このセクションでは、システム情報チャンネルを構成する方法を説明します。

手順

1. タグ管理のナビゲーションエリアで、[タグ管理]ノードのショートカットメニューから、[新規ドライバの追加]エントリを選択します。
2. [システム情報]ドライバを選択します。チャンネルが作成され、[タグ管理]に通信ドライバが表示されます。
3. 関連するシステム情報チャンネルユニットを選択し、ショートカットメニューを呼び出します。このショートカットメニューで、[新規接続]を選択します。
4. 接続の名前を入力します。
5. テーブルエリアの下にある[タグ]タブをクリックします。
6. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
このチャンネルの例を使用する場合は、[タグのコンフィグレーション方法]トピックを続けます。
7. [データタイプ]フィールドで、希望するデータタイプを選択します。

6.16.6 システム情報の評価と表示例

6.16.6.1 システム情報の呼び出しと評価方法

このセクションでは、さまざまな方法でシステム情報を表示し評価する方法を、例を使って説明します。

下記も参照

ステータス表示にプリンタステータスを表示する方法 (ページ 602)

空きディスク容量に関するメッセージの設定方法 (ページ 599)

トレンドウィンドウで CPU 負荷を表示する方法 (ページ 597)

空きディスクスペースを棒グラフで表示する方法 (ページ 596)

I/O フィールドで時間を表示する方法 (ページ 595)

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

6.16.6.2 システム情報チャンネル内のタグをコンフィグレーションする方法

はじめに

このセクションでは、システム情報チャンネル内のタグを設定する方法を説明します。この例ではこれらのタグを使用します。

必要条件


"SystemInfo.chn"チャンネルがインストールされていること。

使用されるデータタイプの表

下記の表はシステム情報チャンネルで使用するタグのタイプとフォーマットを示しています。

例	タグ名	システム情報(ファンクション)	フォーマット	データタイプ
I/O フィールド	Sysinfo_Time	時間	時間:分:秒 (HH:MM:SS)	テキストタグ 8 ビット文字セット
バー、メッセージ	Sysinfo_Drive_C	空きドライブ容量 (ドライブ:C)	数値 0-100% (%の空き)	浮動小数点数 32 ビット IEEE 754
トレンド	Sysinfo_CPU	CPU 負荷	数値 0-100% (%の総負荷)	浮動小数点数 32 ビット IEEE 754
プリンタステータス	Sysinfo_Printerstate	プリンタモニタ	0-n (16 進数) (プリンタステータス)	符号なし 32 ビット値

手順

1. 関連するシステム情報チャンネルユニットのショートカットメニューで、[新規接続]エントリを選択し、"Testinfo"と名前を付けて接続を作成します。
2. テーブルエリアの下にある[タグ]タブをクリックします。
3. [名前]列の先頭の空きセルをクリックします。
タグ名を入力します。
4. [システム情報]ダイアログを開きます。
この目的で、[アドレス]フィールドをクリックしてから、 ボタンをクリックします。
5. 表から、例に適したファンクションおよび表示フォーマットを適用します。
[データタイプ]フィールドは自動的に調整されます。
6. ダイアログを閉じます。

下記も参照

空きディスク容量に関するメッセージの設定方法 (ページ 599)

システム情報チャンネルをコンフィグレーションする方法 (ページ 592)

ステータス表示にプリンタステータスを表示する方法 (ページ 602)

トレンドウィンドウで CPU 負荷を表示する方法 (ページ 597)

空きディスクスペースを棒グラフで表示する方法 (ページ 596)

I/O フィールドで時間を表示する方法 (ページ 595)

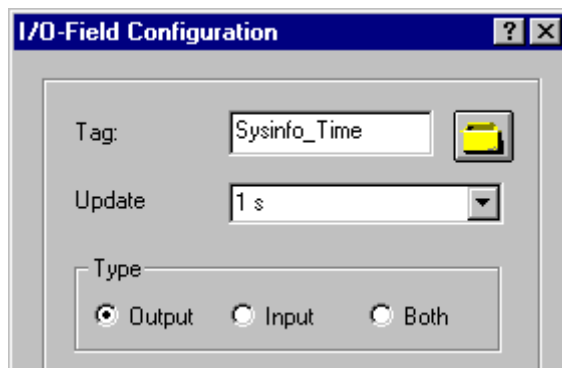
6.16.6.3 I/O フィールドで時間を表示する方法

前提条件

"Sysinfo_Time"タグを"テキストタグ 8 ビット文字セット"データタイプで設定すること。このタグは、"HH:MM:SS"表示フォーマットで"時間"システム情報に割り付けられている必要があります。

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. 画像内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで [I/O フィールド]オブジェクトを選択します。[I/O フィールドコンフィグレーション]ダイアログが開きます。



3. [タグ]フィールドに、名前"Sysinfo_Time"を入力します。
4. 更新を"1 s"に設定します。
5. [出力]にフィールドタイプを設定します。ダイアログを閉じます。
6. I/O フィールドのショートカットメニューの[プロパティ]をクリックして、[オブジェクトプロパティ]ダイアログを開きます。
7. [プロパティ]タブで、[出力/入力]を選択します。[データのフォーマット]属性を[文字列]に設定します。
8. ダイアログを閉じて、画像を保存します。
9. グラフィックデザイナーのツールバーの適切なボタンをクリックしてランタイムを有効にします。

注記

更新サイクルはコンピュータの負荷に影響するため、十分考慮をして選択する必要があります。したがって、250 ミリ秒ごとの時間表示の更新は、システムパフォーマンスには好ましくありません。

下記も参照

ランタイムの開始方法 (ページ 605)

I/O フィールドの挿入方法 (ページ 605)

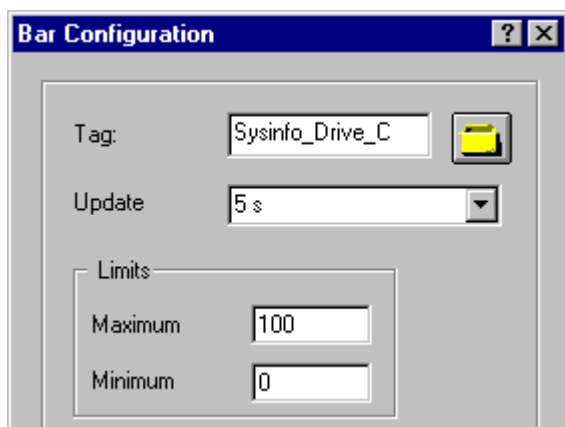
システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

6.16.6.4 空きディスクスペースを棒グラフで表示する方法**前提条件**

"Sysinfo_Drive_C"タグが"32 ビット浮動小数点番号 IEEE754"データタイプでコンフィグレーションされていること。このタグは、"空き容量(%)"表示フォーマットで、ドライブ"C"の"ディスク空き容量"システム情報に割り付けられる必要があります。

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. 画像内にバーグラフを挿入します。これを行うには、オブジェクトパレットの[スマートオブジェクト]から[バー]オブジェクトを選択します。[バーコンフィグレーション]ダイアログを開きます。



3. [タグ]フィールドに、名前"Sysinfo_Drive_C"を入力します。

4. 更新を"5 s"に設定します。
5. 最大値を"100"、最小値を"0"に設定します。 ダイアログを閉じます。
6. 棒グラフのショートカットメニューの[プロパティ]をクリックして、[オブジェクトプロパティ]ダイアログを開きます。
7. [プロパティ]タブで、[軸]を選択します。 [小数位]属性を[0]に設定します。
8. ダイアログを閉じて、画像を保存します。
9. グラフィックデザイナのツールバーの適切なボタンをクリックしてランタイムを有効にします。

注記

更新サイクルはコンピュータの負荷に影響するため、十分考慮をして選択する必要があります。 したがって、使用可能なドライブ領域の表示を每秒更新することは、システムパフォーマンスには好ましくありません。

下記も参照

ランタイムの開始方法 (ページ 605)

バーグラフの挿入方法 (ページ 604)

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

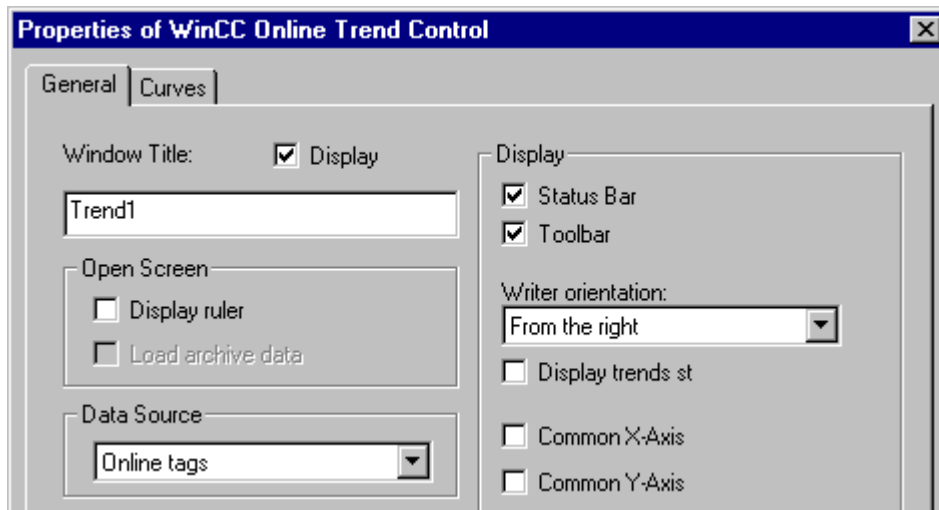
6.16.6.5 トレンドウィンドウで CPU 負荷を表示する方法

前提条件

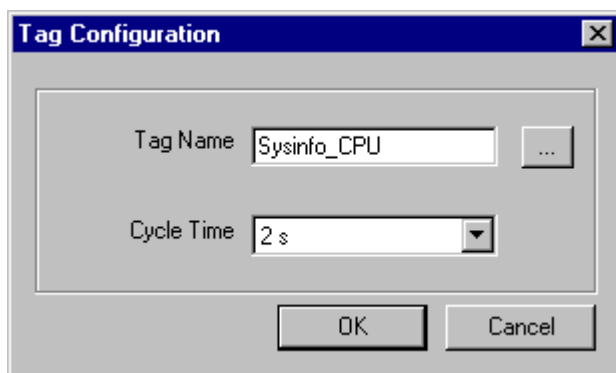
"Sysinfo_CPU"タグが"32 ビット浮動小数点番号 IEEE754"データタイプでコンフィグレーションされていること。このタグは、"総負荷(%)"表示フォーマットで"CPU 負荷"システム情報に割り付けられる必要があります。

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. 画像内にトレンド表示を挿入します。 これを行うには、[コントロール]オブジェクトパレットから[WinCC オンライントレンドコントロール]オブジェクトを選択します。 [WinCC オンライントレンドコントロールのプロパティ]ダイアログが開きます。



3. [一般]タブで、[ウィンドウタイトル]フィールドに[Trend1]と名前を入力します。
4. [データソース]フィールドで、[オンラインタグ]を選択します。
5. [トレンド]タブをクリックし、[選択]ボタンをクリックして[タグの設定]ダイアログを開きます。
6. タグの名前として"Sysinfo_CPU"を入力して、サイクル時間に"2 s"を選択します。 ダイアログを閉じます。



7. [WinCC オンライントレンドコントロールのプロパティ]ダイアログを閉じて、画像を保存します。
8. グラフィックデザイナーのツールバーの適切なボタンをクリックしてランタイムを有効にします。

注記

更新サイクルはコンピュータの負荷に影響するため、十分考慮をして選択する必要があります。したがって、500 ミリ秒ごとの CPU 負荷表示の更新は、システムパフォーマンスには好ましくありません。

下記も参照

ランタイムの開始方法 (ページ 605)

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

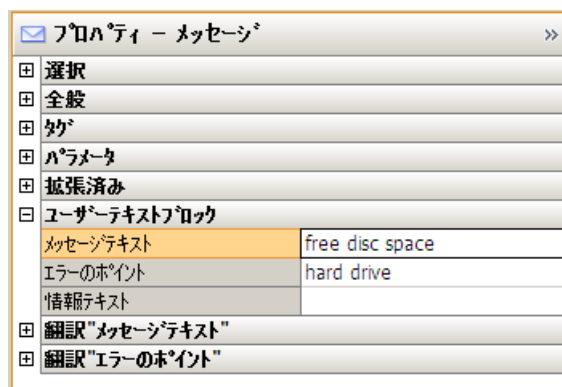
6.16.6.6 空きディスク容量に関するメッセージの設定方法

必要条件

"Sysinfo_Drive_C"タグが"32 ビット浮動小数点番号 IEEE754"データタイプで設定されていること。このタグは、[空き容量(%)]表示フォーマットで、ドライブ「C」の[使用可能なドライブ容量]システム情報に割り付けられる必要があります。

手順

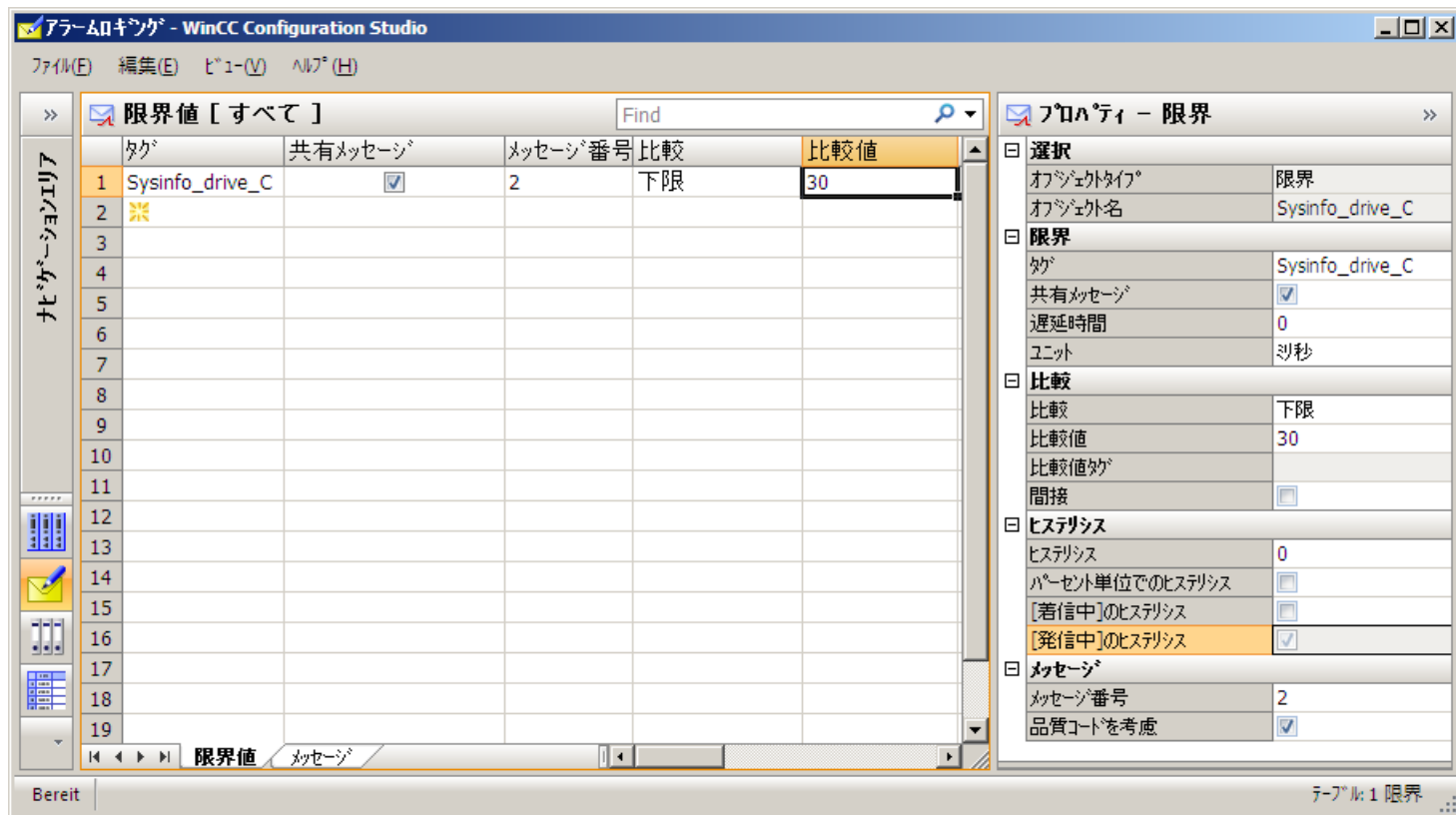
1. "アラームロギング"エディタを開き、新規メッセージを作成します。
2. [プロパティ]エリアで、以下を定義します。
 - 「メッセージテキスト」としてのテキスト「メモリの空き容量」
 - 「エラーのポイント」としてのテキスト「ハードドライブ」



3. アラームロギングのナビゲーションエリアで、[制限値のモニタリング]ノードを選択します。
4. [制限]データウィンドウで、[Sysinfo_Drive_C]タグに新しい制限値を作成します。

6.16 システム情報

5. [共有メッセージ]オプションを有効にします。
6. [メッセージ数]フィールドに、新規に作成されたメッセージの数を入力します。
7. [制限]データウィンドウで、矢印記号をクリックして、[Sysinfo_Drive_C]エントリを拡張します。
 - 新たに挿入された行で、[下限値]を選択します。
 - [比較値]ボックスに値「30」を入力します。



下記も参照

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

利用可能なディスク容量に関するメッセージを表示する方法 (ページ 601)

6.16.6.7 利用可能なディスク容量に関するメッセージを表示する方法

必要条件

- [Sysinfo_Drive_C]タグが、[32 ビット IEEE754 浮動小数点数]データタイプであること。
タグは、[空き容量(%)]表示フォーマットで、ドライブ「C」の[ディスク空き容量]システム情報に割り付けられる必要があります。
- このタグの制限値をモニタリングするための、メッセージテキストおよび下限値が設定されていること。
- コンピュータの起動パラメータに、[アラームロギングランタイム]が設定されていること。

手順

1. [グラフィックデザイナー]を起動し、画像を開きます。
2. 画像に[メッセージ]ウィンドウを挿入します。[コントロール]オブジェクトパレットから[WinCC アラームコントロール]オブジェクトを選択し、画像内に置きます。
3. [プロパティ]ダイアログの[メッセージブロック]タブで、[既存のメッセージブロック]フィールドにある[メッセージテキスト]および[エラーのポイント]チェックボックスを有効にします。
4. [メッセージリスト]タブの[メッセージ行のエレメント]フィールドに、既存のメッセージブロック[メッセージテキスト]および[エラーのポイント]を移動します。
5. ダイアログを閉じて、画像を保存します。
6. グラフィックデザイナーのツールバーの適切なボタンをクリックしてランタイムを有効にします。

下記も参照

WinCC スタートアップパラメータをチェックする方法 (ページ 604)

ランタイムの開始方法 (ページ 605)

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

6.16.6.8 ステータス表示にプリンタステータスを表示する方法

はじめに

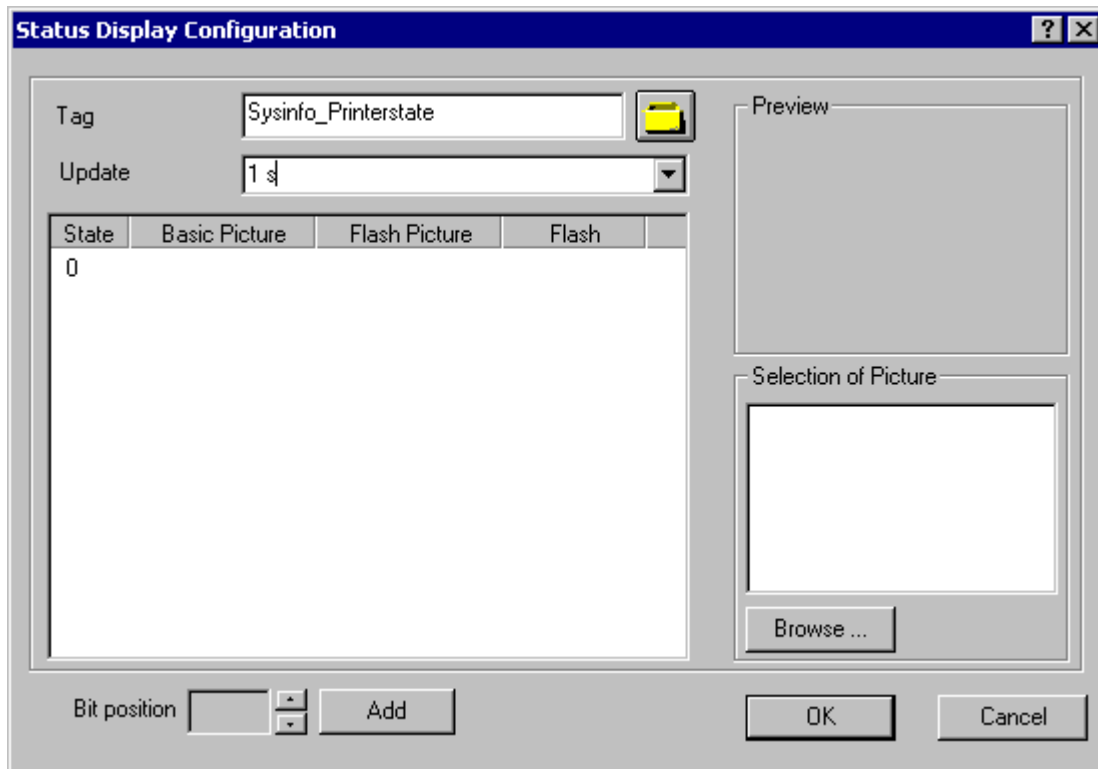
この例では、"プリンタステータス"表示フォーマットに基づいた"プリンタのモニタ"システム情報の可能な評価を示しています。 プリンタまたはジョブステータスは、プリンタまたはジョブステータスの単一ビットによってトリガされるメッセージをコンフィグレーションして評価することもできます。

前提条件

- "Sysinfo_Printerstate"タグが"符号なし 32 ビット値"データタイプにコンフィグレーションされていること。 このタグは、"プリンタステータス"表示フォーマットで"プリンタのモニタ"システム情報に割り付けられる必要があります。
- 使用しているプリンタが、このシステム情報を使用できるようにこれらのステータス表示をサポートしていること。

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. 画像内にステータス表示を挿入します。[スマートオブジェクト]の下オブジェクトパレットから[ステータス表示]オブジェクトを選択します。[ステータス表示コンフィグレーション]ダイアログが開きます。



3. [タグ]フィールドに、名前"Sysinfo_Printerstate"を入力します。
4. 更新を"1 s"に設定します。
5. [ステータス]列で値"0"を選択します。このステータスを[画像の選択]エリア(例: プリンタを示している)からアイコンに割り付けます。希望するアイコンを選択し、マウスを使ってこの画像を"0"行にドラッグして、[基本画像]列にドロップします。[画像の選択]エリアに画像が表示されなかったり、希望する画像が現れない場合は、[参照...]ボタンをクリックして選択ダイアログを開くことができます。
6. また[追加]ボタンでビット位置を追加し、他の画像をこれらのステータスに割り付けることができます。
7. ダイアログを閉じて、画像を保存します。
8. グラフィックデザイナーのツールバーの適切なボタンをクリックしてランタイムを有効にします。

ランタイムでは、利用可能なプリンタが、ステップ 5 でステータス"0"が割り当てられたアイコンで表示されます。プリンタにエラーがある場合、またはステップ 6 を実行して

6.16 システム情報

いない場合、画像は表示されません。ステップ 6 でアイコンを他のビット位置に割り当てた場合は、そのように表示されます。

注記

更新サイクルはコンピュータの負荷に影響するため、十分考慮をして選択する必要があります。したがって、250 ミリ秒ごとの時間表示の更新は、システムパフォーマンスには好ましくありません。

下記も参照

システム情報チャンネル内のタグをコンフィグレーションする方法 (ページ 593)

ランタイムの開始方法 (ページ 605)

6.16.6.9 WinCC スタートアップパラメータをチェックする方法

手順

1. WinCC Explorer で、[コンピュータ]を選択します。
2. ショートカットメニューを開き、[プロパティ]を選択します。
3. [コンピュータリストプロパティ]ダイアログが開きます。[プロパティ]ボタンをクリックします。
4. [コンピュータプロパティ]ダイアログが開きます。
5. [スタートアップ]タブをクリックし、エントリをチェックします。必要に応じ、ランタイムモジュールを有効/無効化するか、または他のアプリケーションをスタートアップリストに追加します。
6. 開いているダイアログを閉じます。

6.16.6.10 バーグラフの挿入方法

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. [スマートオブジェクト]の下の[標準]オブジェクトパレットで、[バー]オブジェクトを選択します。
3. 画像内にバーグラフを挿入します。これを行うには、バーグラフを配置する作業エリア内の位置にマウスポインタを置きます。マウスボタンを押したまま、オブジェクトが希望のサイズになるまでドラッグします。

4. マウスボタンを放すと、[バーコンフィグレーション]ダイアログが開きます。このダイアログに、WinCC タグの名前を入力し、更新仕様および制限を設定します。さらに、[バー方向]を使用して、表示するバーの方向を設定することができます。
5. ダイアログを閉じます。

6.16.6.11 I/O フィールドの挿入方法

手順

1. [グラフィックデザイナー]を開始し、画像を開きます。
2. [スマートオブジェクト]の下の[標準]オブジェクトパレットで、[I/O フィールド]オブジェクトを選択します。
3. 画像内に I/O フィールドを挿入します。これを行うには、I/O フィールドを配置する作業エリア内の位置にマウスポインタを置きます。マウスボタンを押したまま、オブジェクトが希望のサイズになるまでドラッグします。
4. マウスボタンを放すと、[I/O フィールドのコンフィグレーション]ダイアログボックスが開きます。このダイアログボックスに、WinCC タグの名前を入力し、更新およびフィールドタイプを設定します。さらに、値の表示に使用する[フォント]も設定することができます。
5. ダイアログを閉じます。

6.16.6.12 ランタイムの開始方法

前提条件

ランタイムを有効にする前にスタートアップ画像が定義されていること。

手順

1. エディタ内で開いているすべてのファイルを保存してから閉じます。
2. WinCC Explorer を選択します。
3. ツールバーにある[有効にする]ボタンをクリックするか、または[ファイル]メニューから[有効にする]を選択して、プロジェクトを有効にします。

6.16 システム情報

6.16.7 特殊ファンクション

6.16.7.1 マルチユーザーおよびクライアントシステムにおける使用

マルチユーザーおよびクライアントシステムにおける使用

はじめに

マルチユーザーおよびクライアントシステムでは、システム情報チャンネルを使用してサーバーからのシステム情報をクライアントシステム上で処理することができます。このためクライアントシステムでは、単一 WinCC クライアントで複数のサーバーのモニタが可能です。

下記も参照

WinCC クライアント上の複数のサーバーのシステム情報のモニタ (ページ 606)

6.16.7.2 複数のサーバーからのモニタシステム情報の例

WinCC クライアント上の複数のサーバーのシステム情報のモニタ

はじめに

この例では、単一 WinCC クライアントによる 2 つのサーバーのモニタを示しています。モニタするシステム情報(使用可能なディスク容量や CPU 負荷など)が WinCC クライアントのプロセス画像に表示されます。

これを行うには以下のコンフィグレーションが必要です。

第 1 サーバーのコンフィグレーション

第 2 サーバーのコンフィグレーション

WinCC クライアント上のタグのインポート

WinCC クライアント上でプロセス画像をコンフィグレーションする方法

プロジェクトの有効化

前提条件

サーバーおよび WinCC クライアントが Windows ネットワークに接続されていること。

下記も参照

プロジェクトを有効化する方法 (ページ 612)

WinCC クライアント上でプロセス画像をコンフィグレーションする方法 (ページ 611)

WinCC クライアントにタグをインポートする方法 (ページ 610)

第 2 サーバーをコンフィグレーションする方法 (ページ 608)

第 1 サーバーをコンフィグレーションする方法 (ページ 607)

第 1 サーバーをコンフィグレーションする方法

はじめに

このセクションでは、第 1 サーバーのコンフィグレーションを例を示して説明しています。

1. システム情報チャンネルのタグを構成して、使用可能なドライブ容量および CPU 負荷を表します。
2. パッケージの生成。

使用されるデータタイプの表



"システム情報"チャンネル例で使用するタグ名およびフォーマットを下の表に示します。

タグ	機能	データタイプ	フォーマット
Sysvar_1_Drive _C	空きドライブ 容量	浮動小数点数 32 ビット IEEE 754	0-100% (空き%)
Sysvar_1_CPU	CPU 負荷	浮動小数点数 32 ビット IEEE 754	0~100% (総負荷(%))

手順

1. 第 1 サーバー上に"Testinfo_1"と名前を付けてマルチユーザープロジェクトを作成します。
[システム情報] ドライバをプロジェクトにインストールします。
2. 関連するシステム情報チャンネルユニットのショートカットメニューで、[新規接続] エントリを選択し、"Connection1"と名前を付けて接続を作成します。
3. テーブルエリアの下にある[タグ]タブをクリックします。

6.16 システム情報

4. [名前]列の先頭の空きセルをクリックします。
タグに名前"Sysvar_1_Drive_C"を入力します。
5. [データタイプ]を"32 ビット浮動小数点番号 IEEE 754"に設定します。
6. [システム情報]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、ボタンをクリックしてください。
7. [ファンクション]フィールドを[使用可能なドライブ容量]に、[ドライブ]を[C]に、[フォーマット]を[空き容量(%)]にそれぞれ設定します。ダイアログを閉じます。
8. [名前]列の先頭の空きセルをクリックします。
タグに名前"Sysvar_1_CPU"を入力します。
9. [データタイプ]を"32 ビット浮動小数点番号 IEEE 754"に設定します。
10. [システム情報]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、ボタンをクリックしてください。
11. [ファンクション]フィールドの値を[CPU 負荷]に、[フォーマット]を[総負荷(%)]にそれぞれ設定します。ダイアログを閉じます。
12. パッケージを作成します。ナビゲーションウィンドウで[サーバーデータ]を選択してから、ショートカットメニューを開きます。メニュー項目[作成]を選択します。パッケージが作成されたメッセージ状態を確認します。

下記も参照

第 2 サーバーをコンフィグレーションする方法 (ページ 608)

第 2 サーバーをコンフィグレーションする方法

はじめに

このセクションでは、第 2 サーバーのコンフィグレーションを例を示して説明しています。



1. システム情報チャンネルのタグを構成して、使用可能なドライブ容量および CPU 負荷を表します。
2. パッケージの生成。

使用されるデータタイプの表

"システム情報"チャンネル例で使用するタグ名およびフォーマットを下の表に示します。

タグ	機能	データタイプ	フォーマット
Sysvar_2_Drive _C	空きドライブ 容量	浮動小数点数 32 ビット IEEE 754	0-100% (空き%)
Sysvar_2_CPU	CPU 負荷	浮動小数点数 32 ビット IEEE 754	0~100% (総負荷(%))

手順

1. 第 2 サーバー上に"Testinfo_2"と名前を付けてマルチユーザープロジェクトを作成します。
[システム情報]ドライバをプロジェクトにインストールします。
2. 関連するシステム情報チャンネルユニットのショートカットメニューで、[新規接続]エン
トリを選択し、"Connection2"と名前を付けて接続を作成します。
3. テーブルエリアの下にある[タグ]タブをクリックします。
4. [名前]列の先頭の空きセルをクリックします。
タグに名前"Sysvar_2_Drive_C"を入力します。
5. [データタイプ]を"32 ビット浮動小数点番号 IEEE 754"に設定します。
6. [システム情報]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックし
てください。
7. [ファンクション]フィールドを[使用可能なドライブ容量]に、[ドライブ]を[C]に、[フォー
マット]を[空き容量(%)]にそれぞれ設定します。開いているすべてのダイアログを閉じます。
8. [名前]列の先頭の空きセルをクリックします。
"Sysvar_2_CPU"をタグの名前として入力します。
9. 接続のショートカットメニューで[新規タグ]を選択します。
10. [データタイプ]を"32 ビット浮動小数点番号 IEEE 754"に設定します。
11. [システム情報]ダイアログを開きます。
これを実行するには、[アドレス]フィールドをクリックしてから、 ボタンをクリックし
てください。
12. [ファンクション]フィールドの値を[CPU 負荷]に、[フォーマット]を[総負荷(%)]にそれぞれ
設定します。開いているすべてのダイアログを閉じます。
13. パッケージを作成します。ナビゲーションウィンドウで[サーバーデータ]を選択してから、
ショートカットメニューを開きます。メニュー項目[作成]を選択します。パッケージが作成
されたメッセージ状態を確認します。

下記も参照

WinCC クライアントにタグをインポートする方法 (ページ 610)

WinCC クライアントにタグをインポートする方法

はじめに

このセクションでは、WinCC クライアントのコンフィグレーションを例を示して説明しています。

1. 第 1 サーバープロジェクトのパッケージのロード
2. 第 2 サーバープロジェクトのパッケージのロード

前提条件

2 つのサーバープロジェクトパッケージを使用すること。

サーバー	プロジェクト	パッケージ
1	Testinfo_1	Testinfo_1_<computer_name>
2	Testinfo_2	Testinfo_2_<computer_name>

手順

1. WinCC クライアント上に"mc_info"と名前を付けてクライアントプロジェクトを作成します。
2. サーバーデータのショートカットメニューで、[ロード]を選択します。[開く]ダイアログを開きます。
3. 第 1 サーバープロジェクト"Testinfo_1"が検索されるコンピュータを選択します。
4. "<project_name>\<computer_name>\パッケージ"ディレクトリの "Testinfo_1_<computer_name>.pck"パッケージを選択します。
5. [開く]ボタンをクリックして、パッケージが開いた後にメッセージを確認します。
6. [Testinfo_2_<computer_name>.pck]パッケージを第 2 サーバーにロードします。これを行うには、[前提条件]テーブルから取得した第 2 プロジェクトに適切な設定と名前を使用してステップ 2～5 を繰り返します。.

下記も参照

WinCC クライアント上でプロセス画像をコンフィグレーションする方法 (ページ 611)

WinCC クライアント上でプロセス画像をコンフィグレーションする方法

はじめに

このセクションは、WinCC クライアント上のプロセス画像におけるサーバーシステム情報を表示する例を示して、WinCC クライアントのコンフィグレーションについて説明しています。

1. 第 1 サーバーのシステム情報表示のコンフィグレーション
2. 第 2 サーバーのシステム情報表示のコンフィグレーション

前提条件

サーバープロジェクトパッケージがクライアントプロジェクトにロードされていること。

パッケージ	プロジェクト	タグ*
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_Drive_C
Testinfo_1_<computer_name>	Testinfo_1	Sysvar_1_CPU
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_Drive_C
Testinfo_2_<computer_name>	Testinfo_2	Sysvar_2_CPU

手順

1. WinCC クライアント上でグラフィックデザイナを開始して、"p_serverinfo"と名前を付けて画像を作成します。
2. 画像内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで [I/O フィールド]オブジェクトを選択します。[I/O フィールドコンフィグレーション]ダイアログが開きます。
3. タグ選択のボタンをクリックします。[タグ]ダイアログを開きます。
4. 第 1 サーバープロジェクト"Testinfo_1"の"Sysvar_1_Drive_C"タグを選択します。これを行うには、"Testinfo_1_<computer_name>"パッケージの下のディレクトリ構造体を開きます。ダイアログを閉じます。
5. [I/O フィールドのコンフィグレーション]ダイアログで、更新を"5 s"に設定します。
6. [出力]にフィールドタイプを設定します。ダイアログを閉じます。

6.16 システム情報

7. 画像内に第 2I/O フィールドを挿入し、同じプロジェクトの"Sysvar_1_CPU"タグにこれをコンフィグレーションします。これを行うには、[前提条件]テーブルから取得した適切な設定を使用してステップ 2～6 を繰り返します。.
8. ステップ 2～7 を繰り返して、第 2 サーバープロジェクト"Testinfo_2"のタグをコンフィグレーションします。
9. ダイアログを閉じて、画像を保存します。

注記

アップデートサイクルはコンピュータの負荷に影響するため、十分考慮のうえ選択する必要があります。したがって、毎秒の日付表示の更新は、システムパフォーマンスには好ましくありません。

下記も参照

プロジェクトを有効化する方法 (ページ 612)

プロジェクトを有効化する方法

はじめに

このセクションは、サーバーおよび WinCC クライアント上でプロジェクトを有効化する方法を示しています。

1. サーバープロジェクト"Testinfo_1"内のスタートアップ画像の作成およびプロジェクトの有効化
2. サーバープロジェクト"Testinfo_2"内のスタートアップ画像の作成およびプロジェクトの有効化
3. WinCC クライアント上のプロジェクト"mc_info"におけるスタートアップ画像の定義およびプロジェクトの有効化

手順

1. サーバー 1 上のプロジェクト"Testinfo_1"のナビゲーションウィンドウで、[グラフィックデザイン]を選択し、ショートカットメニューを使用して新規画像を作成します。
2. この画像をスタートアップ画像として設定します。この操作をするには、ショートカットメニューで[スタートアップ画像に設定]を選択します。
3. ツールバーの[有効化]ボタンをクリックして、プロジェクトを有効化します。
4. サーバー 2 上のプロジェクト"Testinfo_2"のナビゲーションウィンドウで、[グラフィックデザイン]を選択し、ショートカットメニューを使用して新規画像を作成します。

5. この画像をスタートアップ画像として設定します。この操作をするには、ショートカットメニューで[スタートアップ画像に設定]を選択します。
6. ツールバーの[有効化]ボタンをクリックして、プロジェクトを有効化します。
7. プロジェクト"mc_info"のナビゲーションウィンドウの WinCC クライアント上で、グラフィックデザイナーを選択します。プロセス画像"p_serverinfo"がデータウィンドウ内に表示されます。
8. この画像をスタートアップ画像として設定します。この操作をするには、ショートカットメニューで[スタートアップ画面に設定]を選択します。
9. ツールバーの[有効化]ボタンをクリックして、プロジェクトを有効化します。

通信－診断

7.1 チャンネルおよびタグの診断

このセクションでは、チャンネルおよびそれらのタグの診断、そして内部タグの診断について説明します。

これらの診断は、通信問題または予期しないタグ値が発生した場合に使用できます。

次のドキュメントの内容:

- 通信エラーを認識する方法。
- 次の診断ツールを設定および使用する方法:
 - 「ステータス - 論理接続」
 - 「WinCC チャンネル診断制御」
 - 「パフォーマンス」 タググループのシステムタグ
- チャンネル、接続、それらのタグを診断する方法。
- 内部タグを診断する方法。
- WinCC 通信ハードウェアを確認する方法。

下記も参照

"SIMATIC S5 PROFIBUS FDL"チャンネルの診断オプション (ページ 658)

タグの品質コード (ページ 683)

グローバルアクションによるタグステータスのモニタ (ページ 693)

モニタ接続ステータスへのタグステータスの使用 (ページ 691)

内部タグの作成方法 (ページ 694)

チャンネル診断 (ページ 617)

エラー検出に関する一般情報 (ページ 616)

"OPC"チャンネルの診断のための機能 (ページ 669)

"SIMATIC S7 Protocol Suite"チャンネル - 診断オプション (ページ 645)

"システム情報"チャンネル - 診断オプション (ページ 632)

診断チャンネル「SIMATIC S7-1200/S7-1500」 (ページ 638)

7.2 エラー検出に関する一般情報

通信リンクを確立する間に発生する障害またはエラーは、一般にランタイムではじめて検出されます。

現在のプロセス値では提供できない WinCC タグを使用してダイナミック化されるオブジェクトは、プロセスピクチャ内では無効と表示されます。このような例としては、I/O フィールド、スライダオブジェクト、またはバーグラフがあります。

エラーにより接続のいくつかの WinCC タグが影響を受けなかった場合は、WinCC タグの 1 つが問題の原因であることを示しています。この場合、たとえばタグのアドレス指定だけでなく、グラフィックデザイナーで使用する場合のタグの綴りについてもチェックする必要があります。

エラーにより接続のすべての WinCC タグが影響を受けた場合は、接続自体にエラーがあることを示しています。

以下のセクションでは、どのような方法や手段を使用するとエラーの原因を特定できるかを説明します。

7.3 チャンネル診断

7.3.1 チャンネル診断

チャンネルとその接続の診断には、次の機能を使用できます。

- 「ステータス - 論理接続」機能
- 「パフォーマンス」タググループのシステムタグ
- WinCC 「チャンネル診断」

下記も参照

[ステータス - 論理接続]ファンクションを使用してチャンネルを確認する方法 (ページ 622)

チャンネル診断の原理
(ページ 624)

パフォーマンスタグによる接続の確認 (ページ 617)

7.3.2 パフォーマンスタグによる接続の確認

WinCC は、通信チャンネルを分析するために、次のシステムタグを提供しています。

- @PRF_DMRT_CHNCON_...(タグ管理)
- @PRF_ALGRT_CHNCON_...(アラームロギング)

これらのパフォーマンスタグを使用して、通信の時間動作を評価できます。

@<...>@ConnectionStateEx システムタグを使用して、接続のステータスを決定します。追加情報:

- 「[ステータス - 論理接続]ファンクションを使用してチャンネルを確認する方法 (ページ 622)」
- 「ランタイム時の接続ステータスタグの設定 (ページ 170)」

パフォーマンスタグの作成

通信ドライバで新しい接続を作成するとすぐに、WinCC タグ管理は、対応するパフォーマンスタグを作成します。

7.3 チャンネル診断

接続を名前変更するとき、パフォーマンスタグも自動的に名前変更されます。

タグは、[内部タグ] エリアの [Performance] タググループにあります。

WinCC パフォーマンスタグに関する追加情報:

- [WinCC での作業] > [プロジェクトでの作業] > [ランタイムの設定]:
 - [パフォーマンスタグでのシステム診断]
 - [パフォーマンスタグの概要]

パフォーマンスタグのタイプ

[パフォーマンス]タググループには、次のタグタイプが含まれています。

タグ	データタイプ	アクセス	説明
相対タグ	浮動小数点数 64 ビット IEEE 754	読み取り	<p>現在保留中の値や秒ごとの値など、読み取り時間に相対的に適用される値。</p> <p>リセットタグはこれらの値に影響を与えません。</p> <p>タグ名の接尾辞は次のとおりです。</p> <ul style="list-style-type: none"> • ..._PENDING • ..._SECOND <p>更新サイクル:1 秒</p>
カウンタタグ	浮動小数点数 64 ビット IEEE 754	読み取り	<p>ランタイム有効化以降の絶対値</p> <p>リセットタグを使用して、値を「0」にリセットできます。</p> <p>タグ名の接尾辞は次のとおりです。</p> <ul style="list-style-type: none"> • ..._AVERAGE • ..._PEAK • ..._TOTAL <p>更新サイクル:1 秒</p>
リセットタグ	符号なし 32 ビット値	読み取り 書き込み	<p>次のようにスクリプトからリセットタグの値を設定できます。</p> <ul style="list-style-type: none"> • 0:無効 • 1:すべての関連付けられたカウンタタグの値が「0」にリセットされます。リセットタグの値自体も「0」にリセットされます。 <p>タグ名の接尾辞は次のとおりです。</p> <ul style="list-style-type: none"> • ...RESET

7.3 チャンネル診断

パフォーマンスタグの概要

システムタグ ¹⁾	説明
アラームロギング	
@PRF_ALGRT_RESET	リセットタグが、次のパフォーマンスタグの値をリセットします。 <ul style="list-style-type: none"> • @PRF_ALGRT_CHNCON_..._AVERAGE • @PRF_ALGRT_CHNCON_..._PEAK 通信ドライバへの参照のない@PRF_ALGRT_...カウンタタグもリセットされます。
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND ²⁾	接続を介して 1 秒間に送信される生成メッセージの数
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND_AVERAGE	1 秒間のメッセージの平均数
@PRF_ALGRT_CHNCON_<...>_ALARM_PER_SECOND_PEAK	1 秒間のメッセージの最大数
タグ管理	
@PRF_DMRT_CHNCON_<...>_RESET	リセットタグが、次のパフォーマンスタグの値をリセットします。 <ul style="list-style-type: none"> • @PRF_DMRT_CHNCON_<...>..._TOTAL • @PRF_DMRT_CHNCON_<...>_RESET リセットは、同じ接続に対して作成されたすべてのカウンタタグに適用されます。
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_PER_SECOND ²⁾	読み取られたバイト数/秒 ビットはバイトに切り上げられます。 メタデータ(例、タイムスタンプや SetValue コールバックデータ)は含まれていません。
@PRF_DMRT_CHNCON_<...>_TAG_READ_BYTES_TOTAL	ランタイムの有効化以降に読み取られたバイト数 ビットはバイトに切り上げられます。 メタデータ(例、タイムスタンプや SetValue コールバックデータ)は含まれていません。

システムタグ ¹⁾	説明
@PRF_DMRT_CHNCON_<...>_TAG_READS_PENDING	まだ完了していない開始された読み取りリクエスト値が一定に上昇する場合、システムの過負荷を示します。考えられる原因: <ul style="list-style-type: none"> データソースまたは接続が、過負荷状態またはブロックされているため、十分な速度で読み取り要求を処理していません。
@PRF_DMRT_CHNCON_<...>_TAG_READS_PER_SECOND ²⁾	読み取られたタグ数/秒
@PRF_DMRT_CHNCON_<...>_TAG_READS_TOTAL	ランタイムの有効化以降に読み取られたタグ
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PENDING	まだ完了していない開始された書き込みリクエスト値が一定に上昇する場合、システムの過負荷を示します。考えられる原因: <ul style="list-style-type: none"> データソースまたは接続が、過負荷状態またはブロックされているため、十分な速度で書き込み要求を処理していません。
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_PER_SECOND ²⁾	書き込まれたタグ数/秒
@PRF_DMRT_CHNCON_<...>_TAG_WRITES_TOTAL	ランタイムの有効化以降に書き込まれたタグ
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_PER_SECOND ²⁾	書き込まれたバイト数/秒 ビットはバイトに切り上げられます。 メタデータ(例、タイムスタンプや SetValue コールバックデータ)は含まれていません。
@PRF_DMRT_CHNCON_<...>_TAG_WRITTEN_BYTES_TOTAL	ランタイムの有効化以降に書き込まれたバイト数 ビットはバイトに切り上げられます。 メタデータ(例、タイムスタンプや SetValue コールバックデータ)は含まれていません。

1) <...>は、通信接続の名前を表します。例、

「@PRF_DMRT_CHNCON_S7-417_TAG_READS_PER_SECOND」。

2) [PER_SECOND]の情報は、タグ更新直前の 1 秒に関連しています。

7.3 チャンネル診断

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

[ステータス - 論理接続]ファンクションを使用してチャンネルを確認する方法 (ページ 622)

7.3.3 [ステータス - 論理接続]ファンクションを使用してチャンネルを確認する方法

「論理接続ステータス」機能

WinCC エクスプローラは「論理接続ステータス」機能を使用して、設定されたすべての接続の現在のステータスを簡単な形式で表示するオプションを提供します。

しかし、ステータス表示はランタイム時にのみ可能です。

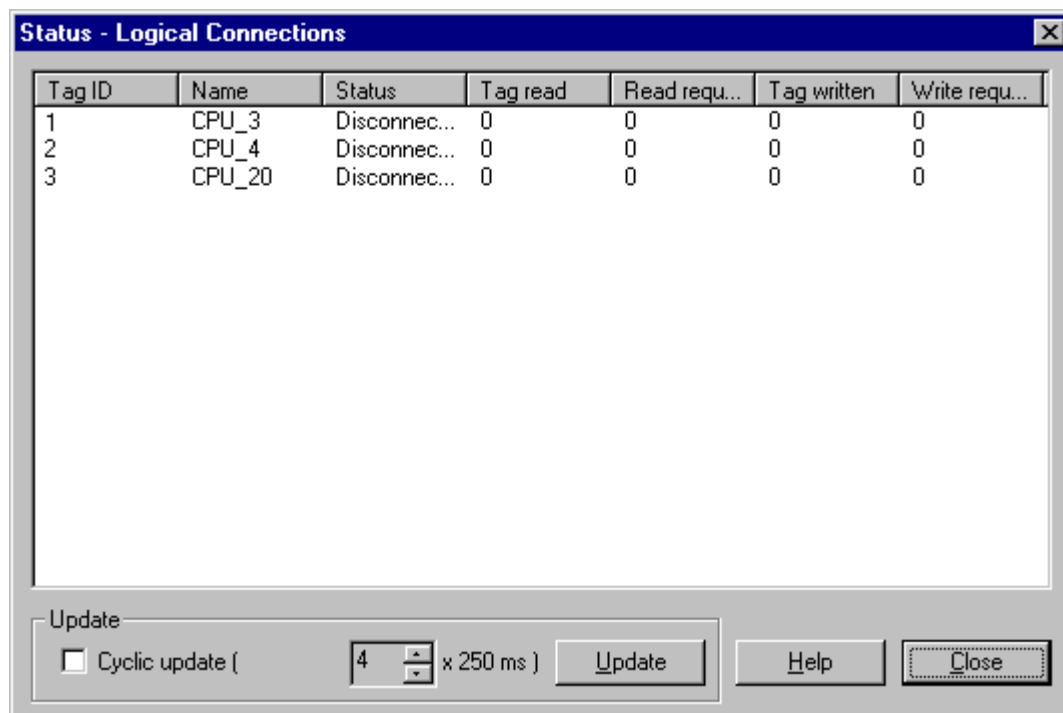
特定の接続が確立または終了しているかどうかだけを照会する場合は、システムタグ「@<Connectionname>@ConnectionStateEx」を使用します。

必要条件

- チャンネルを作成し、このチャンネル内にタグを作成すること。
- WinCC Runtime が有効です。

手順

1. WinCC エクスプローラのメニューバーの「ツール」メニューで、「ドライバ接続のステータス」エントリを選択します。
[ステータス - 論理接続]ダイアログが開きます。



2. 更新を周期的に行うように[更新]エリアで指定できます。
この指定をしない場合、[更新]ボタンをクリックすると手動で表示の更新を開始することができます。
3. 設定された接続は、[名前]列に表示されます。
[ステータス]列には、それぞれの接続の状態が表示されます。
4. [ステータス]列のエントリをチェックします。
ステータス「未接続」は、設定エラーまたはハードウェアエラーがあることを示しています。
関連するチャンネルの詳細情報は、「チャンネル診断」を参照してください。

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

パフォーマンスタグによる接続の確認 (ページ 617)

7.3.4 [チャンネル診断]によるチャンネルの診断

7.3.4.1 チャンネル診断の原理

はじめに

WinCC の[チャンネル診断]を使用すると、WinCC ユーザーはランタイムに有効な接続のステータスに関する概要をすばやく知ることができます。[チャンネル診断]は、一方ではチャンネルユニットに関するステータスと情報を備え、他方では以下の診断出力の設定に対するユーザーインターフェースとしての役割を果たします。

- 通信(たとえばプロセス画像内)のステータスに関する統計または情報の出力
- サービスによるエラー解析と修正のためのログファイルへのテキスト出力
- 通信上の問題の原因を特定するときに、ホットラインを支援するためのトレースファイルへのテキスト出力

診断モジュールは、ActiveX コントロールとしてプロセス画像に挿入したり、独立したアプリケーションとして Windows で開始することができます。

このモジュールには、診断をサポートするチャンネルに対してのみステータス情報が表示されます。

チャンネルのタグの診断については、チャンネル固有の診断の説明に記載されています。

ログブックファイル

[チャンネル診断]では、設定された WinCC チャンネルごとに<ChannelName.log>という名前で呼ばれるログブックファイルが作成されます。これには、重要な情報とエラーが記録されます。テキストの内容の範囲は、チャンネルにより異なります。

ファイルおよび出力テキストの作成を設定することはできません。

ログブックファイルには、開始メッセージと終了メッセージなどの情報、バージョン情報、および通信エラーに関する情報が含まれます。

ファイル内の各エントリは、日付とタイムスタンプ、フラグ名、および説明で構成されます。このファイルは常に入力後すぐに保存され、電圧低下などの事態が発生してもすべての情報が確実に使用できるようになっています。

トレースファイル

追加情報およびエラーが出力される<ChannelName.trc>という名前のトレースファイルは、設定された各 WinCC チャンネルに対して作成できます。トレースファイルを使用するかどうかはランタイムに選択できます。この機能を有効にすると、リンクのランタイムが影響を受けることを警告するメッセージが表示されます。

トレースファイルの各エントリには、タイムスタンプ、続いてフラグ名と説明が含まれます。

トレース機能が有効なとき、ログブックに記録されているすべての情報がトレースファイルにも書き込まれます。

トレースファイルに記録される情報は、通信上の問題の原因を特定するときにホットラインを支援することを目的としています。

注記

トレースファイルエントリおよびログファイルエントリは、英語でのみ記録されます。いずれのファイルも、WinCC ディレクトリ構造の[診断]ディレクトリに保存されます。現在のカウンタ値は、これらのファイルに出力されません。

下記も参照

ActiveX コントロールによるチャンネル診断
(ページ 625)

ActiveX コントロールとしてのチャンネル診断によるチャンネルのチェック方法 (ページ 626)

[チャンネル診断]を使用したチャンネルの診断 (ページ 627)

チャンネル診断によるチャンネルのチェック方法 (ページ 628)

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ランタイムの開始方法 (ページ 631)

7.3.4.2 ActiveX コントロールによるチャンネル診断

はじめに

チャンネルに対するステータス情報は、ActiveX コントロール"WinCC チャンネル診断コントロール"によりプロセス画像に表示することもできます。

7.3 チャンネル診断

ActiveX コントロールはグラフィックデザイナの[コントロール]オブジェクトパレットで見つけることができ、単に画像に挿入されます。したがって、ユーザーは診断プロセス画像などを作成することができます。その中では毎回この配列を再コンフィグレーションする必要なしに、ランタイムに通信のステータスおよびその他の情報を表示することができます。

下記も参照

チャンネル診断によるチャンネルのチェック方法 (ページ 628)

[チャンネル診断]を使用したチャンネルの診断 (ページ 627)

7.3.4.3 ActiveX コントロールとしてのチャンネル診断によるチャンネルのチェック方法

はじめに

このセクションでは、"WinCC チャンネル診断コントロール" ActiveX コントロールを使用するチャンネルの診断方法について示します。

前提条件

- チャンネルを作成し、このチャンネル内にタグを作成すること。

手順

1. [グラフィックデザイナ]を開始し、画像を開きます。
2. "WinCC チャンネル診断コントロール" ActiveX コントロールを画像に挿入します。これは、[コントロール]オブジェクトパレットから[ActiveX コントロール]を選択し、それを画像に挿入して要求されたサイズまでドラッグすることにより実行できます。
3. 画像を保存します。
4. グラフィックデザイナのツールバーの適切なボタンをクリックしてランタイムを有効にします。
5. ActiveX コントロールを挿入した画像を選択します。チャンネルに対するステータス情報が、[チャンネル診断]アプリケーションウィンドウの[チャンネル/接続]タブに表示されます。
6. [コンフィグレーション]タブをクリックします。表示されているチャンネルの 1 つを選択し、関連するログファイルにどのエラーメッセージを記録するかをコンフィグレーションします。
7. トレースファンクションの有効化に関する詳細情報については、「チャンネルのトレースファンクションのコンフィグレーション」を参照してください。

注記

[チャンネル診断]では、チャンネル診断をサポートするチャンネルに対してのみステータス情報が表示されます。

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

7.3.4.4 [チャンネル診断]を使用したチャンネルの診断**はじめに**

チャンネル診断は、WinCC から独立して、Windows プログラムグループ[Siemens Automation]からアプリケーションとして開始することもできます。

したがって、[チャンネル診断]は常に使用可能であり、「WinCC チャンネル診断コントロール」と同様にプロセス画像の選択には左右されません。

WinCC がランタイムの場合、ステータス情報は[チャンネル診断]によってのみ表示されます。

下記も参照

チャンネル診断によるチャンネルのチェック方法 (ページ 628)

ActiveX コントロールによるチャンネル診断
(ページ 625)

7.3.4.5 チャンネル診断によるチャンネルのチェック方法

はじめに

このセクションでは、「チャンネル診断」をアプリケーションとして Windows の[スタート]メニューから開始する方法について説明します。

注記

「チャンネル診断」では、チャンネル診断をサポートするチャンネルに対してのみステータス情報が表示されます。

必要条件

- チャンネルを作成し、このチャンネル内にタグを作成すること。

手順

1. Windows プログラムグループ[Siemens Automation]で、[チャンネル診断]エントリを選択します。
[チャンネル診断]アプリケーションウィンドウが開きます。
ランタイム中の WinCC プロジェクトが現在存在しない場合、[WinCC との接続を確立できません]というメッセージが表示されます。
2. WinCC エクスプローラツールバーからランタイムを有効にします。
3. ActiveX コントロールを挿入した画像を選択します。
チャンネルに対するステータス情報が、[チャンネル診断]アプリケーションウィンドウの[チャンネル/接続]タブに表示されます。
4. [設定]タブでは、表示されているチャンネルの 1 つを選択します。
5. 関連するログファイルに記録されるエラーメッセージを設定します。
トレース機能の有効化に関する詳細情報については、「チャンネルのトレース機能の設定」を参照してください。

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ランタイムの開始方法 (ページ 631)

ActiveX コントロールによるチャンネル診断
(ページ 625)

[チャンネル診断]を使用したチャンネルの診断 (ページ 627)

7.3.4.6 チャンネルのトレースファンクションをコンフィグレーションする方法

はじめに

このセクションでは、ランタイムでのチャンネルのトレースファンクションを設定およびアクティベートする方法を説明します。

通信のエラーおよびステータスに関する追加情報は、トレースファイルに記録されます。

トレースファイル

トレースファイルに記録される情報は、通信上の問題の原因を特定するときにホットラインを支援することを目的としています。

したがって、このセクションではファイルのさらなる評価については説明されません。

OPC トレース

次の FAQ で追加情報を取得できます。

- エントリ ID 99412077:
「WinCC OPC チャンネルのトレースをどのように設定して有効にしますか？」 (<https://support.industry.siemens.com/cs/ww/en/view/99412077>)
- エントリ ID 99412263:
「WinCC OPC DA/HDA/A&E サーバーのトレースをどのように設定して有効にしますか？」 (<https://support.industry.siemens.com/cs/ww/en/view/99412263>)

必要条件

- チャンネルを設定して、その下に接続とタグを作成したこと。
- ランタイムを有効にします。

標準フラグ - 概要

フラグ	意味
致命的なエラー	致命的なエラー(例: ユーザーの介入が必要)
エラー	エラー(例: メッセージエラー)
警告	警告(例: チェックサムエラーへの参照)
情報	情報(例: ファンクション呼び出し)

7.3 チャンネル診断

フラグ	意味
成功	実行の成功(例: ファンクション呼び出しの完了)
ユーザーフラグのチェック	[ユーザーフラグ]チェックボックスを有効にします

手順

1. [スタート]メニューから WinCC チャンネル診断を開始します。
2. [設定]タブで、希望のチャンネル選択します。
3. [フラグ]で、トレースファイルに記録するステータスとエラーメッセージを有効にします。
標準フラグの説明は[標準フラグ - 概要]テーブルにあります。
4. [ユーザーフラグ]をトレースファイルに記録する場合は、[ユーザーフラグのチェック]チェックボックスを選択します。
[ユーザーフラグ]の数と重要性はチャンネルによって異なります。
5. 希望する[ユーザーフラグ]のチェックボックスを選択します。
[設定]ボタンまたは[リセット]ボタンをクリックすると、すべての[ユーザーフラグ]を設定またはリセットできます。
6. [トレースファイル]セクションで、[有効化]チェックボックスを選択します。
これにより、このエリアの他のフィールドも有効になります。
7. [最大ファイル]フィールドには、トレースファイルの最大数を入力します。
8. [最大サイズ]フィールドには、個々のトレースファイルのサイズを設定します。
9. チャンネルの既存トレースファイルを上書きする場合、[上書き]フィールドを有効にします(古いものから始め、ファイルの最大数およびファイルサイズに達するまで)。
10. [保存]をクリックすると、設定を保存して変更を有効にすることができます。

下記も参照

コンフィグレーションデータのチェック方法 (ページ 674)

チャンネルおよび接続のチェック方法 (ページ 676)

インターネット: WinCC V7 OPC DA チャンネルのトレース (<https://support.industry.siemens.com/cs/ww/en/view/99412077>)

インターネット: OPC/OPC UA サーバー(WinCC V7 / WinCC Professional)のトレース (<https://support.industry.siemens.com/cs/ww/en/view/99412263>)

7.3.4.7 ランタイムの開始方法

前提条件

ランタイムを有効にする前にスタートアップ画像が定義されていること。

手順

1. エディタ内で開いているすべてのファイルを保存してから閉じます。
2. WinCC Explorer を選択します。
3. ツールバーにある[有効にする]ボタンをクリックするか、または[ファイル]メニューから[有効にする]を選択して、プロジェクトを有効にします。

下記も参照

チャンネル診断によるチャンネルのチェック方法 (ページ 628)

7.4 "システム情報"チャンネルの診断

7.4.1 "システム情報"チャンネル - 診断オプション

"システム情報"チャンネルまたはそのタグの 1 つを診断するために、以下のオプションが利用可能です。

"チャンネル診断"によるチャンネルの診断

"チャンネル診断"は、ランタイム時のチャンネルおよび接続のステータスを問い合わせます。発生したエラーは、"エラーコード"を使用して表示されます。

チャンネルのタグの診断

[タグ管理]ではランタイムに、現在値、品質コードの現在値、およびタグが最後に変更された時刻を照会することができます。

下記も参照

タグのチェック方法 (ページ 636)

チャンネルおよび接続のチェック方法 (ページ 634)

7.4.2 ログファイルのエントリの説明

はじめに

チャンネルにより、エラーおよびステータスの重要な変更がログファイルに記録されます。これらのエントリを使用すると、通信上の問題を分析できます。

ファイルの各エントリには、日付とタイムスタンプ、続いてフラグ名と説明が含まれています。

ログブックエントリの例:

10.03.00 12:00:210.050 INFO Log starting ...

