

SIEMENS

SIMATIC HMI

WinCC V8.1 Update 2 WinCC:設定および通信(パート 1)


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
WinCC の証明書	5
プロセス通信	6
通信 - 診断	7
OPC - オープンな接続性 (Open Connectivity)	8
WinCC REST 通信	9
WinCC/Cloud Connector	10


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 Aktiengesellschaft 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	13
1.1	WinCC のマルチユーザシステム	13
1.2	WinCC のクライアント/サーバーシステム	14
1.3	標準的な設定	18
1.4	数量構造および特性	20
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	37
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	41
1.7	クライアントコンフィグレーション	45
1.7.1	クライアントコンフィグレーション	45
1.7.2	クライアント側でのプロジェクトの新規作成	47
1.7.3	インポートパッケージのコンフィグレーション	50
1.7.4	標準サーバーの設定方法	53
1.7.5	優先サーバーのコンフィグレーション方法	57
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	60
1.7.7	異なるサーバーからのピクチャの表示	62
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	64
1.7.9	異なるサーバーからのデータの使用	66
1.7.10	異なるサーバーからのメッセージの表示	69
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	70
1.8	ランタイムでのシステム動作	73
1.8.1	ランタイムでのシステム動作	73
1.8.2	サーバーのスタートアップ	77
1.8.3	クライアントのスタートアップ	78
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	79
1.8.5	サーバーのシャットダウン	81
1.8.6	クライアントのシャットダウン	81
1.9	リモートコンフィグレーション	83
1.9.1	リモートコンフィグレーション	83
1.9.2	暗号化通信	85

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.....	88
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.....	92
1.9.5	編集用にプロジェクトを開く方法.....	95
1.9.6	サーバープロジェクトピクチャの編集方法.....	97
1.9.7	プロジェクトの実行方法.....	98
1.9.8	プロジェクトの終了方法.....	100
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.....	102
2	ファイルサーバー.....	105
2.1	ファイルサーバーのセットアップ.....	105
3	WinCC ServiceMode.....	107
3.1	WinCC ServiceMode:標準プロジェクトとサービスプロジェクト.....	107
3.2	サービスプロジェクトの設定.....	109
3.3	サービスプロジェクトの使用と制約.....	110
3.4	WinCC ステータスおよびシステムトレーでのコントロール.....	113
3.5	ファンクションと前提条件.....	116
3.5.1	サービスプロジェクトの操作モード.....	116
3.5.2	サービスプロジェクトの実行の必要条件.....	117
3.6	WinCC ServiceMode の設定.....	120
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.....	120
3.6.2	サービスプロジェクト用の Autostart の設定方法.....	121
3.7	ランタイムのサービスプロジェクト.....	124
3.7.1	サービスプロジェクトを起動する方法.....	124
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.....	126
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.....	126
4	リダンダントシステム.....	129
4.1	2重化.....	129
4.2	WinCC の2重化.....	130
4.3	冗長システムの必要条件.....	133
4.4	2重化の仕組み.....	135
4.5	冗長性システムの構成.....	140
4.5.1	冗長システムのセットアップ方法.....	140
4.5.2	同一ファンクションのコンフィグレーション.....	142
4.5.3	冗長サーバーを構成する方法.....	143
4.5.4	ユーザーアーカイブの同期の設定方法.....	147
4.5.5	冗長サーバーのプロジェクトの複製方法.....	149
4.5.6	ランタイムでの冗長プロジェクトの複製方法.....	152
4.6	WinCC 冗長性のシナリオ.....	154

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.....	154
4.6.2	故障シナリオ	158
4.6.2.1	故障シナリオ	158
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	159
4.6.2.3	シナリオ 2 パートナサーバーへの接続エラー	162
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	164
4.6.2.5	シナリオ 4 プロセス接続の不良	165
4.6.2.6	シナリオ 5 ソフトウェアエラー	166
4.6.3	WinCC リダンダントシステムタグ	167
4.6.4	WinCC リダンダントシステムメッセージ	173
5	WinCC の証明書	179
5.1	WinCC の証明書の概要	179
5.2	基礎	182
5.2.1	通信パートナー	182
5.2.2	信頼関係の作成:.....	183
5.3	WinCC Certificate Manager.....	186
5.3.1	基礎	188
5.3.1.1	認証機関	188
5.3.1.2	必要な証明書	189
5.3.1.3	パスワード要件.....	190
5.3.2	Certificate Manager のインターフェース	191
5.3.2.1	ユーザーインターフェースの構造	191
5.3.2.2	[CA 設定]タブ	193
5.3.2.3	[インストール済み証明書]タブ	194
5.3.2.4	サーフェスのカスタマイズ	194
5.3.2.5	ユーザーインターフェース言語の変更	197
5.3.3	証明書を使用可能にする	198
5.3.4	WinCC 認証機関とルート証明書の作成.....	200
5.3.5	デバイスの追加.....	202
5.3.6	デバイスの削除.....	204
5.3.7	アプリケーション証明書の追加または削除	205
5.3.8	PC のためのエクスポート、インポート、インストール.....	208
5.3.8.1	証明書設定のエクスポート	208
5.3.8.2	証明書設定のインポート	210
5.3.8.3	証明書設定または個別の証明書のインストール	212
5.3.8.4	アプリケーション証明書のアンインストール.....	214
5.3.9	ルート証明書と CRL ファイルのエクスポート	214
5.3.10	アプリケーション証明書を公的証明書としてエクスポート	216
5.3.11	証明書の再作成.....	217
5.3.11.1	アプリケーション証明書の再作成	217
5.3.11.2	設定全体の再作成	218
5.3.11.3	CRL ファイルの更新	219
5.3.12	認証機関のバックアップ	220

5.3.13	IP アドレスのこの後の変更	221
5.4	S7 PLC との通信用証明書	223
5.4.1	[セキュア通信]の設定	223
5.4.2	S7 証明書の基礎.....	228
5.4.3	S7 証明書の新規作成.....	231
5.4.4	S7 通信における信頼関係の確立	232
5.4.5	証明書の有効期限をチェック	236
5.5	OPC UA による通信のための証明書	238
5.5.1	WinCC OPC UA 証明書の基礎	238
5.5.2	OPC UA 通信における信頼関係の確立	240
5.6	WebUX WebNavigator の通信における証明書	242
5.6.1	WebUX/WebNavigator クライアントでルート証明書をインストール	242
6	プロセス通信	245
6.1	通信の基本.....	245
6.2	接続設定の基本規則.....	246
6.3	WinCC プロセス通信	247
6.3.1	WinCC プロセス通信	247
6.3.2	WinCC 通信の原則.....	248
6.3.3	ランタイム時の接続ステータスタグの設定	252
6.3.4	外部タグ	255
6.3.4.1	外部タグ	255
6.3.4.2	新規接続の作成方法.....	258
6.3.4.3	外部タグを以下のように設定します	259
6.3.4.4	WinCC データタイプでソートしたフォーマット調整.....	260
6.3.4.5	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.....	269
6.3.4.6	BinWrite メカニズムの原則.....	283
6.3.4.7	BinWrite 付きタグの設定方法	285
6.3.5	Ethernet 経由のカップリングのポートアドレス	287
6.3.6	SIMATIC S7-PLCSIM Advanced	287
7	通 信 一 診 断	289
7.1	チャンネルおよびタグの診断.....	289
7.2	エラー検出に関する一般情報	290
7.3	チャンネル診断.....	291
7.3.1	チャンネル診断.....	291
7.3.2	パフォーマンスタグによる接続の確認.....	291
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	295
7.3.4	[チャンネル診断]によるチャンネルの診断	297

7.3.4.1	チャンネル診断の原理	297
7.3.4.2	ActiveX コントロールによるチャンネル診断	298
7.3.4.3	「チャンネル診断」による「SIMATIC S7-1200/S7-1500」診断チャンネルの評価	299
7.3.4.4	「チャンネル診断」による「OPC」診断チャンネルの評価	302
7.3.4.5	「チャンネル診断」による「OPC UA」診断チャンネルの評価	304
7.3.4.6	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	306
7.3.4.7	[チャンネル診断]を使用したチャンネルの診断	309
7.3.4.8	チャンネル診断によるチャンネルのチェック方法	310
7.3.4.9	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	311
7.3.4.10	ランタイムの開始方法	313
7.4	"システム情報"チャンネルの診断	314
7.4.1	"システム情報"チャンネル - 診断オプション	314
7.4.2	ログファイルのエントリの説明	314
7.4.3	無効なタグ値の原因の特定方法	315
7.4.3.1	無効なタグの原因の特定方法	315
7.4.3.2	チャンネルおよび接続のチェック方法	316
7.4.3.3	タグのチェック方法	318
7.5	診断チャンネル「SIMATIC S7-1200/S7-1500」	319
7.5.1	SysDiagControl によるシステム診断	319
7.5.2	システム診断の構成方法	323
7.5.3	SysDiagControl の設定の情報	325
7.6	"SIMATIC S7 Protocol Suite"チャンネルの診断	329
7.6.1	"SIMATIC S7 Protocol Suite"チャンネル - 診断オプション	329
7.6.2	ログファイルのエントリの説明	330
7.6.3	無効なタグ値の原因の特定方法	333
7.6.3.1	無効なタグの原因の特定方法	333
7.6.3.2	通信プロセッサの設定確認方法	333
7.6.3.3	SIMATIC NET での通信プロセッサのチェック	335
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	336
7.6.3.5	チャンネルおよび接続のチェック方法	338
7.6.3.6	タグのチェック方法	340
7.7	"SIMATIC S5 Profibus FDL"チャンネルの診断	342
7.7.1	"SIMATIC S5 PROFIBUS FDL"チャンネルの診断オプション	342
7.7.2	ログファイルのエントリの説明	343
7.7.3	無効なタグ値の原因の特定方法	344
7.7.3.1	無効なタグの原因の特定方法	344
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	344
7.7.3.3	SIMATIC NET での通信プロセッサのチェック	346
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	347
7.7.3.5	チャンネルおよび接続のチェック方法	349
7.7.3.6	タグのチェック方法	351

7.8	"OPC"チャンネルの診断.....	353
7.8.1	"OPC"チャンネルの診断のための機能.....	353
7.8.2	ログファイルのエントリの説明.....	354
7.8.2.1	ログファイルのエントリの説明.....	354
7.8.2.2	"INFO"フラグのエントリ.....	355
7.8.2.3	"ERROR"フラグのエントリ.....	356
7.8.3	無効なタグ値の原因の特定方法.....	358
7.8.3.1	無効なタグの原因の特定方法.....	358
7.8.3.2	コンフィグレーションデータのチェック方法.....	358
7.8.3.3	チャンネルおよび接続のチェック方法.....	360
7.8.3.4	タグのチェック方法.....	362
7.9	[OPC UA]チャンネルの診断.....	364
7.9.1	OPC UA 通信のエラー処理.....	364
7.9.2	OPC UA クライアントのトレース機能の設定方法.....	364
7.9.3	OPC UA サーバーのトレース機能の設定方法.....	367
7.10	タグの品質.....	372
7.10.1	タグの品質.....	372
7.10.2	タグの品質コード.....	374
7.10.3	タグステータス.....	383
7.10.4	モニタ接続ステータスへのタグステータスの使用.....	385
7.10.5	グローバルアクションによるタグステータスのモニタ.....	387
7.10.6	内部タグの作成方法.....	388
8	OPC - オープンな接続性(Open Connectivity).....	391
8.1	OPC - オープンな接続性(Open Connectivity).....	391
8.2	OPC の機能.....	392
8.3	OPC 仕様と互換性.....	393
8.4	WinCC における OPC の使用.....	394
8.5	WinCC OPC を使用するために Windows を構成する方法.....	396
8.6	WinCC OPC DA サーバー.....	397
8.6.1	WinCC OPC DA サーバーの機能.....	397
8.6.2	複数の OPC DA サーバーの使用.....	399
8.6.3	OPC DA サーバー名の問い合わせ.....	400
8.6.4	OPC DA 接続の例.....	401
8.6.4.1	WinCC 間の接続.....	401
8.6.4.2	WinCC と SIMATIC NET FMS OPC サーバー間の接続.....	405
8.6.4.3	WinCC と SIMATIC NET S7-OPC サーバー間の接続.....	407
8.6.4.4	WinCC と Microsoft Excel 間の接続.....	413
8.7	WinCC OPC HDA サーバー.....	418
8.7.1	WinCC OPC HDA サーバーの機能.....	418
8.7.2	WinCC OPC HDA サーバーのデータ構造.....	420

8.7.2.1	WinCC OPC HDA サーバーのデータ構造.....	420
8.7.2.2	サポートされている属性の概要	421
8.7.2.3	サポートされている集約の概要	422
8.7.2.4	サポートされている関数の概要	423
8.7.2.5	WinCC OPC HDA サーバーの時間形式	424
8.7.3	品質コード.....	426
8.7.4	サポートされている書き込みアクセス	427
8.7.5	HDA サーバーブラウザ.....	431
8.7.6	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	432
8.8	WinCC OPC A&E サーバー	436
8.8.1	WinCC OPC A&E サーバーの機能.....	436
8.8.2	OPC A&E を使用した WinCC メッセージシステムのマッピング	438
8.8.2.1	OPC A&E を使用した WinCC メッセージシステムのマッピング	438
8.8.2.2	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	439
8.8.2.3	WinCC メッセージの優先度のマッピング	440
8.8.2.4	WinCC メッセージシステムの属性.....	440
8.8.2.5	確認セオリー	443
8.8.3	OPC A&E の品質コード.....	445
8.8.4	階層的アクセス機能付き OPC A&E サーバー	446
8.8.4.1	OPC A&E サーバーの機能	446
8.8.4.2	OPC A&E と階層的アクセス機能付き OPC A&E との違い	448
8.8.4.3	OPC A&E の WinCC メッセージシステムのマッピング	449
8.8.4.4	OPC A&E の品質コード.....	457
8.8.5	アーカイブメッセージの読取り	458
8.8.5.1	アーカイブイベントへのアクセス	458
8.8.5.2	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.....	459
8.8.5.3	アーカイブメッセージの読取りメソッド.....	461
8.8.5.4	アーカイブメッセージの識別.....	462
8.9	WinCC OPC UA サーバー	464
8.9.1	WinCC OPC UA サーバーの動作原理	464
8.9.2	OPC UA のセキュリティ概念	465
8.9.3	セキュリティメカニズムの構成	469
8.9.4	サポートされる OPC UA サービスとプロファイル	472
8.9.5	WinCC OPC UA サーバーの名前領域.....	475
8.9.6	OPC UA データアクセス	479
8.9.7	OPC UA Historical Access.....	479
8.9.8	OPC UA アラームと条件	481
8.9.9	WinCC メッセージシステムの属性.....	486
8.9.10	OPC UA メソッド.....	489
8.9.10.1	[OPC UA メソッド]エディタ	489
8.9.10.2	OPC UA 経由でメソッドへのアクセスを設定する方法	492
8.9.11	WinCC OPC UA サーバーの設定	493
8.9.11.1	WinCC OPC UA サーバーの設定ファイル.....	493
8.9.11.2	OPC UA サーバーのコンフィグレーション方法.....	499

8.9.12	UaExpert	502
8.9.12.1	UaExpert に関する一般情報	502
8.9.12.2	サポートされている OPC UA 機能の概要	503
8.9.12.3	UaExpert でデータを読み取る方法	505
8.10	診断	506
9	WinCC REST 通信	507
9.1	WinCC の REST インターフェースと MQTT プロトコル	507
9.2	WinCC REST サービスの設定方法	512
9.3	WinCC IT コネクタの設定方法	514
9.4	メッセージシステムで REST フィルタを設定するには	524
9.5	メソッドの概要	526
9.6	タグ管理の方法	530
9.6.1	接続の設定データの読み取り	530
9.6.2	すべての接続の設定データの読み取り	530
9.6.3	タググループの設定データの読み取り	532
9.6.4	すべてのタググループの設定データの読み取り	532
9.6.5	構造タイプの設定データの読み取り	534
9.6.6	すべての構造タイプの設定データの読み取り	534
9.6.7	構造タイプのインスタンスの読み取り	537
9.6.8	複数の構造タイプのインスタンスの読み取り	538
9.6.9	タグのランタイム値の読み取り	539
9.6.10	値のタグへの書き込み	540
9.6.11	すべてのタグのランタイム値の読み取り	541
9.6.12	値の複数タグへの書き込み	542
9.6.13	タグの設定データの読み取り	542
9.6.14	すべてのタグの設定データの読み取り	543
9.7	システムアーカイブの方法	545
9.7.1	利用可能なメソッドのリスト	545
9.7.2	すべてのプロセス値アーカイブの設定データの読み取り	545
9.7.3	単一のプロセス値アーカイブの設定データの読み取り	549
9.7.4	単一のプロセス値アーカイブタグの設定データの読み取り	550
9.7.5	プロセス値アーカイブのすべてのタグの設定データの読み取り	551
9.7.6	プロセス値アーカイブのタグのランタイム値の読み取り	554
9.7.7	単一のプロセス値アーカイブの複数タグランタイムデータの読み取り	555
9.7.8	異なるプロセス値アーカイブの複数タグのランタイム値を読み取る	558
9.7.9	アーカイブシステムの単一時間の設定データの読み取り	561
9.7.10	アーカイブシステムの全時間の設定データの読み取り	562
9.7.11	アーカイブシステムタグの設定データの読み取り	565
9.7.12	全アーカイブシステムタグの設定データの読み取り	567
9.8	メッセージシステムのメソッド	569

9.8.1	メッセージの設定データの読み取り	569
9.8.2	すべてのメッセージの設定データの読み取り	570
9.8.3	メッセージクラスの設定データの読み取り	572
9.8.4	すべてのメッセージクラスの設定データの読み取り	574
9.8.5	メッセージタイプの設定データの読み取り	576
9.8.6	すべてのメッセージタイプの設定データの読み取り	577
9.8.7	メッセージブロックの設定データの読み取り	580
9.8.8	すべてのメッセージブロックの設定データの読み取り	580
9.8.9	メッセージグループの設定データの読み取り	582
9.8.10	すべてのメッセージグループの設定データの読み取り	583
9.8.11	タグ制限の設定データの読み取り	585
9.8.12	すべてのタグ制限の設定データの読み取り	586
9.8.13	REST フィルタの設定データの読み取り	592
9.8.14	すべての REST フィルタの設定データの読み取り	593
9.8.15	メッセージリストのランタイムメッセージの読み取り	593
9.8.16	短期アーカイブからのランタイムメッセージの読み取り	594
9.8.17	長期アーカイブからのランタイムメッセージの読み取り	595
9.8.18	ヒットリストからのランタイムメッセージの読み取り	596
9.8.19	ロックリストからのランタイムメッセージの読み取り	597
9.8.20	非表示にするメッセージリストからのランタイムメッセージの読み取り	598
9.8.21	非表示のメッセージリストからのランタイムメッセージの読み取り	599
9.9	方法のフィルタ	601
10	WinCC/Cloud Connector	607
10.1	WinCC/Cloud Connector	607
10.2	ライセンスクラウドコネクタ	610
10.3	MQTT を経由したクラウドへのデータ転送	611
10.4	REST を経由したクラウドへのデータ転送	615
10.5	WinCC タグ管理での設定	617
10.6	WinCC クラウドコネクタの MQTT 設定	619
10.7	WinCC クラウドコネクタの REST 設定	626
10.8	MQTT 経由のクラウド接続の設定方法	628
10.9	MindConnect IoT Extension (MindSphere / Insights Hub)から MindConnect MQTT に アセットを移行する方法	632
10.10	IOT エクステンションにより MindSphere 接続を設定する方法	635
10.11	EU1 ブローカにより MindSphere 接続を設定する方法	638
10.12	REST 経由のクラウド接続の設定方法	642
10.13	クラウド接続の診断	645
	索引	647

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ーシステムの設定：
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。
マルチユーザーシステムの場合、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になっています。
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

以下のようなサーバー障害などの故障時でもシステム作動を維持するために、冗長化サーバーを設定します。リダンダントシステムの設定の正確な手順については、WinCC 文書「リダンダントシステム」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。

1.2 WinCC のクライアント/サーバーシステム

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。
- 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イアウトします)。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。

- 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。
 - 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。
 - 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。
 - 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 102)

リモートコンフィグレーション (ページ 83)

ランタイムでのシステム動作 (ページ 73)

クライアントコンフィグレーション (ページ 45)

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 20)

標準的な設定 (ページ 18)

暗号化通信 (ページ 85)

1.3 標準的な設定

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 20)」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

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。

シンクライアント

シン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 사용할 こともできます。

シン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 66)

異なるサーバーからのピクチャの表示 (ページ 62)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

インポートパッケージのコンフィグレーション (ページ 50)

異なるサーバーからのメッセージの表示 (ページ 69)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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 37)

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 41)

WinCC のクライアント/サーバーシステム (ページ 14)

数量構造および特性 (ページ 20)

暗号化通信 (ページ 85)

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 18)」で参照できます。ハードウェア要件については、インストールに関する注意を参照してください。

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。

次のような要素が、設定中に WinCC プロジェクトのパフォーマンスに影響を与える可能性があります。

- 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、10 のレベルが許可されています。
- 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。

- 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。

1.4 数量構造および特性

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。

システムタグ:サーバーでのアクティブなクライアント接続

システムタグ「@ConnectedRTClients」は、ローカルサーバーまたは RT サーバーへのアクティブなクライアント接続の数を指定します。

アクティブな接続が開始されました。

- スタンドアロンサーバーへのクライアント接続
- 冗長マスターサーバーへのクライアント接続
- 冗長スタンバイサーバーへのクライアント接続(優先サーバーとして入力されている場合)

下記も参照

WinCC のクライアント/サーバーシステム (ページ 14)

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 102)

- リモートコンフィグレーション (ページ 83)
- ランタイムでのシステム動作 (ページ 73)
- クライアントコンフィグレーション (ページ 45)
- 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 18)

1.5 クライアント/サーバーシナリオ

概要

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。

WinCC クライアント:ランタイムでの動作

オペレータオーソリゼーションに応じて、クライアントには次のようなさまざまな機能があります。

- システムのモニタのみ。
- システムのモニタおよび作動の両方。
- サーバープロジェクトをリモートで設定する(例: メンテナンスコンピュータとして)。

クライアントの表示は設定によって異なります:

- たとえば、プロセスをシステム内の複数のポイントから操作できるようにする場合、すべてのクライアントにプロジェクトの同じビューが表示されます。
- クライアントには、メッセージのみやプロセス値のみなど、プロジェクトのさまざまなビューが表示されます。

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 WinCC 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 WinCC 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。

インストール

「WinCC クライアント」は、両方のクライアントタイプにインストールできます。

このためには、次の 2 つのクライアントライセンスのいずれかが必要です。

- 「RT クライアント」
- 「RC クライアント」

Microsoft SQL-Express のインストールが必要です。

WinCC 情報システムの詳細情報:[ライセンス> ライセンスの概要]

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エリアを担当します。
- 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。

1.5 クライアント/サーバーシナリオ

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。これによって、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のが容易になります。

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。たとえば、ミラーリングされたディスクの信頼性を確保できます。

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の値とメッセージは、通常どおりランタイムで、プロセス画像の WinCC OnlineTrendControl または WinCC AlarmControl に表示できます。

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用できるようになります。

サーバー間通信

2 台のサーバー間の通信時には、1 台のサーバーがもう 1 台のサーバーにある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 57)

標準サーバーの設定方法 (ページ 53)

ランタイムでのシステム動作 (ページ 73)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

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は、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
- アラームと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
- 設定データのクライアントへの供給

個々のサーバーで実行されるタスクは、技術的要素または機能的要素によって分散できません。

- 技術面:各サーバーでプロセス/プラントの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の固有タスクを実行します(例、メッセージシステムまたはアーカイブ)。

注記

常に、関連するクライアントの前に、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。

設定手順

マルチユーザシステムの設定

マルチユーザシステムでは、ランタイムでサーバーのビューを表示する複数のクライアントを設定できます。

これらの「独自プロジェクトを持たない WinCC クライアント」は、サーバーから排他的にデータを取得し、独自の設定を持つことはありません。

マルチユーザーシステムのサーバーを設定するには、次の手順が必要です。

1. サーバー上に新しい「マルチユーザープロジェクト」タイプを作成します。
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. サーバーのコンピュータリストでリモートで設定または監視されるクライアントを含みます。

4. リモートで設定されるクライアントのオペレータ権限を割り当てます(「リモートに設定」)。
5. サーバーへの自動パッケージインポートを有効にします。
6. サーバープロジェクトでクライアントのプロパティを設定します(開始画像、キー組み合わせの無効化など)。

分散システムの設定

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できます。

これらの「独自のプロジェクトを使用する WinCC クライアント」は、独自のローカルデータを使用する独自のプロジェクトを持っています。サーバーから更新されたデータはパッケージエクスポート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を設定するには、次の手順が必要です。

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、変数など)を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のみにすることができます。
3. サーバーのコンピュータリストでリモートで設定されるクライアントを含みます。
4. リモートで設定されるクライアントのオペレータ権限を割り当てます。
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. クライアントでクライアントプロジェクトを設定します。
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。

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 57)

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 41)

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 37)

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 45)

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 14)

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できます。

- 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

プロジェクトタイプの変更

既存のプロジェクトは、後で、プロジェクトタイプを修正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。

[コンピュータ]エディタの[プロパティ-プロジェクト]エリアにある[全般]で、設定を変更します。

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。

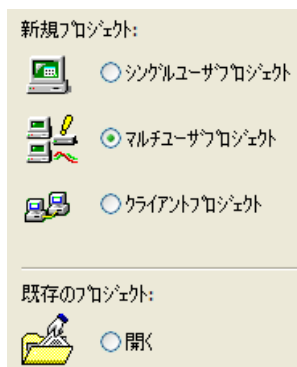
Simatic Shell の通信設定で、特定のポートを指定することもできます。

[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。

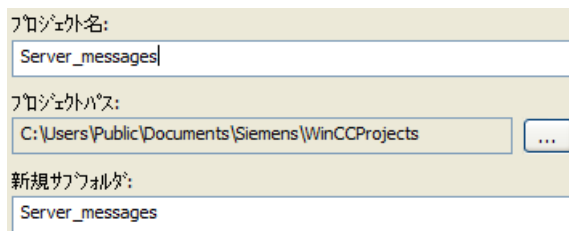
手順

以下の手順はマルチユーザー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 エクスプローラのサーバーで、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. [マルチユーザープロジェクト]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 41)