10.03.00 12:00:210.050 INFO | LogFileName :C:\Siemens\WinCC\Diagnose
\\SYSTEM_INFO_01.LOG


```

10.03.00 12:00:210.050 INFO | LogFileCount :3
10.03.00 12:00:210.050 INFO | LogFileSize :1400000
10.03.00 12:00:210.050 INFO | TraceFlags :fa000001
2000-03-10 12:00:21,050 INFO start timer
2000-03-10 12:00:21,360 ERROR Illegal tag type! tag:"Format_0" correct type:"Text Tag 8-
Bit Character Set"!

```

"INFO"フラグのエントリ

メッセージテキスト	説明
Log starting ...	開始メッセージ
LogFileName :C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	パスを付けたログファイル名
LogFileCount : "n"	チャンネルのログファイル数
LogFileSize : "x"	個々のログファイルのサイズ(バイト単位)
TraceFlags :fa000001	チャンネルにより使用されるフラグ(16進数)
start timer	開始メッセージ

"ERROR"フラグのエントリ

メッセージテキスト	説明
Illegal tag type! tag:"tag" correct type:" データタイプ!"	タグの無効なデータタイプ タグ=無効なデータタイプのタグ名 データタイプ=正しいデータタイプ

7.4.3 無効なタグ値の原因の特定方法

7.4.3.1 無効なタグの原因の特定方法

予期しないタグ値がランタイムに発生した場合は、以下を行い、原因を特定します。

1. チャンネルおよび接続をチェックします
2. チャンネルのタグをチェックします

7.4 "システム情報"チャンネルの診断

下記も参照

タグのチェック方法 (ページ 636)

チャンネルおよび接続のチェック方法 (ページ 634)

7.4.3.2 チャンネルおよび接続のチェック方法





はじめに

このセクションでは、ランタイムでの"システム情報"チャンネルとその接続のチェック方法について説明します。

前提条件

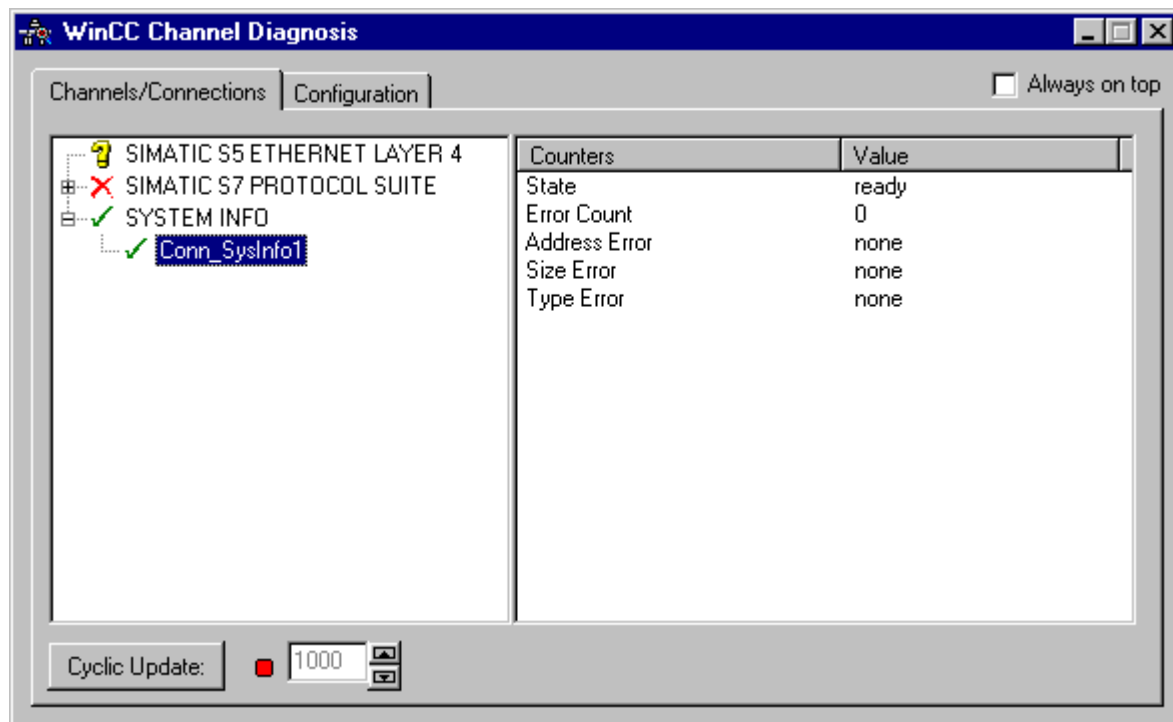
- "システム情報"チャンネルに対する接続およびタグを設定すること。
- WinCC プロジェクトを有効にすること。

ステータスメッセージの概要

アイコン	説明
	チャンネル/接続は無条件で実行可能です。
	チャンネル/接続はいくつかの制限付きで実行可能です。
	チャンネル/接続ステータスに関して可能性のあるステートメントはありません。
	チャンネル/接続は失敗しました。

手順

1. [スタート]メニューから WinCC チャンネル診断を開始します。
2. [チャンネル診断]アプリケーションウィンドウが開きます。インストールされたすべてのチャンネルとその接続に対するステータス情報が、[チャンネル/接続]タブの左に表示されます。



3. "システム情報"という名前のチャンネルとその接続の前にあるアイコンをチェックします。チャンネルと接続のステータスが OK の場合は、それぞれのエントリの前に緑のチェックマークが表示されます。各アイコンの重要性についての詳細は、「ステータスメッセージの概要」の表を参照してください。
4. チャンネル名と接続の前に緑のチェックマークがない場合は、左のウィンドウで接続を選択します。右のウィンドウでは、[アドレスエラー]、[サイズエラー]、および[タイプエラー]のカウンタの値をチェックします。これらの値は検出されたエラーを示します。
5. チャンネル固有のログファイルを確認します。これを行うには、テキストエディタを使用して"Siemens\WinCC\Diagnose"ディレクトリ内のファイルを開きます。"ERROR"フラグが付いている最新のエントリを確認します。詳細については、「ログファイルのエントリの説明」を参照してください。
6. ログファイルを確認してもエラーを特定できない場合は、[トレース]ファンクションを有効にして、カスタマサポートまでお問い合わせください。詳細については、「チャンネルのトレースファンクション設定」を参照してください。

7.4 "システム情報"チャンネルの診断

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ログファイルのエントリの説明 (ページ 632)

タグのチェック方法 (ページ 636)

7.4.3.3 タグのチェック方法

はじめに

外部タグにランタイムに予想される値がない場合、以下の手順を使用してタグをチェックできます。

必要条件

- "システム情報"チャンネルに対する接続およびタグを構成すること。
- WinCC プロジェクトを有効にすること。

手順

1. WinCC エクスプローラ の[タグ管理]で[システム情報]チャンネルを選択します。
2. データウィンドウで、チェックする外部タグを選択します。これを実行するには、タグがテーブルエリアに表示されるまでディレクトリ構造を開きます。
3. マウスポインタをチェックするタグの上へ移動させます。ツールヒントウィンドウが開き、現在のタグ値、品質コード、および値が最後に変更された時刻が表示されます。
4. 品質コードをチェックします。値"80"が表示された場合、タグ値は OK です。他の値についての説明は、「タグの品質コード」に記載されています。
5. 品質コードが"80"でない場合は、[タグ管理]でタグを選択し、ショートカットメニューの[プロパティ]をクリックして[タグプロパティ]ダイアログを開きます。
6. 上限値または下限値、つまり[制限/レポート機能]タブの初期値または置換値に値が設定されているかどうかをチェックします。これらの値は表示に影響することがあります。
7. タグ値が、設定された値の 1 つの影響を受ける場合、プロジェクトを無効にし、制限値または置換値を変更します。

注記

タグ値、品質コードなどは、ランタイム時にだけ表示されます。

下記も参照

タグの品質コード (ページ 683)

7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」

7.5.1 SysDiagControl によるシステム診断

概要

システム診断は、S7-1200 と S7-1500 コントローラの障害とエラーを表示します。

WinCC SysDiagControl を使用すると、WinCC は「SIMATIC S7-1200, S7-1500 Channel」通信チャンネルでの迅速なエラーローカライゼーションの概要を提供します。

コントローラのステータスに関するメッセージから SysDiagControl の診断概要への直接ナビゲーションを設定できます。そこにコントローラエラーの詳細が表示されます。

システム診断表示

WinCC SysDiagControl では、以下のビューを使用できます。

- 診断概要
- 詳細表示
- 診断バッファ表示

システム診断表示では、画面の分割表示もできます。これにより、コントローラおよび関連する詳細情報を一目で確認できます。

上側のエリアに概要が、下側のエリアに診断バッファが表示されます。

下側のエリアに詳細表示が表示されます。


診断概要

診断概要で使用可能な全ての S7-1200/1500 チャンネルが表示されます。

コントローラをダブルクリックすると、詳細表示が開きます。

最初の列のシンボルで、コントローラの現在の状態に関する情報が表示されます。

7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」

						
Diagnostic overview						
Status	Name	Operating mode	Address	Plant designation	Sub-system	
✓	Plant					
✓	S7-1200-St		32*		0	
✓	S7-1200-St		32*		0	
✓	S71500/ET		32*		0	

詳細表示

詳細表示には、選択されたコントローラに関する詳細表示が示されます。

詳細表示のデータが正しいかどうかを点検します。詳細表示のエラーテキストは並べ替えできません。

以下の図に、診断概要と詳細表示の分割表示が示されています。

Diagnostic overview						
Status	Name	Operating mode	Address	Plant designation	Sub-system	
✓	Plant					
✓	S7-1200-St		32*		0	
✓	S7-1200-St		32*		0	
✓	S71500/ET		32*		0	
Property		Value				
>	Status	✓ Good				
>	Name	S71500/ET200MP-Station_1				
>	Operating mode	N/A				
>	Rack					
>	Slot					
>	Type	S71500/ET200MP-Station				
>	Item number					
>	Address	32*				
>	Plant designation					
>	Location identifier					
>	Sub-system	0				
>	Station					
>	Subslot					
>	Sub-address					
>	Software version					
>	Installation					
>	Additional information					
>	Error Text					
>	Manufacturer ID					
>	Hardware version					
>	Profile ID					
>	Specific profile details					
>	I&M data version					
>	Serial number					
>	Revision Counter					

診断バッファ表示

診断バッファ表示には、コントローラの診断バッファからの現在のデータが表示されます。

診断バッファ表示は、診断概要でのみ呼び出すことができます。

診断バッファ表示を更新するには、[更新]ボタンを選択します。



7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」

S71500/ET200MP-Station_1 \ PLC_1 \ Diagnostic buffer						
	No.	Date	Time	Event		
	1	4/3/2015	7:57:01 PM			
	2	4/3/2015	7:54:51 PM			
	3	3/20/2015	7:07:24 PM			
	4	3/20/2015	6:57:44 PM			
	5	3/9/2015	9:50:13 AM			
	6	3/7/2015	4:50:37 PM			
	7	3/7/2015	4:45:22 PM			
	8	3/7/2015	4:45:22 PM			
	9	3/7/2015	4:45:22 PM			
	10	3/7/2015	4:45:17 PM			
	11	3/7/2015	4:44:40 PM			
	12	3/6/2015	3:59:29 PM			
	13	3/2/2015	10:37:09 AM			
	14	3/2/2015	10:39:36 AM			
	15	3/1/2015	3:06:48 AM			
	16	3/1/2015	2:59:38 AM			
	17	2/26/2015	12:18:15 PM			
No.: 1 Date: 4/3/2015 Time: 7:57:01 PM						

システム診断表示のボタン

ボタン	機能
	設定ダイアログを開きます。ここで、SysDiagControl のプロパティを変更できます。
	子デバイスが存在しない場合は、子デバイスまたは詳細表示を開きます。
	親デバイスが存在しない場合は、親デバイスまたは詳細表示を開きます。
	診断概要を開きます。
	診断バッファ表示を開きます 診断概要のみに表示されます。
	診断バッファ表示を更新します。
	表示される診断概要列のユーザー定義並べ替え基準を設定するためのダイアログを開きます。

7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」

ボタン	機能
	表示された値の印刷を開始します。 印刷に使用される印刷ジョブは、設定ダイアログの[全般]タブで定義されます。
	このボタンを使用して、すべてまたは選択したランタイムデータを、「CSV」ファイルにエクスポートします。 [ダイアログの表示]オプションが有効な場合、開いたダイアログにエクスポートの設定が表示され、エクスポートを開始できます。認証ごとに、エクスポート用のファイルとディレクトリを選択できます。 ダイアログが表示されている場合、あらかじめ定義されたファイルへのデータのエクスポートが、すぐに開始されます。

下記も参照

「SIMATIC S7-1200, S7-1500 Channel」チャンネル (ページ 513)

システム診断の構成方法 (ページ 642)

7.5.2 システム診断の構成方法

はじめに

コントローラの故障およびエラーがランタイムでシステム診断の多様な表示に表示されます。

グラフィックデザイナーで、これに対して WinCC SysDiagControl を設定します。

必要条件

- 接続は「SIMATIC S7-1200, S7-1500 Channel」以下の「OMS+」チャンネルユニットで作成されます。
- アラームロギングはサーバーのスタートアップリストで有効です。

- 診断バッファビューに S7-1500 チャンネルのメッセージとテキストを表示するには、次の追加要件が満たされている必要があります。
 - － コントローラの AS メッセージと AS テキストリストが WinCC プロジェクトにロードされます。
 - － アラームロギングの AS テキストリストに対して[使用済み]オプションを選択する必要があります。
 - － メッセージタイプ「Notify_AP」の診断メッセージ用に、アラームロギングで定義された承認原理を設定する必要があります。
診断メッセージは確認応答を必要としないが、「発信」ステータスを持つメッセージタイプに割り付ける必要があります。
- S7-1500 メッセージの自動更新の要件:
 - － コントローラの AS メッセージと AS テキストリストは一度 WinCC プロジェクトにロードされました。
その結果、必要とされるテキストがテキストライブラリに保存されます。

詳細については、[WinCC での作業]>[メッセージシステムのセットアップ]>[メッセージシステムの設定]>[AS メッセージ]を参照してください。

設定手順

1. グラフィックデザイナーのプロセス画像に WinCC SysDiagControl を挿入します。
2. [全般]タブで、SysDiagControl の基本プロパティを設定します。
 - － 診断ウィンドウのプロパティを編集します。
 - － コントロールの全般プロパティ
 - － コントロールのタイムベース
3. [列]タブで、システム診断の表示の列または行として、表示するコントローラデータを指定します。
4. 並べ替えダイアログを使用して、データを並べ替える列を決定します。
以下で、WinCC UserArchiveControl の例を使用して詳細情報を取得できます。
 - － [オプション]>[ユーザーアーカイブ]>[WinCC UserArchiveControl]>[ランタイム時の操作]>[ユーザーアーカイブデータの表示のソート方法]
5. [パラメータ]、[表示]、[マーカ]タブのテーブルの表示とプロパティを設定します。
詳細情報は次の場所で参照できます。
 - － 「WinCC での作業」>「メッセージシステムの設定」>「ランタイム中のメッセージの表示」>「AlarmControl の設定」>「テーブルの表示を設定する方法」
6. それぞれのタブで[テーブル]ウィンドウのツールバーとステータスバーを設定します。
詳細情報は次の場所で参照できます。
 - － 「WinCC での作業」>「メッセージシステムの設定」>「ランタイム中のメッセージの表示」>「AlarmControl の設定」>「ツールバーとステータスバーの設定方法」

7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」

7. WinCC AlarmControl の AS メッセージから直接 WinCC SysDiagControl へジャンプするスクリプトを画像のボタンに設定します。
 - － 画像にボタンを挿入します。
例えばイベントとして、マウスクリックで操作を実行するスクリプトを作成します。
 - － WinCC AlarmControl「AlarmControl_1」と WinCC SysDiagControl「SysDiagControl_1」が同じ画面であるときは、以下のスクリプト例を使用できます。
C の場合:

```
SetPropChar(lpszPictureName,"SysDiagControl_1","NavigateTo",
GetPropChar(lpszPictureName,"AlarmControl_1","DiagnosticsContext"));
```

VBS の場合:

```
ScreenItems("SysDiagControl_1").NavigateTo =
ScreenItems("AlarmControl_1").DiagnosticsContext
```
8. 設定データを保存します。

下記も参照

SysDiagControl によるシステム診断 (ページ 638)

7.6 "SIMATIC S7 Protocol Suite"チャンネルの診断

7.6.1 "SIMATIC S7 Protocol Suite"チャンネル - 診断オプション

エラー検出および"SIMATIC S7 Protocol Suite"チャンネルまたはそのタグの 1 つの診断には次のオプションが利用可能です。

通信プロセッサのコンフィグレーションのチェック

アクセスポイントのチェックに加え、通信プロセッサについてもアプリケーション"PG/PC インターフェース"でテストできます。通信プロセッサは、SIMATIC NET で同じ方法でチェックできます。

接続およびタグのコンフィグレーションのチェック

システムおよび接続パラメータのコンフィグレーションにエラーがあります。タグ値が無効なのは、AS 内のタグの割り付けが間違っていることも原因になっています。

"チャンネル診断"によるチャンネルの診断

"チャンネル診断"は、ランタイム時のチャンネルおよび接続のステータスを問い合わせます。発生したエラーは、"エラーコード"を使用して表示されます。

チャンネルのタグの診断

[タグ管理]ではランタイムに、現在値、品質コードの現在値、およびタグが最後に変更された時刻を照会することができます。

下記も参照

タグのチェック方法 (ページ 656)

チャンネルおよび接続のチェック方法 (ページ 654)

接続およびタグのコンフィグレーションのチェック方法 (ページ 652)

SIMATIC NET での通信プロセッサのチェック (ページ 651)

通信プロセッサの設定確認方法 (ページ 649)

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

7.6.2 ログファイルのエントリの説明

はじめに

チャンネルにより、エラーおよびステータスの重要な変更がログファイルに記録されます。これらのエントリを使用すると、通信上の問題を分析できます。

ファイルの各エントリには、日付とタイムスタンプ、続いてフラグ名と説明が含まれています。

ログブックエントリの例:

```
01.04.99 12:00:240.524 INFO Log starting ...
1999-04-01 12:00:24,524 INFO LogFileName :C:\Siemens\WinCC\Diagnose
\SIMATIC_S7_Protocol_Suite_01.LOG
1999-04-01 12:00:24,524 INFO LogFileCount :3
1999-04-01 12:00:24,524 INFO LogFileSize :1400000
1999-04-01 12:00:24,524 INFO TraceFlags :c4000000
1999-04-01 12:00:24,524 INFO S7 channel DLL started!
1999-04-01 12:00:26,096 ERROR Illegal tag address "nCPU3_1"!
1999-04-01 12:00:27,428 INFO S7DOS release:@(#)TIS-Block Library DLL Version
C5.0.17.3-REL5,0,17,47,3-BASIS
1999-04-01 12:00:27,428 INFO S7DOS version:V5.0 / 0
1999-04-01 12:00:27,428 INFO S7CHN version:V5.0 / Mar 1 1999 / 22:36:40
1999-04-01 12:00:27,428 INFO S7 channel unit "Industrial Ethernet" activated!
1999-04-01 12:00:27,468 ERROR Cannot connect to "CPU_4":Errorcode 0xFFDF 42C2!
1999-04-01 12:00:27,538 INFO S7 channel unit "MPI" activated!
```

"INFO"フラグの最も重要なエントリの記述

メッセージテキスト	説明
LogFileName :C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	パスを付けたログファイル名
LogFileCount : "n"	チャンネルのログファイル数

7.6 "SIMATIC S7 Protocol Suite"チャンネルの診断

メッセージテキスト	説明
LogFileSize : "x"	個々のログファイルのサイズ(バイト単位)
TraceFlags : c4000000	トレースファンクションにより使用されるフラグを16進数で表示します。
S7 channel DLL started!	開始メッセージ
S7 channel DLL terminated!	終了メッセージ
S7 channel unit "unitname" activated!	チャンネルユニットが有効化されました
S7 channel unit "unitname" deactivated!	チャンネルユニットが無効化されました
S7DOS version: versionsstring	バージョン情報
S7CHN version: versionsstring	バージョン情報

"ERROR"フラグの最も重要なエントリの記述

メッセージテキスト	説明
"connectionname"に接続できません。 エラーコード 0xhhhh ffff!	通信エラー WinCC が有効化された後、通信エラーにより AS への接続が直ちに確立できませんでした。過去に少なくとも 1 回エラーなしで接続が確立されている場合、後でエラーが発生すると以下のメッセージが出力されます。 nnn = この接続の切断回数 connectionname = 接続名 hhh = S7DOS / SAPI-S7 (16 進数) での第 1 エラーコード ffff = S7DOS / SAPI-S7 (16 進数) での第 2 エラーコード
"connectionname"に接続できません。 エラーコード 0xhhhh ffff!	通信エラー WinCC が有効化された後、通信エラーにより AS への接続が直ちに確立できませんでした。この接続は過去に少なくとも 1 回エラーなしで確立されています。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

メッセージテキスト	説明
チャンネル API エラー: errorstring	チャンネル API エラー チャンネルにより、エラー文字列 'errorstring' が WinCC Explorer に渡されました。エラーの重要性により、エラー文字列が通知ボックスに表示される場合と表示されない場合があります。エラー文字列の記述については、API エラーテキストを参照してください。
API エラーの最大数になりました - API ログが無効になりました	チャンネル API エラー エラーおよびファンクションにより、API にエラーが周期的に発生する場合があります。これらのエラーメッセージによりログファイルがいっぱいになるのを避けるために、API エラーに対して出力されるのは最大 32 のメッセージです。
ストレージデータを書き込めません! ストレージデータを読み取れません/デフォルトデータを使用します ストレージデータが違法または破壊されています/デフォルトデータを使用します! ストレージデータがありません/デフォルトデータを使用します!	一般のチャンネルエラーメッセージです。
"unitname" ユニット内のデバイス名が "old devicename" から "newdevicename" に変更されました。	初期化メッセージ
logbooksize が最大長になりました - Logbook deactivated	ログファイルが最大長を超えた場合に送信されるメッセージです。 ログ出力は、長さをモニタされます。指定されたサイズに達すると、ログは無効化されます。このメッセージは、メッセージを出力すると最大ファイル長を超える原因になる場合のみ出力されます。ファイル長さがエディタで変更されるか、INI ファイルで最大ファイル長さが縮小された場合、メッセージは出力されません。

7.6.3 無効なタグ値の原因の特定方法

7.6.3.1 無効なタグの原因の特定方法

予期しないタグ値がランタイムに発生した場合は、以下を行い、原因を特定します。

1. 通信プロセッサのコンフィグレーションのチェック
2. SIMATIC NET での通信プロセッサのチェック
3. 接続およびタグのコンフィグレーションのチェック
4. チャンネルおよび接続をチェックします
5. チャンネルのタグをチェックします

下記も参照

タグのチェック方法 (ページ 656)

チャンネルおよび接続のチェック方法 (ページ 654)

接続およびタグのコンフィグレーションのチェック方法 (ページ 652)

SIMATIC NET での通信プロセッサのチェック (ページ 651)

通信プロセッサの設定確認方法 (ページ 649)

7.6.3.2 通信プロセッサの設定確認方法

はじめに

このセクションでは、"PG/PC ポート"プログラムを使用して通信プロセッサを確認する方法について説明します。この例では、PROFIBUS 通信に「CP 5613 A3」タイプのプロセッサを使用します。

必要条件

- CP 5613 A3 をインストールすること。
- 関連した通信ドライバをインストールすること。
- CP 5613 A3 を設定すること。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

手順

1. [コントロールパネル]で[PG/PC ポートの設定]アイコンをクリックします。[PG/PC ポートの設定]ダイアログが開きます。
2. アクセスポイントのエントリをチェックします。CP 5613 A3 がインストールされたときに、Profibus 接続用のアクセスポイント「CP_L2_1:」が自動的に追加されます。このアクセスポイントに対するエントリを選択します。[プロパティ]をクリックして、[プロパティ - CP5613A3.PROFIBUS.1]ダイアログを開きます。



図 7-1 設定例:CP 5623

3. [PROFIBUS]タブのエントリを確認します。
4. [オペレーション状態]タブをクリックします。[テスト]をクリックすると、CP 5613 A3 で機能テストを実行します。テスト結果は、下の出力フィールドに表示されます。テスト結果に応じて、[再起動]ボタンをクリックして、リセットと CP 5613 A3 の完全な再起動を実行することができます。
5. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

接続およびタグのコンフィグレーションのチェック方法 (ページ 652)

SIMATIC NET での通信プロセッサのチェック (ページ 651)

7.6.3.3 SIMATIC NET での通信プロセッサのチェック

はじめに

このセクションでは、SIMATIC NET ソフトウェアで「PC ステーションの設定」プログラムを使用して、通信プロセッサをチェックする方法について説明します。

この例では、「SIMATIC S7 Protocol Suite」チャンネルへの PROFIBUS 通信に、「CP 5613 A3」タイプを使用します。

必要条件

- CP 5613 A3 をインストールすること。
- SIMATIC NET ソフトウェアをインストールすること。
- SIMATIC NET で CP 5613 A3 を設定すること。

手順

1. メニュー項目[PC ステーションの設定]を SIMATIC NET 設定で開きます。
[設定コンソール PC ステーション]ダイアログが開きます。
2. アクセスポイントのエントリをチェックします。ナビゲーションウィンドウで、[アクセスポイント]ディレクトリを選択します。既存のアクセスポイントがデータウィンドウに表示されます。CP 5613 A3 のインストール中、アクセスポイント「CP_L2_1:」は Profibus 接続のために自動的に挿入されます。データウィンドウで、このアクセスポイントを選択します。ショートカットメニューから[プロパティ]メニュー項目を使用して、[CP_L2_1 のプロパティ:]ダイアログを開きます。
3. [割り付けられたインターフェースパラメータ]フィールドのエントリをチェックします。PROFIBUS ネットワークの CP 5613 A3 の場合、エントリ「CP5613A3.PROFIBUS.1」を選択する必要があります。
4. ナビゲーションウィンドウを開き、「Components」ディレクトリを選択し、「CP5613 A3」サブディレクトリを選択します。
5. [ネットワーク診断]ディレクトリを選択します。[テスト]をクリックすると、CP 5613 A3 で機能テストを実行します。結果は出力ウィンドウに表示されます。テスト結果に応じて、「General」ディレクトリの[再起動]をクリックして、リセットと CP 5613 A3 の完全な再起動を実行することができます。
6. [Bus Participants]ディレクトリのリストで、PROFIBUS に接続している参加者のリストをチェックします。この表示に基づき、ユーザー自身のステーションおよび接続している他の参加者の、ファンクションとステータスを特定することができます。
7. ダイアログを閉じます。
8. 通信プロセッサの設定でエラーが検出された場合、修正を行うことができるのは、SIMATIC NET ツールを使用する設定に対してだけです。詳細情報については、SIMATIC NET を参照してください。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

下記も参照

接続およびタグのコンフィグレーションのチェック方法 (ページ 652)

7.6.3.4 接続およびタグのコンフィグレーションのチェック方法

はじめに

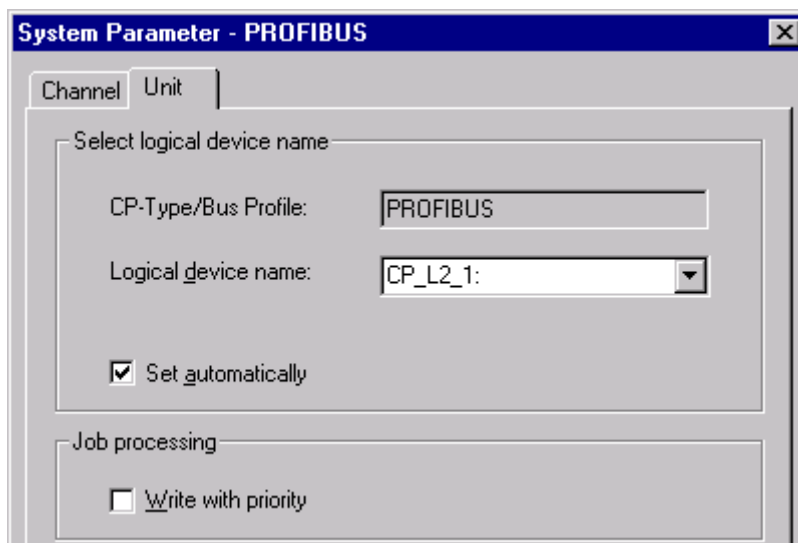
このセクションでは、システムパラメータと、接続およびタグの設定をチェックする方法について説明します。この例では、PROFIBUS 通信に「CP 5613 A3」通信プロセッサを使用します。

必要条件

- CP 5613 A3 をインストールすること。
- 関連した通信ドライバをインストールすること。
- CP 5613 A3 を設定すること。
- "SIMATIC S7 Protocol Suite"チャンネルに対する接続およびタグを設定すること。
- WinCC プロジェクトを有効にすること。

手順

1. WinCC エクスプローラの[タグ管理]で"SIMATIC S7 Protocol Suite"チャンネルを選択します。データウィンドウで、"PROFIBUS"チャンネルユニットを選択します。チャンネルユニットのショートカットメニューで、[システムパラメータ]をクリックします。[システムパラメータ - PROFIBUS]ダイアログが開きます。
2. "ユニット"タブで、[論理デバイス名]フィールドのエントリをチェックします。デフォルトでは、これはアクセスポイント"CP_L2_1:"に設定されています。アクセスポイントは、通信プロセッサの CP 5613 A3 システムへのインストール中に割り付けられます。ダイアログを閉じます。



3. [タグ管理]ナビゲーションウィンドウで、"PROFIBUS"チャンネルユニットを選択します。データウィンドウで、チェックする接続を選択します。ショートカットメニューで[プロパティ]をクリックして、[接続のプロパティ]ダイアログを開きます。
4. [プロパティ]ボタンをクリックして、[接続パラメータ - PROFIBUS]ダイアログを開きます。
5. [接続]タブの設定をチェックします。開いているダイアログを閉じます。
6. ナビゲーションウィンドウで、チェックした接続を選択します。データウィンドウで、チェックするタグを選択します。ショートカットメニューで[プロパティ]をクリックして、[タグのプロパティ]ダイアログを開きます。[タイプ変換]および[データタイプ]フィールドの値を変更します。
7. [選択]ボタンをクリックして、[アドレスプロパティ]ダイアログを開きます。AS のタグをアドレス指定する設定をチェックします。
8. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

チャンネルおよび接続のチェック方法 (ページ 654)

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

7.6.3.5 チャンネルおよび接続のチェック方法





はじめに

このセクションでは、"SIMATIC S7 Protocol Suite"チャンネルおよびその接続をランタイムにチェックする方法について説明します。

必要条件

- WinCC コンピュータに、PROFIBUS 通信用の CP 5613 A3 などの通信プロセッサをインストールすること。
- 関連した通信ドライバをインストールすること。
- CP 5613 A3 を設定すること。
- STEP7 プロジェクトを作成します。
- "SIMATIC S7 Protocol Suite"チャンネルに対する接続およびタグを設定すること。
- WinCC プロジェクトを有効にすること。

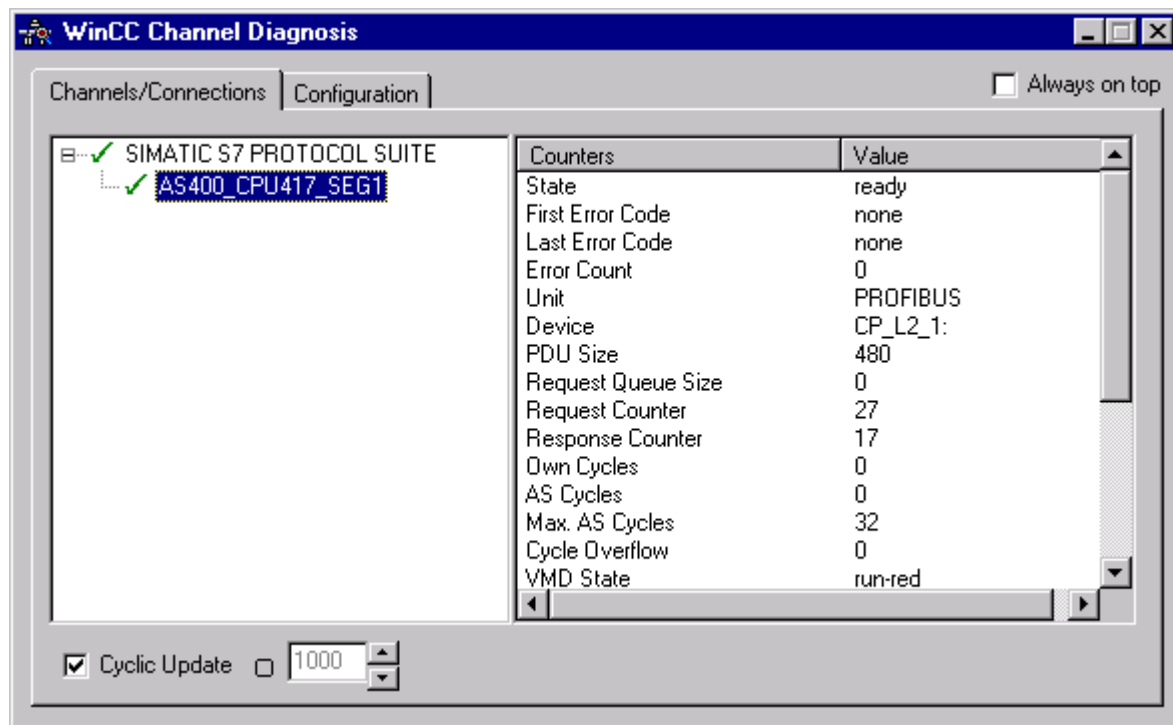
ステータスメッセージの概要

アイコン	説明
	チャンネル/接続は無条件で実行可能です。
	チャンネル/接続はいくつかの制限付きで実行可能です。
	チャンネル/接続ステータスに関して可能性のあるステートメントはありません。
	チャンネル/接続は失敗しました。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

手順

1. [スタート]メニューから WinCC チャンネル診断を開始します。
2. [チャンネル診断]アプリケーションウィンドウが開きます。インストールされたすべてのチャンネルとその接続に対するステータス情報が、[チャンネル/接続]タブの左に表示されます。



3. "SIMATIC S7 Protocol Suite"という名前のチャンネルの前になるアイコンをクリックします。チャンネルと接続が OK の場合は、それぞれのエントリの前に緑のチェックマークが表示されます。各アイコンの重要性についての詳細は、「ステータスメッセージの概要」の表を参照してください。
4. チャンネル名と接続の前に緑のチェックマークがない場合は、左のウィンドウで接続を選択します。右のウィンドウでは、[はじめのエラーコード]および[最新のエラーコード]のカウンタのエントリをチェックします。これらの値は検出されたエラーを示します。表示される値のショートカットメニューをクリックすると[ダイレクトヘルプ]にアクセスできます。
5. [設定]タブで、ログファイルに出力するステータスとエラーメッセージを選択します。これを実行するには、"SIMATIC S7 Protocol Suite"を選択し、エラー表示を設定します。詳細については、「チャンネルのログファイルの設定」を参照してください。
6. チャンネル固有のログファイルを確認します。これを行うには、テキストエディタを使用して"Siemens\WinCC\Diagnose"ディレクトリ内のファイルを開きます。"ERROR"フラグが付いている最新のエントリを確認します。詳細については、「ログファイルのエントリの説明」を参照してください。
7. ログファイルをチェックしてもエラーを特定できない場合は、[トレース]ファンクションを有効にして、カスタマサポートに問い合わせてください。詳細については、「チャンネルのトレースファンクションの設定」を参照してください。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ログファイルのエントリの説明 (ページ 646)

タグのチェック方法 (ページ 656)

7.6.3.6 タグのチェック方法

はじめに

外部タグにランタイムに予想される値がない場合、以下の手順を使用してタグをチェックできます。

"SIMATIC S7 Protocol Suite"チャンネルで、接続固有の内部タグを使用することもできます。この手順は、これらのタグのチェックにも使用できます。

必要条件

- WinCC コンピュータに、MPI 通信用の CP 5613 A3 などの通信モジュールをインストールすること。
- 関連した通信ドライバをインストールすること。
- CP 5613 A3 を設定すること。
- STEP7 プロジェクトを作成します。
- "SIMATIC S7 Protocol Suite"チャンネルに対する接続およびタグを設定すること。
- WinCC プロジェクトを有効にすること。

手順

1. WinCC エクスプローラの[タグ管理]で"SIMATIC S7 Protocol Suite"チャンネルを選択します。
2. データウィンドウで、チェックする外部タグを選択します。これを実行するには、タグがテーブルエリアに表示されるまでディレクトリ構造を開きます。
3. マウスポインタをチェックするタグの上へ移動させます。ツールヒントウィンドウが開き、現在のタグ値、品質コード、および値が最後に変更された時刻が表示されます。
4. 品質コードをチェックします。値"80"が表示された場合、タグ値は OK です。他の値についての説明は、「タグの品質コード」に記載されています。
5. 品質コードが"80"でない場合は、[タグ管理]でタグを選択し、ショートカットメニューの[プロパティ]をクリックして[タグプロパティ]ダイアログを開きます。

7.6 "SIMATIC S7 Protocol Suite" チャンネルの診断

6. 上限値または下限値、つまり[制限/レポート機能]タブの初期値または置換値に値が設定されているかどうかをチェックします。これらの値は表示に影響することがあります。
7. タグ値が、設定された値の 1 つの影響を受ける場合、プロジェクトを無効にし、制限値または置換値を変更します。

注記

ランタイムには、"WinCC チャンネル診断"を使用して、接続固有内部タグの現在値を詳細に表示することができます。メイン接続が選択されると、[カウンタ]列にタグが表示されます。

タグ値、品質コードなどは、ランタイム時にだけ表示されます。

下記も参照

タグの品質コード (ページ 683)

7.7 "SIMATIC S5 Profibus FDL"チャンネルの診断

7.7.1 "SIMATIC S5 PROFIBUS FDL"チャンネルの診断オプション

エラー検出および"SIMATIC S5 PROFIBUS FDL"チャンネルまたはそのタグの 1 つの診断には次のオプションが利用可能です。

通信プロセッサのコンフィグレーションのチェック

アクセスポイントのチェックに加え、通信プロセッサについてもアプリケーション"PG/PC インターフェース"でテストできます。通信プロセッサは、SIMATIC NET で同じ方法でチェックできます。

接続およびタグのコンフィグレーションのチェック

システムおよび接続パラメータのコンフィグレーションにエラーがあります。タグ値が無効なのは、AS 内のタグの割り付けが間違っていることも原因になっています。

"チャンネル診断"によるチャンネルの診断

"チャンネル診断"は、ランタイム時のチャンネルおよび接続のステータスを問い合わせます。発生したエラーは、"エラーコード"を使用して表示されます。

チャンネルのタグの診断

[タグ管理]ではランタイムに、現在値、品質コードの現在値、およびタグが最後に変更された時刻を照会することができます。

下記も参照

タグのチェック方法 (ページ 667)

チャンネルおよび接続のチェック方法 (ページ 665)

接続およびタグのコンフィグレーションのチェック方法 (ページ 663)

SIMATIC NET での通信プロセッサのチェック (ページ 662)

通信プロセッサのコンフィグレーションのチェック (ページ 660)

7.7.2 ログファイルのエントリの説明

はじめに

チャンネルにより、エラーおよびステータスの重要な変更がログファイルに記録されます。これらのエントリを使用すると、通信上の問題を分析できます。

ファイルの各エントリには、日付とタイムスタンプ、続いてフラグ名と説明が含まれています。

ログブックエントリの例:

```
03.05.00 14:43:480.733 INFO Log starting ...
03.05.00 14:43:480.733 INFO | LogFileName :d:\Siemens\WinCC\Diagnose
\SIMATIC_S5_PROFIBUS_FDL_01.LOG
03.05.00 14:43:480.733 INFO | LogFileCount :3
03.05.00 14:43:480.733 INFO | LogFileSize :1400000
03.05.00 14:43:480.733 INFO | TraceFlags :fa017fff
```

"INFO"フラグのエントリの説明

メッセージテキスト	説明
Log starting ...	開始メッセージ
LogFileName :C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	パスを付けたログファイル名
LogFileCount : "n"	チャンネルのログファイル数
LogFileSize : "x"	個々のログファイルのサイズ(バイト単位)
TraceFlags :fa017fff	トレースファンクションにより使用されるフラグを 16 進数で表示します。

7.7.3 無効なタグ値の原因の特定方法

7.7.3.1 無効なタグの原因の特定方法

予期しないタグ値がランタイムに発生した場合は、以下を行い、原因を特定します。

1. 通信プロセッサのコンフィグレーションのチェック
2. SIMATIC NET での通信プロセッサのチェック
3. 接続およびタグのコンフィグレーションのチェック
4. チャンネルおよび接続をチェックします
5. チャンネルのタグをチェックします

下記も参照

接続およびタグのコンフィグレーションのチェック方法 (ページ 663)

タグのチェック方法 (ページ 667)

チャンネルおよび接続のチェック方法 (ページ 665)

SIMATIC NET での通信プロセッサのチェック (ページ 662)

通信プロセッサのコンフィグレーションのチェック (ページ 660)

7.7.3.2 通信プロセッサのコンフィグレーションのチェック

はじめに

このセクションでは、"PG/PC ポート"プログラムを使用して通信プロセッサを確認する方法について説明します。この例では、PROFIBUS 通信に「CP 5613 A3」タイプのプロセッサを使用します。

必要条件

- CP 5613 A3 をインストールすること。
- 関連した通信ドライバをインストールすること。
- CP 5613 A3 を設定すること。

手順

1. [コントロールパネル]で[PG/PC ポートの設定]アイコンをクリックします。[PG/PC ポートの設定]ダイアログが開きます。
2. アクセスポイントのエントリをチェックします。CP 5613 A3 がインストールされたときに、Profibus 接続用のアクセスポイント「CP_L2_1:/SCP」が自動的に追加されます。このアクセスポイントに対するエントリを選択します。[プロパティ]をクリックして、[プロパティ-CP5613A3.PROFIBUS.1]ダイアログを開きます。



図 7-2 設定例:CP 5623

3. [PROFIBUS]タブのエントリを確認します。
4. [オペレーション状態]タブをクリックします。[テスト]をクリックすると、CP 5613 A3 で機能テストを実行します。テスト結果は、下の出力フィールドに表示されます。テスト結果に応じて、[再起動]ボタンをクリックして、リセットと CP 5613 A3 の完全な再起動を実行することができます。
5. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

接続およびタグのコンフィグレーションのチェック方法 (ページ 663)

SIMATIC NET での通信プロセッサのチェック (ページ 662)

7.7.3.3 SIMATIC NET での通信プロセッサのチェック

はじめに

このセクションでは、SIMATIC NET ソフトウェアで「PC ステーションの設定」プログラムを使用して、通信プロセッサをチェックする方法について説明します。

この例では、「SIMATIC S5 PROFIBUS FDL」チャンネルへの PROFIBUS 通信に、「CP 5613 A3」タイプを使用します。

必要条件

- CP 5613 A3 をインストールすること。
- SIMATIC NET ソフトウェアをインストールすること。
- SIMATIC NET で CP 5613 A3 を設定すること。

手順

1. メニュー項目[PC ステーションの設定]を SIMATIC NET 設定で開きます。
[設定コンソール PC ステーション]ダイアログが開きます。
2. アクセスポイントのエントリをチェックします。ナビゲーションウィンドウで、[アクセスポイント]ディレクトリを選択します。既存のアクセスポイントがデータウィンドウに表示されます。CP 5613 A3 のインストール中、アクセスポイント「CP_L2_1:」は Profibus 接続のために自動的に挿入されます。データウィンドウで、このアクセスポイントを選択します。ショートカットメニューから[プロパティ]メニュー項目を使用して、[CP_L2_1 のプロパティ:]ダイアログを開きます。
3. [割り付けられたインターフェースパラメータ]フィールドのエントリをチェックします。PROFIBUS ネットワークの CP 5613 A3 の場合、エントリ「CP5613A3.PROFIBUS.1」を選択する必要があります。
4. ナビゲーションウィンドウを開き、「Components」ディレクトリを選択し、「CP5613 A3」サブディレクトリを選択します。
5. [ネットワーク診断]ディレクトリを選択します。[テスト]をクリックすると、CP 5613 A3 で機能テストを実行します。結果は出力ウィンドウに表示されます。テスト結果に応じて、「General」ディレクトリの[再起動]をクリックして、リセットと CP 5613 A3 の完全な再起動を実行することができます。
6. [Bus Participants]ディレクトリのリストで、PROFIBUS に接続している参加者のリストをチェックします。この表示に基づき、ユーザー自身のステーションおよび接続している他の参加者の、ファンクションとステータスを特定することができます。
7. ダイアログを閉じます。
8. 通信プロセッサの設定でエラーが検出された場合、修正を行うことができるのは、SIMATIC NET ツールを使用する設定に対してだけです。詳細情報については、SIMATIC NET を参照してください。

下記も参照

接続およびタグのコンフィグレーションのチェック方法 (ページ 663)

7.7.3.4 接続およびタグのコンフィグレーションのチェック方法

はじめに

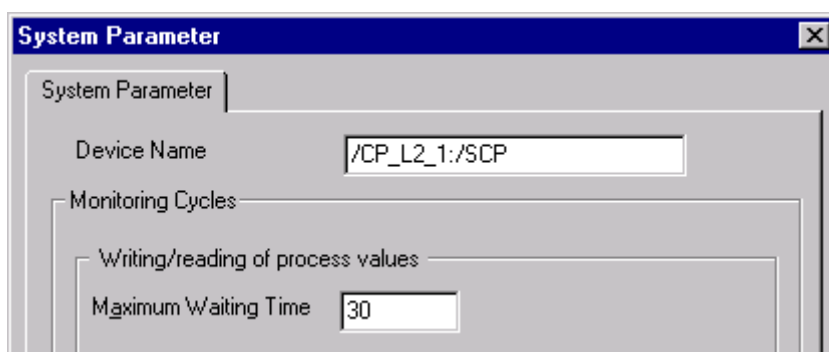
このセクションでは、システムパラメータと、接続およびタグの設定をチェックする方法について説明します。

必要条件

- CP 5613 A3 をインストールすること。
- 通信ドライバをインストールします。
- CP 5613 A3 を設定すること。
- "SIMATIC S5 PROFIBUS FDL"チャンネルに対する接続およびタグを設定します。
- WinCC プロジェクトを有効にすること。

手順

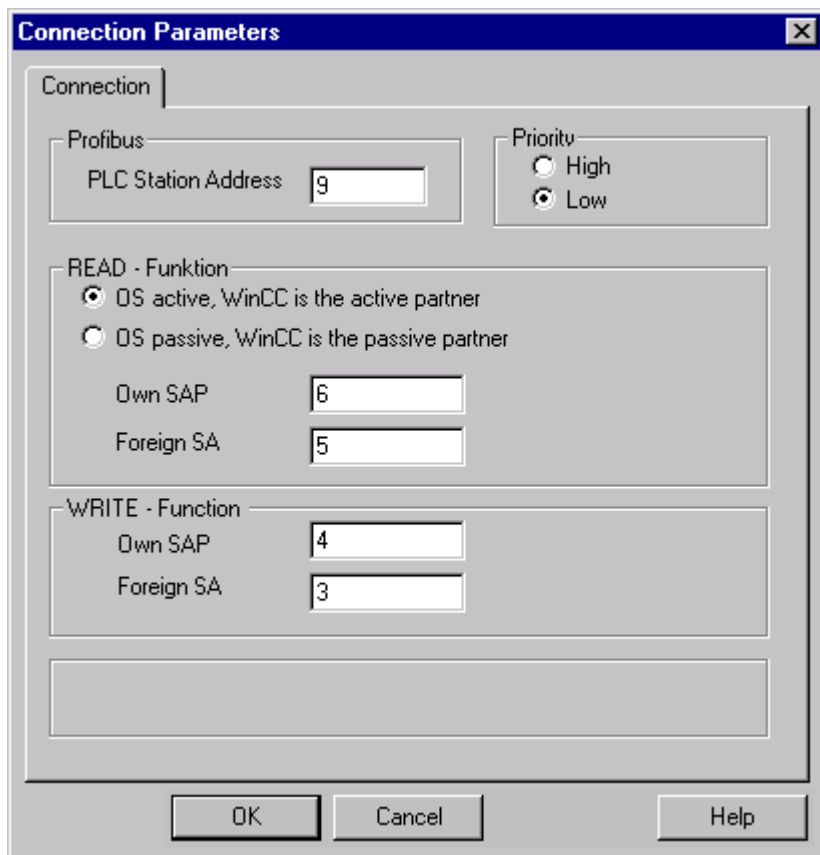
1. WinCC エクスプローラのナビゲーションウィンドウで、[SIMATIC S5 PROFIBUS FDL]アイコンの前にある"プラス符号"をクリックします。[FDL(CP5412/A2-1)]アイコンのショートカットメニューで、[システムパラメータ]をクリックします。[システムパラメータ]ダイアログが開きます。



2. [デバイス名]フィールドのエントリをチェックします。デフォルトでは、アクセスポイント"CP_L2_1:/SCP"が設定されています。アクセスポイントは、通信プロセッサの CP 5613 A3 システムへのインストール中に割り付けられます。ダイアログを閉じます。
3. [FDL(CP5412/A2-1)]アイコンの前にある"プラス符号"をクリックします。テストするタグのショートカットメニューで、[プロパティ]エントリを選択します。[接続プロパティ]ダイアログが開きます。

7.7 "SIMATIC S5 Profibus FDL" チャンネルの診断

4. [接続プロパティ]ダイアログで、[プロパティ]ボタンをクリックします。[接続パラメータ]ダイアログが開きます。



5. [接続]タブの設定をチェックします。開いているダイアログを閉じます。
6. 接続アイコンの前にある"プラス符号"をクリックします。テストするタグのショートカットメニューで、[プロパティ]エントリをクリックします。[タグのプロパティ]ダイアログが開きます。[タイプ変換]および[データタイプ]フィールドのエントリをチェックします。
7. [タグプロパティ]ダイアログで[選択]ボタンをクリックします。[アドレスプロパティ]ダイアログが開きます。設定をチェックします。
8. [OK]をクリックして、開いているダイアログをすべて閉じます。

下記も参照

チャンネルおよび接続のチェック方法 (ページ 665)

7.7.3.5 チャンネルおよび接続のチェック方法





はじめに

このセクションでは、"SIMATIC S5 PROFIBUS FDL"チャンネルおよびその接続をランタイムにチェックする方法について説明します。

必要条件

- CP 5613 A3 をインストールすること。
- 通信ドライバをインストールします。
- CP 5613 A3 を設定すること。
- STEP5 プロジェクトを作成します。
- "SIMATIC S5 PROFIBUS FDL"チャンネルに対する接続およびタグを設定します。
- WinCC プロジェクトを有効にすること。

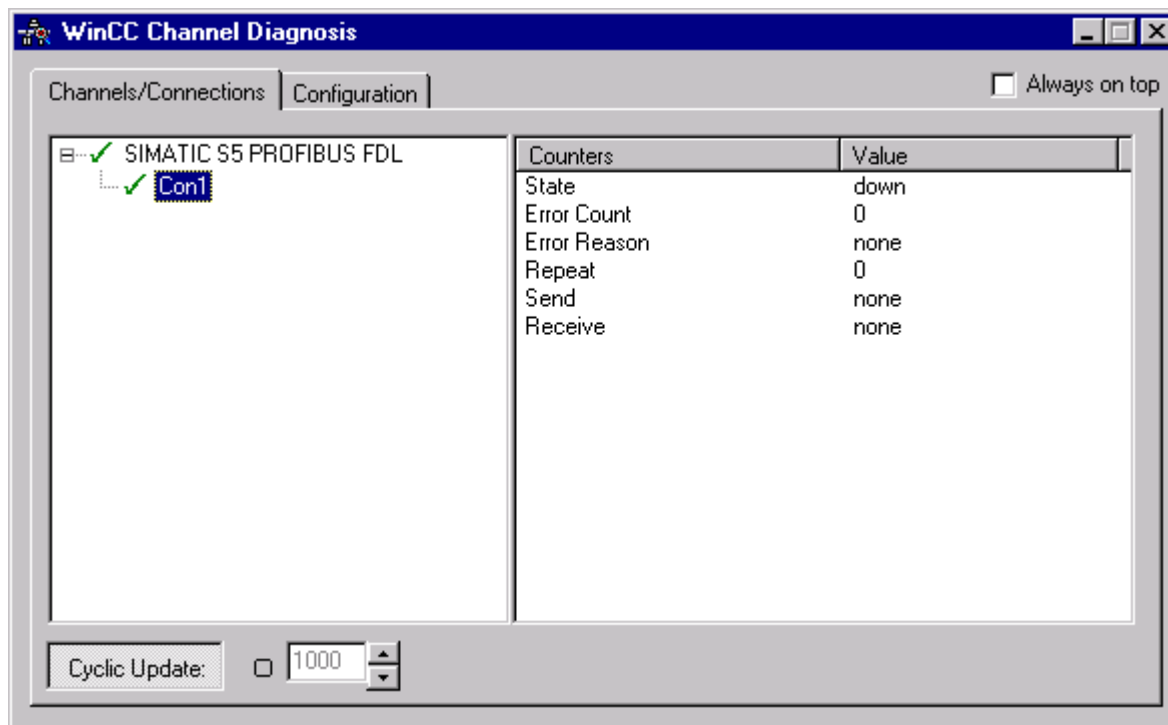
ステータスメッセージの概要

アイコン	説明
	チャンネル/接続は無条件で実行可能です。
	チャンネル/接続はいくつかの制限付きで実行可能です。
	チャンネル/接続ステータスに関して可能性のあるステートメントはありません。
	チャンネル/接続は失敗しました。