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 37)

オペレーター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 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

1.6.3 コンピュータリストへのクライアント登録方法

概要

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。

必要条件

マルチユーザプロジェクトがマルチユーザーシステムまたは分散システムとして作成されたこと。

設定コンピュータは自動的にクライアント/サーバーシステムのサーバーになること。

手順

1. WinCC Configuration Studio で[コンピュータ]エディタを開きます。
2. ナビゲーションエリアでの[プロジェクト]エントリを選択します。
3. [コンピュータ]タブの[名前]列で、最初の空のボックスをクリックします。
4. 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。
5. プロジェクトディレクトリで[フルアクセス]権限を確認します。
そのコンピュータがプロジェクトの登録済みコンピュータのリストに追加されます。
クライアントコンピュータにネットワークを介してアクセスできる場合、[ローカル設定]で WinCC クライアントの自動開始設定を構成できます。
6. クライアントサーバーシステムの現在のサーバーへのアクセスを付与するすべてのコンピュータを追加します。

注記**WinCC クライアント:コンピュータ名の変更**

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 41)
- 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 37)
- 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 45)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 標準的な設定 (ページ 18)

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。

1.6 サーバーコンフィグレーション

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。

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. WinCC Configuration Studio で[コンピュータ]エディタを開きます。
2. [プロパティ-プロジェクト]エリアで、[オプション]にある次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。]
3. 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。

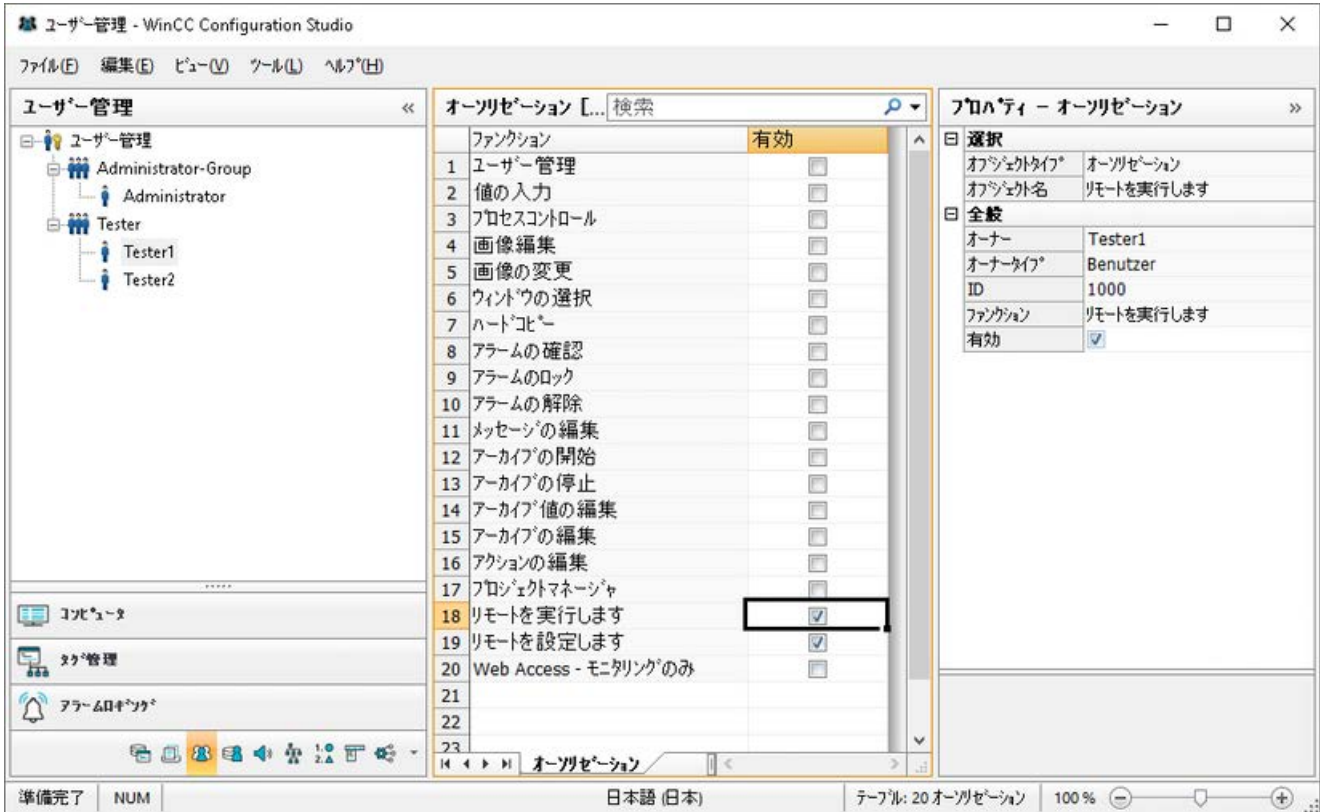
オペレータオーソリゼーションの割り付けについての詳細な情報は、Windows 文書にて提供されます。

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. ナビゲーションエリアでユーザーを選択します。

1.6 サーバーコンフィグレーション

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 41)
- 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 37)
- 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 45)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)

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設定データ(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、初回設定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、設定できます。

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。

設定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[設定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。

1.6 サーバーコンフィグレーション

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	X		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	X		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
モニタが設定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。 		X	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。		X	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		X	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。

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	X		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	X		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設定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。
設定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、すべての設定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

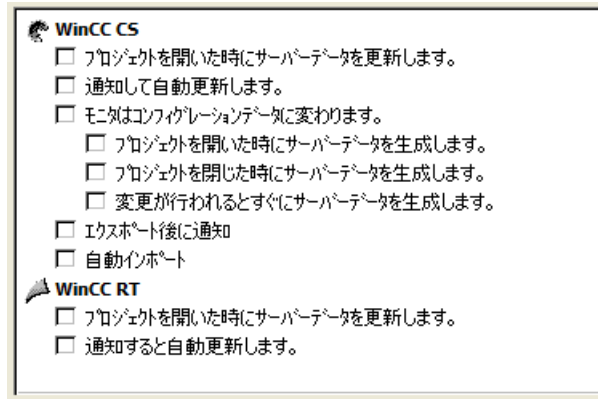
サーバーデータの入ったパッケージは、WinCC エクスプローラ内[サーバーデータ]下のリストに置かれています。パッケージは、ファイルシステムのプロジェクトディレクトリに、<プロジェクト名>\<コンピュータ\パッケージ>*.pck の形式で保存されます。

1.6 サーバーコンフィグレーション

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。

	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)
	ローカルで作成され、独自のプロジェクトに再インポートされたパッケージ。

下記も参照

- クライアントコンフィグレーション (ページ 45)
- 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 41)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 標準的な設定 (ページ 18)

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 サーバー上の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。

初期条件

- 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。

解決法

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。

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. WinCC Configuration Studio で[コンピュータ]エディタを開きます。
2. ナビゲーションエリアの[プロジェクト]で、設定したいクライアントを選択します。
3. [WinCC Runtime の起動時のプロセス]タブで、クライアント上で有効にするアプリケーションを有効にします。
例:スクリプトを使用する場合、[グローバルスクリプトランタイム]を有効にします。
4. [パラメータ]の[プロパティ-コンピュータ]エリアで、クライアントで起動する必要があるランタイムの言語を選択します。
たとえば、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。
必要に応じて、クライアントのデフォルトランタイム言語を変更します。
5. [グラフィック]で、クライアントの開始画面を選択します。
開始画像は、各クライアントに個別に選択できます。
6. 必要な場合、[プロパティ-コンピュータ]エリアでさらに設定を行います(例、[キー]にあるランタイムでの操作のホットキーなど)。
7. WinCC 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. WinCC エクスプローラの[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]エントリを選択します。
設定ダイアログが開きます。
9. [自動インポート]設定を有効にし、[OK]を押して確定します。
- 10.[サーバーデータ]ショートカットメニューで、サーバーパッケージを作成します。

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 37)

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 45)

1.6 サーバコンフィグレーション

クライアント/サーバシナリオ (ページ 24)

数量構造および特性 (ページ 20)

WinCC のクライアント/サーバシステム (ページ 14)

標準的な設定 (ページ 18)

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

概要

クライアント設定は、クライアントが複数のサーバーに表示することができるように、分散システムを設定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、マルチユーザーシステムを設定する場合は、クライアント設定は必要ありません。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設定する場合、各クライアントに個々のクライアントプロジェクトを設定します。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認証に従って、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

1.7 クライアントコンフィグレーション

- 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。

設定のステップ

1. サーバープロジェクトの設定
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設定
4. クライアント側でのクライアントプロジェクトの設定。

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、設定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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

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 60)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

インポートパッケージのコンフィグレーション (ページ 50)

クライアント側でのプロジェクトの新規作成 (ページ 47)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

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せん。

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、後で、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。

[コンピュータ]エディタの[プロパティ-プロジェクト]エリアにある[全般]で、設定を変更します。

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

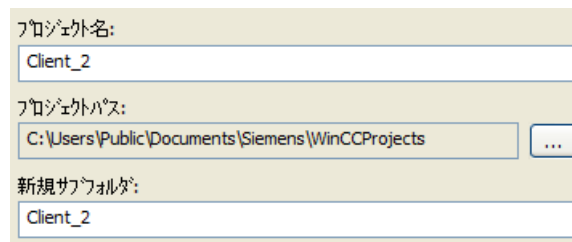
1. WinCC エクスプローラのクライアントで、メニュー項目の[ファイル]>[新規]を選択します。
[WinCC エクスプローラ]ダイアログが開きます。



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。

- 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:
Client_2

プロジェクトパス:
C:\Users\Public\Documents\Siemens\WinCCProjects ...

新規サブフォルダ:
Client_2

- [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 41)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

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 60)

インポートパッケージのコンフィグレーション (ページ 50)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

1.7.3 インポートパッケージのコンフィグレーション

概要

分散システムの種々のサーバーのプロセ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。

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。

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。

1.7 クライアントコンフィグレーション

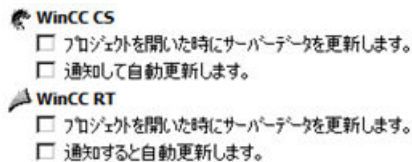
3. ロードするパッケージを選択し、[OK]をクリックします。
パッケージはデフォルトで、ディレクトリ「...\\<サーバープロジェクト名>\<コンピュータ名>\パッケージ」に、名前「<プロジェクト名_コンピュータ名>*.pck」で保存されています。ただし、どのようなデータ媒体に保存されているパッケージにもアクセスできます。
4. [開く]をクリックします。データがロードされます。対応するサーバーを使用できない場合、新規パッケージの要求時に、適切な障害エントリが表示されます。

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]タブが現れます。



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。

	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

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

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 60)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCCのクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

1.7.4 標準サーバーの設定方法

概要

複数のサーバーにアクセスできるクライアント向けに、分散システムで標準サーバーを設定できます。

- たとえば、タグに一意のサーバー接頭語が指定されていない場合、クライアントは標準サーバーからデータを要求します。
- コンポーネントに標準サーバーが設定されていない場合、サーバー接頭語なしのアクセスでは、クライアントのデータ管理(内部タグなど)でローカルにデータを検索しようとしています。
クライアント上のコンポーネントのローカルデータ管理がない場合、メッセージやアーカイブなどのアクセスはエラーで拒否されます。

1.7 クライアントコンフィグレーション

必要条件

標準サーバーのサーバーパッケージがクライアントにインポートされます。

手順

1. クライアントで WinCC エクスプローラが開かれます。
2. 「サーバーデータ」コンポーネントのショートカットメニューで[標準サーバー]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。



ダイアログにリストされるコンポーネントは WinCC の初期設定に応じて異なります。リストには、インストールされたオプションのコンポーネント(例: 「SSM - 分割画面マネージャー」)が含まれる場合があります。

3. 目的のコンポーネントの行で、[シンボリック コンピュータ名]列のエントリをクリックします。
クライアントにパッケージがインポートされているすべてのサーバーのシンボリックコンピュータ名が表示されます。
4. サーバーを選択します。
5. [OK]をクリックし、選択を確定します。

標準サーバーを必要とするコンポーネント

アラーム

オペレータメッセージをクライアント上で生成する場合は、「アラーム」コンポーネントの標準サーバーを指定する必要があります。

アラームログはクライアント自体では設定できません。したがって、メッセージは常にサーバーに出力されます。

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

これらのコンポーネント用に標準サーバーを設定するのは、非常に特殊なアプリケーションの場合にのみ意味があります:

- プロセス画像
- タグ
WinCC/WebNavigator 専用の Web サーバーを使用する場合は、標準サーバーをタグ用に設定する必要があります。
- Archives
- テキストライブラリ
- ユーザーアーカイブ

SIMATIC のマニュアルまたはカスタマサポートによって特定のサーバーを設定するように明示的に求められない場合は、[標準サーバーなし]の設定を維持します。

これらのコンポーネントのデータに対して有効なサーバー接頭語が生成されていない場合は、標準サーバーが使用されます。次に、要求されたデータの検索が標準サーバーで実行されます。

クライアントに標準サーバーが設定されていない場合、サーバー接頭語がないため、このデータに対応するサーバーが見つかりません。

注記

タグ管理:ランタイムでのツールチップ表示

タグに標準サーバーを入力すると、ランタイムの WinCC クライアントのタグ管理にツールチップとしてステータス情報が表示されません。

基本プロセスコントロール

アラーム

アラームには常に標準サーバーを指定する必要があります。

タグ

タグに標準サーバーを指定しないようにする必要があります。

SSM(分割画面マネージャ)

「SSM」コンポーネントには常に標準サーバーを指定する必要があります。

1.7 クライアントコンフィグレーション

トレンドグループ:

- 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冗長化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」コンポーネントの標準サーバーとして指定できます。つまり、コンパイルされたトレンドグループはこれらの WinCC クライアントでも使用できます。
- WinCC クライアントで「SSM」コンポーネント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、このクライアント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、トレンドコントロールでこれらのトレンドグループを表示することはできません。
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。

画面構成:

- 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」コンポーネント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。

冗長システム:

- 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、冗長化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 50)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

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 60)

優先サーバーのコンフィグレーション方法 (ページ 57)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

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。

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。

1.7 クライアントコンフィグレーション

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。



シンボリック	物理的	冗長	優先サーバー
Project_Redundancy_Server_DPC_40	DPC_4005		優先サーバーなし

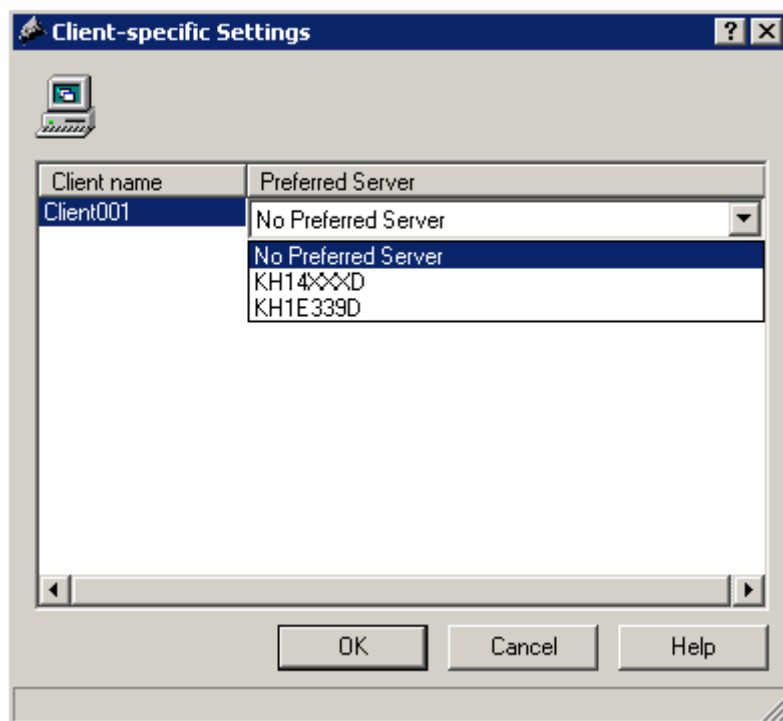
4. [OK]をクリックして、入力を終了します。

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 60)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

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 62)
- 標準サーバーの設定方法 (ページ 53)
- インポートパッケージのコンフィグレーション (ページ 50)
- クライアント側でのプロジェクトの新規作成 (ページ 47)
- クライアントコンフィグレーション (ページ 45)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 標準的な設定 (ページ 18)

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

分散システムの任意のプロセス画像をクライアントの開始画像として使用できます。例:

- サーバーの画像
- 顧客自身の画像の 1 枚
- その他の画像

以下の手順は、サーバーの画像を開始画像として使用する方法について、説明しています。

必要条件

画像を開始画像として使用するサーバーのパッケージはクライアントにインポートされること。

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. [コンピュータ]エディタでコンピュータ名を選択します。
[プロパティ - コンピュータ]エリアが表示されます。
3. [グラフィック]エリアで、[開始画像]フィールドをクリックします。

4. 開始画像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:
 - <Server name>:: - 例:
Server1::StartPicture.pdl
5. 画像を検索するには、[検索]ボタンを使用します。
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画像が表示されます。
6. エントリを OK で完了します。

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

インポートパッケージのコンフィグレーション (ページ 50)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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)

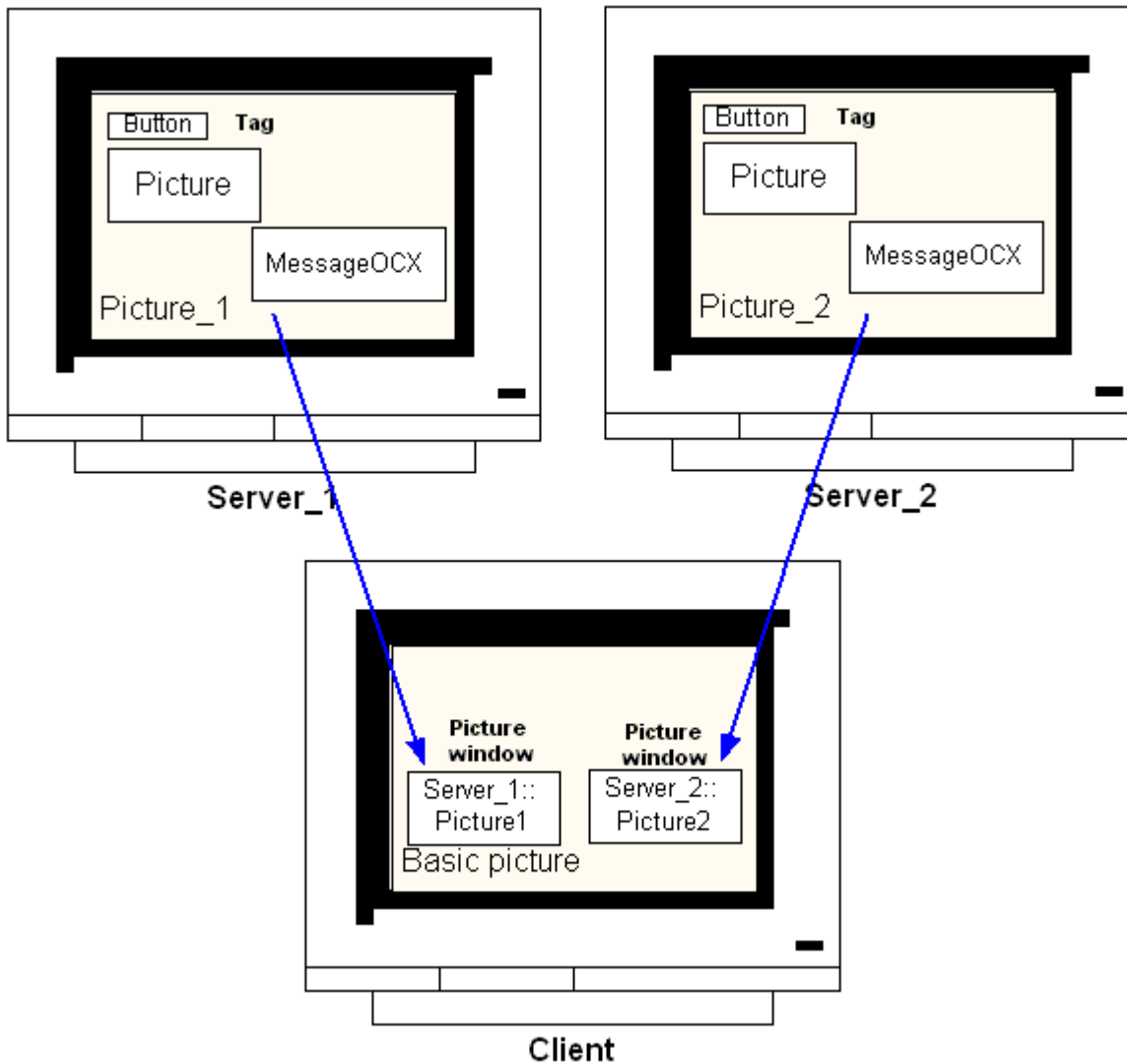
数量構造および特性 (ページ 20)

WinCC のクライアント/サーバーシステム (ページ 14)

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 70)
- 異なるサーバーからのメッセージの表示 (ページ 69)
- 異なるサーバーからのデータの使用 (ページ 66)
- 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 64)
- 異なるサーバーからのピクチャの表示 (ページ 62)
- 優先サーバーのコンフィグレーション方法 (ページ 57)
- 標準サーバーの設定方法 (ページ 53)
- インポートパッケージのコンフィグレーション (ページ 50)
- クライアント側でのプロジェクトの新規作成 (ページ 47)
- クライアントコンフィグレーション (ページ 45)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 標準的な設定 (ページ 18)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

異なるサーバーからのデータの使用 (ページ 66)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

インポートパッケージのコンフィグレーション (ページ 50)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

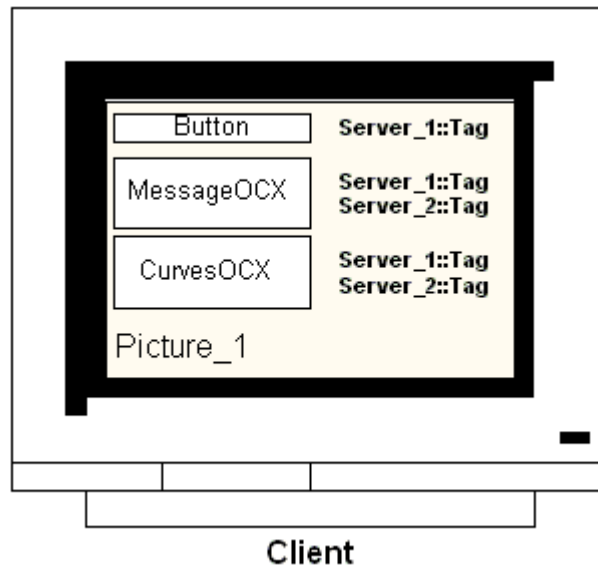
1.7.9 異なるサーバーからのデータの使用

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設定します。

各基本画像の複数のサーバーからのデータにアクセスすることができます。たとえば:

- 2つの出力フィールド:
 - Server_1 のプロセス値の出力フィールド。プラントユニット A を監視します。
 - Server_2 のプロセス値の出力フィールド。プラントの別の部分を監視します。
- トレンドは、異なるプラントユニット/サーバーからのデータを比較を表示します。
- 複数のサーバーのメッセージを表示するメッセージウィンドウです。



あるクライアントの基本画像の設定を別のクライアントにコピーすることもできます。ただし、基本画像でアドレス指定されたサーバーのパッケージは、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

サーバー側で設定され、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。

それらを実行するには、グローバルスクリプトからの C アクションと C ファンクション、または VBS アクションと VBS プロシージャは、クライアント側に存在している必要があります。グローバル C スクリプトおよび VBS スクリプトは、パッケージの一部ではありません。

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、基本画像として使用する画像を設定します。
3. WinCC OnlineTrendControl を、[コントロール]タブのオブジェクトパレットからの基本画像に挿入します。
[WinCC OnlineTrendControl のプロパティ]ダイアログが表示されます。
4. 現在のプロセスを監視する場合、データソースとして[オンラインタグ]を選択します。
5. トレンドタブを有効にします。

1.7 クライアントコンフィグレーション

6. 最初のトレンドの場合、[アーカイブ/タグの選択]にある[選択]ボタンを押して、そのプロセス値を表示するタグを選択します。
7. 以下の形式でタグ名を入力します:
 - 「<ServerPrefix1>::<TagName>」。[OK]をクリックして確定します。
8. トレンドタブで、[+]ボタンを押し、2 番目のトレンドを追加します。
9. 2 番目のトレンドを、次の形式で 2 番目のサーバーからのタグに接続します:
 - 「<Serverprefix2>::<TagName>」。
- 10.[OK]で設定を確定します。

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ウィンドウに表示されます。

- トレンド 1 は、サーバー 1 のデータを表示します。
- トレンド 2 は、サーバー 2 の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 50)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

クライアント側でのプロジェクトの新規作成 (ページ 47)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

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 70)
- 異なるサーバーからのメッセージの表示 (ページ 69)
- 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 64)
- 異なるサーバーからのピクチャの表示 (ページ 62)
- 優先サーバーのコンフィグレーション方法 (ページ 57)
- 標準サーバーの設定方法 (ページ 53)
- インポートパッケージのコンフィグレーション (ページ 50)
- クライアント側でのプロジェクトの新規作成 (ページ 47)
- クライアントコンフィグレーション (ページ 45)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 標準的な設定 (ページ 18)

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がクライアントの基本画像で表示する場合、対応するメッセージシーケンスレポートを出力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、正しい順序で出力されます。

WinCC は、メッセージシーケンスレポート用の事前設定レイアウトおよび印刷ジョブを提供します。

手順

1. WinCC エクスプローラで[レポートデザイナ]をクリックします。
2. データウィンドウで非言語依存レイアウト「@CCAlgrtSequence.RPI」をダブルクリックします。
レイアウトエディタが開きます。
3. [選択]ボタンを押します。
[レポート - テーブル列選択]ダイアログが表示されます。
4. [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がロギングされる必要があるサーバーを、[選択されたサーバー]リストに追加します。
クライアントにインポートされたパッケージを持つサーバーのみが表示されます。
5. 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。
6. [OK]を選択して、入力を確定します。
7. 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。
8. 独自の名前でレイアウトを保存した場合、レイアウトを[レイアウト]リストから選択します。
[ラインプリンタ用ラインレイアウト]オプションを選択します。
9. [プリンタセットアップ]タブで[プリンタ]オプションを有効にします。
10. 接続された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。
11. [OK]で入力内容を確定します。
12. WinCC Configuration Studio で[コンピュータ]エディタを開きます。
13. ナビゲーションエリアの[プロジェクト]で、クライアントコンピュータを選択します。
14. [WinCC Runtime の起動時のプロセス]タブで[メッセージシーケンスレポート]アプリケーションを有効にします。