7.7 "SIMATIC S5 Profibus FDL"チャンネルの診断

手順

1. [スタート]メニューから WinCC チャンネル診断を開始します。
2. [チャンネル診断]アプリケーションウィンドウが開きます。インストールされたすべてのチャンネルとその接続に対するステータス情報が、[チャンネル/接続]タブの左に表示されます。



3. "SIMATIC S5 PROFIBUS FDL"という名前のチャンネルと接続の前にあるアイコンをチェックします。チャンネルと接続のステータスが OK の場合は、それぞれのエントリの前に緑のチェックマークが表示されます。各アイコンの重要性についての詳細は、「ステータスメッセージの概要」の表を参照してください。
4. チャンネル名と接続の前に緑のチェックマークがない場合は、左のウィンドウで接続を選択します。右のウィンドウでは、[状態]、[エラーカウント]、[エラー理由]、[送信]、および[受信]のカウンタをチェックします。これらの値は検出されたエラーを示します。
5. チャンネル固有のログファイルを確認します。これを行うには、テキストエディタを使用して"Siemens\WinCC\Diagnose"ディレクトリ内のファイルを開きます。"ERROR"フラグが付いている最新のエントリを確認します。詳細については、「ログファイルのエントリの説明」を参照してください。
6. ログファイルをチェックしてもエラーを特定できない場合は、[トレース]ファンクションを有効にして、カスタマサポートに問い合わせてください。詳細については、「チャンネルのトレースファンクションの設定」を参照してください。

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ログファイルのエントリの説明 (ページ 659)

タグのチェック方法 (ページ 667)

7.7.3.6 タグのチェック方法

はじめに

外部タグにランタイムに予想される値がない場合、以下の手順を使用してタグをチェックできます。

必要条件

- CP 5613 A3 をインストールすること。
- 通信ドライバをインストールします。
- CP 5613 A3 を設定すること。
- STEP5 プロジェクトを作成します。
- "SIMATIC S5 PROFIBUS FDL"チャンネルに対する接続およびタグを設定します。
- WinCC プロジェクトを有効にすること。

手順

1. WinCC エクスプローラのタグ管理で"SIMATIC S5 PROFIBUS FDL"チャンネルを選択します。
2. データウィンドウで、チェックする外部タグを選択します。これを実行するには、タグがテーブルエリアに表示されるまでディレクトリ構造を開きます。
3. マウスポインタをチェックするタグの上へ移動させます。ツールヒントウィンドウが開き、現在のタグ値、品質コード、および値が最後に変更された時刻が表示されます。
4. 品質コードをチェックします。値"80"が表示された場合、タグ値は OK です。他の値についての説明は、「タグの品質コード」に記載されています。
5. 品質コードが"80"でない場合は、[タグ管理]でタグを選択し、ショートカットメニューの[プロパティ]をクリックして[タグプロパティ]ダイアログを開きます。
6. 上限値または下限値、つまり[制限/レポート機能]タブの初期値または置換値に値が設定されているかどうかをチェックします。これらの値は表示に影響することがあります。
7. タグ値が、設定された値の 1 つの影響を受ける場合、プロジェクトを無効にし、制限値または置換値を変更します。

7.7 "SIMATIC S5 Profibus FDL"チャンネルの診断

注記

タグ値、品質コードなどは、ランタイム時にだけ表示されます。

下記も参照

タグの品質コード (ページ 683)

7.8 "OPC"チャンネルの診断

7.8.1 "OPC"チャンネルの診断のための機能

エラー検出および"OPC"チャンネルまたはそのタグの1つの診断には次の可能性があります。

接続およびタグのコンフィグレーションのチェック

システムおよび接続パラメータのコンフィグレーションにエラーがあります。タグ値が無効なのは、AS内のタグの割り付けが間違っていることも原因になっています。

"チャンネル診断"によるチャンネルの診断

"チャンネル診断"は、ランタイム時のチャンネルおよび接続のステータスを問い合わせます。発生したエラーは、"エラーコード"を使用して表示されます。

チャンネルのタグの診断

[タグ管理]ではランタイムに、現在値、品質コードの現在値、およびタグが最後に変更された時刻を照会することができます。

下記も参照

タグのチェック方法 (ページ 678)

チャンネルおよび接続のチェック方法 (ページ 676)

コンフィグレーションデータのチェック方法 (ページ 674)

7.8 "OPC"チャンネルの診断

7.8.2 ログファイルのエントリの説明

7.8.2.1 ログファイルのエントリの説明

はじめに

チャンネルにより、エラーおよびステータスの重要な変更がログファイルに記録されます。以下のセクションでは、最も重要なエントリのみを説明します。これらのエントリを使用すると、通信上の問題を分析できます。

2つのタイプのエントリを区別する必要があります。

- INFO
- ERROR

エントリ構造

Date/Time Stamp	Flag Name	Description
-----------------	-----------	-------------

ログエントリの例

```
2000-03-24 10:43:18,756 INFO Log starting ...
2000-03-24 10:43:18,756 INFO | LogFileName :C:\Siemens\WinCC\Diagnose\OPC.LOG
2000-03-24 10:43:18,756 INFO | LogFileCount :3
2000-03-24 10:43:18,756 INFO | LogFileSize :1400000
2000-03-24 10:43:18,756 INFO | TraceFlags :fa000007
000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
2000-03-23 10:46:18,756 INFO Process detached at 2000-03-24 10:46:18,746 UTC
2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR
800706ba
2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit
#1)]![OPC_No_Machine]:CoCreateInstance for server "OPCServer.WinCC" on machine
OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba -
RPC_S_SERVER_UNAVAILABLE (RPC サーバーを使用できません))")
```

下記も参照

"ERROR"フラグのエントリ (ページ 672)

"INFO"フラグのエントリ (ページ 671)

7.8.2.2 "INFO"フラグのエントリ

はじめに

ファイルの各エントリには、日付とタイムスタンプ、続いてフラグ名と説明が含まれています。

Date/Time Stamp	Flag Name	Description
-----------------	-----------	-------------

ログエントリの例

2000-03-24 10:43:18,756 INFO Log starting ...
 2000-03-24 10:43:18,756 INFO | LogFileName :C:\Siemens\WinCC\Diagnose\OPC.LOG
 2000-03-24 10:43:18,756 INFO | LogFileCount :3
 2000-03-24 10:43:18,756 INFO | LogFileSize :1400000
 2000-03-24 10:43:18,756 INFO | TraceFlags :fa000007
 000-03-24 10:43:18,756 INFO Process attached at 2000-03-24 09:43:18,746 UTC
 2000-03-23 10:46:18,756 INFO Process detached at 2000-03-24 10:46:18,746 UTC

最も重要なログエントリの説明

メッセージテキスト	説明
Log starting ...	開始メッセージ
LogFileName :C:\ Siemens\ WinCC\ Diagnose\ "channel_name".LOG	パスを付けたログファイル名
LogFileCount : "n"	チャンネルのログファイル数
LogFileSize : "x"	個々のログファイルのサイズ(バイト単位)
TraceFlags :fa000007	トレースファンクションにより使用されるフラグを 16 進数で表示します。

7.8 "OPC"チャンネルの診断

メッセージテキスト	説明
Process attached at 2000-03-24 09:43:18,746 UTC	チャンネルは、WinCC データマネージャによりロードされました。
Process detached at 2000-03-24 10:46:18,746 UTC	チャンネルは、WinCC データマネージャによりアンロードされました。
IOPCChnShutdown::ShutdownRequest was called...Reason:system going down" IOPCChnShutdown::ShutdownRequest	WinCC OPC サーバーの WinCC プロジェクトが無効化されました。WinCC OPC クライアントは WinCC OPC サーバーから接続を切断するよう要求されています。

7.8.2.3 "ERROR"フラグのエントリ

はじめに

ファイルの各エントリには、日付とタイムスタンプ、続いてフラグ名と説明が含まれています。"ERROR"フラグの場合、記述は、メッセージテキスト、エラーコード、およびエラーメッセージのテキストで構成されます。エラーメッセージに対するテキストのないエラーコードもあります。

Date/Time Stamp	Flag Name	Description Message text + error code + error message text
-----------------	-----------	---

ログエントリの例

2000-03-27 13:22:43,390 ERROR ..FOPCData::InitOPC CoCreateInstanceEx- ERROR 800706ba

2000-03-27 13:22:43,390 ERROR - ChannelUnit::SysMessage("[OPC Groups (OPCHN Unit #1)]![OPC_No_Machine]: CoCreateInstance for server \"OPCServer.WinCC\" on machine OPC_No_Machine failed, Error=800706ba (HRESULT = 800706ba - RPC_S_SERVER_UNAVAILABLE (RPC サーバーを使用できません))")

最も重要なログエントリの説明

エラーコード	エラーメッセージテキスト	可能性のある原因
c0040004	"canonicalDatatype"と "requestedDatatype"の間 の変換は、サーバでサポー トされていません。	OPC サーバー上の WinCC タグにアクセスでき ませんでした。 変換は可能ですが、失敗しました。 WinCC タグがサーバー上にないか、設定され たデータタイプが一致しません。
c0040007	サーバーの名前空間に名前 がありません。	サーバーの名前空間に存在しないタグ名で OPC クライアントがアクセスされると、サー バーは常にエラーコードを返します。 例：参照、タグの読み込み、タグの書き込 み、登録へのタグ挿入。
00000001	AddItems	故障した OPC サーバー上の WinCC タグにアク セスしました。 WinCC タグがサーバー上に ないか、設定されたデータタイプが一致しませ ん。 データタイプ WinCC タグ OPC サーバー=デー タタイプ WinCC タグ OPC クライアント。
80004005	サーバー名を解決できませ んでした	WinCC OPC サーバーとして使用されるコンピ ュータが、ネットワークで使用できません。 "OPC"チャンネルによってアクセスされた WinCC OPC サーバーが使用できませんでした。
80040154	クラスが登録されていま せん	WinCC OPC サーバーがシステムに適切に登録 されていません。 WinCC OPC サーバーの WinCC プロジェクトが 有効になっていません。
80070057	パラメータが間違ってい ます	WinCC タグが OPC サーバー上にないか、コン フィグレーションされたデータタイプが一致し ません。
800706ba	RPC サーバーが使用でき ません	起動すべき OPC サーバー上のコンピュータ を、ネットワークで見つけることができません でした。

7.8 "OPC"チャンネルの診断

7.8.3 無効なタグ値の原因の特定方法

7.8.3.1 無効なタグの原因の特定方法

予期しないタグ値がランタイムに発生した場合は、以下を行い、原因を特定します。

1. コンフィグレーションのデータのチェック
2. 接続のチェック
3. チャンネルのタグをチェックします

下記も参照

タグのチェック方法 (ページ 678)

チャンネルおよび接続のチェック方法 (ページ 676)

コンフィグレーションデータのチェック方法 (ページ 674)

7.8.3.2 コンフィグレーションデータのチェック方法

必要条件

- WinCC プロジェクトを備えた WinCC OPC クライアントとしてのコンピュータ
- 「OPC」チャンネルを OPC クライアントの WinCC プロジェクトに統合する必要があります。
- OPC サーバーの WinCC プロジェクトで WinCC タグを設定します。
- OPC クライアントで、接続と作成されたサーバータグと通信する WinCC タグを設定します。
- OPC サーバーと OPC クライアントで、ランタイムを有効にします。

手順

1. OPC クライアントで、WinCC エクスプローラナビゲーションウィンドウの[OPC]アイコンの前にあるプラス記号をクリックします。
[OPC グループ(OPCHN Unit#1)]アイコンの前のプラス記号をクリックします。
2. テストする接続のショートカットメニューで、[プロパティ]エントリを選択します。
[接続プロパティ]ダイアログが開きます。

3. [OPC 接続]タブで、[OPC サーバー名]フィールドの OPC サーバーの ProgID を確認します。
 - － WinCC V5.0 以降のサーバーに接続する際には、「OPCServer.WinCC」を入力する必要があります。
 - － WinCC V4.x 以降のサーバーに接続する際には、「OE.Grouops」を入力する必要があります。
4. OPC サーバーとして使用されているコンピュータ名が[別のコンピュータでサーバーを実行]フィールドに入力されているか確認してください。
5. OPC サーバーへの接続をテストするためには、[サーバーのテスト]をクリックします。ダイアログを閉じます。
6. 接続アイコンの前にあるプラス記号をクリックします。
7. テストするタグのショートカットメニューで、[プロパティ]エントリを選択します。
[タグのプロパティ]ダイアログが開きます。
 - － このタグには OPC サーバー上のタグと同じ[データタイプ]を入力する必要があります。
8. [タグプロパティ]ダイアログで[選択]ボタンをクリックします。
[アドレスプロパティ]ダイアログが開きます。
9. [項目名]および[データタイプ]フィールドのエントリをチェックします。
 - － [アイテム名]は OPC サーバー上のデータタイプと一致する必要があります。
 - － [データタイプ]は、OPC サーバー上のタグのデータタイプと一致する必要があります。
10. チャンネル固有のログファイルを確認します。
これを行うには、テキストエディタを使用して"Siemens\WinCC\Diagnose"ディレクトリ内のファイルを開きます。
「ERROR」フラグが付いている最新のエントリを確認します。
このトピックの詳細については、「ログファイルのエントリの説明 (ページ 670)」を参照してください。
11. ログファイルをチェックしてもエラーを特定できない場合は、トレースファンクションを有効にして、カスタマサポートに問い合わせてください。
詳細情報は、「チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)」を参照してください。

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ログファイルのエントリの説明 (ページ 670)

チャンネルおよび接続のチェック方法 (ページ 676)

7.8 "OPC"チャンネルの診断

7.8.3.3 チャンネルおよび接続のチェック方法





はじめに

このセクションでは、ランタイムでの"OPC"チャンネルとその接続のチェック方法について説明します。

必要条件

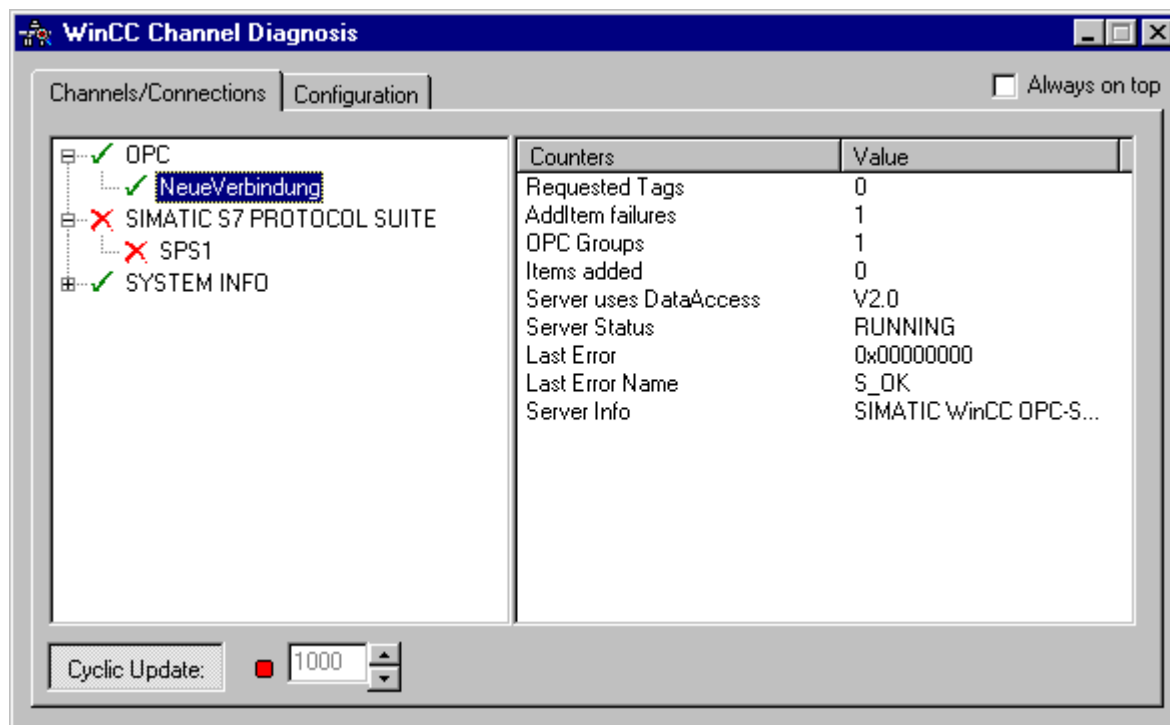
- WinCC プロジェクトを備えた WinCC OPC クライアントとしてのコンピュータ
- 「OPC」チャンネルを OPC クライアントの WinCC プロジェクトに統合する必要があります。
- OPC サーバーの WinCC プロジェクトで WinCC タグを設定します。
- OPC クライアントで、接続と作成されたサーバータグと通信する WinCC タグを設定します。
- OPC サーバーと OPC クライアントで、WinCC プロジェクトを有効にします。

ステータスメッセージの概要

アイコン	説明
	チャンネル/接続は無条件で実行可能です。
	チャンネル/接続はいくつかの制限付きで実行可能です。
	チャンネル/接続ステータスに関して可能性のあるステートメントはありません。
	チャンネル/接続は失敗しました。

手順

1. [スタート]メニューから WinCC チャンネル診断を開始します。
[チャンネル診断]アプリケーションウィンドウが開きます。
インストールされたすべてのチャンネルとその接続に対するステータス情報が、[チャンネル/接続]タブの左に表示されます。



2. OPC 接続の前にあるアイコンをチェックします。
接続のステータスが OK の場合は、それぞれのエントリの前に緑のチェックマークが表示されます。
各アイコンの重要性についての詳細は、「ステータスメッセージの概要」の表を参照してください。
3. 接続名の前に緑のチェックマークがない場合は、左のウィンドウで接続を選択します。
4. 右のウィンドウでは、[AddItemFailures]、[サーバーステータス]、[最新のエラー]、および [最新のエラー名]のカウンタをチェックします。
これらの値は検出されたエラーを示します。
5. チャンネル固有のログファイルを確認します。
これを行うには、テキストエディタを使用して"Siemens\WinCC\Diagnose"ディレクトリ内のファイルを開きます。
「ERROR」フラグが付いている最新のエントリを確認します。
このトピックの詳細については、「ログファイルのエントリの説明 (ページ 670)」を参照してください。
6. ログファイルをチェックしてもエラーを特定できない場合は、トレースファンクションを有効にして、カスタマサポートに問い合わせてください。
詳細情報は、「チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)」を参照してください。

7.8 "OPC"チャンネルの診断

下記も参照

チャンネルのトレースファンクションをコンフィグレーションする方法 (ページ 629)

ログファイルのエントリの説明 (ページ 670)

タグのチェック方法 (ページ 678)

7.8.3.4 タグのチェック方法

はじめに

外部タグにランタイムに予想される値がない場合、以下の手順を使用してタグをチェックできます。

必要条件

- WinCC プロジェクトのある WinCC OPC クライアントとしてのコンピュータ。
- "OPC"チャンネルは、OPC クライアントの WinCC プロジェクトに統合する必要があります。
- OPC サーバーの WinCC プロジェクトに WinCC タグを設定します。
- OPC クライアント上で接続と WinCC タグを設定します。これは作成されたサーバータグと通信します。
- OPC サーバーおよびクライアントで、WinCC プロジェクトを有効にします。

手順

1. WinCC エクスプローラ のタグ管理で"OPC"チャンネルを選択します。
2. データウィンドウで、チェックする外部タグを選択します。これを実行するには、タグがテーブルエリアに表示されるまでディレクトリ構造を開きます。
3. マウスポインタをチェックするタグの上へ移動させます。現在のタグ値、品質値、および最新の変更時刻の表示があるツールヒントウィンドウが開きます。
4. 品質値をチェックします。値"C0"が表示された場合、タグ値は OK です。他の値についての説明は、「タグの品質コード」に記載されています。
5. 品質コードが"C0"でない場合は、[タグ管理]でタグを選択し、ショートカットメニューの[プロパティ]をクリックして[タグプロパティ]ダイアログを開きます。
6. 上限値または下限値、つまり[制限/レポート機能]タブの初期値または置換値に値が設定されているかどうかをチェックします。これらの値は表示に影響することがあります。
7. タグ値が、設定された値の 1 つの影響を受ける場合、プロジェクトを無効にし、制限値または置換値を変更します。

注記

タグ値、品質コードなどは、ランタイム時にだけ表示されます。

下記も参照

タグの品質コード (ページ 683)

7.9 タグの品質

7.9.1 タグの品質

はじめに

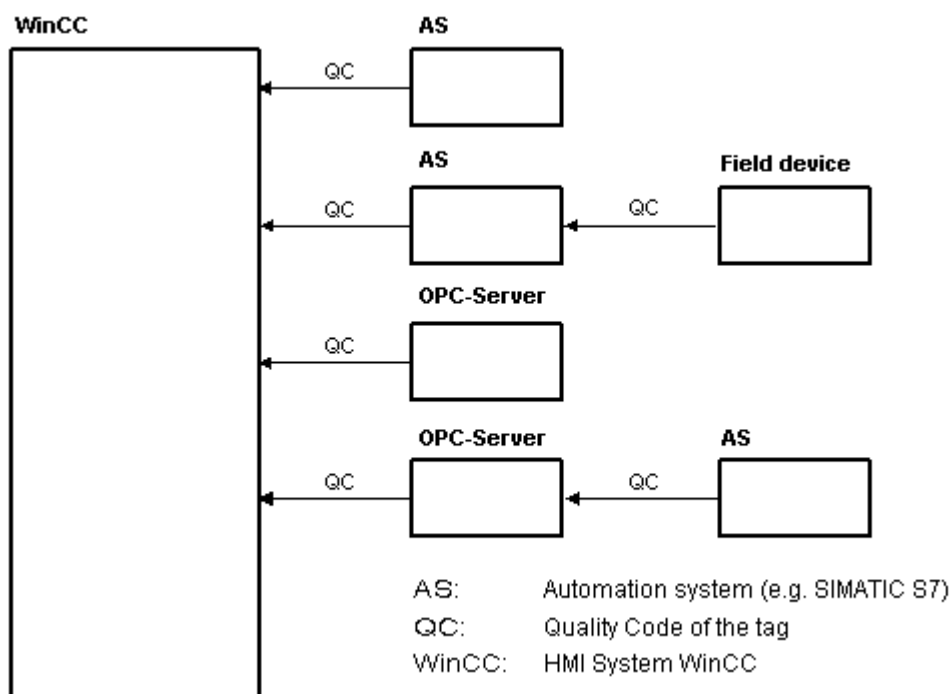
WinCC には 2 つの品質インジケータがあります。このインジケータによってタグの品質を評価できます。この 2 つのインジケータは、タグステータスと品質コードです。

タグステータスは WinCC に形成され、OS 内のコンフィグレーション設定の品質を知らせます。タグステータスはさらに WinCC の通信相手に対する接続ステータスを知らせます。この相手には、自動化システムやサーバーコンピュータが可能です。

品質コードには、タグステータスと同じ情報が含まれます。この情報に加えて、品質ステータスには、タグを評価または処理するパートナーの品質が記述されています。可能なパートナーは以下のとおりです。

- オートメーションシステム
- 現場デバイスを備えるオートメーションシステム
- OPC サーバー
- 下位のオートメーションシステムを備える OPC サーバー

その中で、品質コードは処理経路内を転送されます。処理経路内の 1 点で、複数の品質コードがタグのために保留になっている場合、最悪のコードが転送されます。



品質コードは、コードの形成場所には関係なく、タグの品質を知らせます。

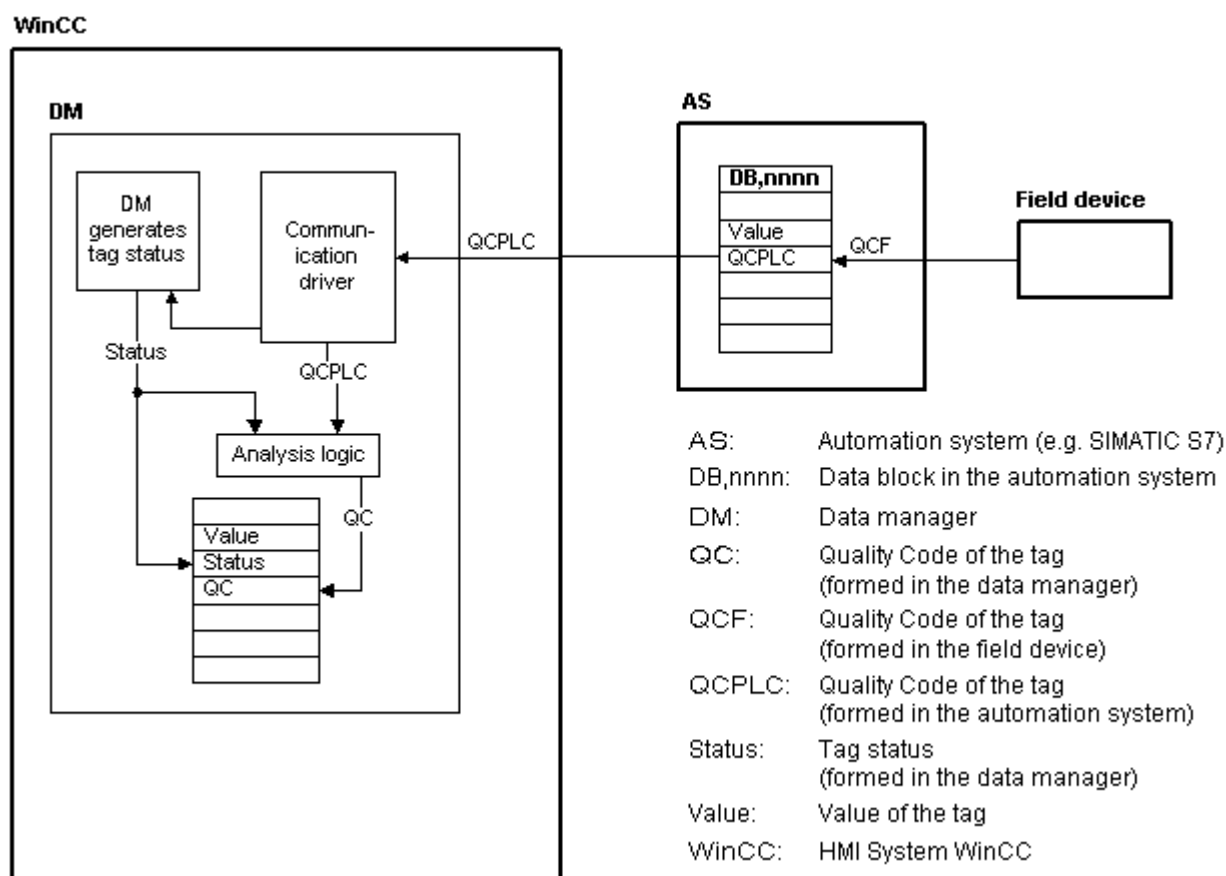
品質コードの重ね表示

接続された現場デバイスを備えるオートメーションシステムの例を使用することによって、品質コードの重ね表示の概要を説明します。

オートメーションシステムでは、フィールドデバイスによって生成される品質コードを読み取ります。解析ロジックを使用して、同じタグのために現在保留中の複数の品質コードを優先度で評価します。最悪のステータスの品質コードがこのタグに割り付けられます。この品質コードが、関連付けられたタグの値のすぐ後ろにあるデータブロックに保存されます。

PCS7 ライブラリのチャンネルモジュールを使用して、解析ロジックを開始できます。PCS7 ライブラリが使用できない場合、オートメーションシステムの解析ロジックを自分で設定する必要があります。

7.9 タグの品質



WinCC は、ランタイム時に、通信ドライバの 1 つを使用してオートメーションデバイスからタグを、関連の品質コードを含めて読み取ります。各タグに対して、タグステータスがデータマネージャに形成されます。これには、たとえば、設定済みの測定有効範囲の違反やオートメーションデバイス間のリンクステータスが含まれます。

データマネージャで解析ロジックを使用すると、データマネージャのタグステータスとオートメーションデバイスの品質コードから品質コードが生成されます。ここでも、最悪ステータスのコードは順送りされ、品質コードとして WinCC によって保存されます。オートメーションシステムで品質コードを持たないタグの場合は、品質コードは常にタグステータスと同一です。

7.9.2 タグの品質コード

はじめに

品質コードは、タグのステータスや品質を確認するために必要です。表示される品質コードには、値の伝達全体の品質とそれぞれのタグの値処理の品質が集約されています。したがって、品質コードを使用すると、たとえば現在値が初期値であるか、または置換値であるかがわかります。

品質コードには優先順位が付けられています。複数のコードが同時に発生した場合、最悪のステータスのコードが表示されます。

品質コードの評価

品質コードは様々な方法で評価できます。

- VB スクリプトでの評価
- C スクリプトでの評価
- ダイナミックダイアログでの評価
- I/O フィールドの[品質コード変更タグ]結果の評価

注記

プロセスタグの完全な値の転送と値の処理を品質コードに含めるには、接続されたオートメーションシステムが品質コードをサポートする必要があります。AS でタグを設定している場合、品質コードのための十分なメモリ領域があることを確認してください。S7 ファミリの AS では、たとえば品質コードにはプロセス値に追加される追加のバイトが必要です。エラーを回避するには、たとえばデータブロックの最後にタグを設定しているとき、このバイトを考慮する必要があります。

品質コードのタグ管理での表示

タグ管理でタグの品質コードを表示することができます。

必要条件：

- WinCC プロジェクトが起動されます。
- タグ管理データエリアに「品質コード」列が表示されます。

7.9 タグの品質



品質コードのプロセス画像での表示

プロセス接続を使ってタグ値をグラフィックオブジェクトに表示する場合、品質コードが表示に影響することがあります。品質タグの値が 0x80（良い）または 0x4c（初期値）である場合、タグ値はグレー表示されません。それ以外の値では、すべてグレー表示されます。さらに、設定した WinCC 設計により、以下のオブジェクトに黄色の三角形の警告が表示されます：

- I/O フィールド
- バー、3D バー
- チェックボックス、ラジオボックス
- グループ表示、ステータス表示
- Slider オブジェクト

構造

構造品質コードは以下のバイナリ構造になっています。

QQSSSSLL

Q: 品質

S: 品質のサブステータス

L: 限界。この値はオプションです。

注記

"品質"テーブルに表示される品質コードは、品質段階の基本の値です。サブステータスと限界エレメントを利用すると、関係のある品質段階に加えて、中間値が生じます。

品質

はじめの 2 桁はタグの品質を指定します。

	Q	Q	S	S	S	S	L	L
	2	2	2	2	2	2	2	2
	7	6	5	4	3	2	1	0
不良 - 値は役に立ちません。	0	0	-	-	-	-	-	-
不確実 - 値の品質は普通より低いですが、値が役に立つ場合もあります。	0	1	-	-	-	-	-	-
良(重ねずに表示) - 値の品質が良です。可能性のあるアラーム条件がサブステータスにより示されます。	1	0	-	-	-	-	-	-
良(重ねて表示) - 値を制御して使用できます。	1	1	-	-	-	-	-	-

サブステータス

品質単独では十分ではありません。個々の品質は、サブステータスに分割されます。品質コードは 2 進コード化されます。品質コードを解析するには、値を 16 進数表示に変換する必要があります。

タグの品質コード

可能な品質コードを、以下の表に一覧表示します。リストの一番上は最低品質コードです。最高品質コードはリストの一番下に示されています。最高品質コードには最低の優先順序が、最高品質コードには最高優先順序が割り付けられます。プロセスで 1 つのタグに複数のステータスが発生した場合は、最低コードが送られます。

コード(16進数)	品質		Q	Q	S	S	S	S	L	L
0x23	不良	デバイスが受動状態-診断 警告が抑止された	0	0	1	0	0	0	1	1
0x3F	不良	機能チェック-ローカル優先	0	0	1	1	1	1	1	1

7.9 タグの品質

コード(16進数)	品質		Q	Q	S	S	S	S	L	L
0x1C	不良	サービス範囲外-ブロックが評価中でなく、コンフィグレーションプランナによる構造の場合があるため、値は信頼できません。ブロックモードが O/S の場合に設定します。	0	0	0	1	1	1	-	-
0x73	不定	シミュレーション値 - 開始	0	1	1	1	0	0	1	1
0x74	不定	シミュレーション値 - 終了	0	1	1	1	0	1	-	-
0x84	良 (重ねずに表示)	更新イベント有効化-値が適切であり、有効な更新イベントがブロックに指定されている場合に設定します。	1	0	0	0	0	1	-	-
0x24	不良	保守アラーム - 詳細診断が可能です	0	0	1	0	0	1	-	-
0x18	不良	通信なし有効値なし-最後のサービス範囲外状態になってから、この値による通信がなかった場合に設定します。	0	0	0	1	1	0	-	-
0x14	不良	最新の有効値による通信なし-この値が失敗した通信により設定された場合に設定します。	0	0	0	1	0	1	-	-
0x0C	不良	デバイス故障-値のソースがデバイス故障により影響を受ける場合に設定します。	0	0	0	0	1	1	-	-
0x10	不良	センサー障害	0	0	0	1	0	0	-	-
0x08	不良	接続なし-この入力接続に必要であり、接続されていない場合に設定します。	0	0	0	0	1	0	-	-
0x04	不良	コンフィグレーションエラー-固有の製造会社が検出できるものに依りて、パラメータ化または設定に関する不一致があるため値が役に立たない場合に設定します。	0	0	0	0	0	1	-	-
0x00	不良	固有でない-値が不良である固有の理由がありません。伝搬に使用されます。	0	0	0	0	0	0	-	-
0x28	不良	プロセス関連 - 置換値	0	0	1	0	1	0	-	-
0x2B	不良	プロセス関連- 保守不要	0	0	1	0	1	0	1	1
0x68	不定	保守要求	0	1	1	0	1	0	-	-
0x60	不定	シミュレートする値-ブロックがマニュアルモードになっている最中にオペレータがプロセス値に書き込む際に設定されます。	0	1	1	0	0	0	-	-
0x64	不定	センサーキャリブレーション	0	1	1	0	0	1	-	-

コード(16進数)	品質		Q	Q	S	S	S	S	L	L
0x5C	不定	コンフィグレーションエラー	0	1	0	1	1	1	-	-
0x58	不定	正常以下	0	1	0	1	1	0	-	-
0x54	不定	技術単位の範囲違反 - 値がこのパラメータに定義された値の集合の外にある場合に設定します。限界には、超過された方向が定義されています。	0	1	0	1	0	1	-	-
0x50	不定	センサー変換が不正確	0	1	0	1	0	0	-	-
0x4B	不定	置換(定数)	0	1	0	0	1	0	1	1
0x78	不定	プロセス関連- 保守不要	0	1	1	1	1	0	-	-
0x4C	不定	初期値 - デバイスまたはパラメータのリセットの間と後の揮発性のパラメータの値です。	0	1	0	0	1	1	-	-
0x48	不定	置換値 - 事前定義された値は計算された値の代わりに使用されます。これはフェールセーフ処理のために使用されます。	0	1	0	0	1	0	-	-
0x44	不定	最新の有効値 - この値を書込んでいるのが何であっても、書込み停止します。これはフェールセーフ処理のために使用されます。	0	1	0	0	0	1	-	-
0x40	不定	固有でない - 値が不確実である固有の理由がありません。	0	1	0	0	0	0	-	-
0xE0	適切 (重ねて表示)	初期のフェールセーフ(IFS) - 下流の出力ブロック (AO など)をフェールセーフ作動させる必要があるブロックからの値です。	1	1	1	0	0	0	-	-
0xD8	適切 (重ねて表示)	ローカル優先(LO) - 値がローカルキースイッチによりロックされたブロック、または連動ロジックが有効な Complex AO/DO であるブロックのものです。普通の制御の失敗は、ホストシステムで実行中のファンクションに伝播される必要があります。これには「呼び出されない」も含まれます。	1	1	0	1	1	0	-	-
0xD4	適切 (重ねて表示)	選択しない(DNS) - 値が、ブロックの条件のため選択できないブロックのものです。	1	1	0	1	0	1	-	-
0xCC	適切 (重ねて表示)	呼び出されない(NI) - この入力を使用する目的のモードがない値です。	1	1	0	0	1	1	-	-

7.9 タグの品質

コード(16進数)	品質		Q	Q	S	S	S	S	L	L
0xC8	適切 (重ねて表示)	初期化要求(IR) - 値がソース(背後の計算の入力パラメータ)の初期値です。下位のループが壊れたか、モードが間違っているためです。	1	1	0	0	1	0	-	-
0xC4	適切 (重ねて表示)	初期化確認(IA) - 値がソース(パラメータへの連続的な入力、リモートの連続的な入力、リモートの出力)の初期化された値です。	1	1	0	0	0	1	-	-
0xC0	適切 (重ねて表示)	OK-エラーがないか、この値に特殊な条件が関連しています。	1	1	0	0	0	0	-	-
0xA0	良 (重ねずに表示)	フェールセーフ開始	1	0	1	0	0	0	-	-
0x98	良 (重ねずに表示)	未確認の重大アラーム - 値が良であり、ブロックに優先度が 8 以上の未確認のアラームがある場合に設定します。	1	0	0	1	1	0	-	-
0x94	良 (重ねずに表示)	未確認の勧告アラーム - 値が良であり、ブロックに優先度が 8 より小さい未確認のアラームがある場合に設定します。	1	0	0	1	0	1	-	-
0x90	良 (重ねずに表示)	未確認の更新イベント - 値が良であり、ブロックに未確認の更新イベントがある場合に設定します。	1	0	0	1	0	0	-	-
0x8C	良 (重ねずに表示)	有効な重大アラーム - 値が良であり、ブロックに優先度が 8 以上の有効なアラームがある場合に設定します。	1	0	0	0	1	1	-	-
0x88	良 (重ねずに表示)	有効な勧告アラーム - 値が良であり、ブロックに優先度が 8 より小さい有効なアラームがある場合に設定します。	1	0	0	0	1	0	-	-
0xA8	良 (重ねずに表示)	保守要求	1	0	1	0	1	0	-	-
0xA4	良 (重ねずに表示)	保守要求	1	0	1	0	0	1	-	-

コード(16進数)	品質		Q	Q	S	S	S	S	L	L
0xBC	良 (重ねずに表示)	機能チェック-ローカル優先	1	0	1	1	1	1	-	-
0x80	良 (重ねずに表示)	OK-エラーがないか、この値に特殊な条件が関連しています。	1	0	0	0	0	0	-	-

限界値

品質コードは、限界によりさらに分類できます。限界は、オプションです。

	Q	Q	S	S	S	S	L	L
OK - 値は自由に移動できます。	-	-	-	-	-	-	0	0
下限 - 値が下限にアクセスしました。	-	-	-	-	-	-	0	1
上限 - 値が上限にアクセスしました。	-	-	-	-	-	-	1	0
定数(上限と下限) - 処理が何であっても、値を移動できません。	-	-	-	-	-	-	1	1

OPC による通信の品質コード

"OPC"チャンネルを介した通信では、OPC がサポートしていない品質コードは変換されません。

WinCC の品質コード	OPC の品質コード
0x48	0x40
0x4C	0x40
0x5C	0x40
0x60	0x40
0x80...0xD4	0xC0
0xD8	0xC0

7.9 タグの品質

7.9.3 タグステータス

はじめに

個々の WinCC タグのタグステータスは、ランタイム時にモニタできます。タグステータスには、特に、設定済みの測定有効範囲の違反と、WinCC とオートメーションデバイス間のリンクステータスに関するデータが含まれます。

品質コードは、コードの形成場所には関係なく、タグの品質を知らせます。その結果、値転送全体のステータスと値処理のステータスが考慮されています。

たとえば、下限値での測定範囲で違反が発生する場合、品質コードレポート"0x55"が通知されます。この測定範囲の違反は、WinCC データマネージャまたはフィールドデバイスで発生した可能性があります。タグステータスによって、この測定範囲違反が WinCC で発生したか、WinCC に値が渡る前に発生したのか調べることができます。

たとえば、タグステータスのレポートがコード 0x0010 の限界値違反場合、その値は WinCC で設定済みの下限レンジ限界より下の状態であることを表します。タグステータスのレポートに限界値違反がない場合は、すでに、その限界値違反は WinCC に渡った品質コードに含まれています。

品質コードの評価

品質コードは様々な方法で評価できます。

- C スクリプトでの評価
- ダイナミックダイアログでの評価
- I/O フィールドの"品質コード変更タグ"結果の評価

WinCC のステータスフラグ

考えられるタグ状態が、以下のテーブルにあります。

フラグ名	値	説明
	0x00 00	エラーなし
DM_VARSTATE_NOT_ESTABLISHED	0x00 01	相手への接続が確立されていない
DM_VARSTATE_HANDSHAKE_ERROR	0x00 02	ハンドシェイクエラー

フラグ名	値	説明
DM_VARSTATE_HARDWARE_ERROR	0x0004	ネットワークモジュールに欠陥があります。
DM_VARSTATE_MAX_LIMIT	0x0008	設定された上限値を超えています。
DM_VARSTATE_MIN_LIMIT	0x0010	設定された下限値を下回っています。
DM_VARSTATE_MAX_RANGE	0x0020	フォーマット上限値を超えています。
DM_VARSTATE_MIN_RANGE	0x0040	フォーマット下限値を下回っています。
DM_VARSTATE_CONVERSION_ERROR	0x0080	変換エラーを表示します(フォーマット限界値の超過に関連して)
DM_VARSTATE_STARTUP_VALUE	0x0100	タグの初期値
DM_VARSTATE_DEFAULT_VALUE	0x0200	タグ置換値
DM_VARSTATE_ADDRESS_ERROR	0x0400	チャンネルアドレス指定エラー
DM_VARSTATE_INVALID_KEY	0x0800	タグが見つかりません/使用できません
DM_VARSTATE_ACCESS_FAULT	0x1000	タグアクセスが許可されていません
DM_VARSTATE_TIMEOUT	0x2000	タイムアウト/チャンネルからのチェックバックメッセージなし
DM_VARSTATE_SERVERDOWN	0x4000	サーバーが使用できない

7.9.4 モニタ接続ステータスへのタグステータスの使用

個々の WinCC タグのステータスはランタイムでモニタでき、関連した接続のステータスに関する情報を知ることができます。

状態モニタの設定

モニタは、グラフィックデザイナーでオブジェクトプロパティとして設定します。

1. [オブジェクトプロパティ]ウィンドウで必要なプロパティを選択します。
2. [ダイナミック]列のショートカットメニューから[ダイナミックダイアログ]エントリを選択します。
[値の範囲]ダイアログが開きます。

3. 設定を指定します。
 - － モニタしたタグ
 - － タグ値:有効範囲の割り付けおよびステータス表示
 - － タグのステータス評価
 - － ステータス：有効範囲の割り付けおよび対応するステータステキスト

タグの現在のステータスに対応する、入力されたステータステキストの1つが、ランタイムで設定されたオブジェクトに表示されます。

下記も参照

ランタイム時の接続ステータスタグの設定 (ページ 170)

7.9.5 グローバルアクションによるタグステータスのモニタ

はじめに

タグのステータスをモニタする方法の 1 つは、グローバルスクリプトエディタの内部ファンクション[GetTagState]および[GetTagStateWait]を利用することです。[GetTag]ファンクションおよび[GetTagWait]ファンクションとは異なり、タグの値だけでなくタグのステータスも返します。このステータス値を評価してさまざまなイベントのトリガに使用できます。関連した接続のステータスの評価に使用することもできます。

グローバルアクションでは、モニタするタグのタイプに対して **GetTagState** ファンクションを使用して、このタグのステータス値を求めます。このようなファンクションはタグタイプごとに存在します。ステータス値"0"は、接続が良でエラーがないことを示しています。このステータスは、必要に応じて評価できます。

7.9 タグの品質

例:

この例では、"符号付き 16 ビット値"タイプの WinCC タグのモニタを示しています。
"GetTagSWordState"ファンクションを使用すると、このタグのステータスを特定できます。最初のファンクションパラメータは、モニタする WinCC タグの名前です。2 番目のパラメータは、返されたステータス値を書込む場所を示しています。

```
#include "apdefap.h"

int gscAction( void )
{
    DWORD dwState = 0;
    GetTagSWordState("Variable_01",&dwState);
    if ( dwState == 0 )
    {
        //Connection OK
        SetTagBit("BINi_E_CONNECTION",FALSE);
    }
    else
    {
        //Connection Error
        SetTagBit("BINi_E_CONNECTION",TRUE);
    }
    return 0;
}
```

タグのステータスは、内部タグ BINi_E_CONNECTION に出力されます。エラーが発生すると、このタグの値は TRUE に設定されます。エラー処理では、たとえばこのタグを使用してアラームをトリガしたり、エラーメッセージを表示できます。

7.9.6 内部タグの作成方法

はじめに

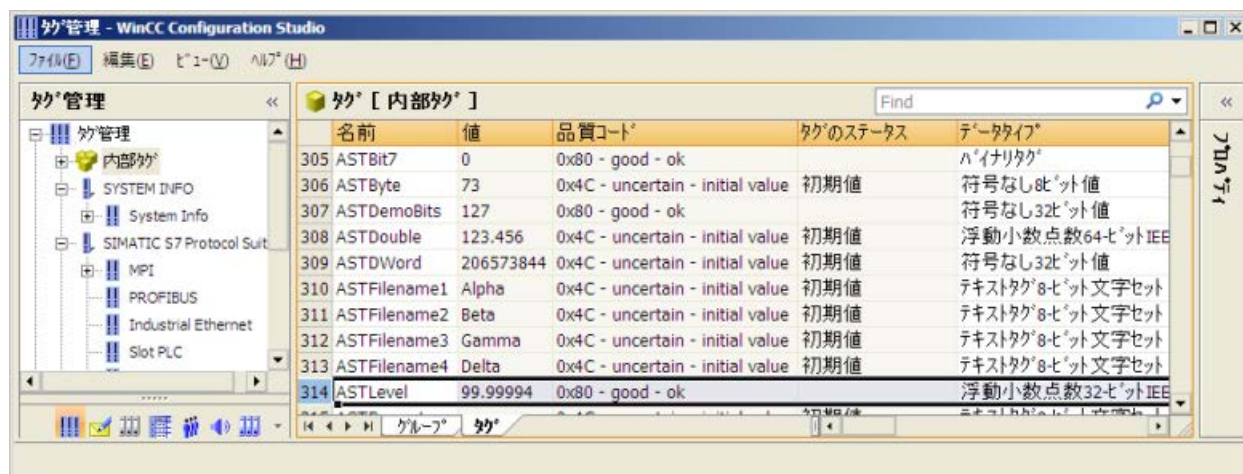
内部タグにランタイムに予想される値がない場合、以下の手順を使用してタグをチェックできます。

必要条件

- 内部タグが構成されている。
- WinCC プロジェクトが起動されます。

手順

1. WinCC エクスプローラの[タグ管理]を開きます。
2. エントリ[内部タグ]とナビゲーションエリアでチェックするタグを選択します。
3. データエリアに「品質コード」と「値」の列を表示するには、[表示]に移動し、列ヘッダーのショートカットメニューでこれらの列を選択する必要があります。
4. 品質コードをチェックします。値"80"が表示された場合、タグ値は OK です。他の値についての説明は、「タグの品質コード」に記載されています。
5. 品質コードが"80"に等しくない場合は、右側のプロパティで設定を確認してください。
6. 上限と下限または初期値に値が設定されているかどうかをチェックします。これらの値は表示に影響することがあります。
7. タグ値が、設定された値の 1 つの影響を受ける場合、プロジェクトを無効にし、制限値または置換値を変更します。



注記

タグ値および品質コードは、ランタイム時にだけ表示されます。

下記も参照

タグの品質コード (ページ 683)

7.9 タグの品質

OPC - オープンな接続性(Open Connectivity)

8.1 OPC - オープンな接続性(Open Connectivity)

内容

標準ソフトウェアインターフェース OPC を使用すれば、各種メーカーのデバイスとアプリケーションを、統一した方法で互いに接続できます。

WinCC は、OPC サーバーとしても OPC クライアントとしても使用できます。「OPC」チャンネルは、WinCC の OPC クライアントアプリケーションです。

このセクションでは、以下について説明します。

- WinCC にある OPC サーバー
- WinCC で OPC を使用する方法
- 各種 OPC DA リンクを設定する方法
- WinCC メッセージシステムへのアクセスを設定する方法
- OPC A&E で WinCC メッセージシステムがマッピングされる方法
- WinCC アーカイブシステムへのアクセスを設定する方法

8.2 OPC の機能

OPC は、オートメーションエンジニアリングでのデータ交換のための、標準化されたメーカーに依存しないソフトウェアインターフェースです。

OPC インターフェースを使用すれば、各種メーカーのデバイスとアプリケーションを標準リンクで接続できます。

OPC は、Windows COM (Component Object Model)および DCOM (Distributed Component Object Model)のテクノロジーをベースとしています。

OPC XML DA は、XML、SOAP および HTTP などのインターネット標準をベースとする追加的なソフトウェアインターフェースを提供します。

OPC UA (Unified Architecture、ユニファイドアーキテクチャ)は、OPC の後継テクノロジーです。OPC UA は、プラットフォームに依存せず、通信媒体としてさまざまなプロトコルをサポートしています。

8.3 OPC 仕様と互換性

概要

OPC は、WinCC で以下のオブジェクトにアクセスするためのインターフェースを指定します。

- プロセス値(OPC Data Access 2.05a, 3.0; OPC XML Data Access 1.01; OPC UA 1.03)
- アーカイブされたプロセス値(OPC Historical Data Access 1.20; OPC UA Historical Access 1.03)
- 時系列順メッセージ(OPC Historical Alarms and Events 1.10)
- メッセージ(OPC Alarms and Events 1.10; OPC UA Alarms and Conditions 1.03)

個々の OPC の仕様の詳細については、OPC 協会 (<http://www.opcfoundation.org>) の Web サイトを参照してください。

互換性

これらの仕様のサポートは、OPC Foundation の「Compliance Test Tool」(CTT)により定期的に監視されます。他のメーカーの OPC 製品との相互運用性は、「OPC Interoperability Workshops」への参加によって保証されます。

提出された結果は、OPC Foundation の Web サイトで公開されます。結果を表示するには、検索用語の「OPC Self-Certified Products」を入力します。

8.4 WinCC における OPC の使用

はじめに

WinCC では、以下の OPC インタフェースでサーバーが使用できます。

- OPC Data Access / OPC XML Data Access: データの WinCC 本体へのアクセス
- OPC Historical Data Access: WinCC アーカイブシステムへのアクセス
- OPC Alarms&Events: WinCC メッセージシステムへのアクセス
- OPC Unified Architecture: データの WinCC 本体とアーカイブシステムへのアクセス

WinCC には、デフォルトで OPC チャンネルが含まれています。OPC チャンネルは、OPC DA、OPC XML DA または OPC UA を介してクライアントとして関連 OPC サーバーにアクセスできます。

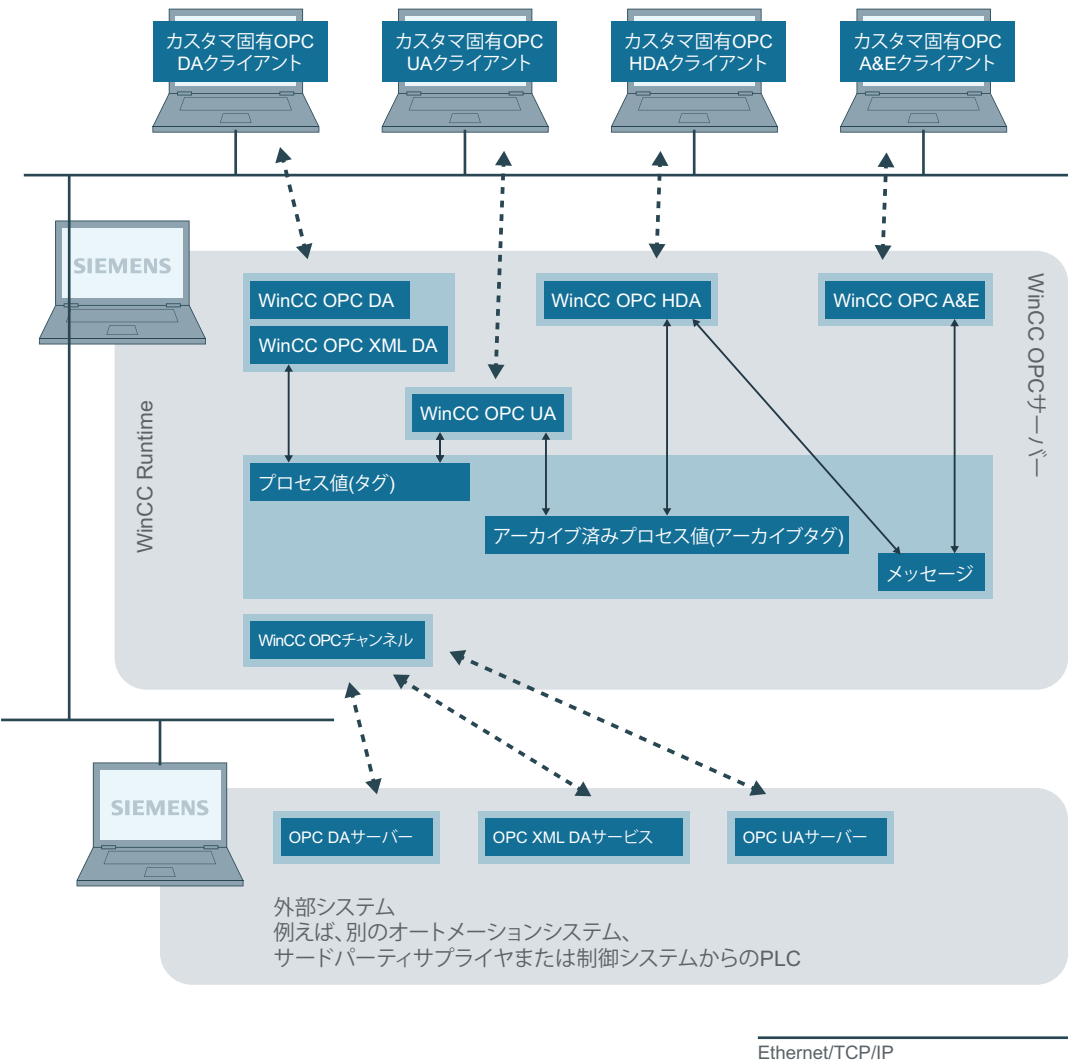
WinCC OPC 通信の概念

WinCC OPC サーバーと OPC クライアント間のデータ交換を、DCOM を使用して完了します。WinCC をインストールすると、WinCC OPC サーバーの DCOM 設定が適切に構成されます。

WinCC OPC サーバーまたはクライアントが外部 OPC システムを使用して通信する場合、対応する調整を実行する必要があります。[ローカルアクセス]および[リモートアクセス]のオーソリゼーションを、クライアントのユーザー管理の[DCOM/作業領域/COM セキュリティ/アクセス権限/デフォルトの編集]のユーザーに入力しなければなりません。

WinCC の OPC XML サーバーは、Web サービスとして実装されます。これにより、インターネット経由で PC へのアクセスが提供されます。そのため、適切なアクセス権限を定義する必要があります。

WinCC OPC 通信の概念を以下に示します。



ライセンス

OPC サーバー	ライセンス
WinCC OPC DA サーバー	WinCC の有効な RT ライセンス
WinCC OPC XML DA サーバー	WinCC の有効な RT ライセンス
WinCC OPC UA サーバー	WinCC オプション接続性パック
WinCC OPC HDA サーバー	
WinCC OPC A&E サーバー	

8.5 WinCC OPC を使用するために Windows を構成する方法

はじめに

OPC クライアントと OPC サーバーは、DCOM アプリケーションです。配布される DCOM アプリケーションは、同じユーザーアカウントでのみ動作します。したがって、OPC サーバーは、OPC クライアントのユーザーアカウントを認識する必要があります。OPC クライアントは、OPC サーバーのユーザーアカウントを認識する必要があります。WinCC OPC サーバーを WinCC OPC クライアントと共に使用すると、インストールの段階で正しく構成されていることが保証されます。

外部 OPC サーバーまたはクライアントを使用している場合のユーザーアカウントの宣言

ユーザー権限の付与の詳細については、Windows のマニュアルを参照してください。

必要条件

ユーザーアクセス権限を構成するために、管理者として WinCC OPC サーバーと OPC クライアントワークステーションの両方にログオンしていること。

手順

1. [コントロールパネル]>[システムとセキュリティ]>[管理ツール]>[コンピュータの管理]>[ローカルユーザーとグループ]の順に選択します。
2. [ユーザー]ショートカットメニューで、[新規ユーザー]を選択します。
[新規ユーザー]ダイアログに、通信パートナのユーザーアカウントの詳細を入力します。[作成]をクリックしてダイアログを閉じます。
3. [ユーザー]アイコンをクリックします。適切なユーザーをダブルクリックします。該当するユーザーの[プロパティ]ダイアログが表示されます。
4. [メンバ]タブをクリックします。[追加]をクリックします。[グループの選択]ダイアログが開きます。
5. [ユーザー]グループを追加します。
WinCC がインストールされたコンピュータを使用している場合、[SIMATIC HMI]グループも追加します。[OK]をクリックして、開いているダイアログをすべて閉じます。

Windows ファイアウォール設定の適用方法

WinCC をインストールすると、WinCC OPC サーバーの Windows ファイアウォール設定が適切に構成されます。

OPC クライアントが別のサブネットの OPC サーバーにアクセスする場合、許可されたネットワークの設定を OPC サーバーに適用する必要があります。

8.6 WinCC OPC XML DA サーバー

8.6.1 WinCC OPC XML DA サーバーの機能

はじめに

WinCC の OPC XML DA サーバーは、Microsoft インターネットインフォメーションサーバー (IIS) の Web サービスとして実現されます。