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 47)

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 70)

異なるサーバーからのメッセージの表示 (ページ 69)

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 64)

異なるサーバーからのピクチャの表示 (ページ 62)

優先サーバーのコンフィグレーション方法 (ページ 57)

標準サーバーの設定方法 (ページ 53)

インポートパッケージのコンフィグレーション (ページ 50)

クライアントコンフィグレーション (ページ 45)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

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

概要

WinCCのクライアント/サーバーシステムは、システムのコンフィグレーションを複数のサーバーに分散させて使用することができ、個々のサーバーに加わる負荷を低減できます。

サーバー上で設定したデータは、クライアントごとに表示できます。クライアントは、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。

表示するデータは、常にアーカイブサーバーからのデータです。

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

ランタイムでクライアントから画像が呼び出されると、最初にグラフィックランタイムはローカルに保存されている画像を検索します。

- 対応する名前の画像がその場所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。
- 画像が見つからない場合は、対応するメッセージを表示します。

1.8 ランタイムでのシステム動作

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

画像キャッシュ:パフォーマンスの向上

クライアントでの画像の構築速度を速くするには、対応するが画像をクライアントにローカルにコピーします。

[コンピュータ]エディタで、クライアントに対応する保存パスを設定します:

- [設定-ローカル設定]エリアの[画像キャッシュ]フィールドで、保存パス用のフォルダを選択します。
- [キャッシュを使用]フィールドで、ローカル保存パスが常に使用されるか、使用を推奨するかを選択します。
この機能を無効化するには、[しない]オプションを選択します。

サーバープロジェクトで画像が修正されている場合は、データを手動で更新する必要があります。修正した画像をクライアントの保存パスにコピーしなおします。

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のコンフィグレーション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。

表示するデータは常にメッセージサーバーからのデータです。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。

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

オペレータの権限が、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。

サーバーのデータは、障害の場合には表示できません。たとえば、すべての操作可能なグラフィックオブジェクトは、無効に切り替えられます。

注記

システム再起動

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。

1.8 ランタイムでのシステム動作

「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。

ソフトウェアエラーが検出されると、ライフビートのモニタリングによって次のアクションがトリガされます。

- プロセスコントロールメッセージによってソフトウェアのエラーがユーザーに通知されます。
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。
- 冗長システム:
 - システムタグ"@RedundantServerState"で、サーバーの状態が「エラー」に変わります。
 - 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。

注記

エラー後に冗長システムサーバーが再起動します

「Application Health Check」ファンクションがソフトウェアエラーを検出し、クライアントの切り替えが開始された場合は、関連するサーバーを再起動する必要があります。サーバーを再起動した後でないと、クライアントをこの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。

下記も参照

サーバーのスタートアップ (ページ 77)

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 79)

クライアントのシャットダウン (ページ 81)

サーバーのシャットダウン (ページ 81)

クライアントのスタートアップ (ページ 78)

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 20)

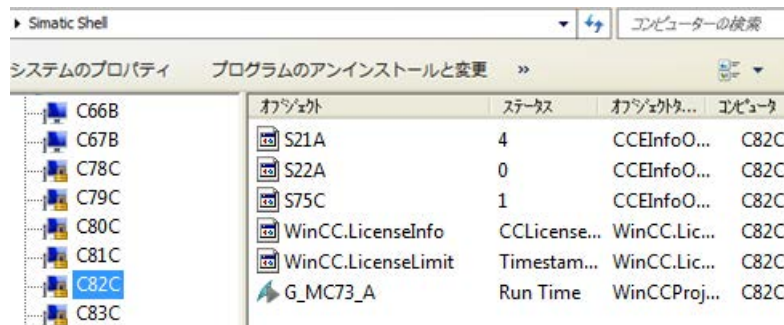
WinCC のクライアント/サーバーシステム (ページ 14)

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 79)

プロジェクトの実行方法 (ページ 98)

クライアントのシャットダウン (ページ 81)

サーバーのシャットダウン (ページ 81)

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 78)

ランタイムでのシステム動作 (ページ 73)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

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 79)

プロジェクトの実行方法 (ページ 98)

クライアントのシャットダウン (ページ 81)

サーバーのシャットダウン (ページ 81)

- サーバーのスタートアップ (ページ 77)
- ランタイムでのシステム動作 (ページ 73)
- 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 20)
- WinCC のクライアント/サーバーシステム (ページ 14)

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

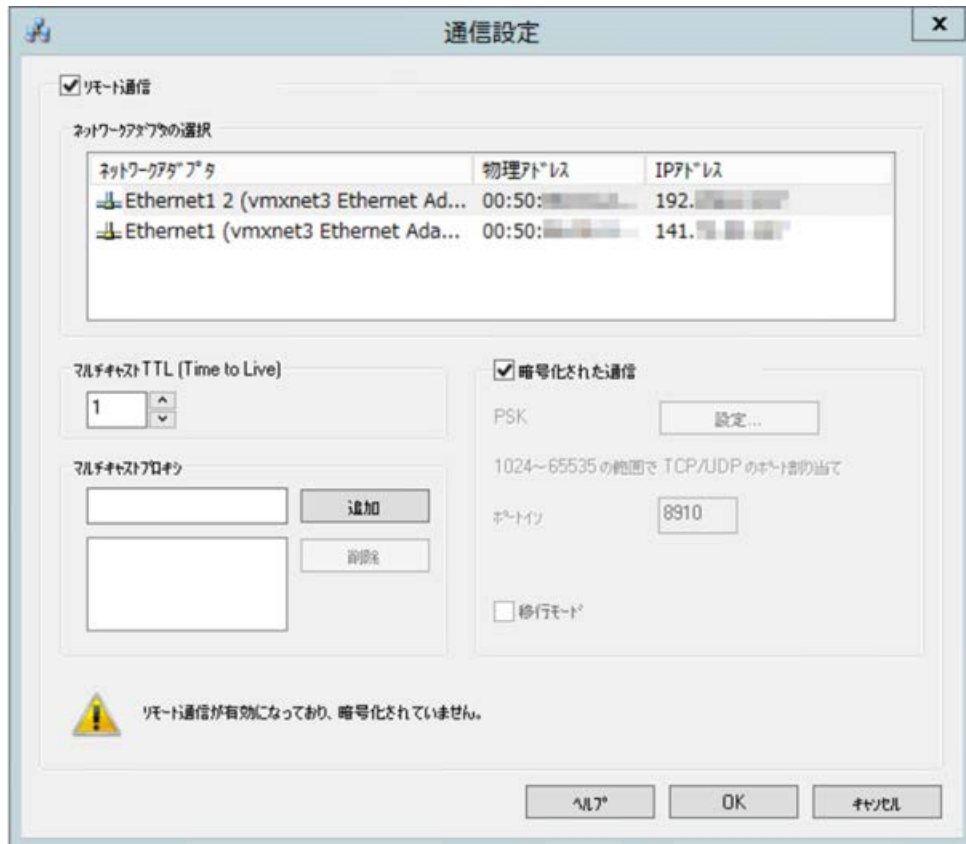
ディレクトリ「Simatic Shell」を使用して、ネットワークカードの設定をチェックできます。

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 88)

リモートコンフィグレーション (ページ 83)

クライアントのスタートアップ (ページ 78)

サーバーのスタートアップ (ページ 77)

ランタイムでのシステム動作 (ページ 73)

WinCC のクライアント/サーバーシステム (ページ 14)

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 78)

プロジェクトの終了方法 (ページ 100)

クライアントのシャットダウン (ページ 81)

サーバーのスタートアップ (ページ 77)

ランタイムでのシステム動作 (ページ 73)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

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 100)

サーバーのシャットダウン (ページ 81)

クライアントのスタートアップ (ページ 78)

サーバーのスタートアップ (ページ 77)

ランタイムでのシステム動作 (ページ 73)

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 20)

WinCC のクライアント/サーバーシステム (ページ 14)

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。

最新のリモートアクセスに関する説明は、FAQ 78463889 で確認できます。

- SiePortal:WinCC ステーションへのリモートアクセス (<http://support.automation.siemens.com/WW/view/en/78463889>)

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、ネットワークで使用できる 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。

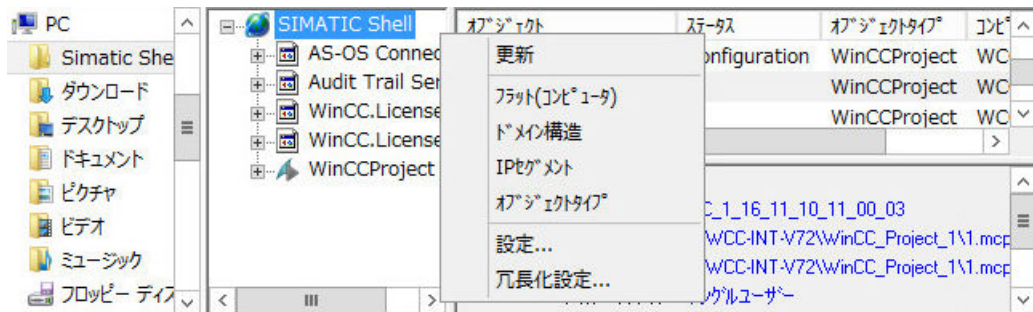
PC および WinCC プロジェクトは、以下の基準に従って構造化表示されます。

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。



1.9 リモートコンフィグレーション

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。



ネットワークの中断のためのアイコン

ネットワークケーブルを取り外した後など、ローカルネットワークアダプターが一時的に使用不可能になった場合、エントリの上に[!]マークが短時間表示されます: 、

必要な場合は、[Simatic Shell]の表示を更新して、接続された PC の変更済みステータスを表示します。

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 100)

プロジェクトの実行方法 (ページ 98)

サーバープロジェクトピクチャの編集方法 (ページ 97)

編集用にプロジェクトを開く方法 (ページ 95)

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 92)

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 88)

WinCC のクライアント/サーバーシステム (ページ 14)

暗号化通信 (ページ 85)

SiePortal:WinCC ステーションへのリモートアクセス (<http://support.automation.siemens.com/WW/view/en/78463889>)

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。

1.9 リモートコンフィグレーション

WinCC の暗号化

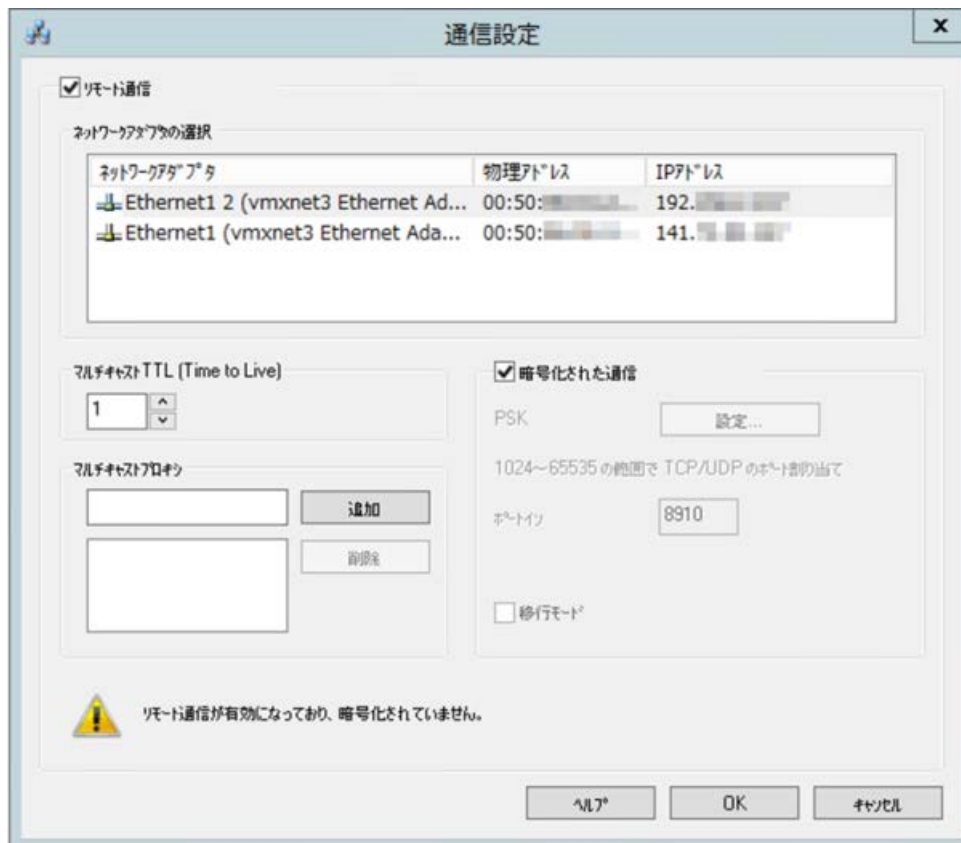
WinCC は Microsoft の「セキュリティサポートプロバイダ」(SSP)インターフェースを使用します。

ドメイン内のコンピュータの場合は、Microsoft の「リモートデスクトッププロトコル (RDP)」の暗号化が使用されます。

ドメイン外では、最先端の対称暗号化が使用されます。

暗号化通信の設定

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。

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 88)」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。

- インターネット:[接続の暗号化のために SQL Server データベースエンジンを設定]
(<https://docs.microsoft.com/ja-jp/sql/database-engine/configure-windows/configure-sql-server-encryption?view=sql-server-ver15>)




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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

標準的な設定 (ページ 18)

インターネット:[接続の暗号化のために SQL Server データベースエンジンを設定] (<https://docs.microsoft.com/ja-jp/sql/database-engine/configure-windows/configure-sql-server-encryption?view=sql-server-ver15>)

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

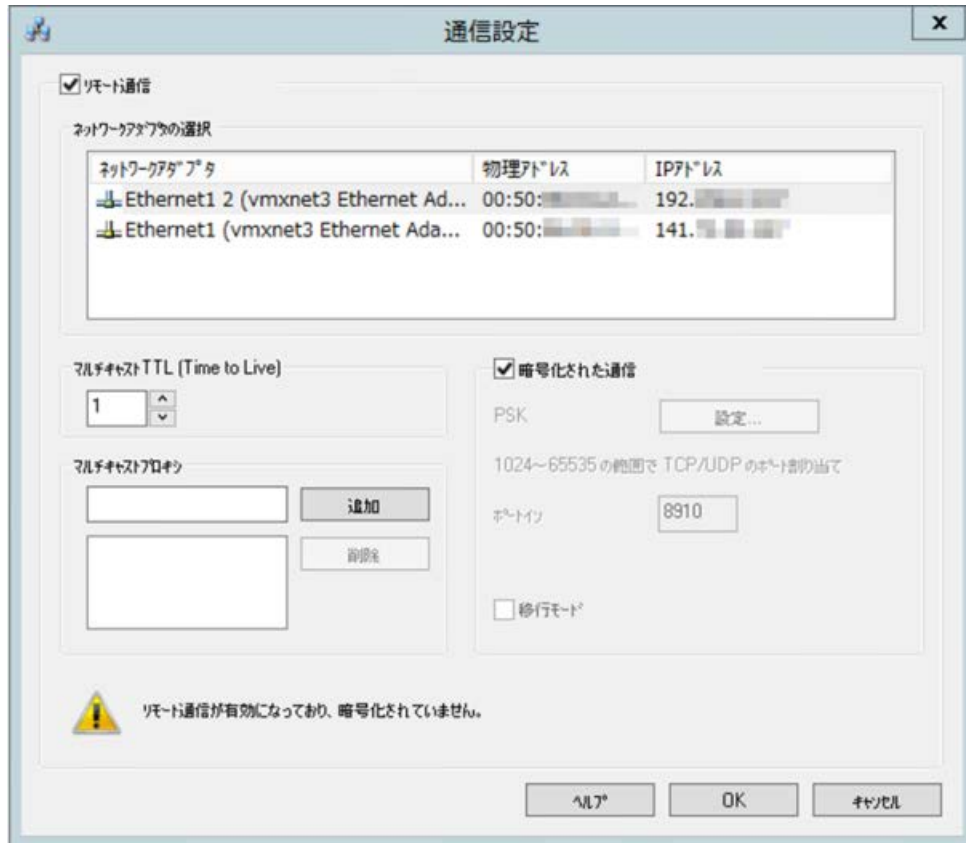
- [リモート通信]オプションが有効になり、ネットワークアダプタが設定されます。

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。

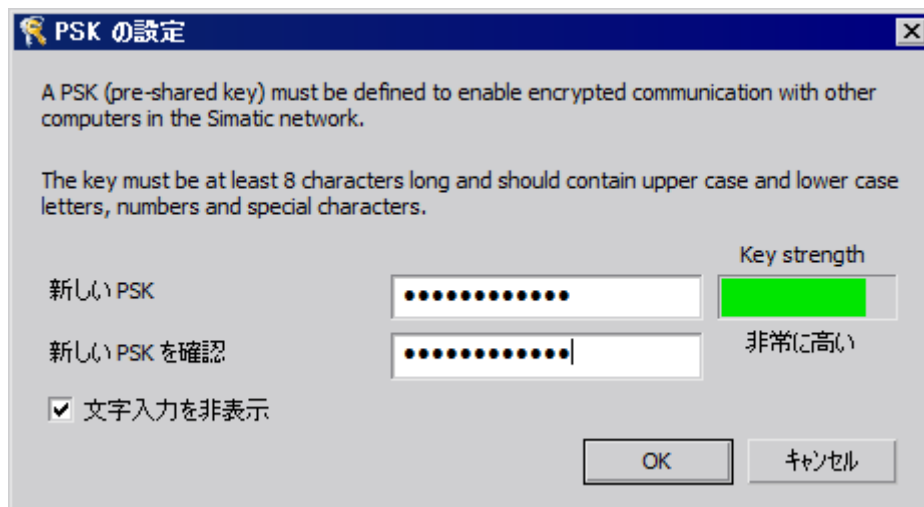
1.9 リモートコンフィグレーション

- [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。



- [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。
- [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。

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 92)
- プロジェクトの終了方法 (ページ 100)
- プロジェクトの実行方法 (ページ 98)
- サーバープロジェクトピクチャの編集方法 (ページ 97)
- 編集用にプロジェクトを開く方法 (ページ 95)
- リモートコンフィグレーション (ページ 83)
- WinCC のクライアント/サーバーシステム (ページ 14)
- 暗号化通信 (ページ 85)

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

設定オプション

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サーバープロジェクトにリモートでアクセスできます。

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)とファイルベースのデータ(ピクチャおよびグラフィック、レポート、スクリプト)が区別されます。

注記

複数のクライアントの編集をオフ

サーバーデータベースからのデータは、複数のクライアントが同時に編集できます。

ただし、同じデータに複数のクライアントがアクセスするとき、最後に保存されたクライアントの変更内容が常に保存されることに注意してください。サーバーデータベースからのデータの場合、個別の値が変更された場合でも、対応するエディタのすべてのデータが常に保存されます。

ファイルに保存されたデータの場合、既に開いたファイルは、それ以上のアクセスがブロックされます。

詳細情報: 「WinCC での作業」 > 「プロジェクトでの作業」 > 「プロジェクトの作成および編集」 > 「マルチユーザーエンジニアリングの使用方法」

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。

タグロギングデータは、ランタイムでは変更できません。

変更内容は、サーバーから関係するすべてのクライアントに分散されます。

画像

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ロックされます。プロジェクトの異なる画像は、異なるクライアントから開くことができます。

画像はランタイムで変更でき、保存すれば次にその画像を選択したときに使用できます。

画像をローカルに保存して編集することもできます。この場合、それらを手動でサーバーと同期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。

メッセージシステムのデータは、ランタイムで変更できます。

変更内容は、サーバーから関係するすべてのクライアントに分散されます。

レポート

レポートは、サーバーのプロジェクトディレクトリで一元的に保存されます。レポートデータは、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)に分割されます。

一度に1台のクライアントのみがサーバー上のレポートシステムを設定できます。

レポートをローカルに保存して編集することもできます。この場合、それらを手動でサーバーと同期させる必要があります。

レポートはランタイムとは独立して実行できるため、ランタイム時にログシステムへの変更を行うことはできません。

スクリプト

スクリプトは、サーバーのプロジェクトディレクトリで一元的に保存されます。

プロジェクト固有のスクリプトは、コンピュータ固有ベースで個別に定義できます。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ーのアクションは画像に保存されます。

編集中、ファイル(スクリプトまたは画像)は他のクライアントに対してロックされます。

サーバーへの接続がない場合、スクリプトをローカルに変更することもできます。この場合、それらを手動でサーバーと同期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す。

サーバーは、変更内容を接続されているすべてのコンピュータに分配します。

注記

スクリプトをローカルに編集

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、スクリプトを適切なサーバー側のディレクトリ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からのテキスト

テキストライブラリからのテキストは、サーバーデータベースに保存されています。

1.9 リモートコンフィグレーション

テキストオブジェクトは個別に保存されます。

テキストは、ランタイムでは変更できません。

サーバーは、変更内容を接続されているすべてのコンピュータに分配します。更新は、ローカルで設定された構成言語で実行されます。

注記

異なるエディタからのアクセス

アラームロギングやユーザー管理者など一部の WinCC エディタは、設定中、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。

そのため、これらのエディタを 1 台のオペレータステーション上だけで同時に編集することができます。

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

ランタイムでの更新

クライアントプロジェクトでのタグの変更が無効化されている場合、変更時に有効であったプロジェクトがあるコンピュータをすべて再起動しないかぎり、その変更内容は有効となりません。

ユーザー管理者

ユーザー管理者のオペレータ権限は、サーバーデータベースに保存されます。

ユーザー管理者データは、ランタイム時に変更できます。

参加し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 98)

プロジェクトの終了方法 (ページ 100)

サーバープロジェクトピクチャの編集方法 (ページ 97)

編集用にプロジェクトを開く方法 (ページ 95)

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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

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。これには、そのモードが現在有効になっている (設定/ランタイム) 選択されたサーバーに関する詳細情報が含まれます。

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

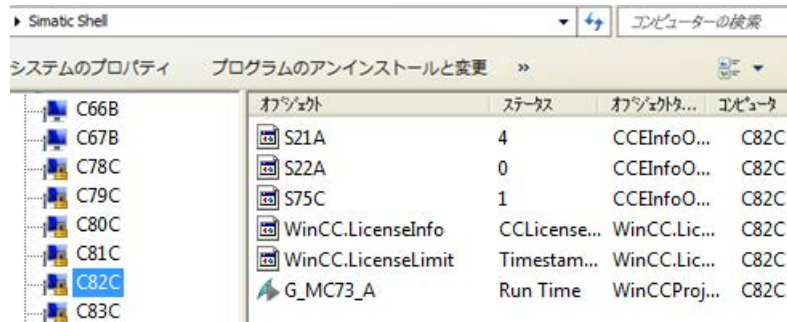
クライアントからサーバープロジェクトを開いてリモート編集する場合、以下の条件を満たす必要があります。

- 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。

1.9 リモートコンフィグレーション

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。
ナビゲーションウィンドウには、ネットワーク上で現在使用可能なすべてのサーバーおよびプロジェクトが含まれます。
独立したウィンドウで [Simatic Shell] ダイアログを開くには、Windows エクスプローラの [Simatic Shell] エントリのショートカットメニューで [新しいウィンドウで開く] 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。
3. プロジェクトリストで、開きたいプロジェクトのショートカットメニューから [開く] エントリを選択します。
ログインダイアログが開きます。
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。
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 100)

プロジェクトの実行方法 (ページ 98)

サーバープロジェクトピクチャの編集方法 (ページ 97)

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 92)

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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

1.9.6 サーバプロジェクトピクチャの編集方法

原理

サーバにある画像は、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。

必要条件

- サーバ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 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 - サーバは利用できません] ダイアログで、[ローカルでサーバを開始する] 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 100)

プロジェクトの実行方法 (ページ 98)

編集用にプロジェクトを開く方法 (ページ 95)

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 92)

1.9 リモートコンフィグレーション

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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

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。

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 100)

サーバープロジェクトピクチャの編集方法 (ページ 97)

編集用にプロジェクトを開く方法 (ページ 95)

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 92)

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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

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設定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。
ネットワーク上で利用可能なすべてのコンピュータとプロジェクトが、現在のステータスも含めて表示されます。
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。

注記

パスワードは、大文字と小文字を区別します。

結果

サーバーでプロジェクトが終了されます。

下記も参照

プロジェクトの実行方法 (ページ 98)

サーバープロジェクトピクチャの編集方法 (ページ 97)

編集用にプロジェクトを開く方法 (ページ 95)

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 92)

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 88)

リモートコンフィグレーション (ページ 83)

WinCC のクライアント/サーバーシステム (ページ 14)

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, 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 1.05.03

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 14)

OPC の機能 (ページ 392)

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

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

概要

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 V8 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 V8 と WinCC ファイルサーバー V8 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]ダイアログの[WinCC]グループで、[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設定 (ページ 109)

サービスプロジェクトの使用と制約 (ページ 110)

サービスプロジェクトの操作モード (ページ 116)

サービスプロジェクトを起動する方法 (ページ 124)

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

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 107)

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。

レポートの印刷

WinCC ServiceMode ではレポートの印刷はできません。

3.3 サービスプロジェクトの使用と制約

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 121)

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 107)

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

概要






WinCC は、トレイエリアと呼ばれるタスクバーの通知エリアに[SIMATIC WinCC]アイコンを表示します。

このアイコン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、このアイコン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]アイコンを示します。

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]アイコン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)

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

- [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]アイコンをクリックします。

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 124)

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。

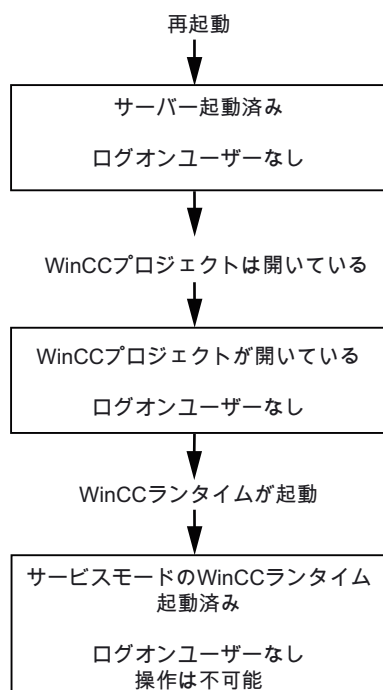
- オペレーティングシステムが起動した後、自動的に。
- ユーザーがログインして、WinCC ランタイムを起動した後に。