WinCC OPC XML DA サーバーは、OPC XML クライアントに Web ページ形式で OPC プロセスデータを提供します。この Web ページは HTTP を使用してインターネットを経由してアクセスできます。WinCC OPC XML DA サーバーのアドレスは次のとおりです。 <http://<xxx>/WinCC-OPC-XML/DAWebservice.asmx>

WinCC OPC XML DA サーバーは WinCC に表示されません。OPC XML クライアントがデータを要求すると、Web サーバーにより Web サービスが自動的に開始されます。

OPC 通信を正常に確立するには、以下の点に準拠してください。

- WinCC OPC XML DA サーバーの WinCC プロジェクトを実行している必要があります。
- HTTP を使用して、WinCC OPC XML DA サーバーのコンピュータにアクセスできる必要があります。

ライセンス

WinCC OPC XML DA サーバーを操作するには、OPC XML サーバーとしてインプリメントされた各 WinCC コンピュータに、以下のライセンスをインストールしておく必要があります。

- WinCC の有効な RT ライセンス
- WinCC オプション接続性パック

"文字列"タイプのタグの専用機能

論理的に浮動小数点の値を表す"文字列"タイプのタグを使用する場合、OPC クライアントが書込みや読取りをしようとする問題が発生するおそれがあります。

説明

OPC クライアントは文字列タグに書込みを行います。新規値は文字列としてではなく、float、double、または小数として入力されます。

問題

小数の値を示すコンマ(ドイツ語の場合)は失われる場合があります。この結果、不正な値になります。

これは、読み取られる値が float、double、または小数として要求されている場合、文字列タグへの読取りアクセスにも影響を与えます。

対策

浮動小数点の値について、それぞれの浮動小数点タグのみを使用してください。文字列タグには文字列形式でのみアクセスしてください。

下記も参照

WinCC OPC XML クライアントの機能 (ページ 275)

サポートされている WinCC データタイプの概要 (ページ 254)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.6.2 WinCC OPC XML DA サーバーのインストール

はじめに

OPC XML を使用すれば、インターネットでプロセスタグにアクセスできます。OPC XML を操作するには、多くのソフトウェアコンポーネントを追加インストールする必要があります。

注記

ここに記載されているインストール手順を順序どおりに行うことが不可欠です。手順を誤ると、インストール中に問題が発生するおそれがあります。

必要条件

- インターネットインフォメーションサーバー(IIS)のインストール
- Microsoft .NET Framework V4.6.2 以降を WinCC 製品 DVD からインストール済み
- WinCC セットアップを使用した WinCC OPC XML サーバーのインストール

インターネットインフォメーションサーバー(IIS)のインストール

Windows Server 2012 R2 / 2016 / 2019 では、関連する役割サービスの「Webserver (IIS)」役割を使用して、サーバーマネージャの設定を構成します。

以下の設定を選択します。

- Web 管理ツール:
 - IIS 管理サービス
 - IIS 管理コンソール
 - IIS 管理スクリプトとツール
- [WWW サービス] > [HTTP 共通機能]または[HTTP 共有機能]:
 - 標準マニュアル
 - 静的コンテンツ
- WWW サービス > アプリケーション開発機能:
 - .NET 拡張性
 - ASP.NET
 - ISAPI 拡張機能
 - ISAPI フィルタ
- WWW サービス > セキュリティ:
 - 要求のフィルタリング
 - 基本認証
 - Windows 認証

注記

Microsoft インターネットインフォメーションサービス(IIS)を常に ASP.NET と共にインストールする

Microsoft インターネットインフォメーションサービス(IIS)をインストールする際には、常に ASP.NET をインストールします。

注記

WinCC OPC XML DA サーバーの Web サービスがポート:80 (HTTP)上で通信を行います。ファイアウォールルール[WWW サービス(HTTP)]が選択されており、必要なネットワークエリアに対して有効になっていることを確認します。

WinCC OPC XML サーバーのインストール

WinCC のインストール中、WinCC OPC XML DA サーバーを選択できます。詳細については、[インストールに関する注意事項]>[WinCC のインストール]の WinCC 情報システムを参照してください。

インストール中、以下の設定を実行する必要があります。

- 仮想ディレクトリ「WinCC-OPC-XML」を作成します。
- このディレクトリにアクセス権限を定義します。

8.6.3 IIS を使用したセキュリティ設定の設定(WinCC OPC XML DA サーバー)

概要

インターネットインフォメーションサービス(IIS)を使用すれば、インターネットを使用して PC にアクセスできます。そのため、適切なアクセス権限を定義する必要があります。

注記

以下の設定に関する疑問点や問題点については、担当のイントラネット/インターネット管理者に問い合わせてください。

手順

1. [コントロールパネル]>[システムとセキュリティ]>[管理ツール]>[コンピュータの管理]>[サービスとアプリケーション]>[インターネット インフォメーション サービス(IIS)マネージャ]の順に選択します。
2. 仮想ディレクトリ「WinCC-OPC-XML」を選択します。
3. 機能表示で[認証]機能を開きます。
インストールされている認証方式のリストが開きます。
4. 匿名アクセスを有効にするには、[匿名認証]を有効にします。
5. 認証アクセスを有効にするには、[Windows 認証]を有効にします。
6. 開いているすべてのダイアログを閉じます。

注記

制限およびリスク

アクセスオプションを設定するときに Microsoft によって示された制限とリスクに注意してください。

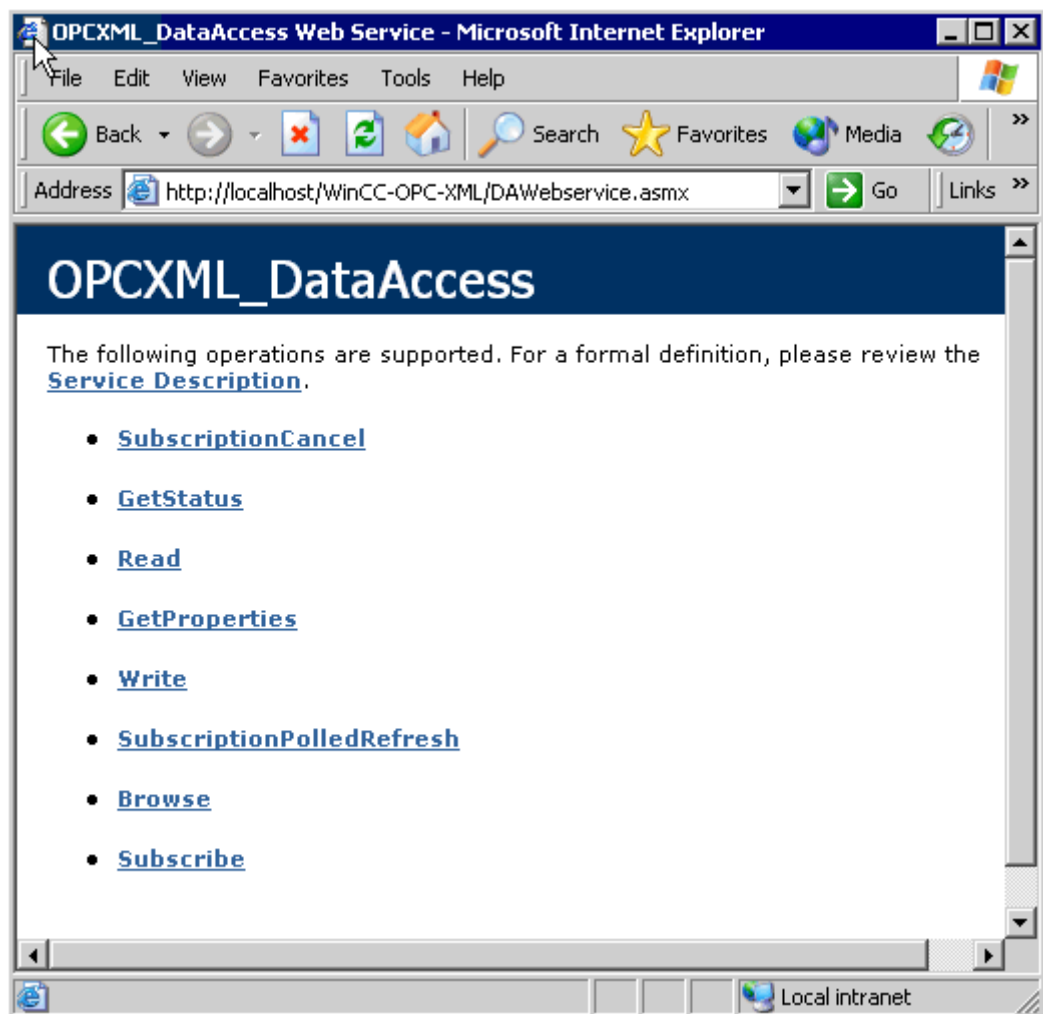
8.6.4 インストールのテスト(WinCC OPC XML DA サーバー)

はじめに

OPC XML-DA を使用すれば、Web ページ形式で OPC プロセスデータを使用できます。この Web ページは HTTP を使用してインターネットを経由してアクセスできます。以下のセクションでは、インストールのテスト方法について説明します。

手順

1. WinCC OPC XML サーバーとして動作しているコンピュータ上で Internet Explorer を起動します。
2. アドレスバーに URL "http://localhost/WinCC-OPC-XML/DAWebservice.asmx"と入力します。Enter キーを押してエントリを確定します。
3. OPC XML DA ファンクション要求が表示されたら、インストールは正常終了しました。



8.7 WinCC OPC DA サーバー

8.7.1 WinCC OPC DA サーバーの機能

はじめに

WinCC OPC DA サーバーは OPC Data Access 仕様 2.05a および 3.00 をサポートします。これは、準拠テストにより確認されています。

WinCC OPC DA サーバーは DCOM アプリケーションの 1 つです。WinCC OPC DA サーバーは、このインターフェースを使用して、WinCC タグに関する必要情報を WinCC クライアントに供給します。

接続を使用して WinCC OPC DA クライアントが WinCC OPC DA サーバーにアクセスすると、WinCC OPC DA サーバーが有効になります。OPC 通信を正常に確立するには、以下の点に注意してください。

- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっている必要があります。
- WinCC OPC DA が実行されるコンピュータに、その IP アドレスを使用してアクセスできる必要があります。

インストール

WinCC のインストール中、WinCC OPC DA サーバーを選択できます。WinCC のインストールが終了すると直ちに、WinCC OPC DA サーバーを使用できます。このとき、それ以上の設定を実行する必要はありません。

WinCC OPC DA サーバーは、WinCC サーバーまたは WinCC クライアントに実装できます。

設定に関する注意

- WinCC プロジェクトの設定のためにタグをタググループに組み立てできます。タグはグループと同じ名前である必要はありません。
- 例えば VBScript またはオブジェクト「IO フィールド」など WinCC で開始される各書き込み要求は、必ず同期「書き込み」呼び出しとして扱われます。[IOPCSyncIO::Write] インターフェイスが、このために WinCC OPC DA サーバーによって使用されます。非同期書き込み機構は、WinCC OPC DA チャンネルには実装されていません。

注記

コンピュータで[接続 -> LAN 設定]の下で設定を自動的に検出するようにインターネットオプションが設定されている場合、Web サービス経由で OPC DA にアクセスすると長い時間がかかります。

下記も参照

OPC DA サーバー名の問い合わせ (ページ 711)

複数の OPC DA サーバーの使用 (ページ 710)

WinCC 間の接続例 (ページ 713)

WinCC と SIMATIC NET FMS OPC サーバー間の接続例 (ページ 717)

WinCC と SIMATIC NET S7 OPC サーバー間の接続例 (ページ 719)

WinCC と Microsoft Excel 間の接続例 (ページ 725)

サポートされている WinCC データタイプの概要 (ページ 254)

www.opcfoundation.org (<http://www.opcfoundation.org>)

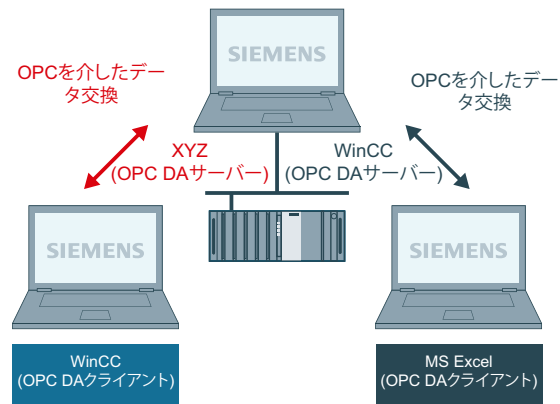
8.7.2 複数の OPC DA サーバーの使用

はじめに

2 台以上の OPC DA サーバーを 1 台のコンピュータにインストールでき、何台でも並列して使用できます。

このように、WinCC の OPC DA サーバーおよび他の(サードパーティの)プロバイダの OPC DA サーバーが、同一コンピュータ上で、互いに独立して動作できます。

WinCC OPC DA クライアントは、サードパーティプロバイダの OPC サーバーを使用して、オートメーションデバイスのプロセスデータにアクセスできます。Microsoft Excel の OPC DA クライアントは、WinCC OPC DA サーバーを使用して WinCC データにアクセスできます。



各種メーカーから数多くの OPC DA サーバーが提供されています。これらの各 OPC DA サーバーには、識別のために一意の名前(ProgID)が指定されています。 OPC DA クライアントは、この名前を使用して OPC サーバーのアドレスを指定する必要があります。

OPC 項目マネージャを使用すれば、OPC DA サーバーの名前を問い合わせることができます。 WinCC V 7 の OPC DA サーバーの名前は"OPCServer.WinCC"です。

下記も参照

OPC DA サーバー名の問い合わせ (ページ 711)

8.7.3 OPC DA サーバー名の問い合わせ

はじめに

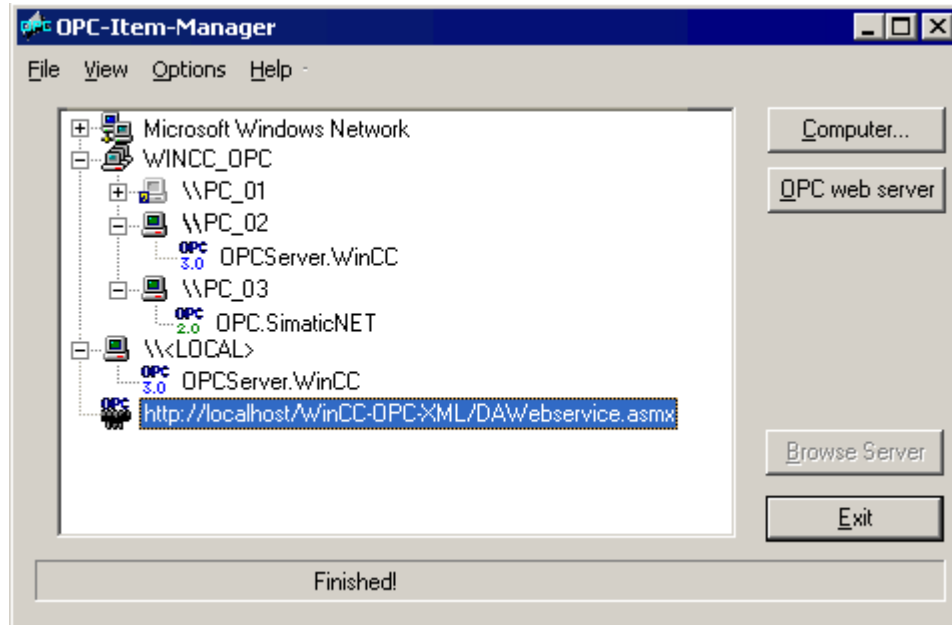
1 台のコンピュータに複数の OPC DA サーバーをインストールできます。 OPC 項目マネージャを使用すると、ワークステーションが使用できる OPC DA サーバーの名前が選択ウィンドウに表示されます。これらの OPC DA サーバーは、同じコンピュータ上、またはネットワーク環境内の各コンピュータ上で実行できます。

必要条件

WinCC OPC DA クライアントの WinCC プロジェクトに"OPC"チャンネルが追加されていること。

手順

1. WinCC OPC DA クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[システムパラメータ]を選択します。"OPC 項目マネージャ"が開きます。
2. OPC 項目マネージャのナビゲーションウィンドウで、アクセスするコンピュータの名前を選択します。
3. OPC 項目マネージャを使用すると、コンピュータが使用できる OPC DA サーバーの名前が選択ウィンドウに表示されます。



下記も参照

OPC 項目マネージャ (ページ 250)

8.7.4 OPC DA 接続の例

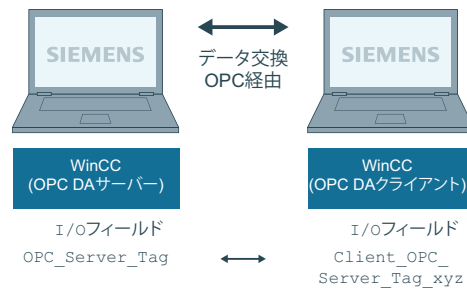
8.7.4.1 WinCC 間の接続

WinCC 間の接続例

はじめに

WinCC 間の接続を確立する場合、WinCC タグ"OPC_Server_Tag"を使用して WinCC OPC DA サーバーと WinCC OPC DA クライアント間でデータを交換します。クライアント上の WinCC タグ"Client_OPC_Server_Tag_xyz"は、サーバー上の WinCC タグ"OPC_Server_Tag"を読み取ります。WinCC OPC サーバー上の"OPC_Server_Tag"タグ値が変更されると、WinCC OPC DA クライアント上の WinCC タグ"Client_OPC_Server_Tag_xyz"も変更されます。クライアント上で変更を実行すると、サーバーにも反映します。

タグ値は、両方のコンピュータの I/O フィールドに表示されます。



必要条件

- WinCC プロジェクトをインストールしたコンピュータが 2 台あること。
- IP アドレスを使用して、どちらのコンピュータにもアクセスできる必要があります。

設定手順

WinCC 間の接続を確立するには、以下の設定が必要です。

1. WinCC OPC DA サーバー上の WinCC プロジェクトの設定
2. WinCC OPC DA クライアント上の WinCC プロジェクトの設定

下記も参照

WinCC OPC DA サーバー上の WinCC プロジェクトの設定方法 (ページ 714)

WinCC OPC DA クライアント上の WinCC プロジェクトの設定 (ページ 715)

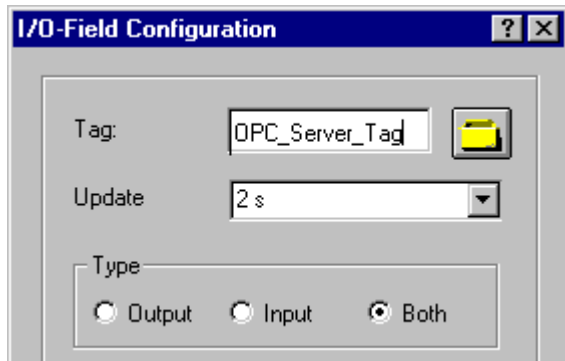
WinCC OPC DA サーバー上の WinCC プロジェクトの設定方法

はじめに

このセクションでは、WinCC OPC DA サーバーの WinCC プロジェクトで WinCC タグを作成し、I/O フィールドに表示します。

手順

1. WinCC OPC DA サーバーで、[内部タグ]アイコンのショートカットメニューで[タグの新規作成]を選択します。"符号付き 16 ビット値"タイプの"OPC_Server_Tag"タグを新たに作成します。
2. グラフィックデザイナを起動し、画像を新たに開きます。
3. 画像内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで[I/O フィールド]オブジェクトを選択します。[I/O フィールド設定]ダイアログが開きます。



4. [タグ]フィールドに名前"OPC_Server_Tag"を入力します。
5. 更新を"2 秒"に、フィールドタイプを[I/O フィールド]に設定します。
6. [OK]をクリックしてダイアログボックスを閉じ、画面を保存します。
7. グラフィックデザイナの[有効]ボタンをクリックして、WinCC プロジェクトを有効にします。

下記も参照

WinCC OPC DA クライアント上の WinCC プロジェクトの設定 (ページ 715)

WinCC OPC DA クライアント上の WinCC プロジェクトの設定

はじめに

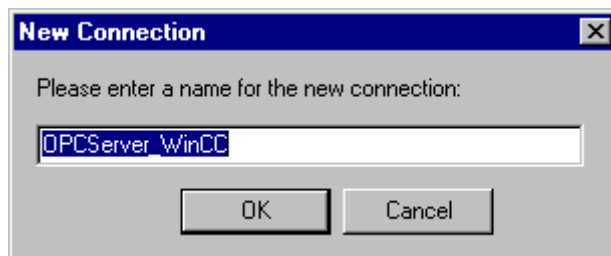
このセクションでは、WinCC OPC DA サーバー上の WinCC タグを読み取るために、WinCC OPC DA クライアント上で WinCC タグを作成します。このタグ値は I/O フィールドに表示されます。

前提条件

- WinCC OPC DA クライアントの WinCC プロジェクトに"OPC"チャンネルが追加されていること。
- WinCC OPC DA サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"OPC_Server_Tag"内部タグが設定されていること。
- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっていること。

手順

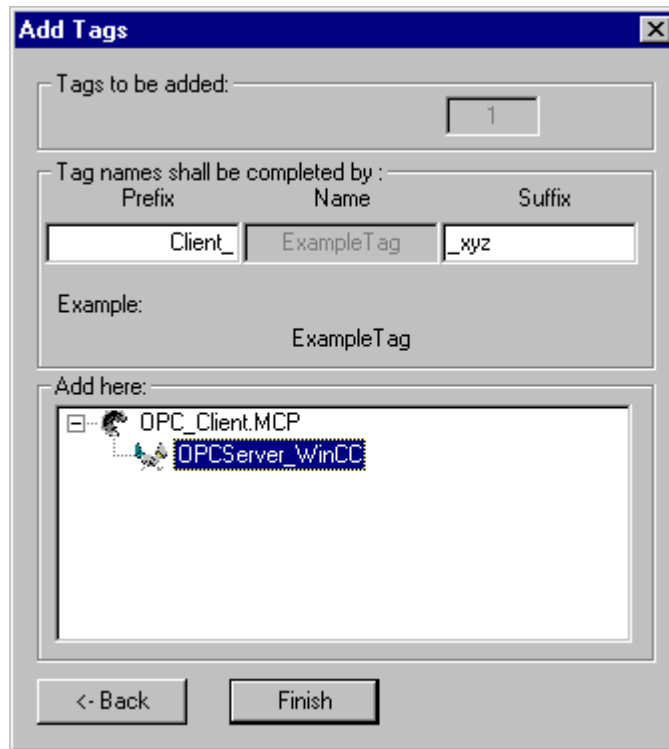
1. WinCC OPC DA クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[システムパラメータ]を選択します。OPC 項目マネージャが開きます。
2. 選択ダイアログで、OPC DA サーバーとして使用するコンピュータの名前を選択します。リストから[OPCServer.WinCC]を選択します。[サーバーの参照]ボタンをクリックします。[フィルタ規準]ダイアログが開きます。
3. [フィルタ規準]ダイアログで、[次へ->]ボタンをクリックします。[OPCServer.WinCC ...]ダイアログの[OPC_Server_Tag]タグを選択します。[項目の追加]ボタンをクリックします。
4. OPC DA サーバーとの接続がすでに存在している場合には、引き続き手順 5 に進みます。まだ接続が設定されていない場合は、対応するメッセージが表示されます。[はい]をクリックします。[新規接続]ダイアログが表示されます。



この接続名として"OPCServer_WinCC"を入力します。[OK]をクリックします。

8.7 WinCC OPC DA サーバー

5. [タグの追加]ダイアログが表示されます。
[接頭語]フィールドに"Client_"を入力し、[接尾語]フィールドには"_xyz"を入力します。
[OPCServer_WinCC]接続を選択します。[終了]をクリックします。



6. [OPCServer.WinCC...]ダイアログで[戻る]ボタンをクリックします。"OPC 項目マネージャ"で[終了]をクリックして OPC 項目マネージャを閉じます。
7. グラフィックデザイナを起動し、画面を新たに開きます。画面内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで[I/O フィールド]オブジェクトを選択します。[I/O フィールド設定]ダイアログが開きます。
8. [タグ]フィールドに名前"Client_OPC_Server_Tag_xyz"を入力します。更新を"2 秒"に設定します。フィールドタイプを[I/O フィールド]に設定します。ダイアログを閉じて、画面を保存します。グラフィックデザイナの[有効]ボタンをクリックして、WinCC プロジェクトを有効にします。
9. WinCC OPC DA サーバーとクライアントの両方の I/O フィールドに、設定したタグの値が表示されます。WinCC OPC DA サーバーの I/O フィールドに値を新たに入力します。新規値は、WinCC OPC DA クライアントの I/O フィールドに表示されます。

下記も参照

WinCC OPC DA サーバー上の WinCC プロジェクトの設定方法 (ページ 714)

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

8.7.4.2 WinCC と SIMATIC NET FMS OPC サーバー間の接続

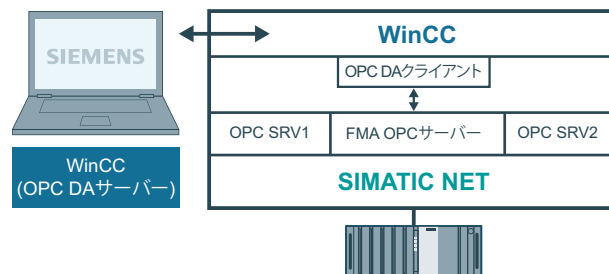
WinCC と SIMATIC NET FMS OPC サーバー間の接続例

はじめに

SIMATIC NET のインストール中に、インストールする OPC サーバーを選択できます。以下の例では、WinCC と SIMATIC NET FMS OPC サーバー間の接続を構成します。オートメーションデバイスのデータは、SIMATIC NET FMS OPC サーバーを使用して WinCC に供給されます。

この例では、WinCC を WinCC OPC DA クライアントとして使用します。OPC 項目マネージャは、オートメーションデバイスに対応して構成されたオブジェクトリストのインデックスを表示します。

タグの現在値は I/O フィールドに表示されます。SIMATIC NET FMS OPC サーバーのタグの値が変更されると直ちに、WinCC OPC DA クライアントのプロセス画像にこの新しい値が反映されます。反対に、I/O フィールドに入力された値はオートメーションデバイスに送られます。



必要条件

- コンピュータに WinCC、SIMATIC NET ソフトウェアがインストールされていること。
- SIMATIC NET FMS OPC サーバーが構成されていること。SIMATIC NET S7 OPC サーバーの設定の詳細については、SIMATIC NET マニュアルを参照してください。

構成手順

WinCC OPC DA クライアントの WinCC プロジェクトでは、以下のコンフィグレーションが必要です。

1. WinCC と SIMATIC NET FMS OPC サーバー間の接続の構成

通信マニュアル

通信マニュアルには、チャンネルコンフィグレーションの追加情報と各種の例が記載されています。このマニュアルは、下記のインターネットからダウンロードできます。

- <http://support.automation.siemens.com/>

アイテム番号による検索:

- A5E00391327

WinCC と SIMATIC NET FMS OPC サーバー間の接続の設定方法

はじめに

このセクションでは、WinCC OPC DA クライアントの WinCC プロジェクトで、FMS インデックスにアクセスする WinCC タグを設定します。このタグ値は I/O フィールドに表示されます。

必要条件

- WinCC OPC DA クライアントの WinCC プロジェクトに"OPC"チャンネルが追加されていること。

手順

1. WinCC OPC DA クライアント上の"OPC グループ(OPCHN Unit#1)"チャンネルユニットのショートカットメニューで[システムパラメータ]を選択します。OPC 項目マネージャが開きます。
2. 選択ダイアログで、OPC DA サーバーとして使用するコンピュータの名前を選択します。リストから[OPC.SIMATICNet]を選択します。
[サーバーの参照]ボタンをクリックします。[フィルタ規準]ダイアログが開きます。
3. [フィルタ規準]ダイアログで、[次へ->]ボタンをクリックします。[OPC.SIMATICNet..]ダイアログが開きます。選択リストに、設定されている FMS インデックスがすべて表示されます。インデックスを 1 つ選択します。[項目の追加]ボタンをクリックします。

4. SIMATIC NET FMS OPC サーバーとの接続がすでに存在している場合には、引き続き手順 5 に進みます。
まだ接続が設定されていない場合は、対応するメッセージが表示されます。
[はい]をクリックします。[新規接続]ダイアログが表示されます。



この接続名として"OPC_SimaticNET"を入力します。[OK]をクリックします。

5. [タグの追加]ダイアログを開きます。
[接頭語]フィールドに"Client_"を入力し、[接尾語]フィールドには"_xyz"を入力します。接続"OPC_SimaticNET"を選択します。[終了]をクリックします。
6. [OPC.SIMATICNet..]ダイアログで[戻る]ボタンをクリックします。"OPC 項目マネージャ"で[終了]をクリックして OPC 項目マネージャを閉じます。
7. グラフィックデザイナを起動し、画面を新たに開きます。画面内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで[I/O フィールド]オブジェクトを選択します。[I/O フィールド設定]ダイアログが開きます。
8. [タグ]フィールドにタグの名前を入力します。更新を[2 秒]に設定します。フィールドタイプを[I/O フィールド]に設定します。
9. [OK]をクリックしてダイアログボックスを閉じ、画面を保存します。グラフィックデザイナの[有効]ボタンをクリックして、WinCC プロジェクトを有効にします。
10. FMS インデックスの現在値は I/O フィールドに表示されます。値は 2 秒ごとに更新されます。I/O フィールドに値を入力します。変更した値は、オートメーションデバイスに引き渡されます。

下記も参照

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

8.7.4.3 WinCC と SIMATIC NET S7-OPC サーバー間の接続

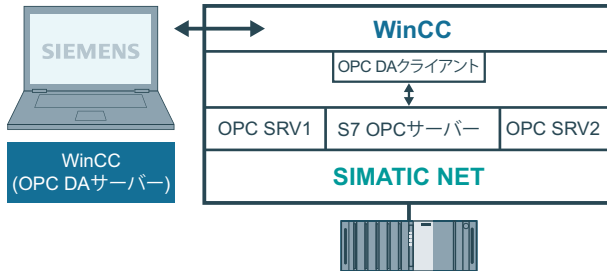
WinCC と SIMATIC NET S7 OPC サーバー間の接続例

SIMATIC NET のインストール中に、インストールする OPC サーバーを選択できます。以下の例では、WinCC と SIMATIC NET S7 OPC サーバー間の接続を構成します。オートメーションデバイスのデータは、SIMATIC NET S7 OPC サーバーを使用して WinCC クライアントに供給されます。

タグの現在値は、WinCC OPC クライアントの I/O フィールドに表示されます。SIMATIC NET S7 OPC サーバーのタグの値が変更されるとすぐに、この変更された値がプロセス画像

8.7 WinCC OPC DA サーバー

に表示されます。反対に、I/O フィールドに入力された値はオートメーションデバイスに送られます。



必要条件

- コンピュータに WinCC、SIMATIC NET ソフトウェアがインストールされていること。
- SIMATIC NET S7 OPC サーバーが構成されていること。SIMATIC NET S7 OPC サーバーの設定の詳細については、SIMATIC NET マニュアルを参照してください。

構成手順

WinCC と SIMATIC NET S7 OPC サーバー間の接続を確立するには、以下のコンフィグレーションが必要です。

1. SIMATIC NET S7 OPC サーバーへのタグの追加
2. SIMATIC NET S7 OPC サーバーのタグへのアクセスの構成

通信マニュアル

通信マニュアルには、チャンネルコンフィグレーションの追加情報と各種の例が記載されています。このマニュアルは、下記のインターネットからダウンロードできます。

- <http://support.automation.siemens.com/>

アイテム番号による検索:

- A5E00391327

SIMATIC NET S7 OPC サーバーへのタグの追加

はじめに

OPC 項目マネージャがタグを表示するには、SIMATIC NET S7 OPC サーバーのアドレススペースにタグを追加する必要があります。設定するには、"OPC Scout"プログラムを使用します。OPC Scout は、SIMATIC NET インストーラを使用して設定されます。この例では、オートメーションデバイス内のキーワード"0"がアドレス指定されます。

使用するパラメータの表

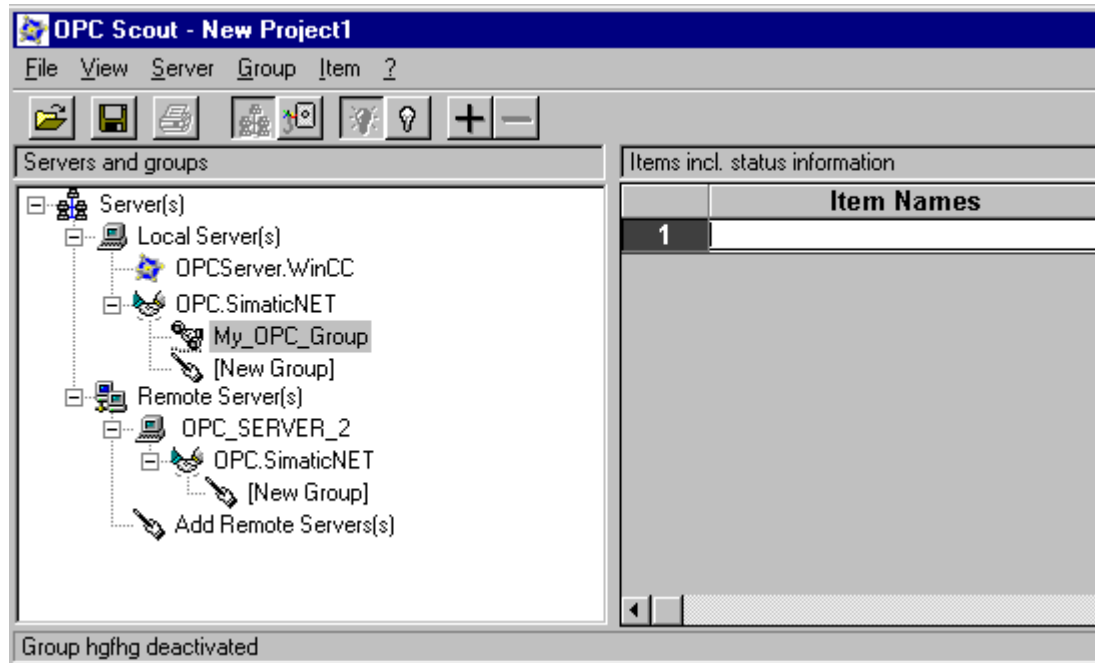
パラメータ	値
データタイプ	W
範囲バイト	0
番号値	1
項目別名	MW0

前提条件

- SIMATIC NET ソフトウェアで S7 接続を設定していること。詳細情報については『SIMATIC NET マニュアル』を参照してください。

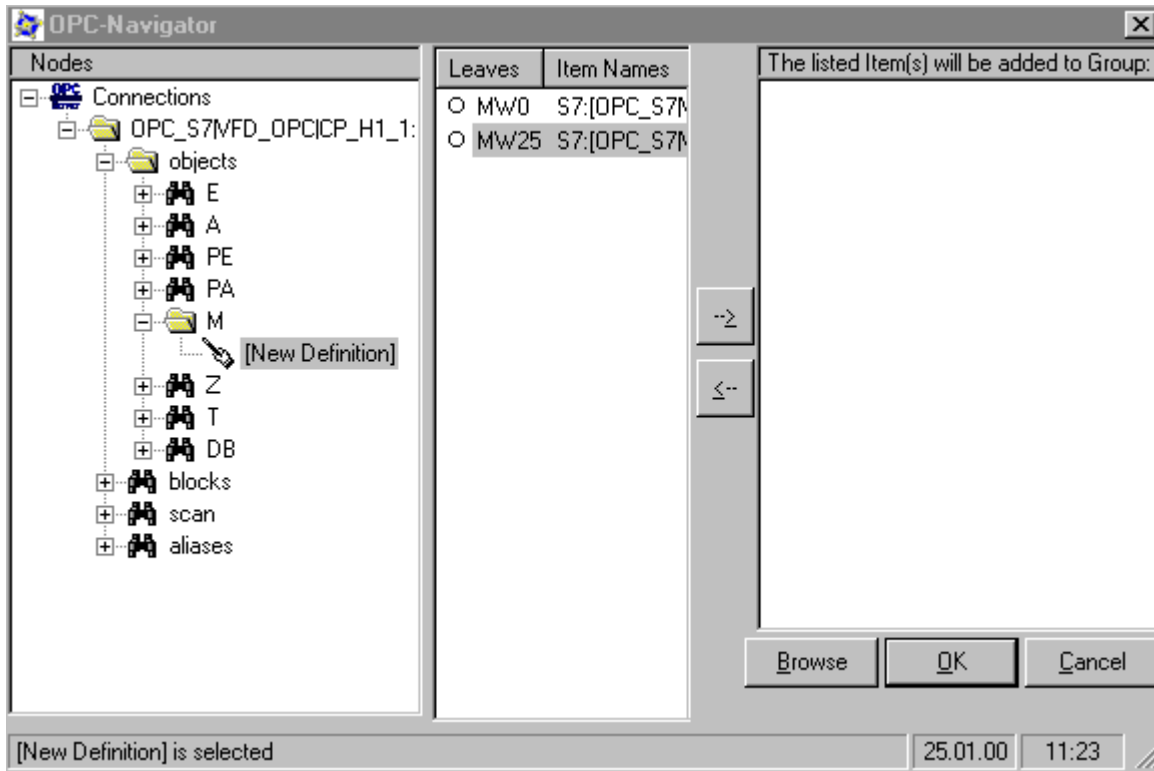
手順

1. [スタート]メニュー → 、[プログラム] → 、[SimaticNet] → 、[OPCServer] → 、[OPCScout]から[OPC Scout]を開きます。

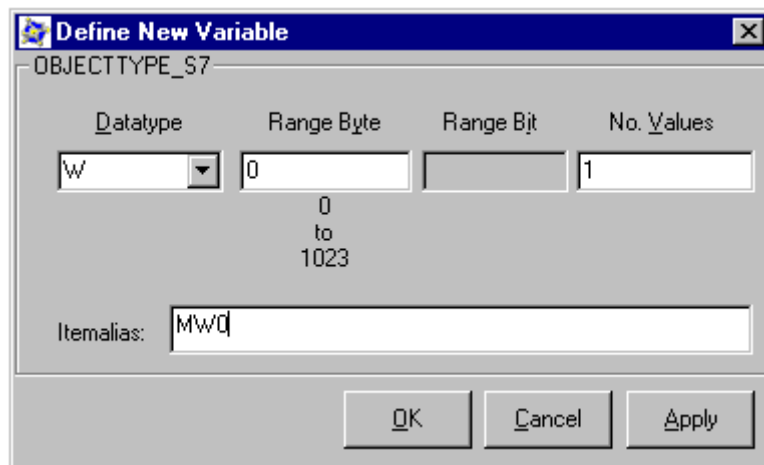


2. [ローカルサーバー]下の[OPC.SimaticNet]を選択します。同じコンピュータ上で SIMATIC S7 OPC サーバーが実行されていない場合ショートカットメニュー[サーバー]で[リモートサーバーの追加]を選択します。[リモートサーバーの追加]ダイアログで、OPC サーバーとして使用するコンピュータの名前を入力した後、[OK]をクリックしてこのダイアログを閉じます。
3. ショートカットメニュー[OPC.SimaticNet]で[接続]を選択します。[グループの追加]ダイアログが表示されます。グループの名前を入力します。[OK]をクリックしてダイアログを閉じます。

4. 追加したグループのショートカットメニューで[項目の追加]を選択します。"OPCNavigator"が開きます。



5. "OPCNavigator"で、[オブジェクト]下の[M] (マーカー)を選択します。[(新規定義)]をダブルクリックします。すると、[新規タグの定義]ダイアログが開きます。
6. [新規タグの定義]ダイアログで、テーブルのパラメータを入力します。



[OK]をクリックして、[新規タグの定義]ダイアログを閉じます。

7. OPCNavigator の[終了]エリアで、[MW0]タグにマークを付けます。[-->]ボタンをクリックします。OPCNavigator で[OK]をクリックします。

下記も参照

SIMATIC NET S7 OPC サーバーのタグへのアクセスの設定 (ページ 724)

SIMATIC NET S7 OPC サーバーのタグへのアクセスの設定

はじめに

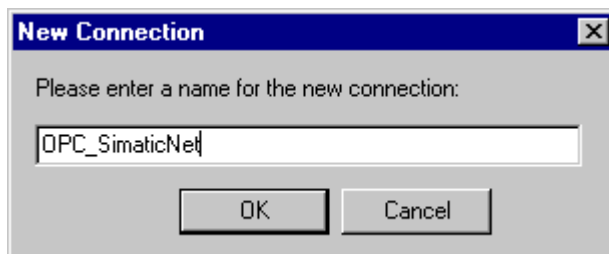
このセクションでは、WinCC OPC DA クライアントの WinCC プロジェクトで WinCC タグを設定します。このタグは、SIMATIC NET S7 OPC サーバーのアドレススペース内の[MW0]タグにアクセスします。このタグ値は I/O フィールドに表示されます。

前提条件

- OPC Scout を使用して[MW0]タグが作成されていること。
- WinCC OPC DA クライアントの WinCC プロジェクトに"OPC"チャンネルが追加されていること。

手順

1. "OPC グループ(OPCHN Unit#1)"のショートカットメニューで[システムパラメータ]を選択します。OPC 項目マネージャが開きます。
2. 選択ダイアログで、OPC サーバーとして使用するコンピュータの名前を選択します。リストから[OPC.SIMATICNet]を選択します。
[サーバーの参照]ボタンをクリックします。[フィルタ規準]ダイアログが開きます。
3. [フィルタ規準]ダイアログで、[次へ->]ボタンをクリックします。[OPC.SIMATICNet...]ダイアログが開きます。[MW0]タグを選択します。[項目の追加]ボタンをクリックします。
4. SIMATIC NET FMS OPC サーバーとの接続がすでに存在している場合には、引き続き手順 5 に進みます。
まだ接続が設定されていない場合は、対応するメッセージが表示されます。
[はい]をクリックします。[新規接続]ダイアログが表示されます。



この接続名として"OPC_SlimaticNET"を入力します。[OK]をクリックします。

5. [タグの追加]ダイアログを開きます。
[接頭語]フィールドに"Client_"を入力し、[接尾語]フィールドには"_xyz"を入力します。接続"OPC_SimaticNET"を選択します。[終了]をクリックします。

6. [OPC.SIMATICNet..]ダイアログで[戻る]ボタンをクリックします。"OPC 項目マネージャ"で[終了]をクリックして OPC 項目マネージャを閉じます。
7. [グラフィックデザイナー]を開始し、画面を開きます。画面内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで[I/O フィールド]オブジェクトを選択します。[I/O フィールド設定]ダイアログが開きます。
8. [タグ]フィールドに名前"Client_MW0_xyz"を入力します。更新を[2 秒]に設定します。フィールドタイプを[I/O フィールド]に設定します。
9. ダイアログを閉じて、画面を保存します。グラフィックデザイナーの[有効化]ボタンをクリックして、WinCC プロジェクトを有効にします。
10. WinCC OPC DA クライアントの I/O フィールドに、S7 タグの現在値が表示されます。値は 2 秒ごとに更新されます。I/O フィールドに値を入力します。変更した値は、オートメーションデバイスに引き渡されます。

下記も参照

SIMATIC NET S7 OPC サーバーへのタグの追加 (ページ 721)

WinCC OPC DA クライアント上での OPC チャンネルの設定 (ページ 258)

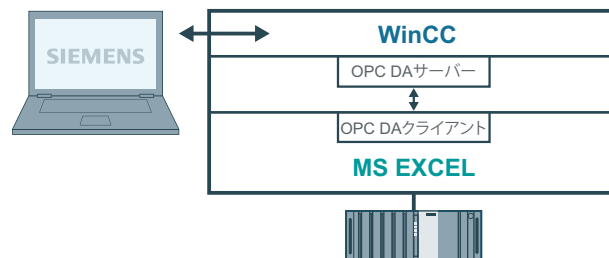
8.7.4.4 WinCC と Microsoft Excel 間の接続

WinCC と Microsoft Excel 間の接続例

はじめに

この例では、Visual Basic Editor を使用して Microsoft Excel で OPC DA クライアントを作成します。OPC DA クライアントは、WinCC OPC DA サーバーの WinCC プロジェクト内の WinCC タグを読み取って、この値をセル内に書き込みます。新しい値がセル内に入力されると、この値は WinCC OPC DA サーバーに引き渡されます。

この接続では、WinCC と Microsoft Excel が両方ともインストールされているコンピュータが使用されます。



設定手順

Microsoft Excel で以下を設定する必要があります。

1. Microsoft Excel の Visual Basic Editor で OPC DA クライアントを作成します。
2. Microsoft Excel の WinCC タグへのアクセスを設定します。

下記も参照

Microsoft Excel での WinCC タグへのアクセスの設定方法 (ページ 729)

Microsoft Excel での OPC DA クライアントの作成 (ページ 726)

Microsoft Excel での OPC DA クライアントの作成

はじめに

OPC DA クライアントとして Microsoft Excel を使用するには、Microsoft Excel の Visual Basic Editor で、専用のスクリプトを作成する必要があります。

必要条件

Microsoft Excel の Visual Basic Editor について基本的な知識を持っていること。

手順

1. Microsoft Excel を開き、新規ブックを表示します。
2. Visual Basic Editor の[ツール]メニューで、 ➡ [マクロ]をクリックします。Microsoft Excel 用の Visual Basic Editor を開きます。
3. Visual Basic Editor の[ツール]メニューで、[参照設定...]を選択します。[参照設定 - VBAProject]タブが表示されます。[参照可能なライブラリファイル]のリストで、[Siemens OPC DAAutomation 2.0]エントリを見つけます。該当するチェックボックスを選択します。[OK]をクリックします。
4. 下記のスクリプトをコピーします。このスクリプトはオンラインヘルプでのみ掲載されています。
5. Visual Basic Editor のプロジェクトウィンドウ内に表示されている[Sheet1]をダブルクリックして、新しいコードウィンドウを開きます。
6. このコードウィンドウにスクリプトを貼り付けます。
7. [ファイル]メニューの[保存]を選択します。[ファイル]メニューで[終了して Microsoft Excel に戻る]を選択します。

スクリプト例

```

Option Explicit
Option Base 1

Const ServerName = "OPCServer.WinCC"

Dim WithEvents MyOPCServer As OpcServer
Dim WithEvents MyOPCGroup As OPCGroup
Dim MyOPCGroupColl As OPCGroups
Dim MyOPCItemColl As OPCItems
Dim MyOPCItems As OPCItems
Dim MyOPCItem As OPCItem

Dim ClientHandles(1) As Long
Dim ServerHandles() As Long
Dim Values(1) As Variant
Dim Errors() As Long
Dim ItemIDs(1) As String
Dim GroupName As String
Dim NodeName As String

'-----
' Sub StartClient()
' 目的: OPC サーバーへの接続、グループの作成と項目の追加
'-----
Sub StartClient()
    ' On Error GoTo ErrorHandler
    '----- ClientHandle および GroupName は自由に選択できます
    ClientHandles(1) = 1
    GroupName = "MyGroup"
    '----- セル"A1"から ItemID を取得
    NodeName = Range("A1").Value
    ItemIDs(1) = Range("A2").Value
    '----- OPC サーバーのインスタンスを取得
    Set MyOPCServer = New OpcServer
    MyOPCServer.Connect ServerName, NodeName

    Set MyOPCGroupColl = MyOPCServer.OPCGroups
    '----- グループ追加のデフォルトアクティブステータスを設定
    MyOPCGroupColl.DefaultGroupIsActive = True
    '----- グループをコレクションに追加
    Set MyOPCGroup = MyOPCGroupColl.Add(GroupName)

    Set MyOPCItemColl = MyOPCGroup.OPCItems
    '----- 1つの項目を追加、ServerHandles が返されます
    MyOPCItemColl.AddItem 1, ItemIDs, ClientHandles, ServerHandles, Errors
    '----- サブスクライブされたグループが非同期の通知を受け取ります
    MyOPCGroup.IsSubscribed = True
    Exit Sub

```

8.7 WinCC OPC DA サーバー

```

ErrorHandler:
    MsgBox "Error: " & Err.Description, vbCritical, "ERROR"
End Sub

'-----
' Sub StopClient()
' 目的: オブジェクトの解放とサーバーからの接続解除
'-----
Sub StopClient()
    '----- グループとサーバーオブジェクトの解放
    MyOPCGroupColl.RemoveAll
    '----- サーバーからの接続解除とクリーンアップ
    MyOPCServer.Disconnect
    Set MyOPCItemColl = Nothing
    Set MyOPCGroup = Nothing
    Set MyOPCGroupColl = Nothing
    Set MyOPCServer = Nothing
End Sub

'-----
' Sub MyOPCGroup_DataChange()
' 目的: このイベントはグループ内の値、品質、タイムスタンプが変更されるとトリガされます
'-----
'----- OPC-DA オートメーション 2.1 がインストールされている場合、使用してください
Private Sub MyOPCGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long,
ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)
    '----- スプレッドシートのセル値を読み取った値に設定します
    Range("B2").Value = CStr(ItemValues(1))
    Range("C2").Value = Hex(Qualities(1))
    Range("D2").Value = CStr(TimeStamps(1))
End Sub

'-----
' Sub worksheet_change()
' 目的: このイベントはワークシートが変更されるとトリガされます。そのため新規値を書き込むことができます
'-----
Private Sub worksheet_change(ByVal Selection As Range)
    '----- セル"B3"が変更された場合にのみこの値を書き込みます
    If Selection <> Range("B3") Then Exit Sub
    Values(1) = Selection.Cells.Value
    '----- 同期モードで新規値を書き込みます
    MyOPCGroup.SyncWrite 1, ServerHandles, Values, Errors
End Sub

```

下記も参照

WinCC OPC DA サーバー上の WinCC プロジェクトの設定方法 (ページ 714)

Microsoft Excel での WinCC タグへのアクセスの設定方法

はじめに

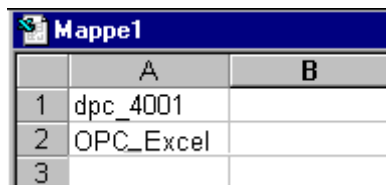
Excel OPC DA クライアントは、WinCC OPC DA サーバーの WinCC タグを読み取って、このタグの値をセル内に書き込みます。WinCC OPC DA サーバーの WinCC プロジェクトでは、I/O フィールドにこのタグの値が表示されます。セル内のタグ値が変更されると、それに応じて WinCC OPC DA サーバーの I/O フィールド内の値が変更されます。

前提条件

- WinCC OPC DA サーバーの WinCC プロジェクトで、"符号付き 16 ビット値"データタイプの"OPC_Excel"内部タグが設定されていること。
- WinCC OPC DA サーバーの WinCC プロジェクトで、I/O フィールドに"OPC_Excel"タグ値が書き込まれていること。
- WinCC OPC DA サーバーの WinCC プロジェクトが有効になっていること。

手順

1. Microsoft Excel で、OPC サーバーとして使用するコンピュータの名前をセル"A1"に入力します。セル"A2"に、タグ名"OPC_Excel"を入力します。



	A	B
1	dpc_4001	
2	OPC_Excel	
3		

2. Excel の[ツール]メニューで[マクロ] → [マクロ...]を選択します。[マクロ]ダイアログが開きます。マクロのリストで[Sheet1.StartClient]を選択します。[実行]をクリックして OPC クライアントを起動します。
3. タグの値はセル"B2"に、品質コードはセル"C2"に、タイムスタンプはセル"D2"に書き込まれます。
4. セル"B 3"に新規値を入力します。変更した値は、WinCC OPC サーバーの I/O フィールドに表示されます。
5. Excel の[ツール]メニューで[マクロ] → [マクロ...]を選択します。[マクロ]ダイアログが開きます。マクロのリストで[Sheet1.StopClient]を選択します。[実行]をクリックして OPC クライアントを停止します。

8.8 WinCC OPC HDA サーバー

8.8.1 WinCC OPC HDA サーバーの機能

はじめに

WinCC OPC HDA サーバーは、DCOM アプリケーションの 1 つであり、アーカイブシステムからの必要データを OPC HDA クライアントに供給します。項目ハンドルを使用してデータにアクセスします。読取りアクセスまたは書込みアクセスが可能です。このデータを分析することもできます。

WinCC OPC HDA サーバーは、OPC Historical Data Access 1.20 の仕様をサポートしています。これは、準拠テストにより確認されています。

以下の章では、WinCC OPC HDA サーバーでサポートされている属性、集約、およびファンクションだけでなく、データ構造の設計についても説明します。詳細な説明ではなく、最も重要な情報についての要約です。詳細については、"OPC Historical Data Access 1.20"の仕様を参照してください。

インストール

WinCC のインストール中、WinCC OPC HDA サーバーを選択できます。WinCC アーカイブシステムへのアクセスを実行するときに書込みファンクションを使用するかどうか選択できます。WinCC のインストールが終了すると直ちに、WinCC OPC DA サーバーを使用できます。このとき、別の設定を実行する必要はありません。

書込みアクセスを指定せずにインストールする場合、WinCC アーカイブシステム内のデータには、読取りと分析以外実行できません。書込みアクセスの場合、WinCC アーカイブシステムのデータを分析、追加、削除、および更新できます。

WinCC OPC HDA サーバーは、WinCC サーバーまたは WinCC クライアント上にインプリメントできます。

ライセンス

WinCC OPC HDA サーバーを操作するには、OPC HDA サーバーとしてインプリメントされた各 WinCC コンピュータに、次のライセンスをインストールしておく必要があります。

- WinCC の有効な RT ライセンス
- WinCC オプション接続性パック

OPC HDA クライアント

OPC Historical Data Access 1.20 仕様に準拠する OPC HDA クライアントはすべて、WinCC OPC HDA サーバーにアクセスできます。OPC HDA クライアントは、ユーザー自身で作成することもできます。独自の OPC HDA クライアントを作成することにより、ユーザー固有の必要条件をほとんど満たすことができます。

OPC HDA クライアントの使用事例には、以下のものがあります。

- アーカイブデータの分析と評価
- 各種 OPC HDA サーバーのアーカイブの統計プロセスコントロール

OPC HDA クライアントを使用して履歴の値を要求するには、設定中、以下に気をつける必要があります。

- 次のクエリが送られる前にクライアントが要求されたデータを受け取ることができるように、クエリサイクルを選択してください。サイクルが短すぎると、データを受け取るまでにかなりの時間がかかる場合があります。
- WinCC サーバーの CPU 負荷は、クエリに対するタグの数によって決まります。

設定されたスワップアウトによるサイクリックアーカイブへの書込みアクセス

ランタイムでは、データは WinCC サーバー上のサイクリックアーカイブで修正されます。

変更がスワップアウトされたアーカイブに受け入れられるのは、データが作成された直後に変更された場合のみです。

サイクリックアーカイブの関連するアーカイブセグメントがすでにスワップアウトされている場合、スワップアウトされたアーカイブでの変更はその後実行されません。WinCC サーバー上でアーカイブセグメントを削除すると、修正されたデータも削除されます。

下記も参照

品質コード (ページ 738)

WinCC OPC HDA サーバーのデータ構造 (ページ 732)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2 WinCC OPC HDA サーバーのデータ構造

8.8.2.1 WinCC OPC HDA サーバーのデータ構造

はじめに

WinCC OPC HDA サーバー上のデータを構造化します。以下に使用可能なデータ構造を一覧表示します。詳細な説明ではなく、最も重要な情報についての要約です。詳細については、"OPC Historical Data Access 1.20"の仕様を参照してください。

データ構造

	説明
属性	未処理データの品質特性を追加指定します。属性には、データタイプ、アーカイブの仕様などが含まれます。詳細については、サポートされている属性の概要を参照してください。
集約	所定の時間間隔に対応する未処理データの要約。総計には、平均、最小、最大値などが含まれます。詳細については、サポートされている総計の概要を参照してください。
開始時間/終了時間	時間間隔の開始ポイントと終了ポイントを設定します。
境界値	境界値は、開始時と終了時に記録される値です。これらの値を使用できない場合、境界値として、該当する時間に近い値が使用されます。
未処理データ	未処理データは、特定の時間間隔に対応した WinCC アーカイブシステムの未処理データです。タイムスタンプおよび品質評価などがあります。
項目ハンドル	項目ハンドルは、WinCC アーカイブタグへの一意の割り付けを指します。
項目 ID	項目 ID は、WinCC アーカイブタグの一意の識別名を指します。項目 ID を使用すれば項目ハンドルを取得できます。

下記も参照

サポートされている関数シグネチャの概要 (ページ 735)

WinCC OPC HDA サーバーの時間形式 (ページ 736)

サポートされている属性の概要 (ページ 733)

サポートされている集約の概要 (ページ 733)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.2 サポートされている属性の概要

はじめに

以下の表は、WinCC OPC HDA サーバーでサポートされている属性を示しています。詳細については、"OPC Historical Data Access 1.20"の仕様を参照してください。

属性

属性	属性 ID	説明
項目 ID	OPCHDA_ITEMID	アクセスする WinCC アーカイブタグを示します。
項目データタイプ	OPCHDA_DATA_TYPE	WinCC アーカイブタグのデータタイプを示します。
説明	OPCHDA_DESCRIPTION	WinCC アーカイブタグの記述を戻します。この記述は、WinCC タグロギングで定義されます。
エンジニアリング単位	OPCHDA_ENG_UNITS	測定単位の表示を設定します。WinCC タグロギングでラベリングが定義されます。

下記も参照

WinCC OPC HDA サーバーのデータ構造 (ページ 732)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.3 サポートされている集約の概要

はじめに

以下の表に、WinCC OPC HDA サーバーでサポートされている集約を一覧表示します。詳細については、"OPC Historical Data Access 1.20"の仕様を参照してください。

集約

集約	説明
OPCHDA_COUNT	指定の時間間隔に対応した未処理データカウントを戻します。
OPCHDA_START	時間間隔の先頭で、未処理データの初期値を戻します。
OPCHDA_END	時間間隔の終わりに、未処理データの最終値を戻します。
OPCHDA_AVERAGE	指定の時間間隔に対応した未処理データの平均値を戻します。
OPCHDA_TIMEAVERAGE	指定の時間間隔に対応した未処理データの時間加重平均を戻します。
OPCHDA_TOTAL	指定の時間間隔に対応した合計値を戻します。
OPCHDA_STDEV	指定の時間間隔に対応した未処理データの標準偏差を戻します。
OPCHDA_MINIMUMACTUALTIME	指定の時間間隔に対応した未処理データとそのタイムスタンプの最小値を戻します。
OPCHDA_MINIMUM	指定の時間間隔に対応した未処理データの最小値を戻します。
OPCHDA_MAXIMUMACTUALTIME	指定の時間間隔に対応した未処理データとそのタイムスタンプの最大値を戻します。
OPCHDA_MAXIMUM	指定の時間間隔に対応した未処理データの最大値を戻します。
OPCHDA_DELTA	指定の時間間隔に対応した未処理データ内の先頭ポイントと最終ポイントの差を戻します。
OPCHDA_REGSLOPE	指定の時間間隔に対応した未処理データの回帰線の勾配を戻します。
OPCHDA_REGCONST	指定の時間間隔に対応した未処理データの回帰値を戻します。
OPCHDA_REGDEV	指定の時間間隔の未処理データに対応した回帰の標準偏差を戻します。
OPCHDA_VARIANCE	指定の時間間隔に対応した未処理データの分散を戻します。
OPCHDA_RANGE	指定の時間間隔に対応した未処理データの OPCHDA_MAXIMUM と OPCHDA_MINIMUM の差を戻します。

集約	説明
OPCHDA_DURATIONGOOD	未処理データの品質が適切だった期間を返します。この期間は、秒単位で指定されます。
OPCHDA_DURATIONBAD	未処理データの品質が不適切だった期間を返します。この期間は、秒単位で指定されます。
OPCHDA_PERCENTGOOD	適切な品質の未処理データの割合を返します。
OPCHDA_PERCENTBAD	不適切な品質の未処理データの割合を返します。
OPCHDA_WORSTQUALITY	指定の時間間隔に対応した未処理データの最悪品質を返します。

下記も参照

WinCC OPC HDA サーバーのデータ構造 (ページ 732)

WinCC OPC HDA サーバーの機能 (ページ 730)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.4 サポートされているファンクションの概要

はじめに

以下の表に、WinCC OPC HDA サーバーでサポートされているファンクションを一覧表示します。これらのファンクションは、データ交換のために OPC HDA クライアントで使用できます 詳細については、"OPC Historical Data Access 1.20"の仕様を参照してください。

読み取り

機能	説明
ReadRaw	指定の時間間隔に対応した未処理データとその品質とタイムスタンプを返します。
ReadProcessed	指定の時間間隔に対応した計算値、値の品質、およびタイムスタンプを返します。計算値は、選択した集約によって違ってきます。
ReadAtTime	特定時間間隔に対応した未処理データとその品質とタイムスタンプを返します。値を使用できない場合、このポイントの値が補間されます。
ReadAttribute	指定の時間間隔に対応した項目属性とタイムスタンプを返します。

下記も参照

WinCC OPC HDA サーバーの機能 (ページ 730)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2.5 WinCC OPC HDA サーバーの時間形式

はじめに

開始時間と終了時間を設定すると、WinCC OPC HDA サーバーで時間間隔が指定されます。指定の時間間隔により、履歴データの観測期間が決定されます。時間を指定する際には、特定形式を保持する必要があります。

時間の指定には、以下のオプションを使用できます。

- UTC に対応した絶対値
- サーバーのローカル時間と相対的な時間

UTC に対応した絶対値

デフォルトでは、WinCC OPC HDA サーバーはそのタイムベースとして協定世界時(UTC)を使用します。この時間は、グリニッジ標準時(つまり、中央ヨーロッパ標準時から 1 時間を引いたもの)に対応します。

時間形式

YYYY/MM/DD hh:mm:ss.msmsms

パラメータ

YYYY = 年

MM = 月

DD = 日

hh = 時間

mm = 分

ss = 秒

ms = ミリ秒

入力例

2002/06/10 09:27:30.000

ローカル時間に相対した時間の指定

このオプションでは、サーバーのローカル時間に相対した時間が入力されます。ローカル時間帯は、コンピュータの[日付と時刻]コントロールパネルで設定されます。

時間形式

キーワード +/-オフセット 1 +/-オフセット(n)

オフセットとは、サーバーのローカル時間との差を指します。

キーワード

NOW = サーバーの現在のローカル時間

SECOND = 現在の秒

MINUTE = 現在の分

HOURL = 現在の時間

DAY = 現在の日

WEEK = 現在の週

MONTH = 現在の月

YEAR = 現在の年

オフセット

+/-S = 秒差

+/-M = 分差

+/-H = 時間差

+/-D = 日差

+/-W = 週差

+/-MO = 月差

+/-Y = 年差

例 :

DAY - 1D = 前日

DAY-1D + 7H30 = 前日の 7:30

MO-1D+5H = 先月の最終日の 5:00

NOW-1H15M = 1 時間 15 分前

YEAR+3MO= 今年の 4 月

下記も参照

WinCC OPC HDA サーバーの機能 (ページ 730)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.3 品質コード

はじめに

品質コードは、未処理データのステータスと品質を評価するのに使用されます。 OPC の品質コードについては、『データアクセスカスタムインターフェース標準バージョン 3.00』仕様の「6.8 OPC 品質タグ」を参照してください。

WinCC OPC HDA サーバーの品質コード

コード	OPC	説明	品質
0x00040000 0	OPCHDA_RAW	未処理データ送信の品質を示します。	GOOD BAD UNCERTAIN
0x00080000 0	OPCHDA_CALCULATED	計算したデータ送信の品質を示します。	GOOD BAD UNCERTAIN
0x00100000 0	OPCHDA_NOBOUND	開始ポイントと終了ポイントで境界値が検出されませんでした。	BAD
0x00200000 0	OPCHDA_NODATA	指定の時間間隔中に未処理データが検出されませんでした。	BAD
0x00400000 0	OPCHDA_DATALOST	選択した間隔中に未処理データの一部がアーカイブされませんでした。	BAD

下記も参照



www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.4 サポートされている書込みアクセス

はじめに



以下の表は、WinCC OPC HDA サーバーでサポートされている書込みアクセスを示しています。

テーブルエレメント:

	説明
サイクリックアーカイブ	アーカイブしたプロセス値はサイクリックアーカイブに保存されます。サイクリックアーカイブは、数量を設定できるデータバッファで構成されます。データバッファはサイズと時間周期(例: 日単位)で定義されます。データバッファがすべて一杯になると、最初のデータバッファ内のプロセスデータが上書きされます。
スワップ後のサイクリックアーカイブ	上書きプロセスによりデータバッファ内のプロセスデータが上書きされないように、プロセスデータをスワップ(エクスポート)できます。
	WinCC でサポートされています。
	WinCC でサポートされていません。

書込みアクセス

後でプロセス値を追加

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
可能	不可		サイクリックアーカイブに時間周期が含まれている場合、後でプロセス値を追加できます。
可能	可能		対応する時間周期のデータバッファが、アーカイブバックアップにスワップされます。アーカイブバックアップに後でプロセス値を追加することはできません。

8.8 WinCC OPC HDA サーバー

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
不可	不可	☹	サイクリックアーカイブを使用できません。プロセス値を保存できません。
不可	可能	☹	サイクリックアーカイブを使用できません。プロセス値を保存できません。

ランタイム中のプロセス値の追加

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
可能	不可	☺	サイクリックアーカイブで現在有効になっているデータバッファにプロセス値が追加されます。

今後のプロセス値の挿入

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
可能	不可	☹	書込みアクセス中に、値は今後何も追加されません。
不可	不可	☹	書込みアクセスでは、値は今後何も追加されません。

プロセス値の削除

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
可能	不可	😊	サイクリックアーカイブに時間周期が指定されている場合、プロセス値を削除できます。
可能	可能	😞	対応する時間周期のデータバッファが、アーカイブバックアップにスワップされます。アーカイブバックアップからプロセス値を削除できます。
不可	不可	😞	サイクリックアーカイブを使用できません。プロセス値を保存できません。
不可	可能	😞	サイクリックアーカイブを使用できません。プロセス値を保存できません。

プロセス値の編集

サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCC でサポートされています。	説明
可能	不可	😊	サイクリックアーカイブに時間周期が指定されている場合、プロセス値を編集できます。
可能	可能	😞	対応する時間周期のデータバッファが、アーカイブバックアップにスワップされます。アーカイブバックアップではプロセス値を編集できません。
不可	不可	😞	サイクリックアーカイブを使用できません。プロセス値を保存できません。
不可	可能	😞	サイクリックアーカイブを使用できません。プロセス値を保存できません。

8.8.5 OPC HDA 接続例

8.8.5.1 OPC HDA 接続例

はじめに

以下の例では、WinCC と OPC HDA クライアント間の接続を設定します。WinCC OPC HDA サーバーを使用すれば、WinCC アーカイブシステムのデータが使用できるようになります。OPC HDA クライアントは、項目ハンドルを使用してデータにアクセスします。設定プロセスを簡単にするために、OPC HDA ブラウザを使用します。

OPC 協会の OPC HDA クライアントを使用します。OPC Historical Data Access 1.20 仕様に準拠する OPC HDA クライアントはすべて、WinCC OPC HDA サーバーにアクセスできます。

前提条件

- WinCC OPC HDA サーバーの WinCC プロジェクトで、"符号なし 16 ビット値"データタイプの"OPC_HDA"内部タグが作成されていること。
- WinCC アーカイブシステムで、"HDA_ProcessValueArchive"プロセス値アーカイブが作成されていること。
- プロセス値アーカイブ"HDA_ProcessValueArchive"で、"OPC_HDA_Tag" WinCC アーカイブタグが作成されていること。WinCC アーカイブタグが内部タグ"OPC_HDA"にリンクされていること。
- ランタイムリストで、タグロギングランタイムを起動しており、グラフィックスランタイムが無効にされていること。
- WinCC OPC HDA サーバーの WinCC プロジェクトを起動していること。

設定の手順

WinCC を OPC HDA と接続するには、以下の設定が必要です。

1. HDA サーバーブラウザを使用した WinCC アーカイブタグへのアクセスの設定
2. WinCC アーカイブタグからの値の読取り

下記も参照

HDA サーバーブラウザを使用した WinCC アーカイブタグへのアクセスの設定方法 (ページ 744)

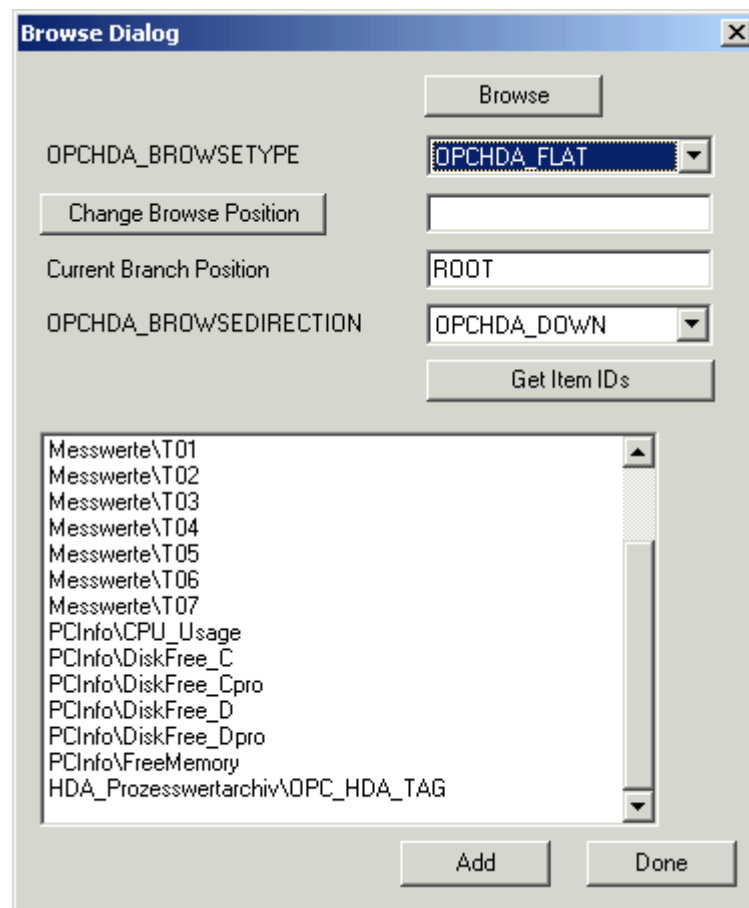
HDA サーバーブラウザ (ページ 743)

WinCC アーカイブタグの値の読取り (ページ 746)

8.8.5.2 HDA サーバーブラウザ

はじめに

OPC HDA クライアントは、項目ハンドルを使用してタグ値にアクセスします。設定を簡単にするために、WinCC OPC HDA サーバーではブラウザ機能をサポートしています。OPC HDA クライアントは、HDA サーバーブラウザを使用すれば、WinCC OPC HDA サーバーのアドレススペースを検索できます。このデータは、プロセス値アーカイブにより階層形式でリストされます。



注記

HDA サーバーブラウザを使用せずに WinCC アーカイブタグにアクセスするには、項目 ID を手動で設定する必要があります。

WinCC アーカイブタグをアドレス指定する場合、パスにはコンピュータ名(サーバー接頭語)が指定されます。項目 ID では、以下の構文が使用されます。Server-prefix::process_value_archive\WinCC_archive_tag.

下記も参照

HDA サーバーブラウザを使用した WinCC アーカイブタグへのアクセスの設定方法 (ページ 744)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.5.3 HDA サーバーブラウザを使用した WinCC アーカイブタグへのアクセスの設定方法

はじめに

このセクションでは、OPC HDA クライアントを使用して、WinCC アーカイブタグにアクセスします。

OPC 協会の OPC HDA クライアントを使用します。

HDA サーバーブラウザを使用してアクセスを設定します。

注記

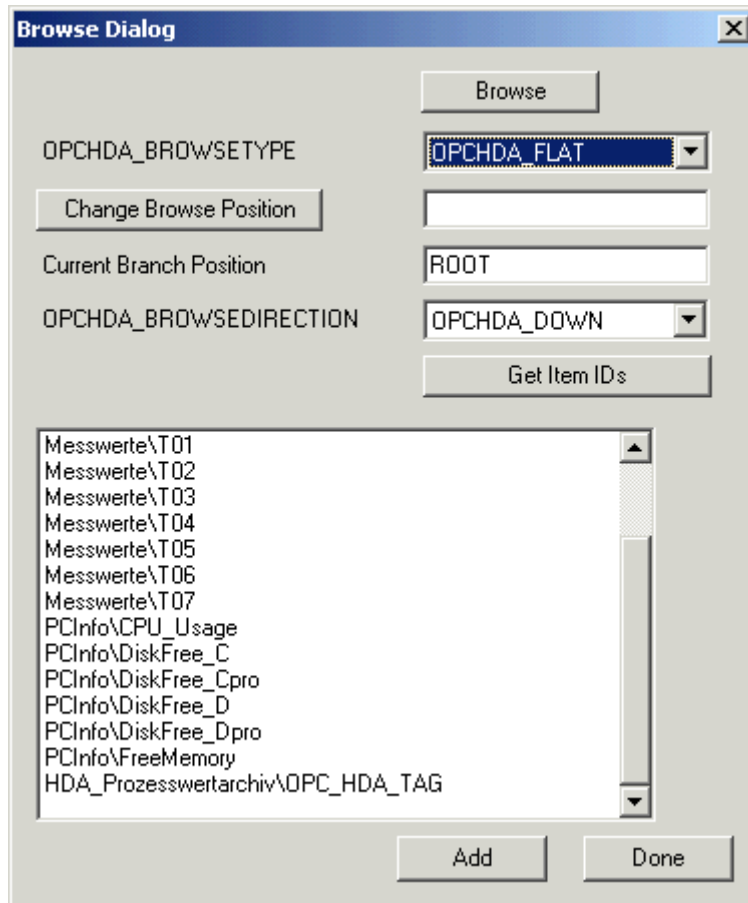
デモクライアント

ここに記述されている OPC HDA クライアントは、OPC 協会のデモクライアントです。このソースは、インターネットで <http://www.opcfoundation.org> を参照してください。

手順

1. WinCC インストールパス「Siemens\WinCC\documents\English」から「SampleClientHDA.exe」ファイルを選択するフォルダへコピーします。
2. "SampleClientHDA.exe"ファイルをダブルクリックします。
[HDA クライアント]プログラムが開きます。
3. [サーバー名]領域でエントリ[OPCServerHDA.WinCC.1]を選択します。

4. [接続]をクリックします。
これによって表示されるダイアログを、確認します。
5. HDA クライアントの[参照]をクリックします。
[ダイアログの参照]ダイアログが開きます。



6. [OPCHDA_BROWSETYPE]フィールドで[OPCHDA_FLAT]を選択します。
7. [選択]ウィンドウでエントリ[HDA_ProcessValueArchive_HDA_TAG]を選択します。
8. [追加]をクリックした後、[完了]をクリックして、ダイアログを閉じます。

詳細については、<http://www.opcfoundation.org> を参照してください。

下記も参照

WinCC アーカイブタグの値の読取り (ページ 746)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.5.4 WinCC アーカイブタグの値の読取り

はじめに

このセクションでは、WinCC アーカイブタグにアクセスして読み取る方法について説明します。

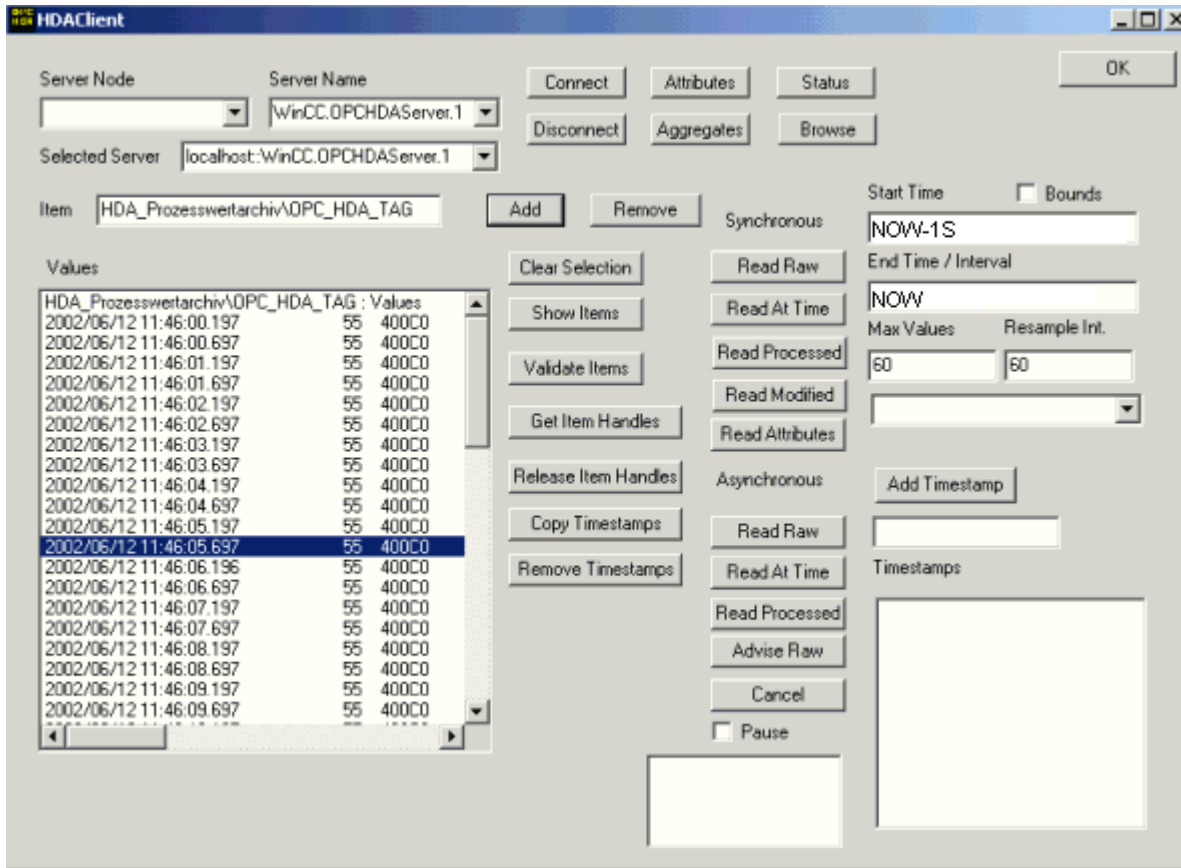
必要条件

- OPC HDA クライアントが実行されていることが必要になります。

手順

1. HDA クライアントで[項目の表示]をクリックします。
2. HDA クライアントで[項目ハンドルの取得]をクリックします。
3. [値]選択フィールドで[HDA_ProcessValueArchive_HDA_TAG]をダブルクリックします。

4. [開始時間]フィールドに"NOW-10S"と入力します。[終了時間]フィールドに"NOW"と入力します。



5. [未処理データの読み込み]をクリックします。これらの値、その品質コード、およびタイムスタンプは、[値]選択フィールドに表示されます。

8.8.6 WinCC の OPC HDA サーバーの非サイクリックロギング用特殊機能

はじめに

WinCC のタグロギングは、サイクリックまたは非サイクリックに実行されます。WinCC OPC HDA サーバーは、タグのロギング方法によって動作が異なります。

- OPC HDA サーバーは、サイクリックにログされたすべての値について、OPC 協議会の HDA 仕様に準拠して動作します。OPC 集約は、直線的に補間されています。
- 非サイクリックにログされるタグは、OPC 協議会の HDA 仕様には含まれていません。OPC 集約は、増分的に補間されています。特に、タグが長時間変更されなかったとき、期間中データは使用できません。有効なデータを取得するには、以下の事項を考慮に入れる必要があります。

注記

OPC HDA サーバーが、非サイクリックにログされたタグに対し OPC に準拠していない。OPC 協議会の HDA 仕様では、非サイクリックにログされたタグは認識されていないため、アーカイブサーバーは非サイクリックにログされたタグを処理できません。サポートされている集約は、OPCHDA 仕様に準拠して計算されます。非明示的な関数の呼び出しはサポートされていません。

注記

プロセス値アーカイブへの書き込みアクセスを有効にすると、意味のない値が追加されることがあります。

非サイクリックにログされるタグの設定

非サイクリックにログされるタグの設定には、[セグメント変更後のアーカイブ]設定をタグ用に有効にする必要があります。これにより、セグメントが変更されたときに、最新の有効な値が新しいログに入力されます。

WinCC の OPC HDA サーバーでサポートされている、非サイクリックロギング用集約

OPC HDA サーバーは以下の集約をサポートします。

- OPCHDA_MINIMUM
- OPCHDA_MAXIMUM
- OPCHDA_AVERAGE
- OPCHDA_END
- OPCHDA_INTERPOLATIVE

- OPCHDA_TIMEAVERAGE
- OPCHDA_TOTAL
- OPCHDA_DURATIONGOOD
- OPCHDA_PERCENTGOOD

WinCC の OPC HDA サーバーでサポートされている、非サイクリックにログされるタグ用機能

- "境界"のみ付き ReadRaw タグの ReadRaw は、ログされた値が変更されていないエリアの、最後に実際に保存された値を見つけるために"境界"付きで実行されます。
- ReadProcessed
- DeleteRaw
- DeleteAtTime
- Insert
- InsertReplace
- Replace

非サイクリックにログされるタグのための集約の計算

集約の計算は、実際に格納された値に加え、計算用の仮想データポイントを含む、拡張 "RawData" データレコードを基に行われます。 WinCC OPC HDA サーバーは、"ReadProcessed" の要件に対応して "RawData" を処理します。 計算に必要な仮想データポイントは、境界の実際のデータポイントから作成されます。 仮想データポイントには、次の重要なポイントが含まれています。

- "StartTime" の値
- "EndTime" の値
- 間隔限界の値

例

"00:59:00"、"01:02:00"、"01:03:00"の値が、非サイクリックタグロギングのタグ用に保存されます。OPC HDA クライアントは、"ReadProcessed"に、以下のパラメータのある集約を要求します。

- StartTime = 01:00:00
- EndTime = 01:04:00
- Interval = 00:02:00

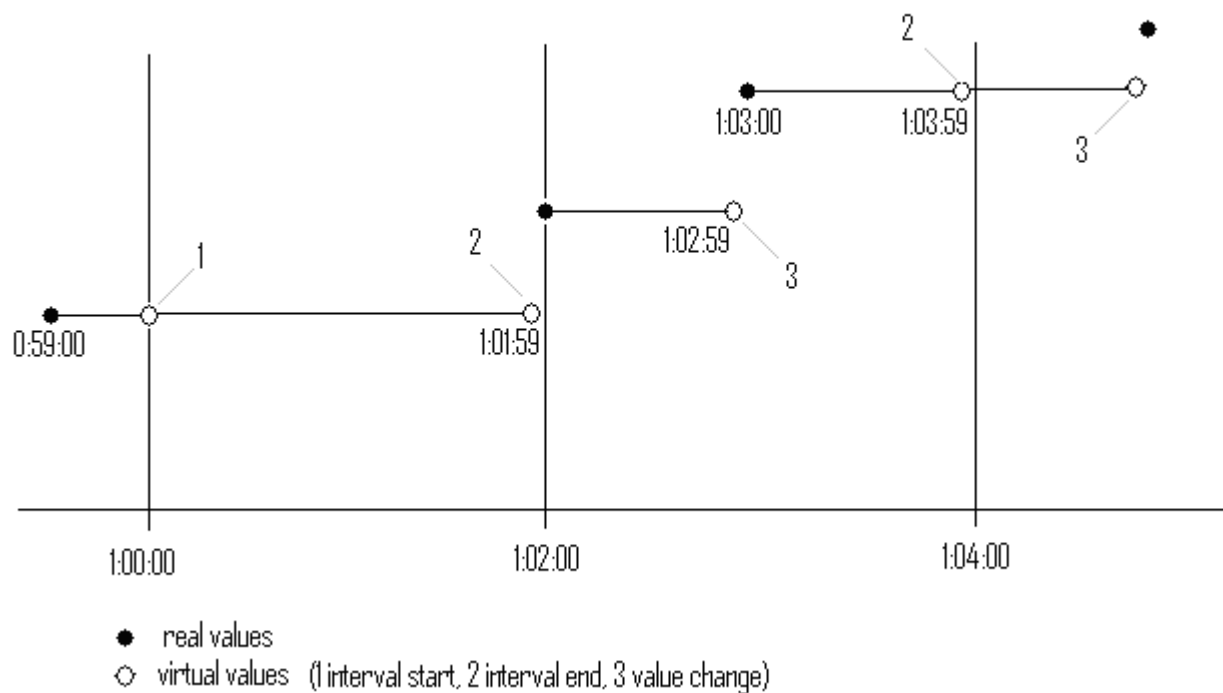
注記

この期間は、限界("EndTime"/"Interval")で仮想値を生成するとき、計算上、限界でのタイムスタンプより常に 1 μs 短くなります。

次の表では、分かりやすいよう、1 秒のデルタが使用されています。次の図に例を示します。

OPC サーバーは、集約の計算に次の"RawData"を使用します。

数値	タイムスタンプ	実際に保存された値	生成された仮想値
1	00:59:00	1.00	
2	01:00:00		1.00
3	01:01:59		1.00
4	01:02:00	2.00	
5	01:02:59		2.00
6	01:03:00	3.00	
7	01:03:59		3.00



8.9 WinCC OPC A&E サーバー

8.9.1 WinCC OPC A&E サーバーの機能

はじめに

WinCC OPC A&E サーバーは DCOM アプリケーションの 1 つです。定期購読によって、OPC A&E クライアントには、WinCC メッセージのステータス変更が通知されます。OPC A&E クライアントは、定期購読にフィルタを適用できます。このフィルタにより、表示するメッセージと属性が決定されます。

WinCC OPC A&E サーバーは、OPC Alarm&Event 1.10 仕様をサポートしています。これは、準拠テストにより確認されています。

以下の章では、WinCC OPC A&E サーバーでサポートされている属性のみでなく、OPC A&E 上での WinCC メッセージシステムの表示についても説明します。詳細な説明ではなく、最も重要な情報についての要約です。詳細については、"OPC Alarms & Events 1.10"の仕様を参照してください。

インストール

WinCC のインストール中、WinCC OPC A&E サーバーを選択できます。インストールが終了すると直ちに、WinCC OPC A&E サーバーを使用できます。このとき、別の設定を追加して実行する必要はありません。

WinCC OPC A&E サーバーは、WinCC サーバーおよび WinCC クライアントに実装できます。

ライセンスング

WinCC OPC A&E サーバーを操作するには、OPC A&E サーバーとして実装された各 WinCC サーバーに、次のライセンスをインストールしておく必要があります。

- WinCC の有効な RT ライセンス
- WinCC オプション接続性パック

サーバーのタイプ

WinCC OPC A&E サーバーは、条件イベントと単純なイベントをサポートしています。さらに、追跡用イベントもあります。

条件関連イベントサーバー

条件関連イベントサーバーを使用して、イベントに条件を関連付けます。たとえば、タグの限界値違反を条件に設定することができます。境界値を超えると直ちに、WinCC にメッセージが表示されます。OPC A&E では、このメッセージはアラームとして表示されます。

単純なイベントサーバー

単純なイベントは、OPC A&E クライアントにイベントを通知するメッセージです。単純なイベントには、たとえばプログラムの起動や終了があります。

注記

冗長システムを使用している場合、以下の点に注意してください。

内部タグと相互接続されている単純イベントは、タグの更新時に 2 回送信されます。

最初のメッセージはマスタによりトリガされ、2 番目のメッセージはスタンバイによりトリガされます。

追跡用イベントサーバー

プロセスで変更が実行されると、OPC A&E クライアントはメッセージを受信します。たとえば、レギュレータの調整が挙げられます。

OPC A&E クライアント

OPC Alarms & Events 1.10 仕様に準拠する OPC A&E クライアントはすべて、WinCC OPC A&E サーバーにアクセスできます。OPC A&E クライアントは、ユーザー自身で作成することもできます。私有の OPC クライアントを作成することにより、ユーザー固有の必要条件をほとんど満たすことができます。たとえば、OPC A&E クライアントは、複数の OPC A&E サーバーからのアラームの分析と共通アーカイブに使用できます。

下記も参照

OPC A&E の品質コード (ページ 761)

OPC A&E を使用した WinCC メッセージシステムのマッピング (ページ 754)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.2 OPC A&E を使用した WinCC メッセージシステムのマッピング

8.9.2.1 OPC A&E を使用した WinCC メッセージシステムのマッピング

概要

WinCC メッセージシステムの設定中、設定を実行して、メッセージを生成するプロセスイベントを決定します。このメッセージは、アラームとして OPC A&E に表示されます。下表に、このアラームの最も重要なパラメータを挙げています。WinCC メッセージシステムが情報を配信する方法についても説明します。詳細については、「アラーム構造」を参照してください。

概要

OPC	WinCC メッセージシステム
[ソース]	メッセージのソースを示します。ソースのフォーマットは"<サーバー接頭辞>::@LOCALMACHINE::"となります。
時間	メッセージの受信、送信、および確認のタイムスタンプを示します。タイムスタンプは、UTC (協定世界時)形式で指定されます。
タイプ	単純なイベント、追跡用イベント、または条件関連付けイベントかを示します。WinCC と OPC A&E サーバー間では、単純なイベント、条件関連付けイベント、追跡イベントをサポートしています。
Severity	WinCC メッセージの重要度を示します。
EventCategory	メッセージのカテゴリを戻します。このトピックの詳細については、「メッセージクラスとメッセージタイプの表示」を参照してください。
メッセージ	対応するメッセージ番号のメッセージテキストを示します。
ConditionName	メッセージ番号を示します。
ChangeMask	メッセージの変更ステータスを示します。詳細については、「確認セオリー」を参照してください。
NewState	メッセージステータスを戻します。詳細については、「確認セオリー」を参照してください。
ConditionQuality	メッセージの品質を戻します。詳細については、「品質コード」を参照してください。
AckRequired	メッセージに確認(受信通知)が必要かどうかを示します。
ActiveTime	メッセージ受信のタイムスタンプを戻します。

OPC	WinCC メッセージシステム
EventAttribute	それぞれのメッセージに必要な属性を一覧表示します。詳細については、「WinCC メッセージシステムの属性」を参照してください。
品質	メッセージの品質コードを戻します。
Cookie	OPC A&E サーバーからクッキーを戻します。クッキーは WinCC アラームシステムのメッセージ番号に対応します。

下記も参照

確認セオリー (ページ 759)

WinCC メッセージシステムの属性 (ページ 756)

WinCC メッセージクラスおよびメッセージタイプのマッピング (ページ 755)

8.9.2.2 WinCC メッセージクラスおよびメッセージタイプのマッピング

はじめに

WinCC メッセージシステムは、プロセス内の妨害とオペレーティング状態をユーザーに通知します。WinCC メッセージは、必ずイベントカテゴリに関連する特定のメッセージクラスおよびメッセージタイプに属します。

OPC 上の WinCC メッセージシステムのマッピングは、"CcAeProvider.ini"ファイルを使って設定されます。

イベントカテゴリ

WinCC OPC A&E サーバーで、メッセージクラスとメッセージタイプの組み合わせごとにイベントカテゴリが 1 つ作成されます、

イベントカテゴリはカテゴリ ID および"カテゴリの説明"によって決定されます。カテゴリ ID はメッセージクラス用の WinCC 内部 ID とメッセージタイプ用の WinCC 内部 ID から構

8.9 WinCC OPC A&E サーバー

成されており、カテゴリの説明は、メッセージクラスとメッセージタイプから構成されています。

注記

OPC A&E サーバーが接続ステーションの WinCC クライアントで実行されている場合、そのステーションに接続されている OS サーバーはメッセージクラスおよびメッセージタイプに関して同一の設定が行われている必要があります。さもなければ、使用された OPC クライアントが OS サーバーに直接アクセスする必要があります。

メッセージクラスおよびメッセージタイプの名前は、アラーム属性"CLASSNAME"および"TYPE_NAME"を使用して正確に確認できます。

8.9.2.3 WinCC メッセージの優先度のマッピング

はじめに

WinCC メッセージの重要度は OPC サーバーによって属性"Severity"に表示されます。

WinCC メッセージシステムのアラームを設定するとき、優先度を 0 と 16 の間に設定できます。OPC A&E 仕様では、1 から 1000 までの範囲の値が重要度に定義されており、1 が重要度が最も低く、1000 は最も高くなっています。

そのため、WinCC の重要度の値は OPC の重要度に合うように変更されて表示されます。標準マッピングでは、WinCC の優先度 0 は、OPC の重要度 1 になります。これ以外の優先度値は、重要度 1000 まで直線的に内挿されます。これ以外の優先度マッピング規則を CcAeProvider.ini ファイルに設定できます。

8.9.2.4 WinCC メッセージシステムの属性

はじめに

以下の表は、WinCC メッセージシステムの OPC 属性の一覧を示しています。属性は、WinCC メッセージシステムで設定します。属性の中には WinCC の内部で使用するのみで、そのため OPC A&E クライアントと関係のないものもあります。これらの属性については一覧表示していません。

属性

OPC 属性	WinCC メッセージシステム	データタイプ
CLASSNAME	メッセージクラス名を返します。	VT_BSTR
TYPENAME	メッセージタイプ名を返します。	VT_BSTR
FORECOLOR	受信、送信、および確認の各メッセージの表示用テキスト色を返します。	VT_I4
BACKCOLOR	受信、送信、および確認の各メッセージの表示用背景色を返します。	VT_I4
FLASHCOLOR	点滅色を返します。	VT_I4
FLAGS	メッセージに確認(受信通知)が必要かどうかを示します。	VT_I4
TEXT01	UserTextBlock01 の内容を返します。	VT_BSTR
TEXT02	UserTextBlock02 の内容を返します。	VT_BSTR
TEXT03	UserTextBlock03 の内容を返します。	VT_BSTR
TEXT04	UserTextBlock04 の内容を返します。	VT_BSTR
TEXT05	UserTextBlock05 の内容を返します。	VT_BSTR
TEXT06	UserTextBlock06 の内容を返します。	VT_BSTR
TEXT07	UserTextBlock07 の内容を返します。	VT_BSTR
TEXT08	UserTextBlock08 の内容を返します。	VT_BSTR
TEXT09	UserTextBlock09 の内容を返します。	VT_BSTR
TEXT10	UserTextBlock10 の内容を返します。	VT_BSTR
PROCESSVALUE 01	ProcessValueBlock01 の内容を返します。	VT_VARIANT
PROCESSVALUE 02	ProcessValueBlock02 の内容を返します。	VT_VARIANT
PROCESSVALUE 03	ProcessValueBlock03 の内容を返します。	VT_VARIANT
PROCESSVALUE 04	ProcessValueBlock04 の内容を返します。	VT_VARIANT
PROCESSVALUE 05	ProcessValueBlock05 の内容を返します。	VT_VARIANT
PROCESSVALUE 06	ProcessValueBlock06 の内容を返します。	VT_VARIANT

OPC 属性	WinCC メッセージシステム	データタイプ
PROCESSVALUE 07	ProcessValueBlock07 の内容を戻します。	VT_VARIANT
PROCESSVALUE 08	ProcessValueBlock08 の内容を戻します。	VT_VARIANT
PROCESSVALUE 09	ProcessValueBlock09 の内容を戻します。	VT_VARIANT
PROCESSVALUE 10	ProcessValueBlock10 の内容を戻します。	VT_VARIANT
STATETEXT	ステータスメッセージを戻します。	VT_BSTR
INFOTEXT	メッセージの情報テキストを戻します。	VT_BSTR
LOOPINALARM	LoopInAlarm が設定されているかどうかを示します。	VT_I4
CLASSID	メッセージクラス ID を戻します。	VT_I4
TYPEID	メッセージタイプ ID を戻します。	VT_I4
MODIFYSTATE	メッセージのステータスタグの値を出力します。	VT_I4
AGNR	メッセージを生成するオートメーションデバイスの番号を戻します。	VT_I2
CPUNR	メッセージを生成する CPU の番号を戻します。	VT_I2
DURATION	メッセージを受信、送信、および確認する期間を示します。	VT_I4
COUNTER	ランタイム開始後のメッセージ数を出力します。	VT_I4
QUITSTATETEXT	メッセージが確認されたかどうかを示します。	VT_BSTR
QUITCOUNT	有効で未確認のメッセージ数を出力します。	VT_I4
PARAMETER	メッセージパラメータを出力します。(メッセージ設定の画像)	VT_BSTR
BLOCKINFO	メッセージブロックの現在の内容を戻します。	VT_BSTR
ALARMCOUNT	保留メッセージ数を出力します。	VT_I4
LOCKCOUNT	ロックされているメッセージの数を出力します。	VT_I4
PRIORITY	メッセージの設定された優先度を示します。	VT_I4
APPLICATION	メッセージをトリガしたアプリケーションを出力します。	VT_BSTR
COMPUTER	メッセージを処理したコンピュータ名を出力します。	VT_BSTR

OPC 属性	WinCC メッセージシステム	データタイプ
USER	メッセージを処理したユーザー名を出力します。	VT_BSTR
COMMENT	メッセージコメントを出力します。	VT_BSTR

8.9.2.5 確認セオリー

はじめに

WinCC での確認についての考え方は、"着信"から"発信"までのメッセージの表示方法と処理方法を指します。WinCC OPC A&E サーバーでは、このメッセージステータスは、"ChangeMask"と"NewState"の各パラメータで管理されます。

条件イベントと単純なイベントと追跡用イベント

通常、WinCC システムからのメッセージは条件イベントとしてクライアントに送信されます。メッセージが単純なイベントとして扱われるようにするには、メッセージクラスの設定中に下記の条件を満たす必要があります。

- [確認の"Came In (着信)"]を有効にしないこと。
- ["Went Out(発信)"ステータスなしのメッセージ]を有効にすること。

マッピング設定によって、メッセージクラス"確認なしのシステム"およびメッセージタイプ"操作メッセージ"のメッセージは OPC 追跡用イベントとして転送されます。

ChangeMask

"ChangeMask"パラメータは、メッセージステータスが変更された場所を追跡します。

パラメータ値:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

"NewState"パラメータは、変更後のメッセージステータスを示します。

パラメータ値:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

概要

WinCC	NewState	ChangeState
メッセージの受信	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
受信通知を指定してメッセージを送信	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
受信通知を指定せずにメッセージを送信	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
メッセージの確認(メッセージは保留中です)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージのロック	-----	OPC_CHANGE_ENABLED_STATE
メッセージのロック解除	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
メッセージの受信と確認	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
受信、受信通知を指定してメッセージを送信	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
受信、受信通知を指定せずにメッセージを送信	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
システムによるメッセージの確認(メッセージは保留されています)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

WinCC	NewState	ChangeState
システムによるメッセージの確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの緊急確認(メッセージは保留されています)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの緊急確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

下記も参照

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.3 OPC A&E の品質コード

はじめに

品質コードは、メッセージのステータスと品質を評価するのに使用されます。OPC の品質コードについては、『データアクセスカスタムインターフェース標準バージョン 3.00』仕様の「6.8 OPC 品質タグ」を参照してください。

品質コード

コード	品質	ステータス
0xC0	OPC_GOOD	実行されます。
0x40	OPC_UNCERTAIN	確認(受信通知)表示の遅延の場合など、不明の場合に戻されます。
0x00	OPC_BAD	ソースとの接続が割り込まれる場合に戻されます。

8.9.4 OPC A&E 接続例

8.9.4.1 OPC A&E 接続例

はじめに

以下の例では、WinCC と OPC A&E クライアント間の接続を設定します。WinCC OPC A&E サーバーを使用すれば、WinCC メッセージシステムのデータが使用できるようになります。

サブスクリプションによって、OPC A&E クライアントには、WinCC メッセージのステータス変更が通知されます。

OPC Alarms&Events 1.10 仕様に準拠する OPC A&E クライアントはすべて、WinCC OPC A&E サーバーにアクセスできます。

設定手順

WinCC と WinCC OPC A&E クライアント間の接続には、以下の設定が必要です。

1. WinCC メッセージシステムへのアクセスの設定

下記も参照

WinCC メッセージシステムへのアクセスの設定方法 (ページ 762)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.4.2 WinCC メッセージシステムへのアクセスの設定方法

はじめに

このセクションでは、OPC 協会の OPC A&E クライアントを使用して、WinCC メッセージシステムにアクセスします。

注記

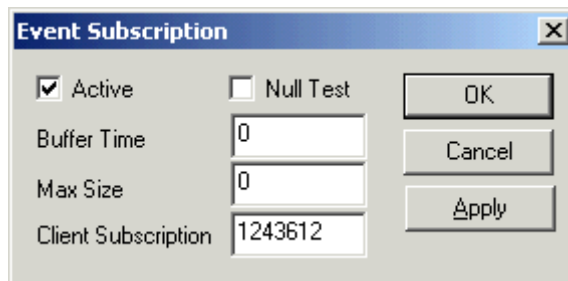
ここに記述されている OPC A&E クライアントは、OPC 協会のデモクライアントです。そのソースコードについては、インターネットサイト <http://www.opcfoundation.org> を参照してください。

必要条件

- WinCC OPC A&E サーバーの WinCC プロジェクトで、2 進データタイプの内部タグが複数作成されていること。
- WinCC OPC A&E サーバーの WinCC プロジェクトで、WinCC メッセージシステムを構成します。メッセージを内部タグにリンクさせます。
- グラフィックデザイナーを使用して画像を構成します。WinCC アラームコントロールおよび I/O フィールドを画像に追加します。メッセージタグをグラフィックオブジェクトにリンクさせます。
- 開始リストの[アラームロギングランタイム]を有効化します。
- WinCC OPC A&E サーバーの WinCC プロジェクトを有効化します。

手順

1. "SampleClientAE.exe"ファイルを"Siemens\WinCC\documents\english"フォルダから選択するフォルダへコピーします。このアプリケーションを使用できるのはオンラインヘルプのみです。
2. メニューバーから[OPC|接続...]を選択します。[OPC アラームサーバー]ダイアログで[OPC.WinCC-AlarmsEven]を選択します。[OK]をクリックしてダイアログを閉じます。
3. メニューバーから[OPC|Event Subscription...]を選択します。[Event Subscription]ダイアログが開きます。
4. ダイアログにある[アクティブ]とラベルの付いたチェックボックスを選択します。"Buffer Time"および"Max Size"フィールドに、"1000"を入力します。[OK]をクリックして、[Event Subscription]ダイアログを閉じます。



5. WinCC メッセージシステムからのメッセージが、[OPC イベントサンプルクライアント]に表示されます。

Mask	NewState	Source	Event Time	Message	Event Type	E Cal	Severity	Condition...	Sub...	Quality	AckReq	Active Time
0x1	ACT ENA	localhost:	13.06.2002 20:02:59	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:59
0x1	ACT ENA	localhost:	13.06.2002 20:02:49	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:49
0x1	ENA	localhost:	13.06.2002 20:02:49	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ENA	localhost:	13.06.2002 20:02:39	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:09
0x1	ACT ENA	localhost:	13.06.2002 20:02:19	Mischer ist angelaufen	COND	65540	200	1	1	0xc0	0	13.06.2002 20:02:19
0x1	ACT ENA	localhost:	13.06.2002 20:02:09	Kessel 1/2 leer	COND	65537	200	2	2	0xc0	0	13.06.2002 20:02:09

6. メニューバーから[OPC]フィルタ]を選択します。[フィルタ]ダイアログが開きます。[イベントカテゴリ]フィールドからカテゴリを選択します。[OK]をクリックして[フィルタ]ダイアログを閉じます。
7. [OPC イベントサンプルクライアント]に、フィルタ規準に合致したメッセージが表示されます。

"Buffer Time"パラメータと"Max Size"パラメータ

OPC 仕様に従って、WinCC ではパラメータ"Buffer Time"および"Max Size"が次のように構成されています。

OPC クライアントが戻り値を要求する	WinCC の使用
Buffer Time < 100 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 100
100 <= buffer time <= 600000 S_OK	Revised buffer time = buffer time
Buffer time > 600000 OPC_S_INVALIDBUFFERTIME	Revised buffer time = 600000
Max Size = 0 OPC_S_INVALIDMAXSIZE	Revised max size = 1000
0 < Max Size < 10 OPC_S_INVALIDMAXSIZE	Revised max size = 10
10 <= Max Size <= 1000 S_OK	Revised max size = max size
Max Size = 1000 OPC_S_INVALIDMAXSIZE	Revised max size = 1000

パラメータはサブスクリプションの作成時に設定されます。ただし、事後に SetState() を使用して既存のサブスクリプションを変更することはできません。

詳細については、<http://www.opcfoundation.org> を参照してください。

下記も参照

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.9.5 階層的アクセス機能付き OPC A&E サーバー

8.9.5.1 OPC A&E サーバーの機能

はじめに

OPC-A&E サーバーは、OPC が可能なアプリケーション間のメッセージの転送に DCOM を使用します。OPC A&E サーバーは、OPC Alarm&Event 1.10 仕様をサポートしています。

以下の章では、階層的アクセス機能付きの OPC A&E 上での WinCC メッセージシステムのマッピングについて、および WinCC OPC A&E サーバーでサポートされている属性を説明します。このドキュメントでは、各情報の概要を説明します。詳細については、"OPC Alarms & Events 1.10"の仕様を参照してください。

動作の原理

OPC-A&E クライアントが、サブスクリプションを介して WinCC メッセージを受信します。サブスクリプションフィルタを使用して、サブスクリプションと共に転送されるイベント数を減らすことができます。OPC-A&E クライアントは、メッセージ属性を表示するすべてのイベントカテゴリに設定できます。

インストール

WinCC のインストール中、WinCC OPC A&E サーバーを選択できます。インストールが終了すると直ちに、WinCC OPC A&E サーバーを使用できます。このとき、別の設定を追加して実行する必要はありません。

WinCC OPC A&E サーバーは、WinCC サーバーおよび WinCC クライアントに実装できます。

ライセンスング

WinCC OPC A&E サーバーを操作するには、OPC A&E サーバーとして実装された各 WinCC サーバーに、次のライセンスをインストールしておく必要があります。

- WinCC の有効な RT ライセンス
- WinCC オプション接続性パック

イベントタイプ

階層的アクセス機能付きの OPC-A&E サーバーは、条件付きイベント、単純イベントおよび追跡用イベントをサポートします。

条件関連イベント

条件関連イベントを使用して、イベントに条件を関連付けます。たとえば、タグの限界値違反を条件に設定することができます。この上限違反は、アラームとして OPC A&E に表示されるメッセージを生成します。

単純イベント

単純なイベントは、OPC A&E クライアントにイベントを通知するメッセージです。単純なイベントには、たとえばプログラムの起動や終了があります。

注記

冗長システムを使用している場合、以下の点に注意してください。

内部タグと相互接続されている単純イベントは、タグの更新時に 2 回送信されます。

最初のメッセージはマスタによりトリガされ、2 番目のメッセージはスタンバイによりトリガされます。

追跡用イベント

追跡用イベントは、オペレータ入力メッセージを使って、OPC A&E クライアントに送信されます。オペレータ入力メッセージは、プロセスの手動介入によってトリガされます。

OPC A&E クライアント

OPC Alarms & Events 1.10 仕様に準拠する OPC A&E クライアントはすべて、OPC A&E サーバーにアクセスできます。OPC A&E クライアントは、ユーザー自身で作成することもできます。私有の OPC クライアントを作成することにより、ユーザー固有の必要条件をほとんど満たすことができます。たとえば、OPC A&E クライアントは、他の OPC A&E サーバーからのアラームの分析と結合アーカイブに使用できます。アーカイブされたメッセージの確認はできません。現在のアラームおよびイベントのみが確認できます。

階層アクセス権限付きの OPC A&E を使用しており、すべての機能を使用する場合は、現在使用中の OPC A&E クライアントを適合する必要があります。

注記

OPC に関するドキュメンテーション

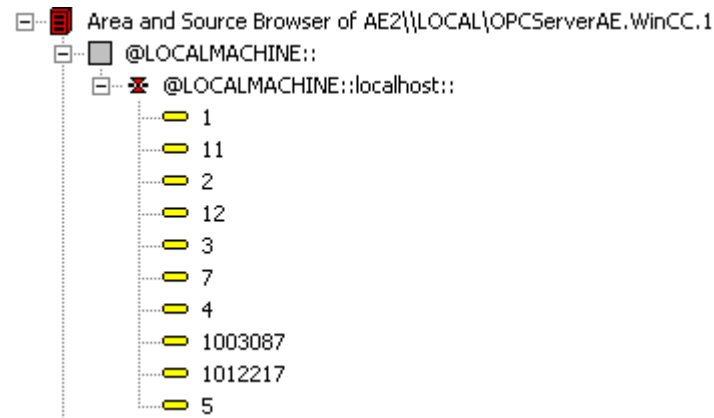
詳細は、「インターフェース」の章の「プロセスコントロールのための OPC - OLE」を参照してください。

8.9.5.2 OPC A&E と階層的アクセス機能付き OPC A&E との違い

OPC A&E のメッセージの表示

OPC A&E サーバーは、メッセージシステムにアクセスするための"条件付きイベント"および"単純イベント"をサポートします。「条件付きイベント」では、ソースごとにメッセージ番号が表示されます。WinCC サーバーはメッセージ番号を多数保持できるため、メッセージの概要の維持が困難です。

OPC ブラウザの表示例を次の図に示します。



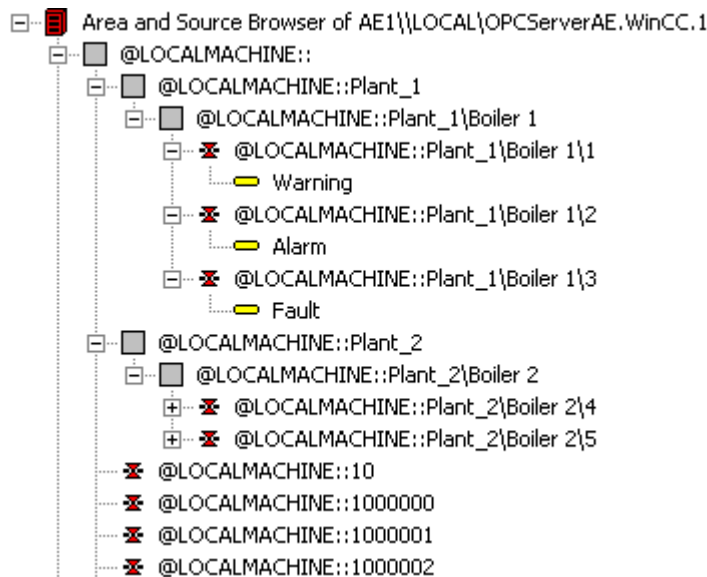
階層アクセス権限を持つ OPC A&E でのメッセージの表示

階層アクセス権限付きの OPC A&E サーバーは、条件付きイベント、単純イベント、追跡用イベントのイベントタイプをサポートします。

ユーザーテキストブロック 2 によって、「条件付きイベント」のメッセージのソースが決定されます。デフォルト設定では、ユーザーテキストブロック 2 は、障害箇所に対応します。メッセージを階層的に表示するには、アラームロギングメッセージで、ユーザー定義のグループメッセージとメッセージを統合する必要があります。グループメッセージの構造は、OPC A&E の領域によって決定されます。

追跡用イベントは、オペレータ入力メッセージがシステム内でトリガされたときに発生します。

OPC ブラウザでの条件付きイベントの表示例を次の図に示します。[領域]と[ソース]に加え[条件]が表示されます。



階層アクセス権限付きの OPC A&E サーバーへの切り替え

新しいプロジェクトの作成時は、階層アクセス権限付き OPC A&E サーバーを使用してください。

既存のプロジェクトでは、OPC A&E サーバーは以前と同じように使うことも、階層アクセス用に変換することもできます。変換は、データを損失することなく元に戻すことができます。

1. プロジェクトフォルダに"CcAeProvider.ini"ファイルをコピーします。ファイルは、WinCC インストールパスの"OPC\AlarmEvent\Hierarchical-Access"フォルダにあります。
2. クライアントを更新するか、OS サーバーの完全ダウンロードを実行します。

8.9.5.3 OPC A&E の WinCC メッセージシステムのマッピング

WinCC メッセージシステムのマッピング

はじめに

設定を基にした WinCC メッセージシステムにより、プロセスのどのイベントがメッセージを生成するかが定義されます。OPC A&E では、このメッセージはイベント告知として表示されます。

階層アクセス権限付きの OPC A&E の WinCC メッセージシステムのマッピング

WinCC ユーザーテキストボックス"2"の OPC ソースおよび WinCC ユーザーテキストブロック[1]の OPC メッセージが、WinCC メッセージシステムのマッピングのデフォルト設定として WinCC で使用されます。

概要

イベント告知の最も重要な属性と、WinCC メッセージシステムからの各情報は次の表のとおりです。

設定属性を使用するイベントは、表の第 3 列に表示されています。

- "S"は単純イベントを指します。
- "C"は条件付きイベントを指します。
- "T"は追跡用イベントを指します。

OPC	WinCC メッセージシステム	イベントタイプ
領域	グループメッセージの構造が、OPC A&E の領域を決定します。メッセージにグループメッセージが設定されていない場合は、サーバー接頭語に対応する OPC 領域のみが使用可能です。	S、C、T
ソース	メッセージのソースを示します。ソースのフォーマットは"<サーバー接頭語>::Arealuser text block 2"です。ローカルコンピュータのサーバー接頭語は"@LOCALMACHINE"です。サーバー接頭語は、常にサーバー階層の一番上の Areas を指します。	S、C、T
時刻	メッセージの受信、送信、確認のタイムスタンプを示します。タイムスタンプは、UTC (協定世界時)形式で指定されます。	S、C、T
タイプ	イベントが、単純イベント、追跡用イベント、または条件付きイベントのどれであることを示します。	S、C、T
重要度	メッセージの優先順位を返します。	S、C、T
EventCategory	メッセージクラスを示します。"イベントカテゴリ"は、"カテゴリ ID"と"カテゴリ説明"で構成されています。"カテゴリ ID"は、メッセージクラスの内蔵 ID に対応します。"カテゴリ説明"は、メッセージクラスの名前に対応します。	S、C、T
メッセージ	対応するメッセージ番号のメッセージテキストを示します。	S、C、T
条件	メッセージタイプを示します。	C
準条件	"条件"パラメータに対応します。	C

OPC	WinCC メッセージシステム	イベントタイプ
ChangeMask	条件の変更を指定します。詳細については、「確認セオリー」を参照してください。	C
NewState	条件の現在のステータスを示します。詳細については、「確認セオリー」を参照してください。	C
ConditionQuality	メッセージの品質を戻します。詳細については、「品質コード」を参照してください。	C
AckRequired	メッセージに確認が必要かどうかを示します。	C
EventAttribute	それぞれのメッセージに必要な属性を一覧表示します。詳細については、「WinCC メッセージシステムの属性」を参照してください。	C
品質	メッセージの品質コードを戻します。	C
クッキー	クライアントが使用できる情報は含まれていません。	C
ActorID	どのユーザーがメッセージを確認したかを示します。	T

注記

ワイルドカードなしのテキストが領域のフィルタとして指定されている場合、領域のメッセージのみが返されます。指定された領域以外にあるソースを含めるには、ワイルドカードを使用する必要があります。