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。



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 117)

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 107)

サービスプロジェクトを起動する方法 (ページ 124)

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。ユーザーは、すべてのサーバーとクライアントで、「SIMATIC HMI」ユーザーグループのメンバーとして作成する必要があります。

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:

- Windows でパスワードを変更する場合、この変更内容をすべてのコンピュータおよび WinCC プロジェクトの設定に適用する必要があります。
- 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。

- [パスワードが期限切れにならない]
- 意図せずに変更されることを防ぐため、[ユーザーがパスワードを変更できない]オプションも無効にします。
矛盾を防ぐため、いずれの場合も、管理者としてパスワード更新を一元的に実行します。

コマンドラインからのユーザーとパスワードの変更

「CCStartStop.exe」アプリケーションは、Windows コマンドプロンプトのコマンドラインから WinCC プロジェクトにアクセスできます。

「/su」パラメータを使用して、WinCC プロジェクトで ServiceMode ユーザーおよび Windows パスワードで更新します。

例

新しい ServiceMode ユーザーの設定:

- CCStartStop /su /domain:plant011 /user:operator02 /password:MYpa\$\$w0rd
「plant011」ユーザーグループのユーザー「operator02」が、パスワード「MYpa\$\$w0rd」を持つ ServiceMode ユーザーとして採用されています。

設定された ServiceMode ユーザーのパスワードの更新:

- CCStartStop /su /password:NEWpa\$\$w0rd
ログインしたユーザーのパスワードを新しいパスワード「NEWpa\$\$w0rd」に変更されます。

詳細情報:

- [WinCC での作業]>[プロジェクトでの作業]>[Windows プロンプトを介して WinCC プロジェクトを管理]

下記も参照

サービスプロジェクトの操作モード (ページ 116)

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 120)

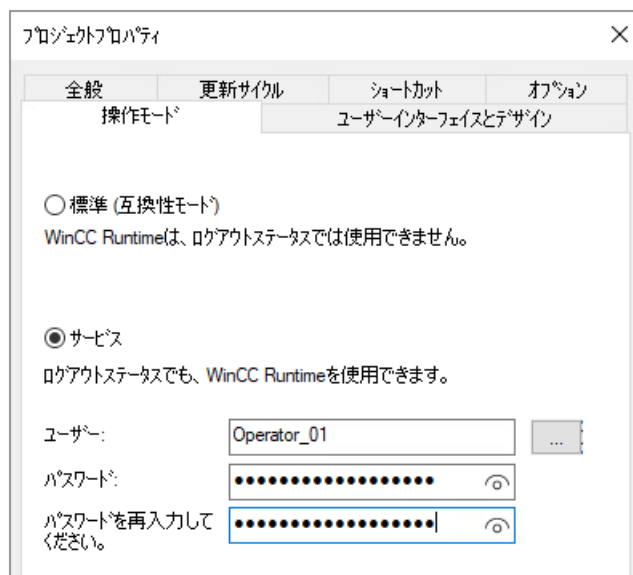
サービスプロジェクト用の Autostart の設定方法 (ページ 121)

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 117)」
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 110)」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 121)

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。

3.6 WinCC ServiceMode の設定

[AutoStart の設定]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。

[起動時のオペレーティングシステムアクセスをオフにする]オプション

このオプションが有効化されている場合、WinCC の開始画面が PC が起動されるとすぐに表示されます。Windows デスクトップは表示されません。

ただし、ServiceMode が起動されているときにユーザーがログインする場合、このオプションのみが ServiceMode で有効化されます。ログインするユーザーがない場合、このオプションを無効化して起動した場合と動作は同じになります。

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 120)

サービスプロジェクトの使用と制約 (ページ 110)

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 113)」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 117)」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。

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 117)

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 (ページ 126)

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 (ページ 126)

WinCC ServiceMode:標準プロジェクトとサービスプロジェクト (ページ 107)

サービスプロジェクトの操作モード (ページ 116)

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 113)

3.7 ランタイムのサービスプロジェクト

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。

下記も参照

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 (ページ 126)

サービスプロジェクトを起動する方法 (ページ 124)

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。

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。

下記も参照

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 (ページ 126)

サービスプロジェクトを起動する方法 (ページ 124)

3.7 ランタイムのサービスプロジェクト

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。

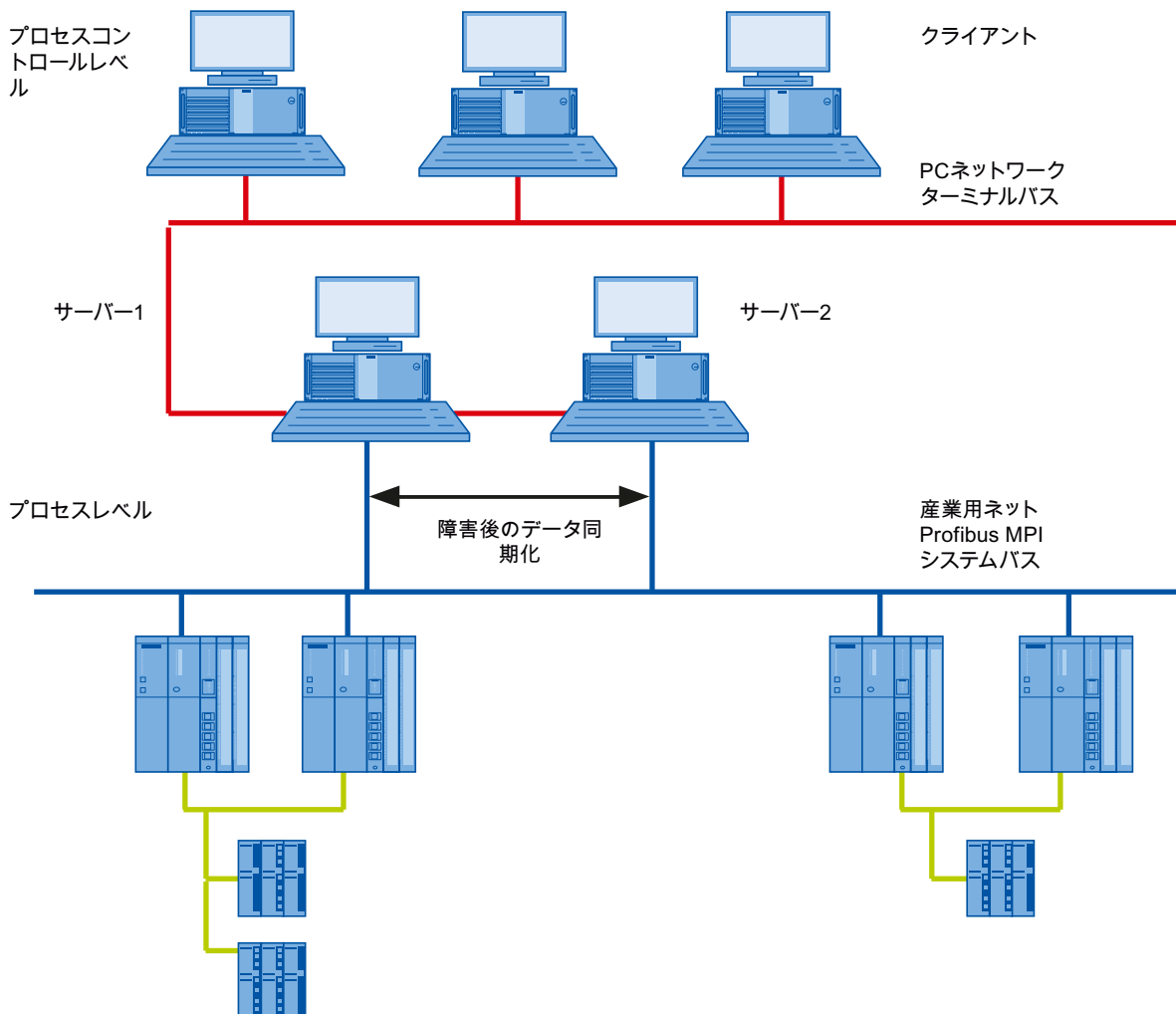
4.2 WinCC の 2 重化

概要

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。

- マスタサーバー
- スタンバイサーバー

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」。
- 「Application Health Check」ファンクションによる WinCC アプリケーションのモニタリング。
- 「SelfDiagnosis」ファンクションによるローカルシステムのハードウェアとソフトウェアのモニタリング。

「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。

ソフトウェアエラーが検出されると、ライフビートのモニタリングによって次のアクションがトリガされます。

- システムタグ"@RedundantServerState"で、サーバーの状態が「エラー」に変わります。
- 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。
- プロセスコントロールメッセージによってソフトウェアのエラーがユーザーに通知されます。
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

エラー後にサーバーが再起動します

「Application Health Check」ファンクションがソフトウェアエラーを検出し、クライアントの切り替えが開始された場合は、関連するサーバーを再起動する必要があります。サーバーを再起動した後でないと、クライアントをこの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。

「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。

- ローカルの 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。

下記も参照

WinCC リダンダントシステムメッセージ (ページ 173)

WinCC リダンダントシステムタグ (ページ 167)

故障シナリオ (ページ 158)

冗長システムのセットアップ方法 (ページ 140)

同一ファンクションのコンフィグレーション (ページ 142)

2重化の仕組み (ページ 135)

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とします。

4.3 冗長システムの必要条件

- 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
サーバーのシリアル接続のケーブルの長さを確認してください。最大ケーブル長は伝送速度によって異なります。
接続ケーブルの長さには通常、次の仕様が適用されます。

伝送速度(bps)	最大長さ(メートル)
2400	900 m
4800	300 m
9600	152 m
19200	15 m
57600	5 m
115200	<2 m

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。

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。

下記も参照

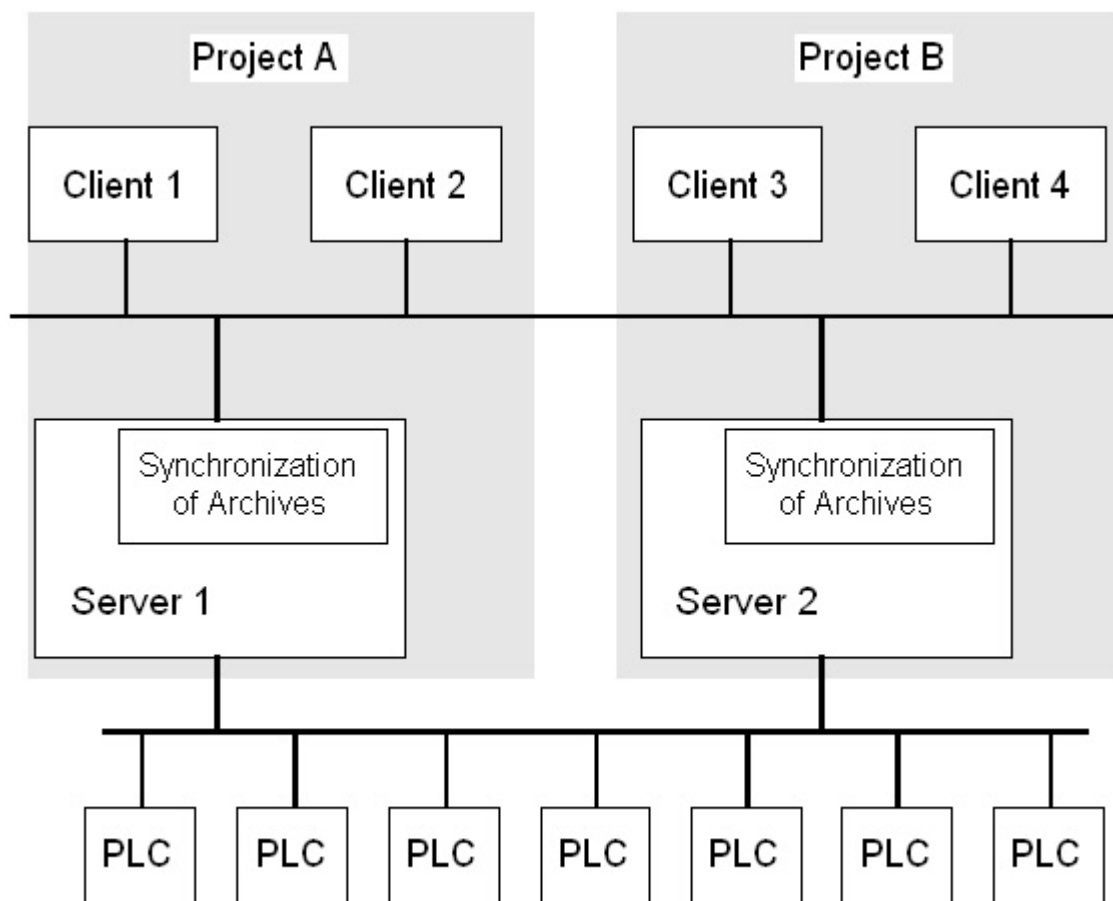
冗長サーバーを構成する方法 (ページ 143)

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、直ちにパートナー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。

下記も参照

WinCC リダンダントシステムメッセージ (ページ 173)