注記

OPC A&E サーバーを以下のように実行する場合、メッセージクラスおよびメッセージタイプは、接続された OS サーバーで同一に設定しなければなりません。

- WinCC クライアント上
- 接続ステーション上

OS サーバーが同一に設定されていないと、使用している OPC クライアントは各 OS サーバーに直接アクセスする必要があります。

メッセージ優先度のマッピング

概要

メッセージの優先度は、OPC サーバーによって属性"Severity(重要度)"にマッピングされます。

メッセージシステムでアラームを設定する場合、重要度を"0"から"16"の値に設定できます。OPC A&E 仕様では、重要度の値範囲は"1"から"1000"に定義されています。この場合、"1"は最低で、"1000"は最高重要度です。

これにより、優先度値が OPC の重要度に適合するように表示されます。標準マッピングでは、優先度"0"は OPC 重要度"1"に、および優先度"16"は OPC 重要度"1000"に割り当てられます。他の優先度値はすべて、"0"から"1000"の間で等間隔に割り当てられます。

WinCC メッセージシステムの属性

はじめに

以下の表は、WinCC メッセージシステムの OPC 属性の一覧を示しています。属性は、WinCC メッセージシステムで設定します。属性の中には WinCC の内部で使用するのみで、そのため OPC A&E クライアントと関係のないものもあります。これらの属性は表には含まれていません。

属性

OPC 属性	WinCC メッセージシステム	データタイプ
CLASSNAME	メッセージクラス名を出力します。	VT_BSTR
TYPENAME	メッセージタイプ名を出力します。	VT_BSTR
FORECOLOR	有効なメッセージ、無効なメッセージ、確認済みメッセージのテキスト色を出力します。	VT_I4
BACKCOLOR	有効なメッセージ、無効なメッセージ、確認済みメッセージの背景色を出力します。	VT_I4
FLASHCOLOR	点滅色を出力します。	VT_I4
FLAGS	確認必須のメッセージであることを示します。	VT_I4
TEXT01	UserTextBlock01 の内容を出力します。	VT_BSTR
TEXT02	UserTextBlock02 の内容を出力します。	VT_BSTR
TEXT03	UserTextBlock03 の内容を出力します。	VT_BSTR
TEXT04	UserTextBlock04 の内容を出力します。	VT_BSTR
TEXT05	UserTextBlock05 の内容を出力します。	VT_BSTR
TEXT06	UserTextBlock06 の内容を出力します。	VT_BSTR
TEXT07	UserTextBlock07 の内容を出力します。	VT_BSTR

OPC 属性	WinCC メッセージシステム	データタイプ
TEXT08	UserTextBlock08 の内容出力します。	VT_BSTR
TEXT09	UserTextBlock09 の内容出力します。	VT_BSTR
TEXT10	UserTextBlock10 の内容出力します。	VT_BSTR
PROCESSVALUE01	ProcessValueBlock01 の内容出力します。	VT_VARIANT
PROCESSVALUE02	ProcessValueBlock02 の内容出力します。	VT_VARIANT
PROCESSVALUE03	ProcessValueBlock03 の内容出力します。	VT_VARIANT
PROCESSVALUE04	ProcessValueBlock04 の内容出力します。	VT_VARIANT
PROCESSVALUE05	ProcessValueBlock05 の内容出力します。	VT_VARIANT
PROCESSVALUE06	ProcessValueBlock06 の内容出力します。	VT_VARIANT
PROCESSVALUE07	ProcessValueBlock07 の内容出力します。	VT_VARIANT
PROCESSVALUE08	ProcessValueBlock08 の内容出力します。	VT_VARIANT
PROCESSVALUE09	ProcessValueBlock09 の内容出力します。	VT_VARIANT
PROCESSVALUE10	ProcessValueBlock10 の内容出力します。	VT_VARIANT
STATETEXT	ステータスメッセージ出力します。	VT_BSTR
INFOTEXT	メッセージ情報テキスト出力します。	VT_BSTR
LOOPINALARM	LoopInAlarm が設定されているかどうかを示します。	VT_I4
CLASSID	メッセージクラス ID 出力します。	VT_I4
TYPEID	メッセージタイプ ID 出力します。	VT_I4
MODIFYSTATE	メッセージのステータスタグの値出力します。	VT_I4
AGNR	メッセージを生成した AS の番号出力します。	VT_I2
CPUNR	メッセージを生成した CPU の番号出力します。	VT_I2
DURATION	メッセージの有効化、無効化、確認の間の間隔出力します。	VT_I4
COUNTER	ランタイム開始後のメッセージ数出力します。	VT_I4
QUITSTATETEXT	メッセージが確認されたかどうかを示します。	VT_BSTR
QUITCOUNT	有効で未確認のメッセージ数出力します。	VT_I4
PARAMETER	メッセージパラメータ出力します。(メッセージ設定の画像)	VT_BSTR
BLOCKINFO	メッセージブロックの現在の内容出力します。	VT_BSTR
ALARMCOUNT	保留メッセージ数出力します。	VT_I4
LOCKCOUNT	ロックされているメッセージの数出力します。	VT_I4
PRIORITY	設定されているメッセージ優先度を示します。	VT_I4
APPLICATION	メッセージをトリガしたアプリケーション出力します。	VT_BSTR

OPC 属性	WinCC メッセージシステム	データタイプ
COMPUTER	メッセージを処理したコンピュータ名を出力します。	VT_BSTR
USER	メッセージを処理したユーザー名を出力します。	VT_BSTR
COMMENT	メッセージコメントを出力します。	VT_BSTR
HIDDEN-COUNT	非表示メッセージ数を出力します。	VT_I4
BIG COUNTER	ランタイム開始後のメッセージ数を出力します。	VT_CY
OS-HIDDEN	メッセージの非表示ステータスを出力します。	VT_BOOL
OS-EVENTID	メッセージに設定されたメッセージ ID を出力します。	VT_I4

確認ポリシー

概要

WinCC の確認ポリシーとは、"着信"から"発信"までのメッセージの表示方法と処理方法を指します。 OPC A&E サーバーでは、このメッセージステータスは、"ChangeMask"と "NewState"の各パラメータに表示されます。

条件付きイベント、単純イベントおよび追跡用イベント

システムからのメッセージは、確認済み条件イベントとしてクライアントに送信されます。

単純イベントとしてメッセージを処理するには、メッセージのメッセージクラスが以下の条件を満たす必要があります。

- "確認済み着信"が有効でないこと。
- "ステータスなしのメッセージの送信"が有効であること。

WinCC では、"システムによる確認必要なし"メッセージクラスで、"オペレータ入力メッセージ"メッセージタイプのメッセージは、追跡用イベントとして転送されます。

注記

"システムによる確認必要なし"メッセージクラスで、"プロセスコントロールシステム"メッセージタイプのメッセージは、"システムメッセージ"イベントカテゴリの単純イベントとして転送されます。

ChangeMask

"ChangeMask"パラメータは、メッセージステータスが変更された場所を追跡します。

パラメータ値:

- OPC_CHANGE_ACTIVE_STATE
- OPC_CHANGE_ENABLE_STATE
- OPC_CHANGE_ACK_STATE

NewState

"NewState"パラメータは、変更後のメッセージステータスを示します。

パラメータ値:

- OPC_CONDITION_ACTIVE
- OPC_CONDITION_ENABLED
- OPC_CONDITION_ACKED

概要

WinCC	NewState	ChangeState
メッセージの受信	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
確認済み送信メッセージ	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
未確認送信メッセージ	OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE
メッセージの確認(メッセージは保留中です)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージのロック	-----	OPC_CHANGE_ENABLED_STATE
メッセージのロック解除	OPC_CONDITION_ENABLED	OPC_CHANGE_ENABLED_STATE
受信、確認済みメッセージ	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACTIVE_STATE

WinCC	NewState	ChangeState
受信、確認済みの送信メッセージ	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
受信、未確認の送信メッセージ	OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
システムによるメッセージの確認(メッセージは保留されています)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
システムによるメッセージの確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの緊急確認(メッセージは保留されています)	OPC_CONDITION_ACTIVE OPC_CONDITION_ACKED OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE
メッセージの緊急確認(メッセージは保留されていません)	OPC_CONDITION_ACTIVE OPC_CONDITION_ENABLED	OPC_CHANGE_ACK_STATE

注記

履歴アラームおよびイベントは確認されません。 OPC A&E 履歴イベントインターフェースは読み取り専用アクセスしかできません。

8.9.5.4 OPC A&E の品質コード

はじめに

品質コードは、メッセージのステータスと品質を評価するのに使用されます。 OPC の品質コードについては、『データアクセスカスタムインターフェース標準バージョン 3.00』仕様の「6.8 OPC 品質タグ」を参照してください。

品質コード

コード	品質	ステータス
0xC0	OPC_GOOD	実行されます。
0x40	OPC_UNCERTAIN	確認(受信通知)表示の遅延の場合など、不明の場合に返されます。
0x00	OPC_BAD	ソースとの接続が割り込まれる場合に返されます。

8.9.6 アーカイブメッセージの読取り

8.9.6.1 アーカイブイベントへのアクセス

はじめに

OPC クライアントを使用して、OPC A&E サーバー経由でアーカイブメッセージにアクセスできます。アーカイブメッセージにアクセスする方法として、以下の 2 つの方法がサポートされています。

- 過去の期間からの出力アーカイブメッセージ
- 期間の終了を明示しない、過去の期間からの出力アーカイブメッセージ。アーカイブメッセージの出力の後、他の全ての新たに生成されたメッセージは、自動的に OPC クライアントに送信されます。

注記

アーカイブメッセージの読み取り後は、返されたメッセージの"ActiveTime"を使用して、メッセージを確認したり、メッセージの転移を追跡することはできません。このために、OPC A&E クライアントは、追加フラグ"OPC_HAE_HISTORICAL_EVENTFLAG"の付いたメッセージの"EventType"を確認する必要があります。アーカイブメッセージの"ActiveTime"が不正です。追加フラグについての詳細は、「アーカイブメッセージの識別」を参照してください。

"履歴アラームおよびイベント"機能の問い合わせ

標準フィルタに加え、以下のファイルが WinCC の拡張された OPC A&E サーバーで提供されています。

フィルタ	フィルタの値	説明
OPC_HAE_FILTER_BY_TIMEFRAME	0x80000000	OPC 履歴データアクセス用の "ReadRaw"関数に相当
OPC_HAE_FILTER_BY_STARTTIME	0x40000000	OPC 履歴データアクセス用の "AdviseRaw"関数に相当

ソースフィルタと履歴アラームリクエスト

アーカイブメッセージを要求するには、OPC クライアントがサブスクリプション機能に対する"SetFilter"をサポートする必要があります。サブスクリプションの"ソースフィルタ"配列にも"OPCHAEsServer"キーワードを挿入すると、OPC サーバーはアーカイブメッセージも送信します。このキーワード以外にも、他のパラメータを使用して、どのメッセージを読み取るかを定義できます。

- メソッド
- 期間
- 制限ありまたは制限なし

フィルタで割り付けられたソースの一覧には、"OPCHAEsServer"ソース以外のソース名も入れます。この場合、サブスクリプションは与えられたソースのイベントの履歴のみを返します。ソース名のシーケンスは重要ではありません。

ソースファイルの設定後、"Refresh"の呼び出しを使って、選択された期間をクライアントから呼び出せます。

8.9.6.2 OPC を使用したアーカイブメッセージへのアクセスの構文

構文

```
OPCHAEsServer hMode=(read|advise) htStartTime=szTime
[hEndTime=szTime] [bBounds=(TRUE|FALSE)]
```

パラメータ

hMode = [read|advise]

このパラメータは必須です。アーカイブメッセージとイベントを読み取る方法を定義します。

読み取り:過去の定義された期間のアーカイブメッセージとイベントを出力します(OPC Historical Data Access の場合の ReadRaw に相当)。

以下は、最近 30 分間の読み取りにフィルタを設定する例です。

```
OPCHAESEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

アドバイス:定義された期間からのアーカイブメッセージとイベントを出力します。すべてのアーカイブメッセージを受け取った後、有効サブスクリプションと同じ方法で新しいメッセージが送信されます(OPC 履歴データアクセスの場合の AdviseRaw に相当)。

次の例では、過去 30 分間のメッセージを読み取ります(サブスクリプションが有効である必要があります)。

```
OPCHAESEServer hMode=advise htStartTime=NOW-30M
```

注記

以下の表記が"htStartTime"および"htEndTime"パラメータでサポートされています。

- 相対的な表記(例: NOW)
- シンボル値(例: NOW、YEAR、MONTH)
- XML 表記法に従った絶対 UTC データ/時刻の仕様 2006-09-01T10:00:00.000Z

シンボル表記の使用は、OPC 履歴データアクセスからの構文に対応します。

htStartTime =

このパラメータは必須です。メッセージとイベントのアーカイブからの読取りを開始する時刻を定義します。

htEndTime =

このパラメータはオプションです。メッセージとイベントがアーカイブからの読み取られる時刻を定義します。"hMode = read"となっている場合、デフォルト値は"NOW"です。

bBounds = [TRUE|FALSE]

このパラメータはオプションです。開始時刻および終了時刻に近いメッセージの処理方法を定義します。この機能は OPC 履歴データアクセスとまったく同様です

bBounds=FALSE:

- 最初に転送されたメッセージのタイムスタンプ \geq htStartTime
- 最後に転送されたメッセージのタイムスタンプ \geq htEndTime

bBounds=TRUE:

- 最初に転送されたメッセージのタイムスタンプ \leq htStartTime
- 最後に転送されたメッセージのタイムスタンプ \geq hEndTime

デフォルト設定は FALSE です。

8.9.6.3 アーカイブメッセージの読取りメソッド

はじめに

2 つの読取りモードのいずれかを使用してアーカイブメッセージを読み取ることができます。

- 読取り
- アドバイス

読取りモード「読取り」

過去の定義された期間からのアーカイブメッセージは、「読取り」モードで読み取られます。アラームが読み取られる各 OS サーバーに関しては、読取りメッセージの順序は常に時系列順になります。開始時間と終了時間を設定することによって、最も古いメッセージを最初に出力するか最後に出力するかを指定できます。開始時間が終了時間より早い場合、最も古いメッセージが出力の最後になります。

「読取り」モードを使用したい場合、サブスクリプションで次の関数を実行します。

1. SetFilter
2. Refresh

更新識別子のあるイベントパケットには、履歴イベントのみが含まれます。これらのイベントはキューにある場合もあります。

履歴メッセージの最後の更新パケットには、[最後に更新]識別子が含まれます。

"Refresh"中の"SetFilter"は拒否されます。"Refresh"中にサブスクリプションを有効にする場合、更新プロセスには何の影響もありません。

履歴イベントは、更新識別子を使用して転送され続けます。

新しく生成されたイベントは、有効なサブスクリプションの標準的な動作に準拠して転送されます。

- 「履歴」ソース"OPCHAESEServer"の例外によるフィルタ値の設定の考慮
- 更新識別子なし

クライアントは更新識別子に基づいて受信したイベントを差別化できます。イベントパケットには、同時に履歴イベントと新規イベントが含まれることはありません。

- 更新識別子のあるイベントパケットには、履歴イベントのみが含まれます。こうしたイベントはキューにある場合もあります。
- 更新識別子のないイベントパケットには、新たに生成されたイベントのみが含まれます。

読み取りモード「アドバイス」

過去の定義された期間から開始されるアーカイブメッセージは、「アドバイス」モードで読み取られます。すべてのアーカイブされたメッセージが読み取られた後、新しいメッセージは、サブスクリプションが有効なときと同様に送信されます。アーカイブメッセージは、各 OS サーバーを参照して時系列順に転送されます。開始時間に開始されたアーカイブメッセージは、最初に転送されます。新しいアーカイブメッセージは、それ以降に転送されます。

「アドバイス」の終了時間を指定しないでください。

有効なサブスクリプションは「アドバイス」モードに使用されます。有効なサブスクリプションで"SetFilter"関数を実行すると、履歴アラームは直ちに転送されます。

無効なサブスクリプションで"SetFilter"関数を実行すると、履歴アラームはサブスクリプションの有効化後に転送されます。無効なサブスクリプションで「アドバイス」モードを使用するには、以下の手順を実行します。

1. SetFilter
2. SetState を使用してサブスクリプションを有効に設定

サブスクリプションを「無効化」に設定すると、転送が終了します。サブスクリプションが有効な場合、"SetFilter"は拒否されます。

「アドバイス」モードの有効な「履歴の」サブスクリプションにおける"Refresh"は、標準のサブスクリプションと同じように機能します。

キューになっているすべての条件関連イベントは、更新識別子の付いたパケットで転送されます。

"Refresh"の呼び出しは、「アドバイス」モードの履歴アラームの読み取りには影響しません。

8.9.6.4 アーカイブメッセージの識別

一般的手順

アーカイブメッセージは、EventType の追加フラグによって識別されます。このフラグは OR リンクを経由して実際の EventType にリンクされています。

名前	EventType	EventType (アーカイブメッセージ)
OPC_SIMPLE_EVENT	0x01	0x81
OPC_CONDITION_EVENT	0x04	0x84
OPC_TRACKING_EVENT	0x02	0x82
OPC_HAE_HISTORICAL_EVENTFLAG		0x80

例

例 1

以下のソースフィルタを使用して、"読取り"モードで最近 30 分のアーカイブメッセージとイベントを出力します。各 OS サーバーの最も古いメッセージが、最初に出力されます。下限値も送信されます。

```
OPCHAEServer hMode=read htStartTime=NOW-30M bBounds=TRUE
```

例 2

以下のソースフィルタは、"読み取り"モードで 2006 年 9 月 1 日の 10 時から 12 時までのアーカイブイベントを出力するために使用します。各 OS サーバーの最新メッセージが、最初に出力されます。この期間の限界も送信されます。

```
OPCHAEServer hMode=read htStartTime=2006-09-01T12:00:00.000Z
htEndTime=2006-09-01T10:00:00.000Z bBounds=TRUE
```

例 3

以下のソースフィルタを使用して、"アドバイス"モードで過去 30 分間のアーカイブメッセージとイベントを出力します。アーカイブメッセージを読み取った後は、新しく生成されたメッセージがアクティブサブスクリプションの場合と同じ方法で送信されます。

```
OPCHAEServer hmode=advise htStartTime=NOW-30M
```

8.10 WinCC OPC UA サーバー

8.10.1 WinCC OPC UA サーバーの動作原理

動作

WinCC OPC UA サーバーは、以下の値を提供します。

- プロセス値
- タグアーカイブからの値
- WinCC メッセージ

WinCC OPC UA サーバーは Windows サービスとしてインストールされ、自動的に起動します。WinCC OPC UA サーバーは、「UA-TCP UA-SC UA Binary」通信プロファイルのみをサポートします。使用されるポート番号を調整することができます。

サポートされる仕様

OPC Unified Architecture はプロセス値、アーカイブデータ、およびメッセージを伝送するための仕様です。WinCC OPC UA サーバーは、OPC UA 仕様 1.03 に対応しています。サポートされる UA 機能に関する詳細情報については、「サポートされる OPC UA サービスとプロファイル (ページ 791)」を参照してください。

インストール

WinCC がインストールされた後、追加の設定をする必要なく、WinCC OPC UA サーバーを即座に使用できます。

WinCC OPC UA サーバーは、WinCC サーバーまたは WinCC クライアント上で使用できます。

WinCC OPC UA サーバーの URL

以下の URL を介して WinCC OPC UA サーバーにアクセスします。

- "opc.tcp://[HostName]:[Port]"

パラメータ	説明
HostName	コンピュータ名のプレースホルダ。自動的に使用されます。
Port	ポート番号。デフォルト設定は、「4862」です。

発見サーバー

「Discovery サーバー」は、OPC 協議会が使用できます。「Discovery サーバー」は、デフォルトで Windows サービスとして HMI デバイスにインストールされます。

「Discovery サーバー」上で OPC UA サーバーを介して、「Discovery サーバー」に登録された UA クライアント情報を入手することができます。

設定に応じて、WinCC OPC UA サーバーはランタイムの起動時に、設定済みで利用可能な「Discovery サーバー」を何も登録しないか、1 台または複数台登録します。その後登録は周期的に繰り返されます。ランタイムを終了すると、WinCC OPC UA サーバーは「Discovery サーバー」から自動的にログオフします。

WinCC アドレス領域でサポートされている言語

WinCC OPC A&E サーバーは、以下の言語で WinCC アドレス領域をサポートします。

- ドイツ語
- 英語
- フランス語
- イタリア語
- スペイン語

8.10.2 OPC UA のセキュリティ概念

はじめに

OPC UA のセキュリティ概念は、主に以下に基づいています。

- アプリケーションの認証と承認および関与するユーザー
- アプリケーション間でやり取りされるメッセージの完全性と機密性の確認

証明書は、OPC UA アプリケーションの認証に使用される方法です。

各アプリケーションは、公開鍵インフラストラクチャで自身を識別する独自のインスタンス証明書を持っています。インスタンス証明書は「アプリケーション証明書」とも呼ばれます。

WinCC OPC UA サーバーの証明書

安全な操作のために、各 WinCC OPC UA サーバーには秘密鍵を使用する独自の証明書、つまりサーバー証明書が必要です。

証明書は、対応するコンピュータ上でのみ有効であり、そのコンピュータにインストールされている WINCC OPC UA サーバーによってのみ使用されることができます。

サーバーの自己署名証明書が作成され、サーバーの証明書フォルダに保存されます。

このサーバー証明書の秘密鍵は、証明書フォルダにも保存されます。秘密鍵が保存されるフォルダへのアクセスは、以下に制限する必要があります。

- サーバー自体
- システム管理者

通知
秘密鍵が保存されるフォルダへのアクセス セキュリティ上の理由から、サーバーとシステム管理者以外の他のユーザーやアプリケーションは、WinCC OPC UA サーバーの秘密鍵へのアクセス権を持っていません。

サイトの管理者は、インストール時に生成されるサーバー証明書および対応する秘密鍵を置き換えることができます。

システムに適用可能なセキュリティコンセプトに従って、新しいサーバー証明書は、自己署名するか、認証機関により発行することができます。

WINCC OPC UA サーバーによって使用される証明書は、「OpcUaServerWinCC.xml」設定ファイルの設定により特定されます。詳細情報は、「WinCC OPC UA サーバーの設定ファイル (ページ 806)」を参照してください。

サーバー証明書の保存

以下のパスで、「WinCC OPC UA サーバー」アプリケーションは保存されます。

保存パス	アプリケーション	設定ファイル
<インストールディレクトリ >WinCC\opc\UAServer\	OpcUaServerWinCC.exe	OpcUaServerWinCC.xml

WinCC OPC UA 証明書は、WinCC インストールパスの以下のフォルダに保存されます。

WinCC OPC UA サーバー	証明書	opc\UAServer\PKI\CA\certs
	秘密鍵	opc\UAServer\PKI\CA\private

設定ファイルの保存場所を変更できます。

信頼されるクライアント証明書

WinCC OPC UA サーバーは、信頼されるクライアントとの間でのみ安全な通信をサポートしています。クライアントは以下の場合に信頼されます:

- クライアントが、WinCC OPC UA サーバーの信頼できる証明書メモリに保存されている有効な自己署名証明書を持っている場合
- または、認証機関により有効なクライアント証明書が発行された場合。
認証機関からの有効な証明書は、WinCC OPC UA サーバーの信頼できる証明書の証明書メモリ内に配置される必要があります。この場合、認証機関からの証明書だけが必要になります。クライアント証明書は、信頼された証明書の証明書保存先に配置する必要はありません。

クライアント証明書の保存

WinCC OPC UA サーバーの設定ファイルを使用して、信頼される証明書の保管設定を指定します。

パラメータ	意味
StoreType	証明書ストレージのタイプ。保管場所は、「ディレクトリ」または「ウィンドウ」のいずれかにできます。
StorePath	信頼されるクライアントの証明書は、このフォルダに保存されます。

「ディレクトリ」保管場所を使用した設定の例

```
<TrustedCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\Trusted</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

この場合、WINCC OPC UA サーバーはサーバー証明書が「...PKI\TrustList\Certs」フォルダにあるすべてのクライアントを信頼します。

「Windows」保管場所を使用した設定の例

```
<TrustedCertificateStore>
  <StoreType>Windows</StoreType>
  <StorePath>UA Applications</StorePath>
  <ValidationOptions />
</TrustedCertificateStore>
```

この保存オプションでは、クライアントの証明書は、「<ローカルコンピュータ>\UA アプリケーション」にあるオペレーティングシステムの証明書保存先になければなりません。

クライアント証明書チェーンを検証するために必要とされる認証機関からの証明書は、認証機関の証明書保存先に保存されます。ここでも、WINCC OPC UA サーバー設定ファイルを使用して保管設定を指定します。

パラメータ	意味
StoreType	証明書ストレージのタイプ。保管場所は、「ディレクトリ」または「ウィンドウ」のいずれかにできます。
StorePath	信頼される認証機関の証明書は、このフォルダに保存されます。

注記

認証機関のメモリからの証明書は自動的に信頼されません。

認証機関が信頼されるには、その証明書が信頼される証明書のためのメモリに存在しなければなりません。

「ディレクトリ」保管場所を使用した設定の例

```
<IssuerCertificateStore>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\CA</StorePath>
  <ValidationOptions />
</IssuerCertificateStore>
```

信頼される認証機関の証明書は、この場合「...\\PKI\\CA\\Certs」フォルダにあります。

「Windows」保管場所を使用した設定の例

```
<IssuerCertificateStore>
  <StoreType>Windows</StoreType>
  <ValidationOptions />
</IssuerCertificateStore>
```

「StorePath」パラメータは関係ありません。認証機関からの証明書は、オペレーティングシステムの要件に従って、Windows の証明書メモリ内に保存する必要があります。

証明書は、次の 2 つの場所のいずれかにある場合に信頼されます。

- <ローカルコンピュータ>\信頼されるルート認証機関
- <ローカルコンピュータ>\サードパーティルート認証機関

注記

保管のために重要なこと

- サーバー証明書の保管場所は、「Directory」でなければなりません。
 - 信頼されるクライアント証明書と認証機関からの証明書のための 2 つの保管場所は、同じ StoreType でなければなりません。つまり、両方の保管場所は「Directory」または「Windows」のいずれかでなければなりません。
-

クライアント証明書が受け付けられない

UA クライアントが信頼される証明書なしで WINCC OPC UA サーバーにアクセスする場合、WINCC OPC UA サーバーは、安全な通信を許可せず、クライアント証明書を拒否された証明書のフォルダにコピーします。

たとえば、WINCC OPC UA サーバーの設定ファイルを使用して、拒否された証明書の保管設定を指定します。

```
<RejectedCertificatesStore>  
  <StoreType>Directory</StoreType>  
  <StorePath>[ApplicationPath]\PKI\OPCUA\rejected</StorePath>  
</RejectedCertificatesStore>
```

注記

ここでも StoreType 「Directory」 のみがサポートされます。

このクライアントとの安全な通信を有効にするには、信頼される証明書の証明書保存先に拒否された証明書を移動する必要があります。

下記も参照

証明書による認証の設定。(ページ 293)

WinCC OPC UA サーバーの設定ファイル (ページ 806)

8.10.3 セキュリティメカニズムの構成

はじめに

以下が通信レベルで確認されます。

- UA アプリケーションの信頼性
- やり取りされるメッセージの機密性
- やり取りされるメッセージの完全性

暗号化と署名のアルゴリズムなど、使用されるセキュリティメカニズムは、標準化されたセキュリティポリシーで定義されます。

WinCC OPC UA サーバーによりサポートされるセキュリティポリシーは、
[ServerConfiguration]と[SecuredApplication]にあるサーバー設定ファイルを使用して設定されます。

ServerConfiguration

[ServerConfiguration]の下にある XML エlement [SecurityPolicies]には、サーバー用のすべての利用できる[Security Profile]と[Message Security Mode]の組み合わせのリストが含まれます。

Security Profile	Message Security Mode	説明
http://opcfoundation.org/UA/SecurityPolicy#None	None	セキュリティで保護されていない通信
http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15	Sign または SignAndEncrypt	安全な通信(署名、または暗号化と署名されたメッセージ)
http://opcfoundation.org/UA/SecurityPolicy#Basic256	Sign または SignAndEncrypt	安全な通信(署名、または暗号化と署名されたメッセージ)
http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256 ¹⁾	Sign または SignAndEncrypt	安全な通信(署名、または暗号化と署名されたメッセージ)

1) セキュリティポリシー「Basic256Sha256」使用の必要条件:署名アルゴリズム「Sha256」と最小長=2048 のキーのインスタンス証明書。

注記

安全な通信の確認

安全な通信には、サーバーとクライアント用のサーバーの証明書、および正しく設定された証明書保存先が必要です。

最大のファンクション適用範囲を使用した設定ファイルの例

```
<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfound">...</SecuredApplication>
  <ServerConfiguration>
    <SecurityPolicies>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</ProfileUri>
        <MessageSecurityModes>None</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <MessageSecurityModes>Sign</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <MessageSecurityModes>Sign</MessageSecurityModes>
      </SecurityPolicy>
      <SecurityPolicy>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <MessageSecurityModes>SignAndEncrypt</MessageSecurityModes>
      </SecurityPolicy>
    </SecurityPolicies>
```

SecuredApplication

OPC UA 仕様に従い、セキュリティメカニズムは、[SecuredApplication]の下の [SecurityProfileUris] エレメントで明示的に有効または無効にされています。

以下の図は、セキュリティで保護されていない通信が無効にされている SecuredApplication を示しています。

```

<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <ApplicationName>OPCUA Server for Simatic WinCC UA Runtime</ApplicationName>
    <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCCUA.Rt</ApplicationUri>
    <ProductName>Simatic WinCC UA</ProductName>
    <ApplicationType>Server</ApplicationType>
    <BaseAddresses>...</BaseAddresses>
    <ApplicationCertificate>...</ApplicationCertificate>
    <TrustedCertificateStore>...</TrustedCertificateStore>
    <TrustedCertificates>...</TrustedCertificates>
    <IssuerCertificateStore>...</IssuerCertificateStore>
    <IssuerCertificates>...</IssuerCertificates>
    <RejectedCertificatesStore>...</RejectedCertificatesStore>
    <SecurityProfileUris>
      <SecurityProfile>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15</ProfileUri>
        <Enabled>true</Enabled>
      </SecurityProfile>
      <SecurityProfile>
        <ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#Basic256</ProfileUri>
        <Enabled>true</Enabled>
      </SecurityProfile>
    </SecurityProfileUris>
  </SecuredApplication>

```

したがって WinCC OPC UA サーバーは、ランタイムに 2 つのセキュリティ戦略「Basic128Rsa15」および「Basic256」をサポートします。[Message Security Modes Sign] と [SignAndEncrypt] を使用しますが、セキュリティで保護されていない通信は使用しません。

通信が確立されると、UA クライアントはこのリストから必要な Policy を選択します。

ユーザー ID

通信レベルのセキュリティメカニズムに加え、WinCC OPC UA サーバーも UserTokenPolicy 「UserName」を使用してクライアントアプリケーションのためにユーザー認証をサポートしています。

クライアントアプリケーションは、通信が確立されるときにユーザー名とパスワードの有効な組み合わせを提供しなければなりません。WinCC OPC UA サーバーは、オペレーティングシステムのユーザー管理でその組み合わせを確認します。

UserTokenPolicy は、WINCC OPC UA サーバーの設定ファイルで設定されます。

```

<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfound">...</SecuredApplication>
  <ServerConfiguration>
  <SecurityPolicies>...</SecurityPolicies>
  <UserTokenPolicies>
    <UserTokenPolicy>
      <TokenType>Anonymous</TokenType>
    </UserTokenPolicy>
    <UserTokenPolicy>
      <TokenType>UserName</TokenType>
    </UserTokenPolicy>
  </UserTokenPolicies>

```

この設定では、WINCC OPC UA サーバーは匿名ユーザーと Policy「UserName」の両方をサポートします。

8.10.4 サポートされる OPC UA サービスとプロファイル

OPC UA サービス

WinCC OPC A&E サーバーは、以下の機能をサポートします。

次の表は、OPC UA サーバー 1.0.9 でサポートされる機能を要約しています。

OPC UA Service Sets	Services	コメント
Discovery Service Set	FindServers GetEndpoints	-
Secure Channel Service Session Service Set	すべて	-
View Service Set	Browse BrowseNext RegisterNodes UnregisterNodes	マッピングされた WinCC データの特定:プロセス値とアーカイブデータ
Attribute Service Set	Read Write HistoryRead HistoryUpdate*)	WinCC タグのみ WinCC タグのみ アーカイブタグのみ アーカイブタグのみ

OPC UA Service Sets	Services	コメント
Subscription Service Set	CreateSubscription SetPublishingMode Publish RePublish DeleteSubscription	
MonitoredItem Service Set	CreateMonitoredItems SetMonitoringMode DeleteMonitoredItems	WinCC タグの「値」属性のみ WinCC メッセージへのアクセス中の Event Notifier
Method Service Set	Call	Acknowledge ConditionRefresh
*):制限については、「サポートされている書込みアクセス (ページ 739)」を参照してください。		

OPC UA プロファイルおよび Conformance Units

WinCC OPC UA サーバーは、無制限に以下の OPC UA プロファイル 1.03 をサポートします。

- 6.5.5 ベースサーバー動作ファセット
- 6.5.16 標準イベントサブスクリプションサーバーファセット
- 6.5.18 A & C ベース条件サーバーファセット
- 6.5.30 メソッドサーバーファセット
- 6.5.36 履歴未処理データサーバーファセット
- 6.5.42 履歴データ更新サーバーファセット
- 6.5.44 履歴データ挿入サーバーファセット
- 6.5.45 履歴データ削除サーバーファセット
- 6.5.131 UA-TCP UA-SC UA 2 進法
- 6.5.149 セキュリティポリシー- Basic256
- 6.5.148 セキュリティポリシー - Basic128Rsa15
- 6.5.147 セキュリティポリシー - なし
- 6.5.150 セキュリティポリシー - Basic256Sha256

WinCC OPC A&E サーバーは、以下の表に示した OPC UA プロファイルをサポートしますが、制限があります。

Profile	"Group"	サポートされていません "Conformance Unit"
6.5.2 Core Server Facet	Attribute Services	Attribute Write Index
6.5.11 Standard DataChange Subscription Server Facet	Monitored Item Services	DeadBand Filter
6.5.12 Enhanced DataChange Subscription Server Facet	Monitored Item Services	-
6.5.14 Data Access Server Facet	Data Access	Data Access Analog Data Access Multistate Data Access PercentDeadBand Data Access Semantic Changes Data Access Two State
6.5.55 Standard UA Server Profile	Attribute Services	Attribute Write StatusCode & Timestamp

8.10.5 WinCC OPC UA サーバーの名前領域

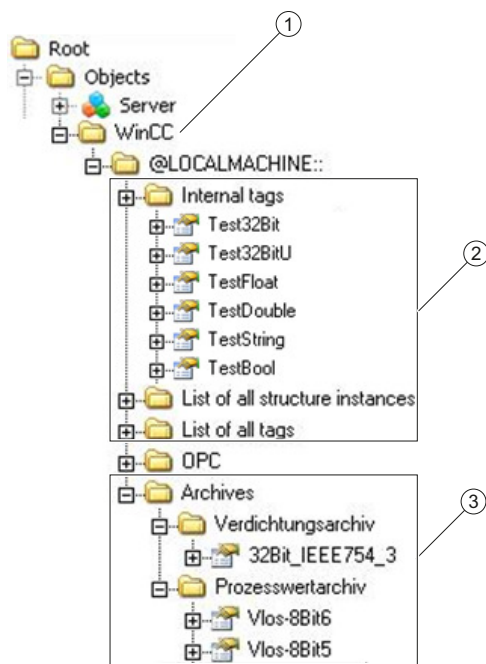
はじめに

WinCC OPC UA サーバーは、階層的名前領域と以下のランタイムデータへのアクセスを OPC UA クライアントに提供します。

- プロセス値(WinCC タグおよび WinCC タググループ)
- ロギングタグを含むデータログ
- WinCC メッセージ

WinCC OPC UA サーバーの名前領域が[Objects]デフォルトフォルダに追加されます。

以下の画面に、ローカル PC("@LOCALMACHINE::")上のアクティブな WinCC プロジェクトの WinCC OPC UA サーバーの名前領域が表示されます。



- ① WinCC の特定の名前領域の開始ノード。
- ② WinCC タグの表示。この構造は WinCC のタグの構造に対応します。
- ③ データログの表示。

WinCC タグの表示

タググループ、通信ドライバ、および接続が「FolderType」タイプの OPC UA オブジェクトによって表示されます。これらの各フォルダには、下位のオブジェクトおよびタグに対する「Organizes」タイプの参照が含まれます。

内部および外部の WinCC タグが「DataItemType」タイプの OPC UA タグによって表示されます。追加で WinCC タグがログされる場合、表示される OPC UA タグには、ログ設定のための「HasHistoricalConfiguration」タイプの参照が追加で含まれます。「Historizing」および「AccessLevel」属性がそれぞれセットされます。

以下の表は、WinCC タグを表す OPC UA タグの最も重要な属性を示しています。属性の完全なリストは、『OPC UA Part 3 - Address Space Model 1.03 Specification』文書の「§5.6」に記載されています。

属性	説明	コメント
NodeId	WinCC タグの一意の指定	-
BrowseName	WinCC タグ名	-
DisplayName	WinCC タグ名	-

属性	説明	コメント
Value	タグ値およびステータス	-
DataType	たとえば以下のような、WinCC タグタイプに対応する OPC UA データタイプ <ul style="list-style-type: none"> • Int32(符号付き 32 ビット値) • UInt32(符号なし 32 ビット値) 	-
AccessLevel	「CurrentRead」 / 「CurrentWrite」 「HistoryRead」 / 「HistoryWrite」	WinCC タグ設定に対応して
ValueRank	常に「Scalar」	-

書き込み保護および読み取り保護

クライアントによるアクセスから WinCC OPC UA サーバータグを保護できます。

WinCC プロジェクトのタグ管理で、[オプション]グループのタグのプロパティエリアで、次の設定を有効化します。

プロパティ	ランタイムでの動作
OPC 書き込み保護	クライアントは、タグ値への読み取りアクセス権のみを持ちます。
OPC 読み取り保護	クライアントは、タグ値の読み取りも書き込みもできません。

WinCC 構造タイプのマッピングなし

WinCC 構造を OPC UA サーバーのタイプとしてマッピングできません。

OPC UA タイプを WinCC 構造タグとリンクできるだけです。

ロギングタグの表示

プロセス値および圧縮ログが「FolderType」タイプの OPC UA オブジェクトによって表示されます。これらの各フォルダには、関連するロギングタグに対する「Organizes」タイプの参照が含まれます。

プロセス値または圧縮ログからのロギングタグが「BaseDateVariableType」タイプの OPC UA オブジェクトによって表示されます。ロギングタグには常に、ログ設定のための「HasHistoricalConfiguration」タイプの参照が含まれます。

8.10 WinCC OPC UA サーバー

以下の表は、WinCC ロギングタグを表す OPC UA タグの最も重要な属性を示しています。属性の完全なリストは、『OPC UA Part 3 - Address Space Model 1.03 Specification』文書の「§5.6」に記載されています。

属性	説明	コメント
NodeId	ロギングタグの一意的指定	-
BrowseName	アーカイブタグの名前	-
DisplayName	アーカイブタグの名前	-
Description	ノードの説明	-
Value	使用不可	ロギングタグの場合、この属性の読み出しも変更もできません。
DataType	たとえば以下のような、WinCC タグタイプに対応する OPC UA データタイプ <ul style="list-style-type: none"> • Double、64 ビット浮動小数点数 • UInt32(符号なし 32 ビット値) 	-
AccessLevel	「HistoryRead」 / 「HistoryWrite」	-
ValueRank	常に「Scalar」	-

WinCC メッセージへのアクセス

WinCC 名前空間の開始ノードは、Event Notifier であり、OPC UA クライアントが Subscriptions 経由でランタイム時に WinCC メッセージのステータス変更を受け取ることができるようにします。

8.10.6 OPC UA データアクセス

内部および外部の WinCC タグが「DataItemType」タイプの OPC UA タグによって表示されます。他の DataAccess タグタイプ(「AnalogItem」や「DiscreteType」)はサポートされていません。

WinCC OPC A&E サーバーは、「DataType」や「AccessLevel」などの OPC UA タグ属性に対する読み出しアクセスをサポートします。書き込みアクセスおよびサブスクリプションは、「Value」属性についてのみサポートされています。

8.10.7 OPC UA Historical Access

はじめに

[OPC Historical Access]は、アーカイブへのアクセスを可能にします。これには、[Historical Data]および[Historical Events]サービスが含まれます。WinCC OPC UA サーバーは、「Historical Data」サービスのみサポートします。

WinCC OPC UA サーバーは、「Services」を介したタグアーカイブの生データへのアクセスを OPC クライアントに提供します。

- HistoryRead (READRAW)
- HistoryUpdate (INSERTDATA、REPLACEDATA、UPDATEDATA、DELETE_RAW)

OPC UA クライアントを使用すると、タグアーカイブにあるアーカイブタグの値を読み出し、制限付きで書き込むことができます。タグアーカイブの設定に応じて、アーカイブタグに生データまたは処理済みのプロセス値のいずれかを格納することができます。

アーカイブタグの特性

WinCC のプロセスタグは、複数のタグのアーカイブ内に保管することができます。この場合、プロセスタグは対応するアーカイブタグのいずれかにリンクされます。

プロパティ/アーカイブ設定の Properties

以下の表は、「HistoricalConfigurationType」タイプの OPC UA タグ設定の Properties を示しています。[Description]プロパティでは、WinCC で設定されたアーカイブタグコメントが表示されます。プロパティの完全なリストが「OPC UA Part 11 - Historical Access 1.03 Specification」文書の「§5.2.2」に記載されています。

Property	説明/値	コメント
Definition	WinCC プロセスタグ名	プロセス値アーカイブの場合
Stepped	True	-

以下のオプションの Properties はサポートされていません。

- MaxTimeInterval
- MinTimeInterval

- ExceptionDeviation
- ExceptionDeviationFormat

Service「HistoryUpdate」の制限

Service「HistoryUpdate」は、プロセス値アーカイブでのみ使用できます。

以下の表は、WinCC OPC UA サーバーでサポートされているファンクションの一覧です。どのファンクションがサポートされるかは、WinCC OPC UA サーバーの設定とプロセス値アーカイブの設定によって決まります。追加情報が『OPC UA Part 11 - Historical Access 1.03 Specification』文書の「§5.5」に記載されています。

Service	ファンクション	説明
HistoryUpdate	INSERTDATA	新規アーカイブ値の挿入
	REPLACEDATA	既存アーカイブ値の置換
	UPDATEDATA	挿入アーカイブ値の置換
	DELETE_RAW	アーカイブ値の削除

8.10.8 OPC UA アラームと条件

概要

WinCC 7.3 の時点で、OPC UA サーバーは WinCC メッセージシステムのメッセージへのアクセスを提供しています。

OPC UA サーバーは、WinCC メッセージステータスの変更を Subscriptions と Monitored Event Items 経由で、WinCC-Event-Notifications を使用して OPC UA クライアントに転送しますが、その名前空間に Condition インスタンスを保持しません。

使用する Event Notifier ノードは、WinCC 名前領域の開始ノードです。

UA クライアントはメッセージをフィルタリングし、返されるメッセージ属性のリストを定義します。

OPC UA サーバーは、"OPC UA Alarms & Conditions 1.03"仕様をサポートします。

次のセクションでは、WinCC メッセージシステムの OPC UA へのマッピングについて説明しています。詳細情報については、「Part 9: Alarms and Conditions 1.03 Specification」の仕様を参照してください。

OPC UA イベントタイプにマッピングされる WinCC メッセージシステム

BaseEventType

「BaseEventType」は、OPC UA Event タイプ「WinCCEventType」と「WinCCAlarmConditionType」の派生元である基本タイプです。

注記

フィルタはすべての WinCC メッセージを表示します

「BaseEventType」にフィルタを適用すると、すべての WinCC メッセージが表示されます。

WinCC メッセージは次の OPC UA イベントタイプにマップされます。

WinCCEventType

このタイプは「BaseEventType」に基づいており、次の確認理論を使用して「簡単な」WinCC メッセージにマッピングされます。

- [発信ステータスなしのメッセージ]が有効であること。
- [確認着信]が有効でないこと。

このタイプのメッセージの例は、モーターの開始と停止です。

WinCCAlarmConditionType

このタイプは「AlarmConditionType」に基づいており、たとえば確認可能メッセージやステータスが「着信」および「発信」になっているメッセージのように、「WinCCEventType」にマップできないすべてのメッセージをマップします。

「WinCCAlarmConditionType」タイプのメッセージでは、イベントは条件にリンクされます。たとえば、WinCC はタグ制限が違反されるとすぐにメッセージを生成します。OPC UA 内のこのメッセージは、Alarm Condition と同等です。

WinCC メッセージ属性

2 つの Event タイプが、WinCC 固有のメッセージ属性を基本的なタイプに追加します。属性は UA Event Properties として 1:1 でマッピングされます。詳細はセクション「WinCC メッセージシステムの属性」で説明しています。

メッセージクラスとメッセージタイプ

WinCC メッセージシステムは、プロセス内の妨害とオペレーティング状態をユーザーに通知します。WinCC メッセージは常に、特定のメッセージクラスとメッセージタイプに属します。これは、対応する UA Events の "CLASSID"、"TYPEID"、"CLASSNAME"、および "TYPENAME" 属性で指定されます。

優先度

WinCC メッセージシステムでメッセージを設定するとき、"0"~"16"の優先度を構成できます。OPC UA 仕様では、Severity の値の範囲は"1"から"1000"に定義されています。この場合、"1"は最低で、"1000"は最高の Severity です。

したがって優先度の値は、OPC 重要度に適切にマッピングされる必要があります。標準のマッピングでは、"0"の優先度は OPC-Severity "1"に割り付けられ、"16"の優先度は OPC-Severity "1000"に割り当てられます。他の値はすべて、"0"から"1000"の間で等間隔に割り付けられます。

OPC UA のマッピングルール

WinCC メッセージシステムの設定中、設定を実行して、メッセージを生成するプロセスイベントを決定します。このメッセージは通常、OPC UA 内の Event として表示されます。

次の表には、Events の最も重要な Properties と、WinCC メッセージシステムが情報を提供する方法を示しています。

OPC UA プロパティ	WinCC メッセージシステムでのマッピング
すべてのイベントタイプ:	
EventID	一意のメッセージ指定
EventType	イベントタイプ:WinCCAlarmConditionType ノードまたは WinCCEventType ノードのノード ID
SourceNode	該当なし
SourceName	メッセージのソースを示します。以下にマッピングの詳細を説明します。
Message	対応するメッセージ番号のメッセージテキスト。
Time	イベントの時刻。タイムスタンプは UTC で提供されます
Severity	WinCC メッセージの優先度
WinCCAlarmConditionType の場合のみ:	
ConditionName	メッセージだけで無く出力されるテキストを設定します。テキストの出力は、設定されるマッピングルールによって異なります: <ul style="list-style-type: none"> 「モード 1」と「モード 2」:メッセージ番号 「モード 3」:メッセージクラス、たとえば「プロセスコントロールメッセージ」
Quality	メッセージの品質を返します
ConditionClassId	「ProcessConditionClassType」ノードのノード ID

OPC UA プロパティ	WinCC メッセージシステムでのマッピング
ConditionClassName	「ProcessConditionClassType」
Retain	保留メッセージの場合、「TRUE」
NodeId	ConditionId:UA-Condition の一意の指定、たとえばアラーム。サポートされる Condition インスタンスがない場合でも、確認が必要
EnabledState	メッセージが有効な場合、「TRUE」
ActiveState/Id	メッセージが着信している場合、「TRUE」
AckedState/Id	メッセージが確認済みの場合、「TRUE」
ClientUserId	ログオンしているユーザーを示します

注記

次の OPC UA Condition と Alarm Properties は、OPC UA サーバーによりサポートされていません。

- BranchId
- LastSeverity
- InputNode
- ConfirmedState
- SuppressedState
- ShelvingState
- SuppressedOrShelved
- MaxTimeShelved

メッセージステータス/確認ステータス

次の表は、WinCC メッセージステータスと対応する WinCCAlarmConditionType - Properties へのマッピングを示します。

メッセージステータス	EnabledState/Id	ActiveState/Id	AckedState/Id
ロックされたメッセージ	FALSE	-	-
有効なメッセージ	TRUE		
受信されたメッセージ	TRUE	TRUE	FALSE
送信されたメッセージ(確認あり)	TRUE	FALSE	TRUE
送信されたメッセージ(確認なし)	TRUE	FALSE	FALSE

メッセージステータス	EnabledState/Id	ActiveState/Id	AckedState/Id
確認されたメッセージ(メッセージは保留中)	TRUE	TRUE	TRUE
確認されたメッセージ(メッセージはもう保留中ではない)	TRUE	FALSE	TRUE
受信済み、確認されたメッセージ	TRUE	TRUE	TRUE
受信済み、送信されたメッセージ(確認あり)	TRUE	FALSE	TRUE
受信済み、送信されたメッセージ(確認なし)	TRUE	FALSE	FALSE
システムによる確認されたメッセージ(メッセージは保留中)	TRUE	TRUE	TRUE
システムによる確認されたメッセージ(メッセージはもう保留中ではない)	TRUE	FALSE	TRUE
緊急確認されたメッセージ(メッセージは保留中)	TRUE	TRUE	TRUE
緊急確認されたメッセージ(メッセージはもう保留中ではない)	TRUE	FALSE	TRUE

WinCC メッセージシステムのマッピングの設定

OPC UA サーバーの設定は、メッセージの Properties 「SourceName」と「Message」のマッピングに関して、OPC UA サーバーにも適用されます。

- 階層的アクセス機能付き OPC A&E サーバーの場合:

SourceName	メッセージのソースを示します。Source のフォーマットは「<サーバー接頭語>::Area\UserTextBlock 2」です。ローカルコンピュータのサーバー接頭語は「@LOCALMACHINE」です。
Message	対応するメッセージ番号のメッセージテキストを返します

- 階層的アクセス機能のない OPC A&E サーバーの場合:

SourceName	メッセージのソースを示します。Source のフォーマットは「<サーバー接頭語>::localhost::」となります。ローカルコンピュータのサーバー接頭語は「@LOCALMACHINE」です。
Message	対応するメッセージ番号のメッセージテキストを返します

アラームグループ

WinCC 7.3 では、WinCC アラームグループは名前領域に表示されません。

サポートされるイベントメソッド

確認

WinCC メッセージは、標準 OPC UA 情報モデルの「AcknowledgeableConditionType」ノードの「Acknowledge」メソッドを使用して確認されます。

「WinCCAlarmConditionType」タイプのメッセージのみを確認できます。

ConditionRefresh

まだ保留中のメッセージは、標準 OPC UA 情報モデルの「ConditionType」ノードの「ConditionRefresh」メソッドを使用して確認されます。

フィルタ

OPC UA クライアントは、Monitored Event Items のフィルタを定義できます。

8.10 WinCC OPC UA サーバー

ただし、次のオペレータは、OPC UA サーバーによりサポートされていません。

- FilterOperator_Cast
- FilterOperator_BitwiseAnd
- FilterOperator_BitwiseOr
- FilterOperator_RelatedTo
- FilterOperator_InView

下記も参照

WinCC メッセージシステムの属性 (ページ 804)

8.10.9 WinCC メッセージシステムの属性

概要

以下の表は、WinCC メッセージシステムの構成可能な属性の一覧を示しています。属性は、UA Event Properties として 1:1 でマッピングされます。

WinCC メッセージ属性	意味	データタイプ
CLASSNAME	メッセージクラス名	String
TYPENAME	メッセージタイプの名前	String
FORECOLOR	着信、送信、および確認メッセージの前景色。	Int32
BACKCOLOR	着信、送信、および確認メッセージの背景色。	Int32
FLASHCOLOR	点滅色	Int32
FLAGS	確認必須のメッセージであることを示します。	Int32
TEXT01...TEXT10	ユーザーテキストブロック#1....#10 のコンテンツ	String
PROCESSVALUE01...PROCESSVALUE10	プロセス値ブロック#1....#10 のコンテンツ	
STATETEXT	ステータスメッセージ	String
INFOTEXT	メッセージの情報テキスト	String

WinCC メッセージ属性	意味	データタイプ
LOOPINALARM	LoopInAlarm が構成されているかどうかを示します。	Int32
CLASSID	メッセージクラス ID	Int32
TYPEID	メッセージタイプ ID	Int32
MODIFYSTATE	メッセージステータスタグの値	Int32
AGNR	メッセージを生成した自動システムの番号を出力します。	Int16
CPUNR	メッセージを生成した CPU の番号を出力します。	Int16
DURATION	メッセージの着信状態、送信状態および確認の間の期間を出力します	Int32
COUNTER	ランタイムの開始後のメッセージの数	Int32
QUITSTATETEXT	メッセージが確認されたかどうか示します。	String
QUITCOUNT	未処理で未確認のメッセージの数	Int32
PARAMETER	メッセージのコンフィグレーションパラメータ	Int32
BLOCKINFO	メッセージブロックの現在のコンテンツ	String
ALARMCOUNT	保留中メッセージの数	Int32
LOCKCOUNT	ロックされたメッセージの数	Int32
PRIORITY	メッセージの優先度	Int32
APPLICATION	メッセージをトリガしたアプリケーションを出力します。	String
COMPUTER	メッセージを処理したコンピュータ名を出力します。	String
USER	メッセージを処理したユーザー名を出力します。	String
COMMENT	メッセージコメント	String
HIDDEN-COUNT	非表示メッセージの数	Int32
OS-HIDDEN	メッセージが非表示であることを示します。	Boolean
OS_EVENTID	WinCC メッセージ番号	Int32
BIG_COUNTER	メッセージカウンタ	Int64

下記も参照

OPC UA アラームと条件 (ページ 798)

8.10.10 WinCC OPC UA サーバーの設定

8.10.10.1 WinCC OPC UA サーバーの設定ファイル

はじめに

WinCC OPC UA サーバーは、設定ファイル「OPCUAServerWinCC.xml」を使用して構成します。

設定ファイルは複数のセクションに分かれています。このセクションでは、設定ファイルのレイアウトについて説明します。

「OPC UA サーバーのコンフィグレーション方法 (ページ 811)」の章では、WinCC OPC UA サーバーを設定する方法について説明します。

設定ファイルのパス

WinCC OPC UA サーバーには、2 つの設定ファイル「OPCUAServerWinCC.xml」があります。

設定ファイル	ストレージパス
サーバー固有設定ファイル	<WinCC インストールパス>\opc\UAServer\
プロジェクト固有設定ファイル	<WinCC プロジェクトフォルダ>\OPC UAServer

設定ファイルの編集

設定ファイルの変更を実行するには、次の認証が必要です。

サーバー固有設定ファイル	Windows 管理者権限
プロジェクト固有設定ファイル	ユーザーは、「SIMATIC HMI」ユーザーグループのメンバーでなければなりません。

注記

同一のパラメータ:ファイルの優先度

一部のパラメータは両方の設定ファイルに含まれています。

両方のパラメータが一致しない場合、プロジェクト固有の設定ファイルの設定がより高い優先度を持ちます。

構造:<SecuredApplication>セクション

本セクションでは、OPC UA アプリケーションのセキュリティが、OPC UA 仕様/パート 6/§「セキュリティ設定管理」に準拠して設定されます。

「OPC UA のセキュリティ概念 (ページ 783)」の URL で追加情報を参照できます。

<SecuredApplication>	
<BaseAddresses> <...></...> </BaseAddresses>	WinCC OPC UA サーバーの URL の設定
<SecurityProfileUris> <SecurityProfile> <...></...> </SecurityProfile> ... </SecurityProfileUris>	サポートされているセキュリティポリシーの設定 [none]設定は、試験および診断の目的でのみ使用します。
<ApplicationCertificate> <TrustedCertificateStore> > <TrustedCertificates> <...>	OPC UA 仕様/パート 6 によるデフォルトの証明書設定の改訂。 (オプション) これらのパラメータは、サーバー固有の設定ファイルにのみ含まれています。
</SecuredApplication>	

例:OPC UA アプリケーションセキュリティ

```
<OPCUA_Server_WinCCUA
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ua="http://opcfoundation.org/UA/2008/02/Types.xsd"
  xmlns:s1="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
  <SecuredApplication xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <ApplicationName>OPCUA Server for Simatic WinCC UA Runtime</ApplicationName>
    <ApplicationUri>urn:[HostName]:Siemens.Automation.WinCCUA.Rt</ApplicationUri>
    <ProductName>Simatic WinCC UA</ProductName>
    <ApplicationType>Server</ApplicationType>
    <ApplicationCertificate>...</ApplicationCertificate>
    <TrustedCertificateStore>...</TrustedCertificateStore>
    <IssuerCertificateStore>...</IssuerCertificateStore>
    <RejectedCertificatesStore>...</RejectedCertificatesStore>
```

構造:<ServerConfiguration>セクション

サーバー固有パラメータがこのセクションで設定されます。

メッセージセキュリティモードに関する詳細情報については、「OPC UA のセキュリティ概念 (ページ 783)」を参照してください。

<ServerConfiguration>	
<SecurityPolicies> <SecurityPolicy> <...></...> </SecurityPolicy> ... </SecurityPolicies>	メッセージセキュリティモードの設定 [none]設定は、試験および診断の目的でのみ使用します。
<UserTokenPolicies> <UserTokenPolicy> <...></...> </UserTokenPolicy> ... </UserTokenPolicies>	ユーザー ID の設定 [Anonymous]設定は、試験および診断の目的でのみ使用します。

<pre><FastInsert> <Users> <...></...> </Users> <Clients> <...></...> </Clients> </FastInsert></pre>	最適化した WinCC アーカイブ書き込みアクセスの設定
</ServerConfiguration>	

構造:<CertificateDescriptor>セクション

<ServerConfiguration>セクションの<CertificateDescriptor>ヘッダーの下で、WinCC OPC UA サーバーの証明書パラメータを指定します。

これらのパラメータは、サーバー固有の設定ファイルにのみ含まれています。

「OPC UA のセキュリティ概念 (ページ 783)」でインスタンス証明書に関する追加情報を参照できます。

<ServerConfiguration>	
<CertificateDescriptor>	
<pre><OrganizationUnit>...</...> <Organization>...</...> <Country>...</...></pre>	<p>説明エレメント</p> <p>パラメータは変更でき、アプリケーションの機能には影響はありません。</p>
<KeyLength>...</...>	<p>証明書を作成するときに使われた秘密鍵の長さ</p> <p>長さは署名のアルゴリズムに依存します。</p> <ul style="list-style-type: none"> 1024:OPC UA を介した安全な通信のための最短の長さ 2048:Sha256 が使用されるときに最短の長さ ¹⁾
<pre><SignatureAlgorithm>... </...></pre>	<p>証明書に署名するために使用される署名アルゴリズム</p> <ul style="list-style-type: none"> 可能な値:Sha1、Sha224、Sha256、Sha384、Sha512 通常値:Sha1、Sha256 デフォルト値:鍵の長さが 2048 の Sha256 ¹⁾

<pre><LifetimeInMonths>...</ ...></pre>	<p>証明書の有効期間(月単位)</p> <p>指定された時間が経過した後、サーバーはこの証明書を使用して操作できなくなります。</p> <ul style="list-style-type: none"> デフォルト値:60
<pre></ CertificateDescriptor> </ServerConfiguration></pre>	

1) セキュリティポリシー「Basic256Sha256」を使用した安全な接続を確立するには、このサーバーおよび OPC UA クライアントには次の値を持つ証明書が必要です。

- KeyLength:最低 2048
- SignatureAlgorithm:Sha256

例:証明書の制御のためのパラメータ

```
<ServerConfiguration>
  <CertificateDescriptor>
    <OrganizationUnit>DF PL DER HMI</OrganizationUnit>
    <Organization>Siemens AG</Organization>
    <Country>DE</Country>
    <KeyLength>2048</KeyLength>
    <SignatureAlgorithm>SHA256</SignatureAlgorithm>
    <LifetimeInMonths>60</LifetimeInMonths>
  </CertificateDescriptor>
```

サーバー証明書のストレージパスの変更

必要に応じて、WinCC OPC UA サーバーの証明書の保管場所を、プラントの管理者が変更することができます。

これらのパラメータをサーバー固有の設定ファイルでのみ変更することができます。

パラメータ	値	意味
StoreType	Directory	証明書ストレージのタイプ。 保管場所は、「ディレクトリ」でなければなりません。
StorePath	[ApplicationPath]\PKI WINCC-OPC-UA-Server	証明書と秘密鍵は、このフォルダに保存されます。

例:サーバー証明書のストレージパス

```
<ApplicationCertificate>
  <StoreType>Directory</StoreType>
  <StorePath>[ApplicationPath]\PKI\OPCUA</StorePath>
  <SubjectName>OPCUA Server for Simatic WinCC UA Runtime</SubjectName>
  <Thumbprint />
</ApplicationCertificate>
```

新しいサーバー証明書の作成

OPC UA サーバーに対して新しい証明書を作成するには、管理者権限が必要です。

1. バックアップを作成します。
2. 該当するフォルダにある既存の証明書と関連する秘密鍵を削除します。
3. 設定ファイルで、証明書パラメータを更新し、XML ファイルを保存します。
4. Windows で管理者権限を使用して DOS ウィンドウ「cmd.exe」を開きます。
5. 証明書を作成するには、OPC UA アプリケーションのインストールパスに進みます。
6. 次の呼び出しを入力します。
 - OpcUaServerWinCC.exe /CreateCertificateストレージパスに新しい証明書と秘密鍵が作成されます。

8.10.10.2 OPC UA サーバーのコンフィグレーション方法

前提条件

WinCC プロジェクトが作成されました。

コンフィグレーションファイルを開く

1. Windows エクスプローラを開きます。ディレクトリ「<WinCC プロジェクトフォルダ>OPC \UAServer」に移動します。
2. 「OPCUAServerWinCC.xml」コンフィグレーションファイルを開きます。詳細情報については、「WinCC OPC UA サーバーの設定ファイル (ページ 806)」を参照してください。

WinCC OPC UA サーバーのポート番号の変更

1. 必要であれば、[<BaseAddresses>]にあるポート番号 **4862** を変更します。
すでに別のアプリケーションに割り当てられているポート番号を使用しないでください。
パラメータ[HostName]はコンピュータ名のプレースホルダであり、ランタイム時に決まります。
例:

```
<BaseAddresses>
<ua:String>opc.tcp://[HostName]:5210</ua:String>
</BaseAddresses>
```

セキュリティ設定の指定

1. 通信用のセキュリティ設定を指定します。詳細については、「OPC UA のセキュリティ概念 (ページ 783)」を参照してください。
2. [<SecurityProfileUri>]で、サポートされている「セキュリティポリシー」をコンフィグレーションします。
 - [真]で設定を有効にします。
 - [偽]で設定を無効にします。
 例:


```
<SecurityProfile>
<ProfileUri>http://opcfoundation.org/UA/
SecurityPolicy#None</ProfileUri>
<Enabled>>false</Enabled>
</SecurityProfile>
```
3. [<SecurityPolicies>]で、関連する[メッセージセキュリティモード]をコンフィグレーションします。
設定を無効にするには、エントリ<SecurityPolicy>... </Security Policy>全体を削除します。
例:


```
<SecurityPolicy>
<ProfileUri>http://opcfoundation.org/UA/SecurityPolicy#None</
ProfileUri>
<MessageSecurityModes>None</MessageSecurityModes>
</SecurityPolicy>
```

注記

クライアントとサーバー間のセキュリティが確保されない通信

[none]設定は、試験および診断の目的でのみ使用します。

プロダクションモードの安全なクライアント/サーバー通信では、少なくとも以下の設定を使用する必要があります。

- SecurityPolicy::Basic128Rsa15
MessageSecurityMode::Sign

ユーザー ID の指定

1. [<UserTokenPolicies>]にある接続を設定するには、ユーザー ID を指定します。詳細情報については、「OPC UA のセキュリティ概念 (ページ 783)」を参照してください。設定を無効にするには、エントリ全体を削除します。

例

```
<UserTokenPolicy>  
<TokenType>Anonymous</TokenType>  
</UserTokenPolicy>
```

最適 WinCC アーカイブ書き込みアクセスのコンフィグレーション

1. 必要な場合、 [<FastInsert>]にある最適 WinCC アーカイブ書き込みアクセスをコンフィグレーションします。
 - [真]を設定して、すべての OPC UA クライアントの WinCC アーカイブへの最適書き込みアクセスを有効にします。
 - [偽]を設定して、特定の Windows ユーザーまたは OPC UA クライアントの最適 WinCC アーカイブ書き込みアクセスを設定します。
[<Users>]にある Windows ユーザーを指定します。
[<Clients>]にある OPC UA クライアントを指定します。[ClientName]としてクライアント証明書に入力される[Common Name]を使用します。

例:

```
<EnabledByDefault>>false</EnabledByDefault>  
<Users>  
<User>domain\user1</User>  
</Users>  
<Clients>  
<Client>ClientName1</Client>  
</Clients>
```

8.11 診断

8.11 診断

トレースファイル

すべてのサーバーが、テストの目的およびトラブルシューティングのために、診断データの出力を有効にする可能性を提供します。

サーバーのデータはトレースファイルに書き込まれます。

設定

各サーバーの設定ファイルで診断データの出力を指定します。

詳細情報については SIMATIC カスタマーサポートに問い合わせてください。

REST インターフェース

9.1 WinCC の REST インターフェース

WinCC REST (REpresentational State Transfer)サービス

WinCC REST サービスは、ポートをモニタするための自己ホスト型サービスです。このサービスでは、設定された URL と選択されたポートのモニタに Microsoft C++ SDK が使用されます。

WinCC は、REST 通信を介して、ランタイムデータと設定データへのアクセスをサポートします。

- クラウドへのランタイムデータの送信は、REST プロトコルを介してタグ値の形式で行います。追加情報:
 - [インターフェース] > [WinCC/Cloud Connector] > [REST を経由したクラウドへのデータ転送 (ページ 849)]
- REST サービスとしての WinCC は、HTTPS などの柔軟な認証メカニズムを介して安全な通信をサポートします。
- REST サービスとしての WinCC を使用すると、外部アプリケーションが WinCC データにアクセスできるようになります。
外部アプリケーションは、REST インターフェースを介して WinCC の設定データとタグ値の読み取りと書き込みを行うことができます。
以下の操作がサポートされます。
 - データの照会
 - データの編集アクセスには、リソースの通常の HTTP メソッドと JSON 表示を使用します。

制約:サポートされるデータタイプ

タグにアクセスするとき、次のデータタイプはサポートされません:

- 未処理データタグ
- テキスト参照

API 構造

WinCC は、WinCC ランタイムデータと設定データを送信するために HTTP リクエストと HTTP レスポンスを使用します。

9.1 WinCC の REST インターフェース

これらのリクエストとレスポンスは以下の要素から構成されます。

Header	リクエストのメタデータ
Body	<ul style="list-style-type: none"> リクエスト本体として WinCC API に送信されるデータ レスポンス本体として WinCC API によって送信されるデータ
Service Endpoint	サービスエンドポイント: WinCC API サービスのネットワークアドレスを含むベース URL

HTTP メソッド

メソッド	説明
GET	サーバーからリソースを読み取ります。 WinCC プロジェクトからランタイム値と設定データを読み取るために使用されます。
POST	サーバーからリソースを読み取ります。 Body 内のリソースのアドレスを指定します。 複数の値を送信するために使用されます。
PUT	書き込みアクセス URL を介してリソースのアドレスを指定します。 個々の WinCC 値をクラウドに送信するために使用されます。

エラーコード

レスポンスコード	説明	コメント
200	OK	WinCC API はリクエストを正常に処理しました。
400	Bad Request	無効なリクエスト
401	Unauthorized	アクセス許可なし
402	Payment Required	有効な WinCC ライセンスが見つかりませんでした
404	Not Found	レスポンスが見つからない URL が WinCC API のリソースタイプと一致しません。 URL の形式を修正してください。

レスポンスコード	説明	コメント
416	RangeNotSatisfiable	無効なリクエスト URL 構造が不正です。要求された範囲は対応されません。 この理由は、「GET」が呼び出されるときに、アドレス指定されたリソースの名前が欠落していることなどです。
423	Locked	リソースがロックされています。 この理由は、アドレス指定されたデータの読み取りが現在進行していることなどです。 待機時間が経過した後でリクエストをもう一度送信してください。
500	InternalError	内部エラー サーバーによって予期しないエラーが報告されました。
501	NotImplemented	無効なリクエスト サーバーは要求された機能をサポートしていません。

下記も参照

REST を経由したクラウドへのデータ転送 (ページ 849)

メソッドの概要 (ページ 820)

設定データのフィルタ (ページ 837)

WinCC クラウドコネクタの REST 設定 (ページ 858)

WinCC REST サービスの設定方法 (ページ 818)

9.2 WinCC REST サービスの設定方法

[コンピュータのプロパティ]ダイアログの[REST 設定]タブで、REST サーバーへの接続を設定します。

HTTPS 経由の安全な通信を使用するには、REST サーバーおよびそのサーバーにインストールされた証明書を選択します。

WinCC ユーザーと権限

タグ管理:タグアクセスの権限

[タグ管理]では、タグごとに値の読み取りまたは書き込みを特定の権限に制限できます。

[WinCC クラウド]グループのタグプロパティの以下のフィールドで、適切な権限を選択します。

- 読み取り権限
- 書き込み権限

ユーザー管理者:権限の割り当て

REST サービスとしての WinCC を介したアクセスを許可するには、ユーザー管理者で WinCC ユーザーを作成します。

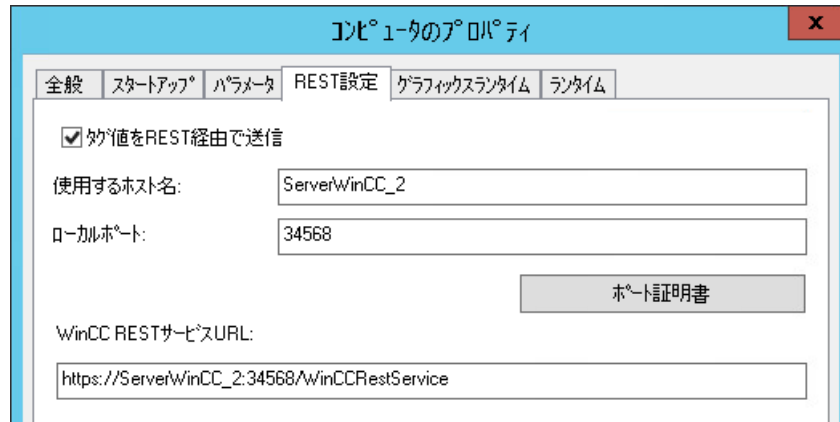
タグにアクセスするには、WinCC ユーザーにタグプロパティで定義されている権限を付与します。

REST 設定

設定	説明
使用されるホスト名	WinCC サーバー
ローカルポート	アクセスに使用されるポート番号
ポート証明書	ボタンをクリックしてインストールされた証明書を選択します。
WinCC REST サービス URL	HTTPS パスは、設定から形成されます。

手順

1. [コンピュータのプロパティ]ダイアログを開きます。
2. [REST 設定]タブで、[REST 経由でタグ値を送信]オプションを有効にします。



3. 接続データを指定します:
 - 使用されるホスト名
 - ローカルポートサーバーへのパスが、[WinCC REST サービス URL]フィールドに表示されます。
4. 認証された接続をセットアップするには、[ポート証明書]ボタンをクリックします。
[証明書]ダイアログが開かれます。
5. 証明書を選択し、[OK]をクリックして確定します。
6. [OK]を押してダイアログボックスを閉じます。
7. [REST Service]アプリケーションがコンピュータのスタートアップリストでまだ有効化されていない場合、それを有効化するためのプロンプトが表示されます。
[はい]をクリックして、有効化を確定します。
このアプリケーションが無効に設定されている場合、REST サービスは WinCC Runtime で実行されません。

下記も参照

WinCC の REST インターフェース (ページ 815)

メソッドの概要 (ページ 820)

9.3 メソッドの概要

PUT:サポートされているリクエスト

- 値のタグへの書き込み (ページ 835)
- 値の複数タグへの書き込み (ページ 835)

GET / POST:サポートされているリクエスト

- タグのランタイム値の読み取り (ページ 821)
- すべてのタグのランタイム値の読み取り (ページ 821)
- 接続の設定データの読み取り (ページ 822)
- すべての接続の設定データの読み取り (ページ 823)
- タグの設定データの読み取り (ページ 824)
- すべてのタグの設定データの読み取り (ページ 825)
- タググループの設定データの読み取り (ページ 827)
- すべてのタググループの設定データの読み取り (ページ 827)
- 構造タイプの設定データの読み取り (ページ 829)
- すべての構造タイプの設定データの読み取り (ページ 829)
- 構造タイプのインスタンスの読み取り (ページ 832)
- 複数の構造タイプのインスタンスの読み取り (ページ 833)

下記も参照

WinCC の REST インターフェース (ページ 815)

設定データのフィルタ (ページ 837)

WinCC REST サービスの設定方法 (ページ 818)

9.4 GET / POST メソッド

9.4.1 タグのランタイム値の読み取り

説明

ランタイムで、プロセスタグまたは内部タグのタグ値を読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/TagManagement/Value/<タグの名称>

本文

なし

戻り値:例

```
{ "variableName": "Tag2",  
  "dataType": 2,  
  "value": "0",  
  "timestamp": "2020-03-20T09:12:38.5830Z",  
  "qualitycode": "0x4C - uncertain - initial value",  
  "errorcode": 0 }
```

9.4.2 すべてのタグのランタイム値の読み取り

説明

ランタイムですべての登録されたプロセスタグおよび内部タグのタグ値を読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/Values

本文

```
{
  "variableNames" : [ "タグ名 1", "タグ名 2", "タグ名 3", "タグ名 4" ]
}
```

戻り値:例

```
[
  { "variableName": "Tag1",
    "dataType": 1,
    "value": "false",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorCode": 0 },
  { "variableName": "Tag2",
    "dataType": 2,
    "value": "0",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorCode": 0 },
  { "variableName": "Tag10",
    "dataType": 5,
    "value": "この変数にはテキスト値が含まれます",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": "0x4C - uncertain - initial value",
    "errorCode": 0 },
  { "variableName": "Tag",
    "error": "見つかりませんでした" }
]
```

9.4.3 接続の設定データの読み取り**説明**

通信ドライバの下に作成された接続の設定データを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
Connection/<接続名>

本文

なし

戻り値:例

```
{  
  "connectionName": "NewConnection_Name",  
  "channelUnit": "OMS+",  
  "channelName": "SIMATIC S7-1200、S7-1500 チャンネル",  
  "lastChange": "2020-01-30 15:08:31.000"  
}
```

9.4.4 すべての接続の設定データの読み取り**説明**

タグ管理の通信ドライバの下に作成される全接続の設定データを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
Connections

本文

なし

戻り値:例

```
{ "connections": [
  { "connectionName": "内部タグ",
    "channelUnit": "内部タグ",
    "channelName": "内部タグ",
    "lastChange": "2020-01-30 15:01:30.297" },
  { "connectionName": "NewConnection_Name",
    "channelUnit": "OMS+",
    "channelName": "SIMATIC S7-1200、S7-1500 チャンネル",
    "lastChange": "2020-01-30 15:08:31.000" }
]}
```

代替アドレス指定**URL**

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
Connections?itemLimit=20
```

戻り値:例

```
{ "continuationPoint": 2,
  "moreAvailable": false,
  "connections": [
    { "connectionName": "内部タグ",
      "channelUnit": "内部タグ",
      "channelName": "内部タグ",
      "lastChange": "2020-01-30 15:01:30.297" },
    { "connectionName": "NewConnection_Name",
      "channelUnit": "OMS+",
      "channelName": "SIMATIC S7-1200、S7-1500 チャンネル",
      "lastChange": "2020-01-30 15:08:31.000" }
  ]
}
```

9.4.5 タグの設定データの読み取り**説明**

プロセスタグまたは内部タグの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/variable/<タグの名前>

本文

なし

戻り値:例

```
{ "variableName": "Tag2",  
  "dataType": 2,  
  "comment": "Tag2 のコメント",  
  "canRead": true,  
  "canWrite": true,  
  "lastChange": "2020-03-03 13:48:29.000" }
```

9.4.6 すべてのタグの設定データの読み取り**説明**

すべての登録されたプロセスタグおよび内部タグの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/variables

本文

なし

戻り値:例

```
{ "variables":  
  [  
    { "variableName": "@SCRIPT_COUNT_TAGS",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    { "variableName": "@SCRIPT_COUNT_REQUESTS_IN_QUEUES",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    ... ]
```

代替アドレス指定**URL**

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/variables?  
itemLimit=2&continuationPoint=0
```

戻り値:例

```
{ "continuationPoint": 3,  
  "moreAvailable": true,  
  "variables": [  
    { "variableName": "@SCRIPT_COUNT_TAGS",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"},  
    { "variableName": "@SCRIPT_COUNT_REQUESTS_IN_QUEUES",  
      "dataType": 2,  
      "comment": "",  
      "canRead": true,  
      "canWrite": true,  
      "lastChange": "2020-03-20 09:12:37.000"}  
  ] }
```

9.4.7 タググループの設定データの読み取り

説明

通信ドライバの下に作成されたタググループの設定データまたは内部グループとして読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/Group/<グループ名>

本文

なし

戻り値:例

```
{
  "groupName": "ProcessHistorian",
  "connectionName": "内部タグ",
  "lastChange": "2020-06-08 14:13:14.467"
}
```

9.4.8 すべてのタググループの設定データの読み取り

説明

通信ドライバの下に作成されたタググループの設定データまたは内部グループとして読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/Groups

本文

なし

戻り値:例

```
{ "groups": [
  { "groupName": "Script",
    "connectionName": "内部タグ",
    "lastChange": "2020-06-08 14:13:14.463" },
  { "groupName": "TagLoggingRt",
    "connectionName": "内部タグ",
    "lastChange": "2020-06-08 14:13:14.467" },
  { "groupName": "ProcessHistorian",
    "connectionName": "内部タグ",
    "lastChange": "2020-06-08 14:13:14.467" },
  { "groupName": "Performance",
    "connectionName": "内部タグ",
    "lastChange": "2020-06-08 14:13:14.470" },
  { "groupName": "Connection1_Group_1",
    "connectionName": "NewConnection_1",
    "lastChange": "2020-06-08 15:23:57" },
  { "groupName": "Connection2_Group_1",
    "connectionName": "NewConnection_2",
    "lastChange": "2020-06-08 15:24:39" }
] }
```

代替アドレス指定**URL**

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
Connections?itemLimit=2&continuationPoint=0
```

戻り値:例

```
{ "continuationPoint": 3, "moreAvailable": true,
  "groups": [
    {
      "groupName": "Script",
      "connectionName": "内部タグ",
      "lastChange": "2020-06-08 14:13:14.463" },
    { "groupName": "TagLoggingRt",
      "connectionName": "内部タグ",
      "lastChange": "2020-06-08 14:13:14.467" }
  ] }
```

9.4.9 構造タイプの設定データの読み取り

説明

[構造タグ]の下に作成された構造タイプの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement /
StructureType/<構造の名前>

本文

なし

戻り値:例

```
{ "typeName": "NewStructure_1",  
  "typeMembers": [  
    { "memberName": "NewElement_1",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:06.000" },  
    { "memberName": "NewElement_2",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:07.000" },  
    { "memberName": "NewElement_3",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:08.000" },  
    { "memberName": "NewElement_4",  
      "dataType": 2,  
      "lastChange": "2020-03-19 08:46:09.000" }  
  ] }
```

9.4.10 すべての構造タイプの設定データの読み取り

説明

[構造タグ]の下に作成されたすべての構造タイプの設定データを読み取ります。

9.4 GET / POST メソッド

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
StructureTypes

本文

なし

戻り値:例

```
{ "structures":  
  [  
    { "typeName": "NewStructure_1", "typeMembers": [  
      { "memberName": "NewElement_1",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:06.000"},  
      { "memberName": "NewElement_2",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:07.000"},  
      { "memberName": "NewElement_3",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:08.000"},  
      { "memberName": "NewElement_4",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:09.000"} ] },  
    { "typeName": "NewStructure_2", "typeMembers": [  
      { "memberName": "NewElement_1",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:12.000"},  
      { "memberName": "NewElement_2",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:13.000"},  
      { "memberName": "NewElement_3",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:14.000"} ] },  
    { "typeName": "NewStructure_3", "typeMembers": [  
      { "memberName": "NewElement_1",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:18.000"},  
      { "memberName": "NewElement_2",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:20.000"},  
      { "memberName": "NewElement_3",  
        "dataType": 2,  
        "lastChange": "2020-03-19 08:46:21.000"}  
    ] }  
  ]  
}
```

代替アドレス指定

URL

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/  
StructureTypes?itemLimit=2&continuationPoint=0
```

戻り値:例

```
{
  "continuationPoint":1026,
  "moreAvailable":true,
  "structures":[
    {
      "typeName":"NewStructure_1",
      "typeMembers":[
        {
          "memberName":"NewElement_1",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:06.000"},
        {
          "memberName":"NewElement_2",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:07.000"},
        {
          "memberName":"NewElement_3",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:08.000"},
        {
          "memberName":"NewElement_4",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:09.000"}
      ]
    },
    {
      "typeName":"NewStructure_2",
      "typeMembers":[
        {
          "memberName":"NewElement_1",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:12.000"},
        {
          "memberName":"NewElement_2",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:13.000"},
        {
          "memberName":"NewElement_3",
          "dataType":2,
          "lastChange":"2020-03-19 08:46:14.000"}
      ]
    }
  ]
}
```

9.4.11 構造タイプのインスタンスの読み取り**説明**

[構造タグ]の下に作成された構造タイプの構造タグエレメントを読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/
StructureVariable/<構造タイプの名前>

本文

なし

戻り値:例

```
{ "structureVariables":  
  [  
    { "variableName": "StructureInstance1_1",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:15.000"},  
    { "variableName": "StructureInstance1_2",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"},  
    { "variableName": "StructureInstance1_3",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"}  
  ]  
}
```

9.4.12 複数の構造タイプのインスタンスの読み取り

説明

[構造タグ]の下に作成された複数の構造タイプの構造タグエレメントを読み取ります。

呼び出し

GET / POST

URL

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/  
StructureVariables
```

本文

```
{  
  "typeNames" : [ "NewStructure_1", "NewStructure_2" ]  
}
```

戻り値:例

```
{ "structureVariables":  
  [  
    { "variableName": "StructureInstance1_1",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:15.000"},  
    { "variableName": "StructureInstance1_2",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"},  
    { "variableName": "StructureInstance1_3",  
      "typeName": "NewStructure_1",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:20.000"},  
    { "variableName": "StructureInstance2_1",  
      "typeName": "NewStructure_2",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:32.000"},  
    { "variableName": "StructureInstance2_2",  
      "typeName": "NewStructure_2",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:36.000"},  
    { "variableName": "StructureInstance2_3",  
      "typeName": "NewStructure_2",  
      "connectionName": "内部タグ",  
      "comment": "",  
      "lastChange": "2020-03-19 17:38:36.000"}  
  ]  
}
```

9.5 PUT メソッド

9.5.1 値のタグへの書き込み

説明

同期的に値をタグに書き込みます。

呼び出し

PUT

URL

https://<ホスト>:<ポート>/WinCCRestService/TagManagement/Value/<タグ
の名称>

本文

```
{"value": "テキスト値"}
```

戻り値:例

```
{"variableName": "Tag11",  
  "errorcode": 0}
```

9.5.2 値の複数タグへの書き込み

説明

同期的に一覧表示されたタグに値を書き込みます。

各タグに対して書き込まれる値を指定します。

呼び出し

PUT

9.5 PUT メソッド

URL

https://<ホスト>:<ポート>/WinCCRestService/TagManagement/Values

本文:例

```
[  
  {"variableName":"タグ名 1","value":true},  
  {"variableName":"タグ名 2","value":6},  
  {"variableName":"タグ名 3","value":8},  
  {"variableName":"タグ名 4","value":10}  
]
```

戻り値:例

```
[  
  {"variableName":"タグ名 1","errorcode":0},  
  {"variableName":"タグ名 2","errorcode":0},  
  {"variableName":"タグ名 3","error":"見つかりませんでした"},  
  {"variableName":"タグ名 4","errorcode":0}  
]
```

9.6 設定データのフィルタ

フィルタを使用してリクエストの範囲を絞ることができます。
これにより転送されるデータの量を低減します。

フィルタの挿入

フィルタは、URL の一部として転送されます:

- <URL>?<フィルタ基準>

例:

- https://<URL>/WinCCRestService/tagManagement/variables?canWrite

フィルタ文字列は、大文字と小文字を区別しません。

リンク

複数の多様なフィルタを使用できます。

論理演算では、オペランド「&」を使用します。例:

- <URL>?<フィルタ基準_1>&<フィルタ基準_2>&<フィルタ基準_3>

関連付けるフィルタを増やして、検索結果をさら限定します。OR 論理演算は使用できません。

1 つのリクエストで、各フィルタパラメータを 1 回のみ使用することができます。

無効なフィルタの組み合わせの例:

- variables?variableName=mot*&variableName=!motor

フィルタ基準:パラメータ

パラメータ	オブジェクト	説明
canRead	タグ	ログオンユーザーが読み取りアクセスを持つタグの名前を返します。
canWrite	タグ	ログオンユーザーが書き込みアクセスを持つタグの名前を返します。
variableName	タグ	検索された文字列を含むすべてのタグの名前を返します。 プレースホルダ「?」および「*」を使用することができます。

9.6 設定データのフィルタ

パラメータ	オブジェクト	説明
connection	タグ:接続データ	タグ、構造タグまたはタググループが下に作成される接続に関する情報を提供します。
structureType	構造タグ	構造タイプに対応するすべての構造タイプエレメントの名前を返します。
structureVariable	構造タグ	構造タイプに対応するすべての構造タグエレメントの名前を返します。
group	タググループ	タググループにあるすべてのタグの名前を返します。
channel	通信チャンネル	チャンネルの下にあるすべての接続の名前を返します。
changed_After	設定データ	指定された日付以降に変更された設定データを返します。 日付フォーマット:YYYY-MM-DD hh:mm:ss:ms 例:2015-12-31 16:30:00:000
itemLimit	設定データ	返されるエレメント数を制限します。 これにより、大規模な WinCC プロジェクトのパフォーマンスを向上します。
continuationPoint	設定データ	[itemLimit]経由で制限されたリクエストを続行します。 見つかったエレメント数が[itemLimit]経由で設定された値よりも大きい場合、出力は最大値に到達したときに終了されます。追加のリクエストでは、見つかったすべてのエレメントが返されるまで出力を続けることができます。
* ?	ワイルドカード	タグ名にフィルタを掛けるとき、ワイルドカードのみを使用することができます (variableName)。 <ul style="list-style-type: none"> ? :文字列の任意の文字 * :文字列の先頭または末尾にある任意の数
!	オペランド:否定 (NOT)	リクエストごとに、1つの NOT フィルタのみを使用することができます。
&	オペランド:論理演算 (AND)	各リクエストに対して、複数の AND 論理演算を使用できます。

例:ワイルドカード

名前に[Motor]を含むタグをリクエストします:

- /variables?variableName=*motor*

名前が[parfum]または[perfum]で始まるタグをリクエストします:

- /variables?variableName=p?rfum*

例: channel

チャンネル[SIMATIC S7-1200, S7-1500 Channel]の下にある接続名をリクエストします:

- /connections?channel=SIMATIC S7-1200, S7-1500 Channel

WinCC タグ管理に表示されるときに、通信チャンネルの名前を引用します。

例:itemLimit / continuationPoint

リクエストは、ログオンユーザーが書き込みアクセスを持つすべてのタグを返す必要があります。

合計で 250 タグが見つっていますが、[itemLimit]を使用して最大 90 タグまでに制限します。

- /variables?canWrite&itemLimit=90

応答本文には先頭の 90 タグが含まれます。

後続の 90 タグを照会するには、リクエストを繰り返し、[continuationPoint]を追加します。

- /variables?canWrite&itemLimit=90&continuationPoint

応答本文には後続の 90 タグが含まれます。

このリクエストを再度繰り返して、最後の 70 タグを照会します。

例: ! (NOT)

changed_After:指定された時刻前

特定の時刻までに作成されたすべての接続をリクエストするには、[changed_After]の否定を選択します:

- /connections?changed_After!=2015-11-26 10:30:02.000

9.6 設定データのフィルタ

connection:接続名の除外

次のタグの接続データをリクエストしたい場合:

- タグ名には、文字列「_S7_」が含まれます。
- タグは、接続「S7_1」の下には作成されません。

フィルタ:

- /variables?Connection!=S7_1&variableName=*_S7_*

下記も参照

WinCC の REST インターフェース (ページ 815)

メソッドの概要 (ページ 820)

WinCC/Cloud Connector

10.1 WinCC/Cloud Connector

WinCC/Cloud Connector を使用すると、WinCC ステーションからクラウドに自動的にタグを転送できます。

より詳細な分析のためにクラウドに保存されたデータを使用したり、ダッシュボードなどを介して、タグ値を出力することができます。

クラウドプロバイダ

WinCC/Cloud Connector は、次のプロバイダをサポートします:

- Siemens MindSphere - MindConnect IoT Extension
- Amazon: AWS
- Microsoft: Azure
- Alibaba Cloud: PolarDB, AnalyticDB (Aliyun、中国)
- 汎用 MQTT
- REST プロトコル

Cloud Connector の動作方法

[Cloud Connector]は、タグ値の転送に、[MQTT]または[REST]プロトコルを使用します。

通信のセキュリティを向上するため、証明書ハンドシェイクによって暗号化された接続を使用します。

Cloud Connector を使用したデータの送信のみが可能です。データは、WinCC では受信されません。

Message Queue Telemetry Transport (MQTT)

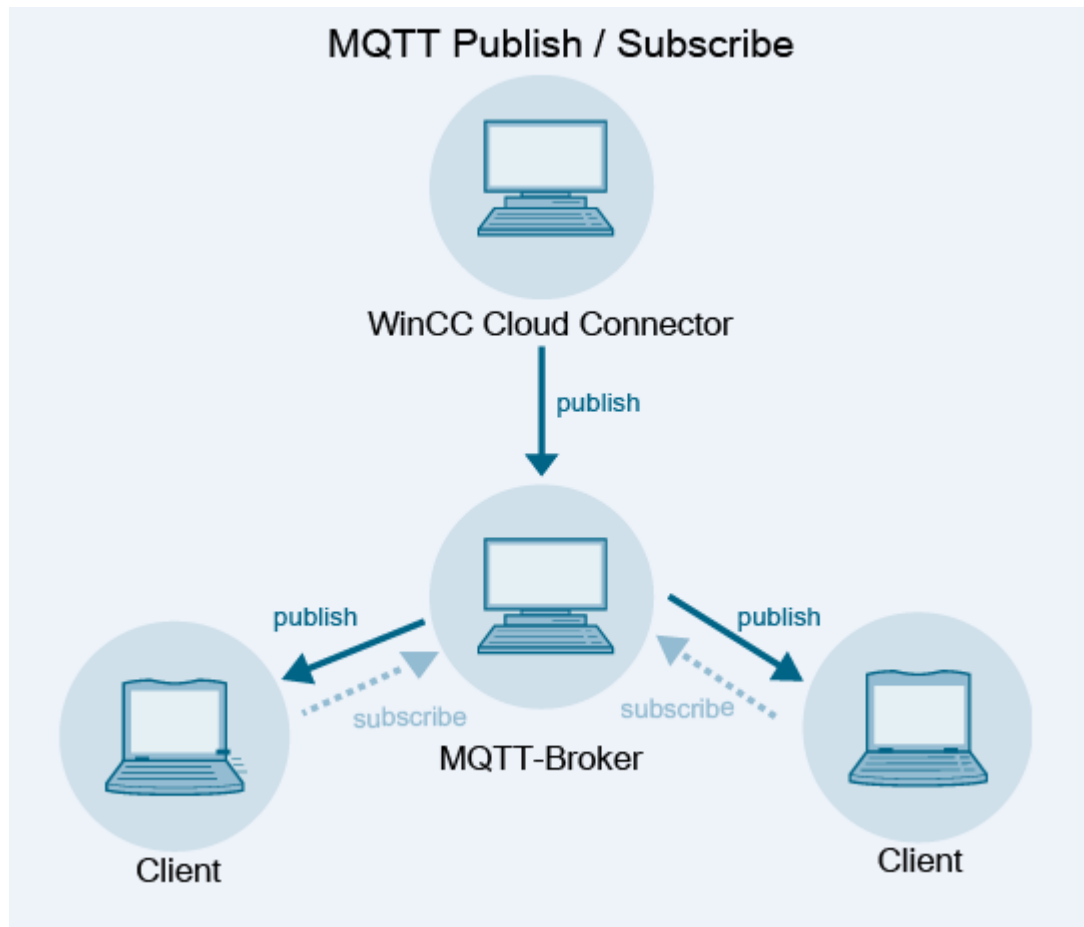
MQTT プロトコルを使用すると、中央サーバーである MQTT ブローカがデータ転送に使用されます。

送信デバイスと受信デバイス間のデータ転送は、MQTT ブローカを介して排他的に実行されます。

MQTT プロトコルに関する追加情報は、インターネットで参照できます:

- <https://mqtt.org/> (<https://mqtt.org/>)

MQTT による通信



Representational State Transfer (REST)

REST プロトコルは、分散クライアントサーバーシステムの通信に HTTP または HTTPS を使用します。

REST を介したデータ交換

MQTT と同様、データは中央サーバーに送信されます。

任意のプロバイダをアドレス指定できます。プロバイダでユーザー名とパスワードを使用して、確認を行います。

REST サービスとしての WinCC

REST サービスとしての WinCC を使用すると、外部アプリケーションが WinCC データにアクセスできるようになります。

HTTP メソッドを介した、WinCC の REST インターフェースへのアクセスについての詳細:

- [インターフェース] > [REST インターフェース (ページ 815)]

下記も参照

WinCC クラウドコネクタの REST 設定 (ページ 858)

WinCC の REST インターフェース (ページ 815)

MQTT プロトコル (<https://mqtt.org/>)

10.2 ライセンシングクラウドコネクタ

ライセンス

WinCC/Cloud Connector 用に別個のライセンスが必要です:

- SIMATIC WinCC Cloud Connect

ライセンスなしでは、テスト目的で最大 5 つのタグを転送できます。

10.3 MQTT を経由したクラウドへのデータ転送

クラウドプロバイダ

データは、Cloud Connector 経由で、MQTT プロトコルを使用して書き込まれます。

次のクラウドプロバイダが現在サポートされています。

- Siemens MindSphere - MindConnect IoT Extension (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud: PolarDB, AnalyticDB (MQTT)

CCCloudConnect サービス

Windows サービス CCloudConnect は、WinCC プロジェクトとクラウドシステム間の接続を確立するために使用されます。

CCCloudConnect サービスは、標準ポート 8883 または 443 からデータを送信するために、クラウドの MQTT ブローカに接続する MQTT クライアントです。

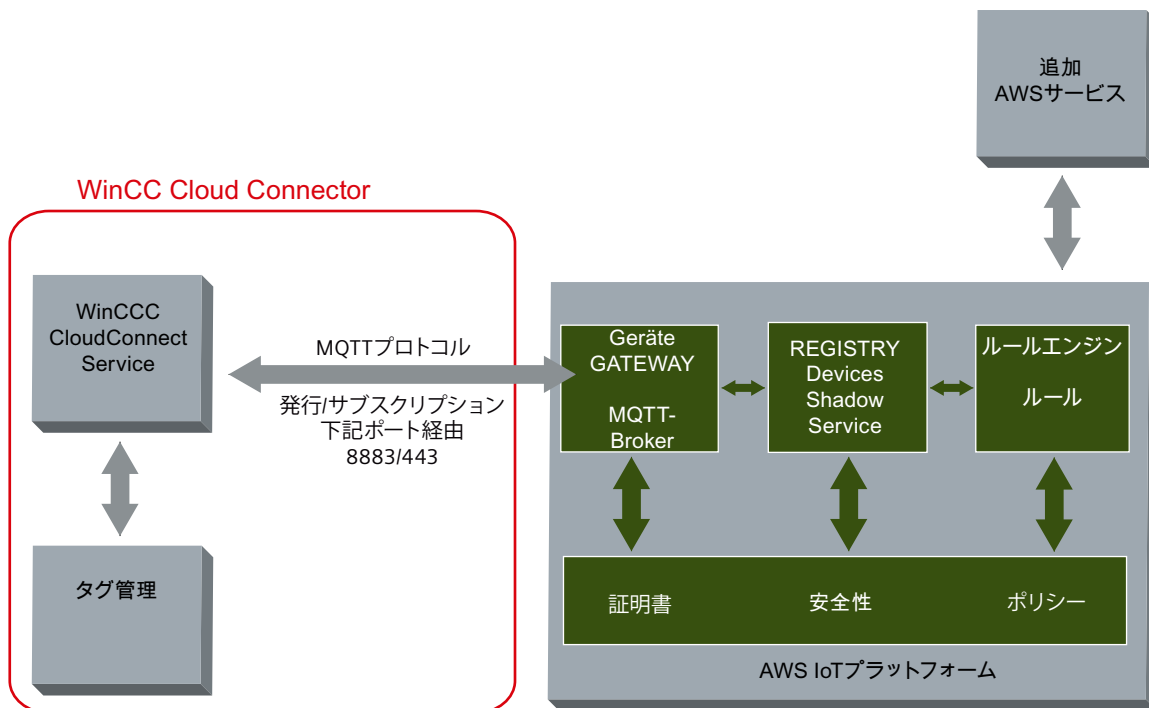
WinCC では、CCCloudConnect サービスは、WinCC タグの値の変更をログ記録します。値はクラウドに書き込まれます。

CCCloudConnect がタグ管理からの値の変化を受信した場合、サービスがメッセージを作成します。サービスは、MQTT ブローカにこのメッセージを転送します。

10.3 MQTT を経由したクラウドへのデータ転送

例:WinCC/Cloud Connector および AWS-MQTT

次の図は、Cloud Connector と AWS プラットフォーム間のデータ転送を表示しています。



MQTT トピック

命名規則

MQTT ブローカに送信される各タグに対して、個別の MQTT トピックが作成されます。

ブローカからこのトピックを受信したいすべてのクライアントは、トピックを認識する必要があります。

WinCC タグの命名規則は次のとおりです。

- <ステーション名><WinCC プロジェクト名><タグ名>

これらの値を受信することを希望する MQTT のクライアントは、適切なパスのある MQTT トピックに登録する必要があります。

Cloud Connector 設定でデフォルトステーション名「WinCC」を変更できます。

MindSphere の MQTT トピック

プロバイダ「MindSphere - MindConnect IoT Extension」を選択するとき、WinCC からタグ名を適用します。

ステーション名として、MindSphere からデバイス名を使用します。

例

WinCC プロジェクトが「MyWinCCProject」という名前で作成されています。

WinCC プロジェクトでクラウドに対して「MyTag1」と「MyTag2」という名前の 2 つのタグが有効にされます。

CCCloudConnect は、これらのタグに対して次の MQTT トピックを送信します:

- WinCC/MyWinCCProject/MyTag1
- WinCC/MyWinCCProject/MyTag2

MindSphere の例

デバイスが「WinCCStation1」という名前で作成されています。

「MyTag1」および「MyTag2」タグは、デバイス「WinCCStation1」上で最初に転送された後に表示されます。

キュー

Cloud Connector 経由でのデータ転送中、メッセージは、キュー原則に従って送信されます。

最初に追加されるメッセージが最初に送信されます。

データタイプ

ほとんどのデータタイプが転送用に許可されています。

例外

- 構造化されているデータタイプはサポートされていません。例: STRUCT や ARRAY。
- 追加の制限はそれぞれのクラウドプロバイダにより異なります。

タグタイプ[日付/時間]

[日付/時間]タグタイプのフォーマットは、使用されるクラウドに依存しています。

MindSphere のタグタイプ

MindSphere では、以下のデータタイプは WinCC タグに対してサポートされません。

- テキストタグ、8 ビットフォント
- テキストタグ、16 ビットフォント
- テキスト参照
- 日付/時刻

10.3 MQTT を経由したクラウドへのデータ転送

タイムスタンプ

タイムスタンプは WinCC ステーションによって生成され、クラウドに送信されます。

クラウドプロバイダは、タイムスタンプに対して協定世界時(UTC)を使用します。

接続割り込みに対応するためのデータバッファ

データを失うことなくクラウドへの短時間の接続割り込みをやり過ごすため、MQTT を介した通信ではデータバッファが作成されます。

転送されるタグ値は一時的にこのデータバッファに格納されます。

- 最大サイズ:1 000 タグ

接続割り込みの後、接続が復旧するとすぐに、データバッファ内のタグ値がクラウドに保存されます。

WinCC Runtime の無効化

WinCC Runtime を無効にすると、クラウドへの接続も終了します。

クラウドに送信された最後のメッセージは、診断ファイル「CCCloudConnect.log」に保存されます。

下記も参照

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

クラウド接続の診断 (ページ 870)

WinCC タグ管理での設定 (ページ 851)

10.4 REST を経由したクラウドへのデータ転送

データは、Cloud Connector 経由で、REST プロトコルを使用して書き込まれます。

Representational State Transfer (REST)

CCCloudConnect Windows サービスを介した通信とは対照的に、HTTP または HTTPS を経由して通信を確立します。

タグ値を読み取り、それらを変更時および周期的にクラウドに書き込むことができます。

値は、HTTP リクエストおよび HTTP 応答として転送されます。

REST を介したデータ交換

WinCC は、MQTT ではなく、REST を介してクラウドに値を送信します。

MQTT と同様、データは中央サーバーに送信されます。

任意のプロバイダをアドレス指定できます。

プロバイダでユーザー名とパスワードを使用して、確認を行います。

キュー

Cloud Connector 経由でのデータ転送中、メッセージは、キュー原則に従って送信されます。

最初に追加されるメッセージが最初に送信されます。

データタイプ

ほとんどのデータタイプが転送用に許可されています。

例外

- 構造化されているデータタイプはサポートされていません。例: STRUCT や ARRAY。
- 追加の制限はそれぞれのクラウドプロバイダにより異なります。

タイムスタンプ

タイムスタンプは WinCC ステーションによって生成され、クラウドに送信されます。

クラウドプロバイダは、タイムスタンプに対して協定世界時(UTC)を使用します。

10.4 REST を経由したクラウドへのデータ転送

WinCC Runtime の無効化

WinCC Runtime を無効にすると、クラウドへの接続も終了します。

クラウドに送信された最後のメッセージは、診断ファイル「CCCloudConnect.log」に保存されます。

下記も参照

WinCC の REST インターフェース (ページ 815)

10.5 WinCC タグ管理での設定

WinCC タグ管理では、転送されるタグを指定します。

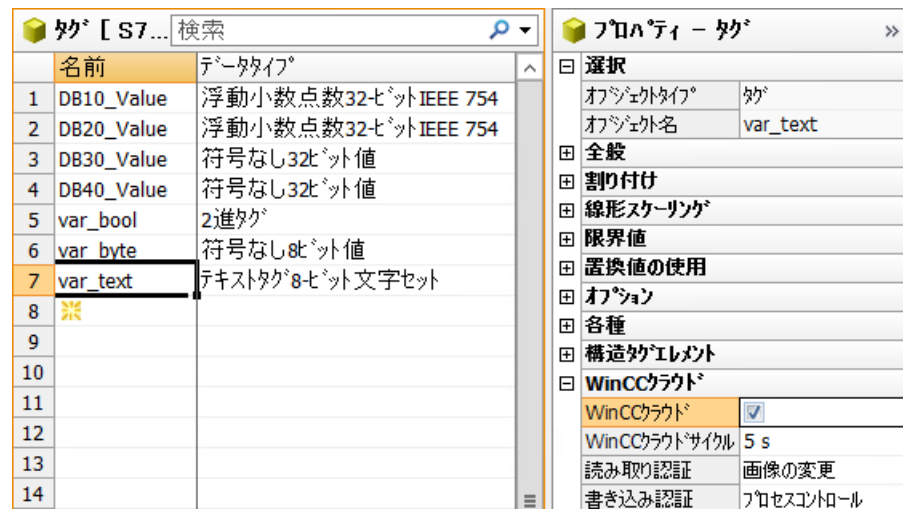
クラウドプロバイダは、それぞれ異なるタグタイプをサポートします。制限事項に関する情報は、「MQTT を経由したクラウドへのデータ転送 (ページ 845)」を参照してください。

クラウド転送の有効化

タグの値をクラウドに送信するには、タグ管理で[WinCC クラウド]タグプロパティを有効にします。

適切な取得サイクルを選択します。

権限の割り当ては、WinCC を REST サービスとして使用している場合にのみ関係します。



サイクルタイムを設定する

「WinCC クラウドサイクル」は、クラウドに転送される各タグに対して個々に設定することができます。

10.5 WinCC タグ管理での設定

この設定は、WinCC タグロギングのサイクルタイム設定に相当します。

- サイクルタイムに対して何の設定も行わない場合、1 分のサイクルが初期設定として使用されます。
- 可能な最短のサイクルタイムは 1 秒です。
- サイクルタイムに対して、タグサイクルのリストから、[値の変更時]または固定値を選択します。
 - 1 / 2 / 5 / 10 秒
 - 1 / 5 / 10 分
 - 1 時間

注記

ODK / VBA

サイクルタイムは、VBA または ODK 経由で変更できません。

下記も参照

MQTT を経由したクラウドへのデータ転送 (ページ 845)

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

クラウド接続の診断 (ページ 870)

MindSphere 接続の設定方法 (ページ 864)

REST 経由のクラウド接続の設定方法 (ページ 867)

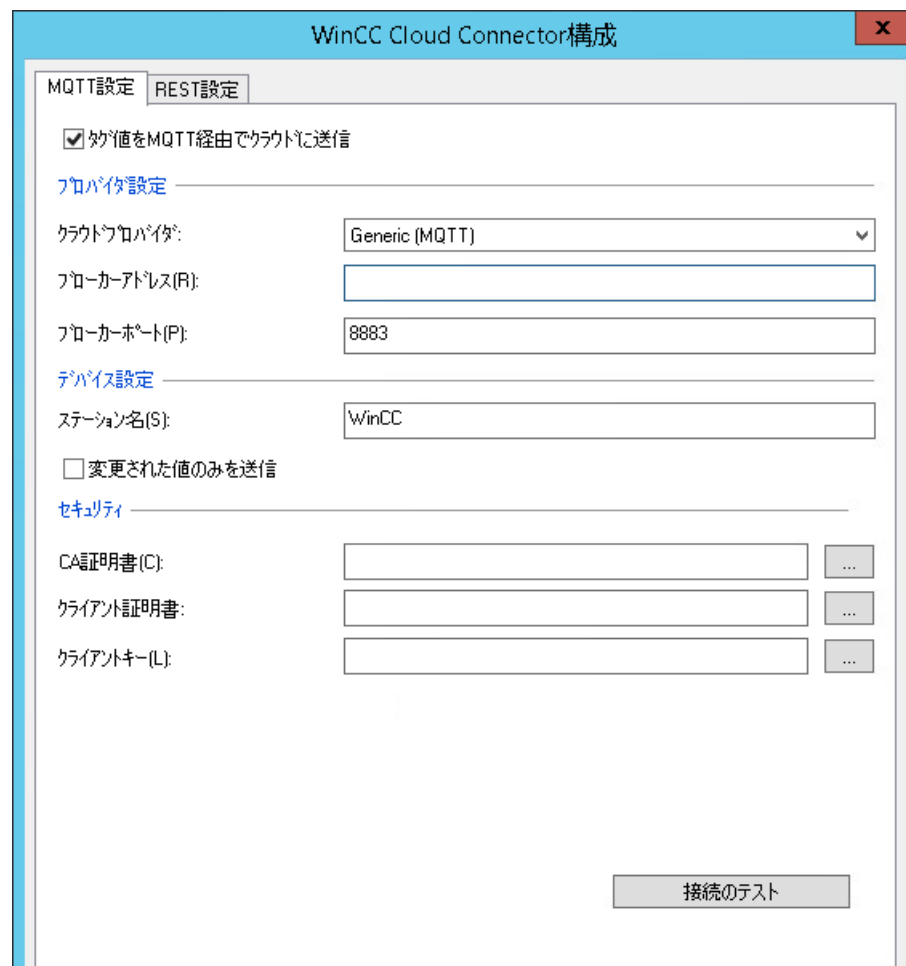
MQTT 経由のクラウド接続の設定方法 (ページ 860)

10.6 WinCC クラウドコネクタの MQTT 設定

概要

[WinCC/Cloud Connector 設定]ダイアログボックスで、使用されるクラウドの URL およびアクセス設定を構成します。

MQTT プロトコルを使用するには、[MQTT 設定]タブで[MQTT 経由でタグ値をクラウドに送信]オプションを選択します。



The image shows the 'WinCC Cloud Connector構成' (WinCC Cloud Connector Configuration) dialog box, specifically the 'MQTT設定' (MQTT Settings) tab. The dialog has a title bar with a close button (X). Inside, there are two tabs: 'MQTT設定' and 'REST設定'. The 'MQTT設定' tab is active. It contains several sections: 1. A checkbox labeled 'タグ値をMQTT経由でクラウドに送信' (Send tag values to cloud via MQTT) which is checked. 2. A section titled 'クラウドプロバイダ設定' (Cloud Provider Settings) with fields for 'クラウドプロバイダ:' (set to 'Generic (MQTT)'), 'ブローカーアドレス(R):' (empty), and 'ブローカーポート(P):' (set to '8883'). 3. A section titled 'デバイス設定' (Device Settings) with a field for 'ステーション名(S):' (set to 'WinCC'). 4. A checkbox labeled '変更された値のみを送信' (Send only changed values) which is unchecked. 5. A section titled 'セキュリティ' (Security) with fields for 'CA証明書(C):', 'クライアント証明書:', and 'クライアントキー(L):', each followed by a browse button (...). At the bottom right, there is a button labeled '接続のテスト' (Test Connection).

クラウドプロバイダ

[汎用 MQTT]またはプロバイダを選択します。

10.6 WinCC クラウドコネクタの MQTT 設定

次のプロバイダがサポートされます。

- Siemens MindSphere - MindConnect IoT Extension (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud: PolarDB, AnalyticDB (MQTT)

アプリケーションの例

多様なクラウドプロバイダに対してデータ接続を設定する方法の詳細な例については、インターネットで参照できます:

- アプリケーション例 109760955: 「クラウドの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)」

ブローカアドレス

MQTT クラウドの終了ポイントです。

MQTT ブローカアドレスは、クラウドプロバイダによって使用可能です。

ブローカポート

標準ポートがサポートされています。

- 8883
- 443

ステーション名

クライアントに一意の名前を指定します。

タグ転送中に、クライアントの名前が、MQTT トピックのパスとして使用されます。

MindSphere のステーション名

プロバイダ「MindSphere - MindConnect IoT Extension (MQTT)」を使用するとき、ステーション名がデバイス名として使用されます。

MindSphere でデバイスを登録するとき、名前がデバイス ID として指定されます。

変更された値のみ送信

このオプションを選択すると、変更されたデータのみがクラウドに送信されます。

これにより、データ転送中に負荷を減らすことができます。

CA 証明書

クラウドプロバイダからの CA 証明書を取得できます。「CA」は、「認証局」を表します。

WinCC ステーション上に証明書をローカルに保存します。

WinCC デフォルトパス:

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Certificate

Alibaba 証明書

Alibaba クラウドにアクセスするには、プロバイダの独自の証明書を使用します。

AWS 証明書

Amazon Web Services は、AWS IoT によって生成される証明書または CA 認証のある証明書を使用してデバイスを識別します。

AWS IoT 証明書は、次の CA 証明書によって署名されます。

- RSA 2048 bit key: VeriSign Class 3 Public Primary G5 root CA certificate
- RSA 2048 bit key: Amazon Root CA 1
- RSA 4096 bit key: Amazon Root CA 2
- ECC 256 bit key: Amazon Root CA 3
- ECC 384 bit key: Amazon Root CA 4

AWS IoT サーバー証明書でデバイスを認証するには、AWS は、WinCC ステーションに 5 つの証明書をすべてインストールすることをお勧めします。

Azure 証明書

Microsoft Azure クラウドは、一時的な証明書を使用します。

MindSphere 証明書

MindSphere の CA 証明書は、WinCC のインストール中にインストールされます。

[MindSphere - MindConnect IoT Extension (MQTT)]プロバイダを選択するとき、[セキュリティ]領域が非表示になります。

[MindSphere]領域が表示され、MindSphere 証明書が表示されます。

クライアント証明書

クラウドプロバイダから受け取った証明書を WinCC ステーション上にローカルに保存します。

証明書には、次のファイル拡張子があります。

- .cer
- .crt
- .pem

入力フィールドに、証明書へのパスを入力します。

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Certificate

MindSphere 証明書

[MindSphere - MindConnect IoT Extension (MQTT)]プロバイダを選択するとき、[セキュリティ]領域が非表示になります。

クライアントキー

クラウドプロバイダからクライアントキーを取得します。キーを WinCC ステーションにローカルに保存します。

クライアントキーは、[クライアント/デバイスキー]とも呼ばれ、ファイル拡張子「.key」が付きます。

入力フィールドに、クライアントキーへのパスを入力します。

- \Program Files (x86)\Siemens\WinCC\CloudConnector\Private

MindSphere 証明書

[MindSphere - MindConnect IoT Extension (MQTT)]プロバイダを選択するとき、[セキュリティ]領域が非表示になります。

登録/登録解除

このボタンは、プロバイダ「MindSphere - MindConnect IoT Extension (MQTT)」を選択したときにのみ表示されます。

[登録]ボタンは、MindSphere のデバイス登録を開始します。

登録に成功した後、このボタンはグレーアウトされ、[登録解除]ボタンが代わりに有効になります。

WinCC ステーションの設定を変更するには、[登録解除]をクリックします。デバイスは引き続き MindSphere で作成され、既存のデータが保持されます。

ユーザー名/パスワード

このフィールドは、プロバイダ「MindSphere - MindConnect IoT Extension (MQTT)」を選択したときにのみ表示されます。

ユーザー名およびパスワードは、MindSphere の登録中に指定され、Cloud Connector に表示されます。

MindSphere のパスワードを変更するとき、Cloud Connector で新しいパスワードを適用する必要があります。

下記も参照

MQTT を経由したクラウドへのデータ転送 (ページ 845)

MQTT 経由のクラウド接続の設定方法 (ページ 860)

クラウド接続の診断 (ページ 870)

WinCC タグ管理での設定 (ページ 851)

MindSphere 接続の設定方法 (ページ 864)

REST 経由のクラウド接続の設定方法 (ページ 867)

WinCC クラウドコネクタの REST 設定 (ページ 858)

適用例:クラウドへの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)

10.7 WinCC クラウドコネクタの REST 設定

概要

[WinCC/Cloud Connector 設定]ダイアログボックスで、使用されるクラウドの URL およびアクセス設定を構成します。

REST プロトコルを使用するには、[REST 設定]タブで[REST 経由でタグ値をクラウドに送信]オプションを選択します。

WinCC Cloud Connector構成

MQTT設定REST設定

☒タグ値をREST経由でクラウドに送信

プロバイダ設定

サービスアドレス(S):

サービスポート(P):

8080

サービスパス(P):

送信方式:

すべてのタグ値を"POST"コメントを使用して送信します。

☐変更された値のみを送信する

基本認証

ユーザー名(U):

パスワード(P):

プロバイダ設定

プロバイダの接続データの入力:

設定	説明
サービスアドレス	プロバイダの HTTP アドレス
サービスポート	アクセスに使用されるポート番号 デフォルトでは、HTTP ポート「8080」が設定されています。
サービスパス	サーバーディレクトリへのパス
送信方法	転送方法: <ul style="list-style-type: none">PUT:タグのすべての値を送信POST:すべてのタグのタグ値を送信

変更された値のみ送信

このオプションを選択すると、変更されたデータのみがクラウドに送信されます。

これにより、データ転送中に負荷を減らすことができます。

基本認証

REST サーバーでのアクセス保護のためにユーザー名とパスワードを設定します。

[ユーザー名]フィールドと[パスワード]フィールドに REST サーバー用のアクセスデータを入力してください。

下記も参照

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

WinCC/Cloud Connector (ページ 841)

REST 経由のクラウド接続の設定方法 (ページ 867)

WinCC の REST インターフェース (ページ 815)

10.8 MQTT 経由のクラウド接続の設定方法

概要

[WinCC クラウドコネクタ設定]ダイアログボックスで使用する、クラウドの URL およびアクセス設定を指定します。

アプリケーションの例

多様なクラウドプロバイダに対してデータ接続を設定する方法の詳細な例については、インターネットで参照できます:

- アプリケーション例 109760955: 「クラウドの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)」

その他の手順

[MindSphere (MindConnect IoT Extension)]をクラウドプロバイダとして選択する場合、次の指示に従います:

- MindSphere 接続の設定方法 (ページ 864)

REST 経由のクラウド接続を設定するには、下記の指示に従います:

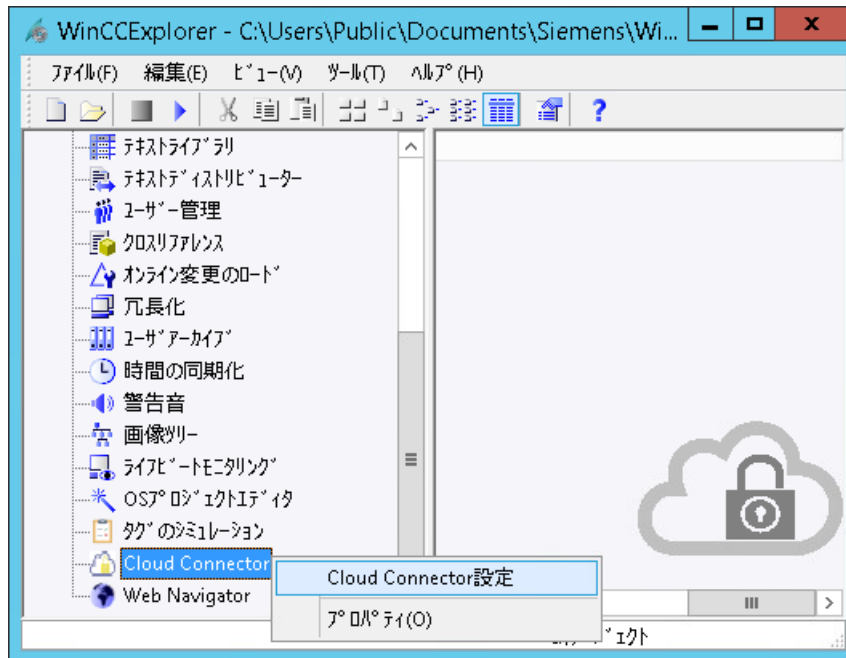
- REST 経由のクラウド接続の設定方法 (ページ 867)

必要条件

- タグを転送するためには、WinCC タグ管理で[WinCC クラウド]オプションが有効に設定されている必要があります。

手順

1. WinCC エクスプローラを開きます。
2. [Cloud Connector]のショートカットメニューで、[クラウドコネクタ設定]エントリを選択します。



[WinCC クラウドコネクタ設定]ダイアログボックスが開きます。

10.8 MQTT 経由のクラウド接続の設定方法

3. [MQTT 設定]タブで、[MQTT 経由でタグ値をクラウドに送信]オプションを選択します。

The image shows the 'WinCC Cloud Connector構成' (WinCC Cloud Connector Configuration) dialog box. It has two tabs: 'MQTT設定' (MQTT Settings) and 'REST設定' (REST Settings). The 'MQTT設定' tab is active. Inside the tab, there is a checkbox labeled 'タグ値をMQTT経由でクラウドに送信' (Send tag values to cloud via MQTT), which is checked. Below this, there are three sections: 'クラウドプロバイダ設定' (Cloud Provider Settings), 'デバイス設定' (Device Settings), and 'セキュリティ' (Security). Under 'クラウドプロバイダ設定', there are three fields: 'クラウドプロバイダ:' (Cloud Provider) with a dropdown menu showing 'Generic (MQTT)', 'ブローカーアドレス(R):' (Broker Address), and 'ブローカーポート(P):' (Broker Port) with the value '8883'. Under 'デバイス設定', there is one field: 'ステーション名(S):' (Station Name) with the value 'WinCC'. Under 'セキュリティ', there is a checkbox labeled '変更された値のみを送信' (Send only changed values), which is unchecked. Below this, there are three fields for certificates/keys: 'CA証明書(C):' (CA Certificate), 'クライアント証明書:' (Client Certificate), and 'クライアントキー(L):' (Client Key), each with a text box and a browse button (...). At the bottom right of the dialog, there is a button labeled '接続のテスト' (Test Connection).

4. 接続データを指定します:
- クラウドプロバイダ
 - ブローカーアドレス
 - ブローカーポート
 - ステーション名
- データ転送中の負荷を低減するには、[変更された値のみを送信]オプションを選択します。
5. 証明書およびクライアントキーを選択します。
6. 接続設定をテストするには、[接続のテスト]ボタンをクリックします。
7. [OK]を押してダイアログボックスを閉じます。
8. [Cloud Connector]アプリケーションがコンピュータのスタートアップリストでまだ有効化されていない場合、それを有効化するためのプロンプトが表示されます。
[はい]をクリックして、有効化を確定します。
アプリケーションを無効化する場合、Cloud Connector サービスは WinCC Runtime で実行されません。

下記も参照

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

MQTT を経由したクラウドへのデータ転送 (ページ 845)

MindSphere 接続の設定方法 (ページ 864)

REST 経由のクラウド接続の設定方法 (ページ 867)

WinCC タグ管理での設定 (ページ 851)

適用例:クラウドへの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)

10.9 MindSphere 接続の設定方法

概要

[WinCC クラウドコネクタ設定]ダイアログボックスで使用する、クラウドの URL およびアクセス設定を指定します。

MindSphere に接続しているときに、WinCC と一緒にインストールされた CA 証明書が使用されます。

追加の認証のため、MindSphere は、ユーザー名およびパスワードを作成します。

アプリケーションの例

多様なクラウドプロバイダに対してデータ接続を設定する方法の詳細な例については、インターネットで参照できます:

- アプリケーション例 109760955: 「クラウドの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)」

必要条件

- タグを転送するためには、WinCC タグ管理で[WinCC クラウド]オプションが有効に設定されている必要があります。

手順

1. WinCC エクスプローラを開きます。
2. [Cloud Connector]のショートカットメニューで、[クラウドコネクタ設定]エントリを選択します。
[WinCC クラウドコネクタ設定]ダイアログボックスが開きます。

3. [MQTT 設定]タブで、[MQTT 経由でタグ値をクラウドに送信]オプションを選択します。

The image shows the 'WinCC Cloud Connector構成' (WinCC Cloud Connector Configuration) dialog box. It has two tabs: 'MQTT設定' (MQTT Settings) and 'REST設定' (REST Settings). The 'MQTT設定' tab is active. It contains the following fields and options:

- ☒ タグ値をMQTT経由でクラウドに送信
- クラウドプロバイダ設定**
 - クラウドプロバイダ: MindSphere (MindConnect IoT Extension) (dropdown menu)
 - ブローカーアドレス: (empty text box)
 - ブローカーポート: 8883 (text box)
- デバイス設定**
 - ステーション名(S): WinCC (text box)
 - ☐ 変更された値のみを送信
- MindSphere**
 - CA証明書(C): Program Files (x86)\Siemens\WinCC\bin\CertRootCA2G3.crt (text box with browse button)
 - WinCCをMindConnect IoT Extensionのデバイスとして登録し、接続を確立します。
MindConnectデバイスの登録時にステーション名はデバイスIDに対応します。
 - 登録後、構成は変更できなくなります。
 - 登録 (button)
 - 登録を解除 (button)
 - ユーザー名(U): (empty text box)
 - パスワード(A): (empty text box)
 - 接続のテスト (button)

4. 接続データを指定します:
- クラウドプロバイダ: [MindSphere (MindConnect IoT Extension)]
 - ブローカーアドレス
 - ステーション名
- デフォルトポート 8883 は、変更できません。
データ転送中の負荷を低減するには、[変更された値のみを送信]オプションを選択します。
5. ブラウザの MindSphere 設定を開きます。
6. [MindConnect IoT Extension]のエディタに切り替えます。
[デバイス]でデバイス登録を選択します。
7. 新しいデバイスを作成するには、ステーション名を入力します。
[Cloud Connector]のステーション名および[MindConnect IoT Extension]エディタのデバイス名が一致する必要があります。
- WinCC ステーションがデバイスとして作成されます。
 - [接続の待機]ステータスが表示されます。
8. [WinCC クラウドコネクタ設定]ダイアログボックスを開き、[登録]ボタンをクリックします。

10.9 MindSphere 接続の設定方法

9. ブラウザウィンドウで、[MindConnect IoT Extension]エディタに戻ります。
登録を完了するには、[承諾]をクリックします。
 - 登録が承諾されている場合、[登録]ボタンが Cloud Connector でグレイアウトされます。
 - [登録解除]ボタンが有効になります。
 - ユーザー名およびパスワードは、[MindConnect IoT Extension]エディタによって設定され、クラウドコネクタで表示されます。
[MindConnect IoT Extension]エディタでパスワードを変更するとき、クラウドコネクタで新しいパスワードを適用する必要があります。
10. 接続設定をテストするには、Cloud Connector の[接続のテスト]ボタンをクリックします。
11. [OK]をクリックしてダイアログを閉じます。
12. [Cloud Connector]アプリケーションがコンピュータのスタートアップリストでまだ有効化されていない場合、それを有効化するためのプロンプトが表示されます。
[はい]をクリックして、有効化を確定します。
アプリケーションを無効化する場合、Cloud Connector サービスは WinCC Runtime で実行されません。

下記も参照

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

WinCC タグ管理での設定 (ページ 851)

MQTT 経由のクラウド接続の設定方法 (ページ 860)

REST 経由のクラウド接続の設定方法 (ページ 867)

適用例:クラウドへの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)

10.10 REST 経由のクラウド接続の設定方法

概要

REST プロトコルを使用して、データを HTTP プロバイダに送信できます。

[WinCC クラウドコネクタ設定]ダイアログボックスで使用する、クラウドの URL およびアクセス設定を指定します。

その他の手順

MQTT 経由のクラウド接続を設定するには、下記の指示に従います:

- MQTT 経由のクラウド接続の設定方法 (ページ 860)

[MindSphere (MindConnect IoT Extension)]をクラウドプロバイダとして選択する場合、次の指示に従います:

- MindSphere 接続の設定方法 (ページ 864)

必要条件

- タグを転送するためには、WinCC タグ管理で[WinCC クラウド]オプションが有効に設定されている必要があります。

手順

1. WinCC エクスプローラを開きます。
2. [Cloud Connector]のショートカットメニューで、[クラウドコネクタ設定]エントリを選択します。
[WinCC クラウドコネクタ設定]ダイアログボックスが開きます。

10.10 REST 経由のクラウド接続の設定方法

3. [REST 設定]タブで、[REST 経由でタグ値をクラウドに送信]オプションを選択します。

The screenshot shows the 'WinCC Cloud Connector 構成' (WinCC Cloud Connector Configuration) dialog box with the 'REST 設定' (REST Settings) tab selected. The 'MQTT 設定' (MQTT Settings) tab is also visible. The 'REST 設定' tab contains the following options and fields:

- ☒ 効値をREST経由でクラウドに送信 (Send tag values to cloud via REST)
- プロパティ設定** (Property Settings)
 - サービスアドレス(S): [Empty text box]
 - サービスポート(P): [8080]
 - サービスパス(P): [Empty text box]
 - 送信方式: [すべての効値を"POST"コマンドを使用して送信します。] (Send all tag values using "POST" command)
 - ☐ 変更された値のみを送信する (Send only changed values)
- 基本認証** (Basic Authentication)
 - ユーザー名(U): [Empty text box]
 - パスワード(P): [Empty text box]

4. 接続データを指定します:
- サービスアドレス
 - サービスポート
 - サービスパス
 - 送信方法
- データ転送中の負荷を低減するには、[変更された値のみを送信]オプションを選択します。
5. [ユーザー名]フィールドと[パスワード]フィールドに REST サーバー用のアクセスデータを入力してください。
REST サーバーでパスワードを変更する場合は、Cloud Connector で新しいパスワードを適用する必要があります。
6. [OK]を押してダイアログボックスを閉じます。
7. [Cloud Connector]アプリケーションがコンピュータのスタートアップリストでまだ有効化されていない場合、それを有効化するためのプロンプトが表示されます。
[はい]をクリックして、有効化を確定します。
アプリケーションを無効化する場合、Cloud Connector サービスは WinCC Runtime で実行されません。

下記も参照

MQTT 経由のクラウド接続の設定方法 (ページ 860)

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

WinCC タグ管理での設定 (ページ 851)

MindSphere 接続の設定方法 (ページ 864)

WinCC クラウドコネクタの REST 設定 (ページ 858)

10.11 クラウド接続の診断

概要

WinCC は、パフォーマンスタグの診断およびログファイルのメッセージの出力をサポートします。

接続監視用のインターフェースタグ

WinCC プロジェクトを作成するとき、システムタグが内部[Performance]タググループに作成されます。追加情報:

- [WinCC での作業] > [プロジェクトでの作業] > [ランタイムの設定] > [パフォーマンスタグでのシステム診断]

パフォーマンス分析用に次のシステムタグを選択できます。

システムタグ	説明
@PRF_CLDCN_RESET	リセットタグが、次のパフォーマンスタグの値をリセットします。 <ul style="list-style-type: none"> • @PRF_CLDCN_TAG_FAILED_WRITES_TOTAL • @PRF_CLDCN_TAG_WRITES_TOTAL
@PRF_CLDCN_TAG_FAILED_WRITE S_TOTAL	クラウドによって確認されなかった転送タグ数
@PRF_CLDCN_TAG_WRITES_PER_S ECOND	1 秒間で転送されるタグの数
@PRF_CLDCN_TAG_WRITES_TOTA L	接続を経由して転送されるタグの合計数

診断ファイル

「CCCloudConnect.log」ファイルが、「診断」フォルダの WinCC インストールパスに作成されます。

Eclipse Mosquitto

Eclipse Mosquitto エラーコードは、診断ファイルで「MOSQ」が付いて示されます。

詳細は、「libmosquitto API」ドキュメントを参照してください。

- <https://mosquitto.org/> (<https://mosquitto.org/>)

下記も参照

MQTT を経由したクラウドへのデータ転送 (ページ 845)

WinCC クラウドコネクタの MQTT 設定 (ページ 853)

WinCC タグ管理での設定 (ページ 851)

Internet: <https://mosquitto.org/> (<https://mosquitto.org/>)

10.11 クラウド接続の診断

索引

"

"システム情報"チャンネル, 632
診断オプション, 632
"スロット PLC"チャンネルユニット, 438
接続の構成, 438

[

[新規接続], 259

「

「MPI」チャンネルユニット
接続の設定, 429

5

505 Ethernet (CP 1413-x)
チャンネルユニット, 546

A

A&E サーバー, 752, 754, 765, 767, 768
WinCC メッセージクラスおよびメッセージタイプ
のマッピング, 768
WinCC メッセージシステムのマッピング, 754,
768
階層アクセス, 767
条件関連付けイベント, 752
条件付きイベント, 765
単純イベント, 765
追跡用イベント, 752, 765
ActiveX コントロールとしての, 625
Application Health Check, 120, 153
AR_SENDAR_SEND-FunctioninWinCC, 458
AS511, 390
AS シンボル, 416, 423, 525, 532
AS 構造, 421, 530
インポート, 416, 525
AS データタイプ, 173
タイプ変換, 187
Autostart, 113

B

BinWrite メカニズム, 201
一般的な手順, 201
Buffer Time, 762

C

CCCloudConnect サービス, 845
Channel DiagnosisChannelDiagnosis
KanalDiagnosismitChannelDiagnosis, 627
CP1434 TF, 556, 557
CPU 負荷, 586
システム情報の例,

E

ERROR フラグ, 632, 646
E サーバー
WinCC OPC A&OPC の機能, 703

I

I/O フィールド
S5 Profibus FDL の例, 388
システム情報の例,
I/O フィールド設定, 714
OPC DA の例, 714
INFO フラグ, 632, 646, 659, 670

M

Max Size, 762
MELSEC FX3U シリーズ, (三菱製 Ethernet を参照)
MELSEC iQ-R シリーズ, (三菱製 Ethernet を参照)
MELSEC Q シリーズ, (三菱製 Ethernet を参照)
MPI チャンネルユニット, 428
MQTT プロトコル, 841

N

NAT ルータ, 292

O

OPC

[新規接続], 259

HDA サーバブラウザ, 743

HDA サーバブラウザを使用した WinCC アーカイブタグへのアクセスの設定, 744

OPC A&E サーバの機能, 765

OPC DA クライアントとしての WinCC, 248, 269

OPC DA 通信に障害が発生した場合のエラー処理, 269, 334

OPC UA, 283

OPC UA クライアントとしての WinCC, 334

OPC XML クライアントとしての WinCC, 248

OPC1, 263, 281

OPCServer.WinCC-(DPC_4001), 259

OPC コンピュータでのユーザーアカウントの設定, 702

OPC 項目マネージャ, 250

OPC 項目マネージャの機能概要, 250

OPC 項目マネージャを使用した WinCC タグへのアクセスの設定, 259

OPC 項目マネージャを使用しない WinCC タグへのアクセスの設定, 263, 281

ProglID, 711

ProglID の問い合わせ, 250

WinCC OPC A&E サーバの機能, 752

WinCC OPC DA クライアントの機能, 255

WinCC OPC DA クライアント上での OPC チャンネルの設定, 258

WinCC OPC DA クライアント上での構造体の使用, 265, 268

WinCC OPC DA サーバの機能, 709

WinCC OPC HDA サーバの機能, 730

WinCC OPC UA サーバの動作原理, 782

WinCC OPC XML クライアントの機能, 275

WinCC OPC XML サーバの機能, 703

WinCC の OPC, 248, 700

WinCC メッセージシステムへのアクセスの構成, 762

アドレスプロパティ, 263, 281

サポートされる WinCC データタイプ, 254

サポートされる WinCC データタイプの概要, 254
タグの選択, 250

タグの追加, 250, 259

タグへのアクセス, 257

チャンネル診断, 669

トレース, 814

フィルタ基準, 259

互換性, 699

構造体タグの設定, 266

構造体の設定, 266

構造体プロパティ, 266

仕様, 699

新規ユーザー, 702

接続の作成, 250

通信の概念, 700

複数の OPC DA サーバの使用, 710

例, 717, 718, 720, 721, 724, 725, 726, 729

例:, 713, 714, 715, 742

OPC A&E クライアント

OPC A&E の例, 762

OPC A&E サーバ, 703, 752, 755, 762, 765, 767, 768

WinCC メッセージシステムへのアクセスの構成, 762

品質コード, 761, 775

OPC A&E サーバのメッセージクラス, 768

OPC A&E サーバのメッセージタイプ, 768

OPC A&E サーバの機能

条件付きイベント, 765

単純イベント, 765

追跡用イベント, 765

OPC A&未処理データ, 761

OPC DA クライアント, 255

WinCC OPC DA クライアントの機能, 255

OPC DA サーバ, 709

WinCC OPC DA サーバの機能, 709

複数の OPC DA サーバの使用, 710

OPC HDA サーバ, 730

WinCC OPC HDA サーバの時間形式, 736

サポートされているファンクション, 735

データ構造, 732, 733

境界値, 732

項目 ID, 732

項目ハンドル, 732

集約, 733

書込みアクセス, 739

属性, 733

動作の原則, 730

品質コード, 738, 761

未処理データ, 730

OPC UA

OPC UA クライアントとしての WinCC, 283

WinCC タグの表示, 794

タグへのアクセス, 285

データタイプ, 284

プロファイル, 792

ロギングタグの表示, 795

適合ユニット, 792

OPC UA Historical Access, 797

OPC UA クライアント, 283

OPC UA サーバー

- URL, 782
- アプリケーション証明書, 784
- インスタンス証明書, 784
- サポートされる仕様, 782
- セキュリティポリシー, 783
- 証明書, 784
- 信頼されるクライアント証明書, 785
- 設定ファイル, 806
- 通信プロファイル, 782
- 動作, 782

OPC UA データアクセス, 796

OPC XML, 706, 707

- IIS のセキュリティ設定, 706
- インストールのテスト, 707
- タグへのアクセス, 276

OPC XML クライアント, 275

- WinCC OPC XML クライアントの機能, 275

OPC XML サーバー, 703

- インストール, 704

OPC1, 263, 281

OPCScout 新規プロジェクト 1

- OPC DA の例, 721

OPCServer.WinCC-(DPC_4001), 259

OPC インターフェース, 97

- クライアント/サーバーシステムで, 97

OPC チャンネル, 669

- WinCC OPC DA クライアント上での OPC チャンネルの設定, 258
- 診断オプション, 669

OPC 項目マネージャ, 250, 259, 711

- OPC 項目マネージャの機能概要, 250

OPC 通信, 269, 334

- OPC DA 通信に障害が発生した場合のエラー処理, 269

- OPC UA 通信のエラー処理, 334

P

PROFIBUS システムパラメータ, 652

PROFIBUS チャンネルユニット, 434

- 接続の設定, 435

ProgID, 250

- ProgID の問い合わせ, 250

- 問い合わせ, 711

PSK キー, 19, 81, 83

R

REST サービス, 849

REST プロトコル, 841

S

S5 AS511, 391

- タグアドレス, 393

- タグの設定, 393

- データタイプ, 390

- バイト単位でアクセスするタグの定義, 397

- ビット単位でアクセスするタグの構成, 396

- 設定, 391

S5 Ethernet 第 4 層, 340

- iNA960 メッセージ, 360

- SCI メッセージ, 361

- タグアドレス, 345

- タグの設定, 345

- チャンネルユニットのシステムパラメータ, 353

- データタイプ, 341

- デバイス名, 353, 354

- トランスポートパラメータ, 353, 355

- バイト単位でアクセスするタグの構成, 348

- ビット単位でアクセスするタグの構成, 347

- ワード単位でアクセスするタグの構成, 350

- 接続パラメータ, 342

- 接続障害中のエラーコード, 356

- 設定, 342

- 内部エラーコードと定数, 356

- 未処理データタグ, 351

S5 Profibus FDL, 365

- S5 タグへのアクセス, 366

- SIMATIC S5 Profibus FDL の構成, 370

- アクティブなデータ転送の構成, 381

- アクティブ接続, 365

- アドレスプロパティ, 373, 374, 375, 380

- サービスアクセスポイント, 365

- サイクリックブロック, 385

- サポートされるデータタイプ, 366

- システムパラメータ, 377, 378

- スタートアップブロック, 385

- タイプ変換, 368

- タグの設定, 373

- タグパラメータ, 370

- タグプロパティ, 373, 374, 375, 380

- チャンネル SIMATIC S5 Profibus FDL の構成, 370

- チャンネルの特徴, 368

- チャンネルユニット FDL (CP5412/A2-1), 365

- チャンネルユニットのシステムパラメータ, 377

- チャンネル診断, 658

- データタイプ, 366

- データタイプがバイトの S5 タグへのアクセス, 366

- データタイプがバイト配列の S5 タグへのアクセス, 366

- データタイプがビットの S5 タグへのアクセス, 366
- データ処理ブロックの設定, 385
- データ領域, 366
- デバイス名, 377
- デバイス名の変更, 377
- バイト単位でアクセスするタグの構成, 374
- バイト配列としての未処理データタグ, 379
- パッシブなデータ転送の構成, 383
- パッシブ接続, 365
- ビット単位でアクセスするタグの構成, 373
- ブロック, 385
- ワード単位でアクセスするタグの構成, 375
- 書き込み/読み取りモニタ時間, 377
- 書き込み/読み取りモニタ時間の変更, 378
- 新規ドライバの追加, 370
- 接続の構成, 371
- 接続パラメータ, 370, 371, 381, 383
- 接続プロパティ, 371, 381, 383
- 通信タイプの設定, 381
- 特殊ファンクション, 379
- 未処理データタグの構成, 380
- 例, 385, 388
- S5 Serial 3964R, 399
 - タグのデータタイプ, 399
 - タグの設定, 402
 - バイト単位でアクセスするタグの構成, 406
 - ビット単位でアクセスするタグの構成, 404
 - 設定, 400
- S5 タグ, 366
 - S5 タグへのアクセス, 366
 - データタイプがバイトのタグへのアクセス, 366
 - データタイプがバイト配列の S5 タグへのアクセス, 366
 - データタイプがビットの S5 タグへのアクセス, 366
- S7 Plus, (以下の項目を参照してください。SIMATIC S7-1200、S7-1500 Channel)
- S7 Protocol Suite, 408
 - "BSEND/BRCV"ファンクション用の未処理データタグの構成, 501
 - AR_SEND 変数形のプロパティの概要, 467
 - AR_SEND ファンクションのデータブロック構造とパラメータ, 460
 - BSEND/BRCV ファンクション用の未処理データタグ, 498
 - FB S7ProtocolSuite とのデータ交換 : AR_SEND 変数形, 458
 - WinCC でのソフトウェアの二重化の削除, 511
 - アーカイブタグが 1 つの場合の AR_SEND 変数形, 470
 - アーカイブタグが 1 つの場合の AR_SEND 変数形を構成する方法, 489
 - アーカイブタグが複数の場合の AR_SEND 変数形, 475
 - アーカイブタグが複数の場合の AR_SEND 変数形 (最適化) , 487
 - アーカイブタグが複数の場合の AR_SEND 変数形を構成する方法, 491
 - サイクリック読み取りサービス, 453
 - サポートされるデータタイプ, 413
 - システムパラメータ, 451
 - システムパラメータのコンフィグレーション, 454
 - ソフトウェアの二重化, 502
 - ソフトウェアの二重化 - 接続固有内部タグ, 504
 - ソフトウェア冗長化の設定, 509
 - チャンネルユニット, 410, 425
 - チャンネル診断, 645
 - データタイプ, 413
 - データブロック構造体 AR_SEND の例, 471, 472, 473, 474, 479, 481, 483, 485, 488
 - テキストタグの構成, 449
 - バイト単位でアクセスするタグの構成, 446
 - バイト配列としての未処理データタグ, 494
 - バイト配列としての未処理データタグの構成, 496
 - ビット単位でアクセスするタグの構成, 445
 - ワード単位でアクセスするタグの構成, 448
 - 接続の構成, 438, 440
 - 接続の設定, 426, 429, 433, 435, 443
 - 接続障害のエラーコード, 512
 - 設定, 415
 - 未処理データタグ, 493
 - 論理デバイス名の変更方法, 456
- S7+, (以下の項目を参照してください。SIMATIC S7-1200、S7-1500 Channel)
- S7-1200 コントローラ, 513
- S7-1500 R/H-CPU, 534, 537, 540
- S7-1500 コントローラ, 513
- S7DOS, 167
- SCADA Export, 417, 526
- SelfDiagnosis, 120
- ServiceMode, 101
- Siemens SIMATIC SCADA Export, 417, 526
- SIMATIC S5 PROFIBIS FDL チャンネル, 658
 - 診断オプション, 645
- SIMATIC S5 Profibus FDL, 365
 - S5 Profibus FDL, 365
 - 特殊ファンクション, 379
 - 未処理データタグ, 379
- SIMATIC S5 Programmers Port AS511, 390
- SIMATIC S5 Serial 3964R, 399, 400

SIMATIC S7 Protocol Suite, 408
 SIMATIC S7 Protocol Suite チャンネル, 645
 診断オプション, 645
 SIMATIC S7-1200, S7-1500 Channel, 513
 システム診断の設定, 642
 チャンネル診断, 638
 データタイプ, 514
 接続の設定, 519
 設定, 519
 SIMATIC S7-1200、S7-1500 Channel
 インポート, 416, 525
 設定, 514, 517
 SIMATIC SCADA Export, 417, 526
 SIMATIC Shell, 79, 83
 SIMATIC TI Ethernet Layer 4, 546
 チャンネル, 546
 SIMATIC TI Ethernet 第 4 層
 トランスポートパラメータ, 555
 SIMOTION チャンネル, 568
 システムパラメータ, 576, 578
 データタイプ, 568
 故障診断, 580
 構成, 574, 575
 設定, 569, 570, 571, 572

T

TCP/IP チャンネルユニット, 441
 接続の設定, 443
 TI Ethernet 第 4 層, 546
 システムパラメータ, 546, 555, 556, 557
 タグのアドレス, 550
 タグの設定, 550
 データタイプ, 547
 デバイス名, 555
 トランスポートパラメータ, 555
 バイト単位でアクセスするタグの構成, 554
 バイト単位のアクセス, 554
 ビット単位でアクセスするタグの構成, 552
 ビット単位のアクセス, 552
 接続パラメータ, 546, 547
 設定, 547
 TIA Portal
 AS 構造, 421, 530
 インポート, 416, 525
 TI シリアル, 559
 タグアドレス, 562
 タグのアドレス, 562
 タグの設定, 562
 データタイプ, 559
 バイト単位でアクセスするタグの構成, 566
 ビット単位でアクセスするタグの構成, 564

 接続パラメータ, 560
 設定, 560
 TTY インターフェース, 390

U

URL
 OPC UA サーバー, 782

W

WinAC Basis, 440
 WinAC Pro, 438
 WinCC, 16
 OPC DA クライアントとして, 248, 269
 OPC UA クライアントとして, 283, 334
 OPC XML クライアントとして, 248
 ServiceMode, 101
 WinCC の OPC, 248, 269, 334, 700
 WinCC の中の OPC UA, 283
 クライアント/サーバーシステム, 16
 サービスモード, 101
 WinCC - OPC A&E クライアントの接続, 762
 OPC A&E の例, 762
 WinCC Explorer-OPC_Client.MPC, 711
 WinCC OPC A&E サーバー
 階層アクセス, 767
 WinCC OPC A&E サーバーの機能
 単純イベント, 752
 WinCC OPC UA サーバー, 782
 コンフィグレーション, 811
 発見サーバー, 783
 WinCC ServiceMode, 101, 109
 WinCC クラウド, (以下の項目を参照してください: クラウドコネクタ)
 WinCC サービス, 101
 WinCC タイプ変換, 173
 WinCC タグ@RM_MASTER, 146
 WinCC タグ@RM_MASTER_NAME, 146
 WinCC チャンネル診断, 634, 654, 665, 676
 WinCC データタイプ, 173, 178
 WinCC データタイプ, 178
 タイプ変換, 178, 187
 WinCC と Microsoft Excel 間の接続, 725
 OPC DA の例, 725
 WinCC と OPC HDA クライアント間の接続, 742
 OPC HDA の例, 742
 WinCC と SIMATIC NET FMS OPC サーバー間の接続, 717
 OPC DA の例, 717

WinCC と SIMATIC NET S7 OPC サーバー間の接続, 720

OPC DA の例, 720

WinCC の AR_SEND ファンクション

データブロック構造とパラメータ, 460

WinCC の AR_SEND-ファンクション

アーカイブタグが複数の場合の変数形を構成する方法, 491

アーカイブタグの変数形, 470

アーカイブタグの変数形の構成, 489

データブロック構造体の例, 471, 472, 473, 474, 479, 481, 483, 485, 488

複数のアーカイブタグの変数形, 475

複数のアーカイブタグの変数形 (最適化), 487

変数形のプロパティの概要, 467

WinCC プロジェクト, 79

WinCC プロジェクトダイアログ, 73

リモートの実行, 93

リモート終了, 95

呼び出し, 73

WinCC プロセス通信, 167

WinCC メッセージシステム

OPC A&E サーバー上の WinCC メッセージクラス, 755

OPC A&OPC-A&WinCC メッセージシステム上の, 754

WinCC のメッセージクラスとメッセージタイプのマッピング, 755, 756

WinCC メッセージシステムへのアクセスの構成, 762

属性, 756, 771

WinCC 間の接続, 713

OPC DA の例, 713

WinCC 通信, 168

原理, 168

Windows

タスクバー, 106

トレイエリア, 106

ア

アーカイブ ` メモリ, 147, 149

アドレスタイプ

三菱製 Ethernet, 234

アドレスプロパティ, 263, 281, 373, 374, 375, 380

S5 Profibus FDL の例, 388

イ

インスタンス証明書, 293

インストール, 99

ファイルサーバー, 99

インポート, 49

パッケージ, 49

ウ

ウェブクライアント, 19

エ

エクスポート, 36

パッケージ, 36

エクスポートツール, (以下の項目を参照してください。Siemens SIMATIC SCADA Export)

エラーのポイント, 599

オ

オフライン設定, 416, 525

オペレータの特権, 33

オペレーティングシステム, 33

クライアント, 33

オペレーティングシステム, 33

オペレータオーソリゼーションの設定, 33

冗長システム, 123

オンラインでの変更のロード

冗長システムのユーザーアーカイブの変更, 137

カ

カウンタ, 586

ク

クライアント, 19, 44

インポートパッケージ, 49

ウェブクライアント, 19

オペレータ権限, 33

コンピュータのプロパティの設定, 40

コンフィグレーション, 44

サーバーでプロジェクトを開く, 89

サーバーのコンピュータリストへの入力, 32

サーバープロジェクトにおける設定, 40

サーバープロジェクトの終了, 95

サーバープロジェクトへのアクセス, 87

サーバーへの表示付き, 40

サーバー画像の編集, 91

サーバー上での同時起動, 22

シャットダウン, 77

- シンククライアント, 19
- ピクチャの開始のコンフィグレーション, 58
- ピクチャの変更のコンフィグレーション, 62
- ブート, 74
- プロジェクト新規作成, 46
- マルチユーザーシステムにおけるスタートプロパティの定義, 40
- メッセージシーケンスレポートのコンフィグレーション, 68
- リモートサーバープロジェクトの実行, 93
- 複数サーバーのピクチャの表示, 60
- 複数サーバーのメッセージの表示, 67
- 複数サーバーの表示, 64
- 複数ステーションシステム, 40
- 分散システムにおけるコンフィグレーション, 44
- クライアント/サーバーシステム, 16
 - OPC インターフェースのアプリケーション, 97
 - クライアントおよびサーバーの可能数, 16
 - クライアントタイプ, 19
 - クライアントのシャットダウン, 77
 - クライアントのスタートアップ, 74
 - サーバーのシャットダウン, 77
 - サーバーのスタートアップ, 73
 - サーバー間通信, 24
 - ファイルサーバー, 24
 - マルチユーザーシステム, 24
 - ランタイムでの動作, 70
 - 使用方法, 16
 - 冗長性, 16, 24
 - 数量構造, 22
 - 設定, 19
 - 必要条件, 16
 - 標準サーバー, 24
 - 複合設定, 19
 - 複数ステーションシステム, 16
 - 分散システム, 16, 24
- クライアント/サーバーシナリオ, 24
- クライアントプロジェクト, 46
- クライアント証明書, 293
- クラウドコネクタ, 841

コ

- コンピュータ
 - ステータス, 106
- コンピュータリスト, 32
 - クライアントの受入れ, 32
- コンフィグレーション, 28, 44
 - WinCC OPC UA サーバー, 811
 - クライアント, 44
 - サーバー, 28
 - セキュリティ設定, 812

- ポート番号, 812
- マルチユーザーシステム, 28
- ユーザー ID, 813
- 最適 WinCC アーカイブ書き込みアクセス, 813
- 分散システム, 28

サ

- サーバー, 24, 28
 - SOFTNET ドライバの特殊機能, 75
 - オペレータ権限, 33
 - クライアントからのプロジェクトの実行, 93, 95
 - クライアント側からの画像編集, 91
 - コンフィグレーション, 28
 - サーバープロジェクトにおけるクライアントの設定, 40
 - サーバー間通信, 24
 - シャットダウン, 77
 - ファイルサーバー, 24
 - ブート, 73, 75
 - プロジェクト新規作成, 30
 - リモート設定, 33
 - リモート設定(前提条件), 32
 - 標準サーバー, 24
 - 標準サーバーの設定, 52
 - 複数クライアントへのプロジェクトアクセス, 87
 - 複数のネットワークカードを使った特殊機能, 75
 - 優先サーバーの構成, 55
- サーバーデータ, 36
 - パッケージインポート, 49
 - パッケージエクスポート, 36
 - 標準サーバーの設定, 52
 - 優先サーバーの構成, 55
- サーバーでの SOFTNET ドライバ常時使用, 75
- サーバー間通信, 24
- サーバー証明書, 293
- サーバー接頭語, 58, 60, 62, 64
- サービスアクセスポイント, 365
- サービスプロジェクト, 101, 109
 - Autostart, 113
 - インタラクティブな操作の起動, 117
 - インタラクティブな操作の終了, 118
 - スクリプト, 105
 - スタートアップリスト, 105
 - ログオフ, 116
 - ログオン, 116
 - 診断情報, 105
 - 制約, 104
 - 接続ステーション, 105
 - 設定, 103
 - 定義, 112
 - 非リリース構成要素, 105

- 必要条件, 110
- 標準プロジェクトへの変更, 113
- 用途, 104
- サービスモード, 101
- サイクリック読み取りサービス, 453
- サブネット, 83, (サブネットを参照)
- サポートされるシステム情報, 586
 - 概要, 586

シ

- システムエラー, 70
- システムタグ, 155
- システムパラメータ, 168, 353, 451, 576, 578, 663
 - S5 Ethernet 第4層, 340
 - S5 Profibus FDL, 365
 - SIMATIC S7 Protocol Suite, 408
 - TI Ethernet 第4層, 546, 555, 556, 557
 - コンフィグレーション, 454, 456
 - チャンネルユニット, 555, 556
- システムメッセージ, 160
- システム情報, 584
 - I/O フィールドコンフィグレーション, 595
 - エラーのポイント, 599
 - サポートされるシステム情報の概要, 586
 - ステータス表示コンフィグレーション, 602
 - チャンネル診断, 632
 - ドライブの空き容量, 586
 - バーコンフィグレーション, 596
 - メッセージテキスト, 599
 - モニタするタグ, 599
 - 構成, 592
 - 使用されるデータタイプ, 593, 607
 - 他のソフトウェアの構成要素との違い, 591
 - 例, 593, 601, 607, 608, 611, 612
 - 例:, 595, 596, 597, 599, 602, 610
- システム情報システム情報
 - マルチユーザーシステムにおける使用, 606
- システム診断, 642
 - デバイス表示, 638
 - 詳細表示, 639
 - 診断バッファ表示, 640
- システム診断表示, 641
- システム動作, 70
- シナリオ, 24, 146
 - クライアントおよびサーバー, 24
 - 中央アーカイブサーバー, 25
- シャットダウン, 77
 - クライアントの, 77
 - サーバーの, 77
- シリアルケーブル, 123
- シンククライアント, 19

ス

- スクリプト
 - サービスプロジェクト, 105
- スタートアップブロック, 385
- スタートアップリスト
 - サービスプロジェクト, 105
- スタートアップ動作, 73, 74
 - クライアントの, 74
 - サーバーの, 73
- ステータス表示, 106, 171, 634, 654, 665, 676
 - システム情報の例,

ソ

- ソフト PLC チャンネルユニット, 440
 - 接続の構成, 440
- ソフトウェアの冗長化, 540
 - WinCC での設定, 509
- ソフトウェアの二重化, 502
 - WinCC での削除, 511
 - 接続固有内部タグ, 504

タ

- タイプ変換, 173, 187, 368
- タイマ, 586
- タイムスタンプ, 460
- タグ, 173, 345, 593, 678
 - BinWrite 付きタグの設定, 202
 - Check, 636, 678
 - HDA サーバブラウザの設定, 744
 - OPC HDA の例, 744
 - OPC 項目マネージャ, 259
 - OPC 項目マネージャの設定, 259, 263, 281
 - インポート, 416, 525
 - エクスポート, 423, 532
 - オートメーションシステムの外部タグのアドレス指定, 173
 - システム情報の例, 593
 - タグアドレスの構成, 403
 - タグの設定のテスト, 652, 663
 - タグの選択, 250
 - タグの追加, 250
 - テキストタグの構成, 177
 - テキストタグの長さ情報, 173
 - バイト単位のアクセスの構成, 348, 374, 406, 446, 554, 566
 - パワータグ, 173
 - パワータグの構成, 177

- ビット単位のアクセスの構成, 347, 373, 404, 445, 552, 564
 - ワード単位でアクセスするための構成, 350, 375, 448
 - 外部, 173
 - 外部タグの構成, 177
 - 確認, 656, 667
 - 設定, 345, 373, 550, 562
 - 線形スケーリングの構成, 177
 - 内部タグのチェック, 694
 - 品質コード, 683
 - タグ, 680
 - HDA サーバーブラウザ, 743
 - OPC DA の例, 721, 724, 729
 - OPC HDA の例, 746
 - システム情報の例, ステータス, 680, 690
 - 品質, 680
 - 品質コード, 680
 - タグアドレス, 345
 - S5 Ethernet 第 4 層, 340
 - TI シリアル, 559
 - タグステータス, 680, 690, 692, 693
 - グローバルアクションによるタグステータスのモニタ, 693
 - ダイナミックダイアログでのタグステータスのモニタ, 692
 - タグのアドレス, 550
 - TI Ethernet 第 4 層, 550
 - タグの設定
 - TI Ethernet 第 4 層, 550
 - タグの追加, 250, 259
 - OPC DA の例, 715
 - タグの同期化, 128
 - タグの品質コード, 680, 683
 - タグプロパティ, 177, 373, 374, 375, 380
 - S5 Profibus FDL の例, 388
 - タグ名
 - 接頭語, 176
 - 接尾語, 176
 - タスクバー, 106
- チ**
- チャンネル, 168, 340, 399, 408, 676
 - S5 PROFIBUS FDL の例, 385
 - SIMATIC S5 Ethernet Layer 4, 340
 - SIMATIC S5 PROFIBUS FDL の構成, 370
 - SIMATIC S5 Programmers Port AS511, 390
 - SIMATIC S5 Serial 3964R, 399
 - SIMATIC S7 Protocol Suite, 408
 - SIMATIC S7-1200, S7-1500 Channel, 513
 - SIMATIC TI Ethernet Layer 4, 546
 - TI シリアル, 559
 - テスト, 676
 - 確認, 634, 654, 665
 - 三菱製 Ethernet, 226
 - 接続の確立/終了, 171
 - 論理接続ステータス, 171, 622
 - チャンネル S5Serial3964R のライフビートのモニタ
 - ライフビートのモニタ, 400
 - チャンネルユニット, 168, 340, 370, 410
 - 505 Ethernet (CP 1413-x), 546
 - 505 シリアルユニット#1, 559
 - FDL (CP5412/A2-1), 370
 - Industrial Ethernet, 425
 - MPI, 428
 - PROFIBUS, 434
 - S5 Transport (CP1413-x), 342
 - S5 Transport (TCP/IP), 342
 - S5-AS511, 390, 391
 - S5-RK512 (3964R), 399
 - SIMATIC S7-1200, S7-1500 Channel, 513
 - SIMATIC S7 Protocol Suite S7 Protocol Suite チャンネルの選択: チャンネルユニットの選択, 410
 - TCP/IP, 441
 - システムパラメータ, 353, 377, 451, 555, 556
 - スロット PLC, 438
 - ソフト PLC, 440
 - 三菱製 Ethernet, 226
 - 接続パラメータ, 547
 - 名前指定接続, 431
 - チャンネルユニット「Industrial Ethernet」, 425
 - 接続の設定, 426
 - チャンネルユニット FDL (CP5412/A2-1), 365, 370
 - チャンネル診断, 624
 - "システム情報"チャンネル, 632
 - OPC チャンネル, 669
 - PROFIBUS システムパラメータ, 652
 - SIMATIC S5 PROFIBUS FDL チャンネル, 658
 - SIMATIC S7 Protocol Suite チャンネル, 645
 - SIMATIC S7-1200, S7-1500 Channel, 638
 - WinCC チャンネル診断, 634, 654, 665, 676
 - Windows アプリケーションとしてのチャンネル診断, 627
 - グローバルアクションによるタグステータスのモニタ, 693
 - システムパラメータ, 663
 - ステータスメッセージの概要, 634, 654, 665, 676
 - ダイナミックダイアログでのタグステータスのモニタ, 692
 - タグのチェック, 636, 678
 - タグの確認, 656, 667

チャンネルのトレースファンクションの設定, 629
 チャンネルのテスト, 634, 654, 665, 676
 チャンネル診断, 624, 625, 627
 トレースファンクションの標準フラグ, 629
 によるチャンネル,
 によるチャンネル診断, 625
 プロパティ - CP5613, 649, 651, 660, 662
 ログファイル, 632, 646, 659, 670
 ログブックエントリの ERROR フラグ,
 ログブックエントリの INFO フラグ,
 次によるチャンネル, 628
 次によるチャンネル診断, 624
 診断オプション, 632, 645, 658, 669
 接続ステータス, 622
 接続の確認, 634, 654, 665, 676
 接続パラメータ, 663
 設定データのチェック, 674
 通信プロセッサの設定確認, 649, 651, 660, 662
 内部タグのチェック, 694
 例, 693
 論理接続ステータス, 622
 チャンネル診断チャンネル
 テスト, 626, 628
 チャンネル診断チャンネル診断
 チャンネル診断によるチャンネル診断, 625
 チャンネル診断トレースファイル, 624

テ

ディスク容量, 586
 システム情報の例,
 ドライブの空き容量, 586, 591
 データ, 60
 複数サーバーの表示, 60
 複数のサーバーからの表示, 64
 データタイプ, 173, 254, 284, 341, 413, 514, 586
 OPC UA, 284
 S5 AS511, 390
 S5 Ethernet 第 4 層, 341
 S5 Profibus FDL, 366
 S5 Serial 3964R, 399
 S7 Protocol Suite, 413
 SIMATIC S7-1200, S7-1500 Channel, 514
 TI Ethernet 第 4 層, 547
 サポートされる WinCC データタイプ, 254
 サポートされる WinCC データタイプの概要, 254
 タイプ変換, 187
 三菱製 Ethernet, 226, 234
 使用されるデータタイプ, 586, 593
 データ転送, 381
 アクティブなデータ転送の構成, 381
 パッシブなデータ転送の構成, 383

データ領域, 366
 テキストタグ, 173, 445
 構成, 177, 449
 デバイス名, 353, 377, 555
 TI Ethernet 第 4 層, 555
 変更, 377

ト

トランスポートパラメータ, 168, 353
 SIMATIC S5 Ethernet 第 4 層, 353
 SIMATIC TI Ethernet 第 4 層, 555
 TI Ethernet 第 4 層, 555
 トレイエリア, 106
 トレースファンクション, 629
 設定, 629

ネ

ネットワーク, 83, 165
 サブネット, 83
 ルータ, 83

ハ

バー

 システム情報の例,
 バイト単位でアクセスするタグの構成
 TI Ethernet 第 4 層, 554
 バイト単位のアクセス, 201, 348, 445
 S5 AS511, 397
 S5 Ethernet 第 4 層, 348
 S5 Profibus FDL, 374
 S5 Serial 3964R, 406
 S7 Protocol Suite, 446
 TI Ethernet 第 4 層, 554
 TI シリアル, 566
 バイト単位のアクセスの構成
 タグ, 554
 バイト配列, 379
 パッケージ, 36, 49
 Configuring export エクスポートのコンフィグレーション, 36
 WinCC エクスプローラの表示, 36, 49
 インポートの設定, 49
 ファイルシステムにおける保存, 36
 自動更新, 36
 パフォーマンス, 22

ヒ

ビットアクセス/バイトアクセス, 201
 ビットタグ/バイトタグ, 202
 ビット単位でアクセスするタグの構成
 TI Ethernet 第4層, 552
 ビット単位のアクセス, 201, 347, 445
 S5 AS511, 396
 S5 Ethernet 第4層, 347
 S5 Profibus FDL, 373
 S5 Serial 3964R, 404
 S7 Protocol Suite, 445
 TI Ethernet 第4層, 552
 TI シリアル, 564
 ビット単位のアクセスの構成
 タグ, 552

フ

ファイルサーバー, 24, 99
 インストール, 99
 設定, 99
 フィルタ基準, 259
 ブート, 73, 74
 クライアントの, 74
 サーバーの, 73, 75
 プリンタモニタ, 586
 システム情報の例,
 プログラマブルコントローラ, 368
 プロジェクト, 30, 46
 OPC DA の例, 715
 WinCC のステータス, 106
 クライアントプロジェクトの新規作成, 46
 サーバー側でのプロジェクトの新規作成, 30
 リモートアクセス, 87
 リモートオープン, 89
 実行, 93
 終了, 95
 複製, 138, 140
 プロジェクトデュプリケータ, 138, 140
 プロセス画像
 システム情報の例,
 ブロック, 385
 サイクリックブロック, 385
 プロパティ - CP5613, 649, 651, 660, 662

マ

マルチユーザーシステム, 24, 606
 クライアントの設定, 40
 チャンネルの使用, 606
 マルチユーザープロジェクト, 30
 作成, 30
 マルチユーザシステム
 コンフィグレーション, 28

メ

メッセージ, 67
 インポート, 416, 525
 クライアント側でのメッセージシーケンスレポー
 トのコンフィグレーション, 68
 システム情報の例, 599, 601
 複数のサーバーからの表示, 67
 メッセージクラスのロック, 123
 パッシブ, 123
 メッセージシーケンスレポート, 68
 メッセージテキスト, 599
 メッセージのロック, 123
 アクティブ, 123
 パッシブ, 123
 メッセージのロック(アクティブ), 123
 メッセージのロック(パッシブ), 123
 メディアの冗長性, 535

モ

モニタするタグ, 599

ユ

ユーザーアカウント, 702
 OPC コンピュータへの通知, 702

ラ

ランタイム, 70
 アーカイブ, 70
 グラフィック, 70
 システムエラー時の動作, 70
 システム動作, 70
 スクリプト, 70
 テキストライブラリ, 70
 メッセージ, 70

ユーザー管理者, 70
レポート, 70

リ

リダンダントサーバーの起動, 130
リダンダントサーバー間のシリアル接続, 133
リモートコンフィグレーション
 サーバープロジェクトへのアクセス, 87
 プロジェクトの実行, 93
 プロジェクトの終了, 95
リモート設定, 79
 プロジェクトを開く, 89
 画像の編集, 91

ル

ルータ, 83

ロ

ログファイル, 632, 646, 659, 670
ログブックエントリ, 632, 646, 659, 670
ログブックファイル, 624

ワ

ワード単位のアクセス, 445
 S7 Protocol Suite, 448
ワイルドカード, 837

安

安全な通信, 83

暗

暗号化通信, 19

夏

夏時間/標準時間, 460

画

画像, 58
 クライアント側でのピクチャ変更のコンフィグレーション, 62

クライアント側での複数サーバーの表示形式, 60
サーバープロジェクトの編集, 91
サーバー接頭語, 58
ピクチャの開始のコンフィグレーション, 58

開

開く
 プロジェクト, 89

確

確認セオリー, 759
確認ポリシー, 773

境

境界値, 732

故

故障診断
 チャンネル, 580

互

互換性, 699

構

構造
 WinCC OPC DA クライアント上での構造体の使用, 265
 WinCC OPC クライアントの使用, 268
 構造体の設定, 266
構造タイプ
 AS 構造, 421, 530
構造タグ
 AS 構造, 421, 530
構造体タグ, 266
 構造体タグの設定, 266
構造体プロパティ, 266

項

項目 ID, 732
項目ハンドル, 732

三

三菱製 Ethernet, 226
 アドレスタイプ, 234
 サポートされているコントローラ, 227
 タグ, 237
 タグの設定, 227, 234, 237
 チャンネルユニット, 227, 228, 230, 231, 233
 データタイプ, 226, 234
 接続の設定, 227, 228, 230, 231, 233

試

試運転, 814

時

時間, 586, 591
 システム情報の例,
時間同期化, 123
時系列順メッセージ, 123

自

自動更新, 36, 49

実

実行, 93
 プロジェクト, 93

初

初期画面, 58
 クライアントのコンフィグレーション, 58

書

書き込み/読み取りモニタ時間, 377, 378
 変更, 378

証

証明書, 293

冗

冗長システム
 プロジェクトの複製, 138, 140
 設定, 132
冗長システムの必要条件, 123
冗長化の設定, 130
冗長性, 24, 120
 Application Health Check, 120, 153
 CPU 冗長性, 534
 アーカイブの切り替え, 120
 オペレーティングシステム, 123
 サーバーの起動, 120
 サーバー間のシリアル接続, 133, 146
 システムメッセージ, 147
 ソフトウェアの冗長化, 540
 時間同期化, 123
 時系列順メッセージ, 123
 冗長化 PLC, 534, 537
 設定, 130
 必要条件, 123
 優先サーバー, 24
 優先サーバーの構成, 55

新

新規タグの定義, 721
 OPC DA の例, 721
新規ドライバの追加, 370
新規ユーザー, 702

診

診断
 接続ステータス, 171, 622
診断オプション, 632, 645, 658, 669
 "システム情報"チャンネル, 632
 OPC チャンネル, 669
 SIMATIC S5 PROFIBIS FDL チャンネル, 658
 SIMATIC S7 Protocol Suite チャンネル, 645
診断情報
 サービスプロジェクト, 105
診断能力
 SIMATIC S7-1200, S7-1500 Channel, 638

数

数値タグタイプ, 173
 リニアスケーリング, 173

数量構造, 22

切

切り捨て, 837

接

接続, 168, 425, 676

アクティブ接続, 365

テスト, 676

パッシブ接続, 365

ランタイムでのステータス, 171, 622

確認, 634, 654, 665

確立/終了, 171

構成, 371, 438, 440

新規接続の作成, 176

接続の作成, 250

接続の設定のテスト, 652, 663

設定, 426, 429, 433, 435, 443

接続ステーション

サービスプロジェクト, 105

接続ステータス, 106

テスト, 622

接続のモニタ, 255

接続パラメータ, 168, 342, 370, 371, 381, 383, 425, 663

S5 AS511, 390, 391

S5 Ethernet 第4層, 340

S7 Protocol Suite, 408

SIMATIC S5 Serial 3964R, 399

TI Ethernet 第4層, 546, 547

TI シリアル, 559

チャンネルユニット, 547

接続プロパティ, 176, 371, 381, 383

接頭語, 176

接尾語, 176

設

設定, 19, 99, 130, 132

TI Ethernet 第4層, 547

クライアント/サーバーシステム, 19

タグ, 550

ファイルサーバー, 99

リモート, 79

設定データ, 674

設定ファイル

WinCC OPC UA サーバーの URL, 806

WinCC ログイン書き込みアクセス, 806

セキュリティ設定, 806

ポート番号, 806

ユーザー ID, 806

レイアウト, 806

証明書, 806

保存先, 806

値

値の範囲, 187

中

中央アーカイブサーバー, 25

通

通信, 165

AS データタイプ, 177, 187

AS データタイプによるフォーマット調整のソート, 187

BinWrite メカニズム, 201

BinWrite メカニズムの原則, 201

BinWrite 付きタグの設定, 202

IPv4 プロトコル, 167

WinCC タイプ変換, 173

WinCC データタイプ, 173

WinCC データタイプによるフォーマット調整のソート, 178

WinCC とオートメーションシステム間の通信, 168

WinCC プロセス通信, 167

WinCC 通信の原則, 168

オートメーションシステムのアドレス指定, 173

タイプ変換, 173, 187

タグプロパティ, 177

チャンネルユニット, 168

データタイプ, 173, 187

テキストタグの長さ情報, 173

パワータグ, 173

パワータグの構成, 177

ビットアクセス/バイトアクセス, 201

ビットタグ/バイトタグ, 202

基本, 165

新規接続の作成, 176

数値タグタイプのリニアスケールリング, 173

接続, 168

接続プロパティ, 176

値の範囲, 187

通信ドライバ, 168

通信タイプ, 381

設定, 381

通信ドライバ, 168, (チャンネルを参照)
 接続の確立/終了, 171
通信プロセッサ, 649, 651, 660, 662
 通信プロセッサの設定確認, 649, 651, 660, 662

特

特殊ファンクション, 379

日

日付, 586

標

標準サーバー, 24
 設定, 52
標準フラグ, 629
標準プロジェクト, 101, 109
 サービスプロジェクトへの変更, 113

表

表示, 60
 複数サーバーから, 60
 複数のサーバーからのデータ, 64

品

品質コード, 738, 761, 775

複

複合設定, 22
複数ステーションシステム, 16

分

分散システム, 16, 24
 クライアントコンフィグレーション, 44
 クライアントプロジェクトの作成, 46
 クライアント側データ出力のコンフィグレーション, 64
 クライアント側でのサーバーピクチャの表示, 60
 クライアント側でのピクチャ変更のコンフィグレーション, 62
 クライアント側でのメッセージシーケンスレポートのコンフィグレーション, 68
 クライアント側でのメッセージの表示, 67

コンフィグレーション, 28
パッケージインポート, 49
リモート設定, 79

変

変更の転送, 453
変数, 445

未

未処理データ, 730
未処理データタグ, 379, 493
 BSEND/BRCV ファンクション用, 498
 BSEND/BRCV ファンクション用の構成, 501
 バイト配列として, 379, 494
 バイト配列としての構成, 496
 構成, 380

無

無停電電源装置(UPS), 123

名

名前指定接続チャンネルユニット, 431
 接続の設定, 433

優

優先サーバー, 24
 構成, 55

例

例:, 693

論

論理デバイス名, 451
 modify, 456