WinCC リダンダントシステムタグ (ページ 167)

故障シナリオ (ページ 158)

冗長システムのセットアップ方法 (ページ 140)

同一ファンクションのコンフィグレーション (ページ 142)

WinCC の 2 重化 (ページ 130)

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

故障シナリオ (ページ 158)

WinCC の 2 重化 (ページ 130)

同一ファンクションのコンフィグレーション (ページ 142)

冗長サーバーを構成する方法 (ページ 143)

ユーザーアーカイブの同期の設定方法 (ページ 147)

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。

2台のサーバーには同等のアーカイブがあり、追加の測定ポイントまたはアーカイブを追加する形式で追加できます。

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、ハードディスクにある次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になっ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。

機能的に同等の WinCC プロジェクトを確実に冗長化パートナーサーバー上で実行するには、毎回の変更後に、「プロジェクトデュプリケータ」を使ってプロジェクトを複製します。

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。

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 130)

冗長システムのセットアップ方法 (ページ 140)

4.5.3 冗長サーバーを構成する方法

概要

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ーとアーカイブの同期を設定します。

4.5 冗長性システムの構成

必要条件

- 2つの冗長性サーバーの機能が等しくなるように設定すること。

Simatic Shell:冗長パートナーへの接続の設定とテスト

「Simatic Shell」のショートカットメニューを使用して、[冗長性設定]ダイアログを開きます。

このダイアログは、冗長パートナーサーバーへの接続を設定する別の方法を提供します。

開いている場合、「冗長性」エディタからの既存の設定が採用されます。

接続設定のテスト

[ネットワーク接続の拡張チェック(ターミナルバス)]を使用して、冗長パートナーおよび個々のクライアントへの接続をテストします。

- 「デフォルトゲートウェイ」の利用可能性
- ステーションの利用可能性

接続をテストする PC の名前または IP アドレスを入力します。

名前と IP アドレスは、スペースなしでセミコロンで区切ります。次に例を示します:

「hostsv1;hostsv2;123.456.78.9」

利用可能性をテストするには、[チェック]ボタンをクリックします。

[ターミナルバス情報]ダイアログに、選択した PC の結果が表示されます。

- 名前または IP アドレス
- Ping による利用可能性:
- 応答時間

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]タブに変更します。
[サーバー]フィールドは、WinCC の冗長性を設定している現在のコンピュータの名前を含みます。

The screenshot shows the 'Redundancy' configuration dialog box in WinCC, with the 'General' (全般) tab selected. The 'Server' (サーバー) field contains 'WCC-DEV'. The 'Default Master' (デフォルトマスタ) checkbox is unchecked. The 'Redundant Partner Server' (冗長パートナーサーバー) field contains 'SV_REDUND_07', with a 'Reference' (参照(B)...) button next to it. The 'Local Computer Settings' (ローカルコンピュータ設定) section includes a dropdown for 'Network Adapter Used for Redundant Partner Connection' (ネットワークアダプタを使用した冗長パートナーとの接続) set to 'None' (なし). Below this, there are radio buttons for 'Dynamic' (ダイナミック) and 'Static' (スタティック), with 'Dynamic' selected. A 'Port' (ポート) field is set to '1765'. There is also a dropdown for 'Serial Interface Used for Redundant Partner Connection' (シリアルインターフェイスを使用した冗長パートナーとの接続) set to 'COM1'. The 'Options' (オプション) section has several checkboxes: 'Synchronization of alarm logs after partner server returns online' (パートナーサーバーがオンラインに戻った後のアラームロギングの同期化) is checked; 'Synchronization of alarm logs after partner server returns online' (パートナーサーバーがオンラインに戻った後のアラームロギングの同期化) is checked; 'Online synchronization of alarm logs' (アラームロギングのオンライン同期化) is checked; 'Synchronization of logs after disconnection' (切断後の同期化) is unchecked; and 'Switching WinCC client when connection fails' (接続エラーの場合の WinCCクライアントの切り替え) is checked. At the bottom, there is a checkbox 'Enable redundancy' (冗長化を実行する) which is checked.

2. WinCC の冗長性を設定するには、ダイアログの下部にある[冗長性を有効化]オプションを選択します。
3. 起動時にデフォルトでサーバーをマスタとして有効化するには、[デフォルトマスタ]オプションを選択します。
このオプションを選択しないと、サーバーはスタンバイサーバーとして起動します。

<p>通知</p> <p>「デフォルトマスタ」に指定できるのは 1 台の冗長サーバーのみです</p> <p>[デフォルトマスタ]オプションが 2 台の冗長パートナーサーバーのうち 1 台でのみ有効になっていることを確認してください。</p> <p>両方のパートナーサーバーをデフォルトマスタとして設定すると、クライアントの冗長性切り替えで不具合が発生する可能性があります。</p>
--

4. パートナサーバーのコンピュータ名を入力します。
あるいは、[参照]をクリックします。
パートナーサーバーのポートの指定は書き込み禁止になっており、変更できません。

4.5 冗長性システムの構成

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。
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。
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。
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。サーバーのシリアル接続のケーブルの長さを確認してください。最大ケーブル長は伝送速度によって異なります。
あるいは、「Simatic Shell」を使用して接続設定を設定します。
 - 「Simatic Shell」フォルダのショートカットメニューから、Microsoft Windows Explorer の[冗長性設定]ダイアログを開きます。
 - シリアルインターフェース、ネットワークアダプタ、およびネットワークアドレスを選択します。
 - 必要な場合、拡張ネットワーク接続チェックにより接続設定をテストします。
6. リターンまたは障害時の同期動作を定義するには、必要なオプションを選択します。
 - 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、タグ接続のないメッセージ、および内部メッセージタグ付きのメッセージの同期
 - 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):
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 154)」で説明します。
7. [OK]をクリックして設定を保存します。
8. ランタイムで設定を適用するには、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 154)

ユーザーアーカイブの同期の設定方法 (ページ 147)

WinCC の 2 重化 (ページ 130)

WinCC リダンダントシステムメッセージ (ページ 173)

冗長システムのセットアップ方法 (ページ 140)

冗長システムの必要条件 (ページ 133)

故障シナリオ (ページ 158)

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。

注記

変更された設定データ:同期なしと変更のオンラインロード

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。

アーカイブ同期への変更後のランタイムの再起動

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

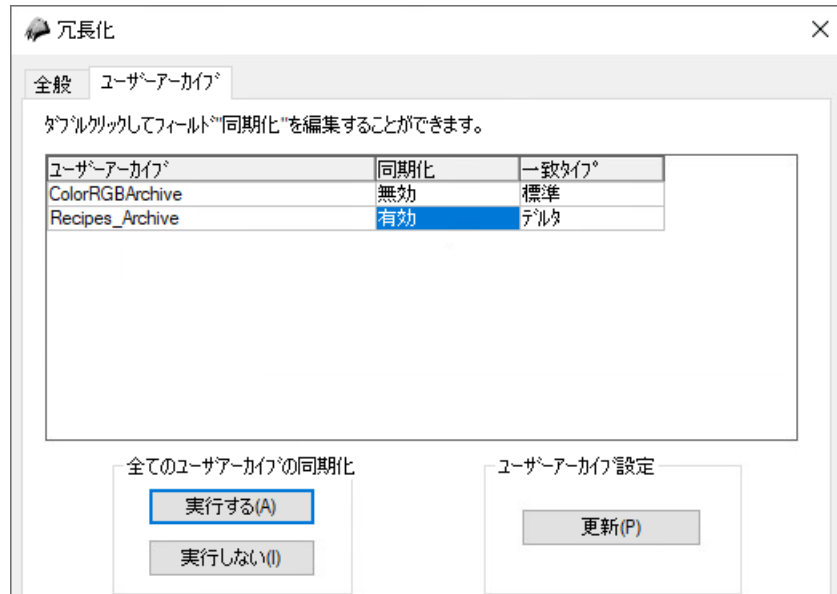
必要条件

- ユーザーアーカイブのコンフィグレーションは 2 台の冗長サーバーで同一であること。これにはプロジェクトデュプリケータを使用すること。

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。
[ユーザーアーカイブ]タブの[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。



2. [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を設定する場合、[更新]ボタンをクリックします。
ユーザーアーカイブの現在の設定が適用されます。
5. 100 000 以上のデータレコードを含む非常に大規模なユーザーアーカイブがプロジェクトに含まれる場合、差分同期を使用して、冗長化同期中のパフォーマンスを向上できます。
通常の設定制限の場合、デフォルトの設定をそのまま使用します。
[差分]オプションを有効にするには、[マッチングタイプ]列をダブルクリックします。
差分同期の要件:
 - 差分同期が冗長化パートナーサーバーでも有効になっているか確認してください。
両方のサーバーを一致させるため、[マッチングタイプ]オプションが設定されている必要があります。
 - ユーザーアーカイブは、対応するプライマリサーバー(マスター)でのみ設定します。
WinCC プロジェクトの各変更後、WinCC プロジェクトデuplicータを使用して冗長化パートナープロジェクトを生成します。
6. [OK]をクリックして設定を保存します。
7. 機能的に同等の WinCC プロジェクトを確実に冗長化パートナーサーバー上で実行するには、「プロジェクトデuplicータ」を使ってプロジェクトを複製します。

ユーザーアーカイブの並行編集

冗長化ユーザーアーカイブのデータレコードの並行挿入用の境界条件:

- 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。同期が完了しない場合、スクリプトまたはユーザーアーカイブコントロールでエラーメッセージを受け取ります。
- 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

両方のサーバーの障害:再起動シーケンス

両方の冗長化サーバーに障害があるか両方のコンピュータが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。

上記の順序を守らないと、変更内容が失われることがあります。

下記も参照

冗長サーバーを構成する方法 (ページ 143)

WinCC の 2 重化 (ページ 130)

冗長システムのセットアップ方法 (ページ 140)

WinCC リダンダントシステムメッセージ (ページ 173)

WinCC リダンダントシステムタグ (ページ 167)

故障シナリオ (ページ 158)

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、同一のハードウェアおよびソフトウェア機能を使用して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 プロジェクトデュプリケータを使用して冗長化パートナープロジェクトを生成します。

4.5 冗長性システムの構成

プロジェクトデuplicエ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化パートナーにコピーする。
- 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uplicエ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。

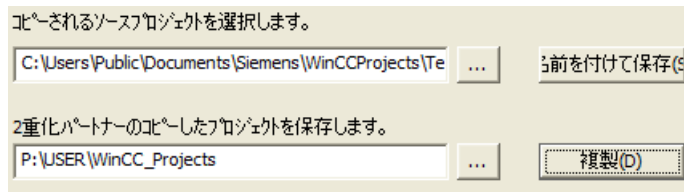
必要条件

- 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。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. [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。例:
 - コンピュータ名
 - 冗長化エディタの設定
 - 他のエディタの設定
 - 自動起動が WinCC プロジェクトで設定されているときの自動起動設定

4.5 冗長性システムの構成

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。

下記も参照

ランタイムでの冗長プロジェクトの複製方法 (ページ 152)

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。

プロジェクトデュプリケ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、プロジェクトデュプリケータを使用して、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レータを起動します。
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。

下記も参照

冗長サーバーのプロジェクトの複製方法 (ページ 149)

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。

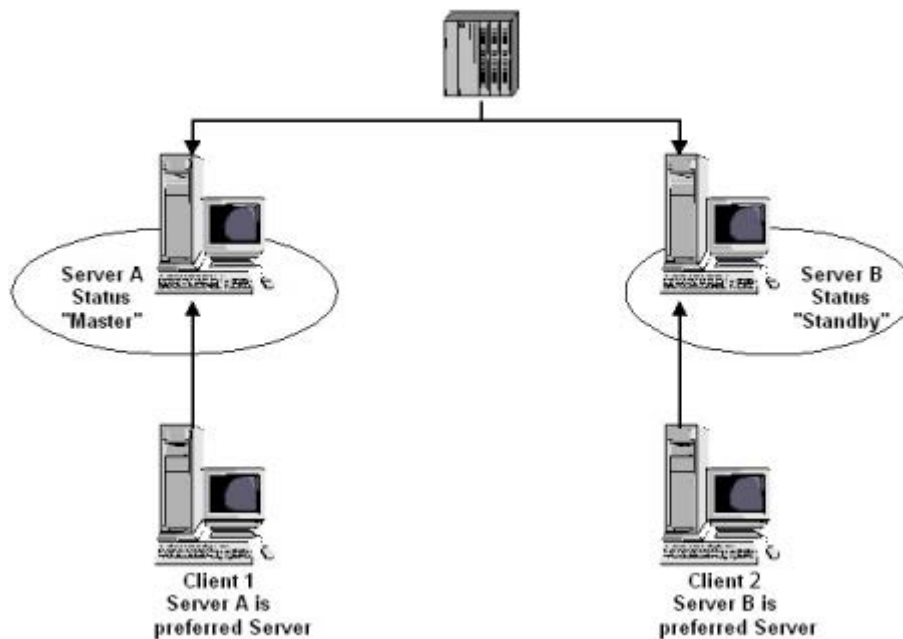
「マスタ」サーバーの論理接続不具合数が冗長パートナーサーバーよりも多い場合、サーバーのステータスは「@RedundantServerState」システムタグで「エラー」になります。クライアントの接続は、「マスタ」ステータスを持つ冗長パートナーサーバーに切り替えられます。

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

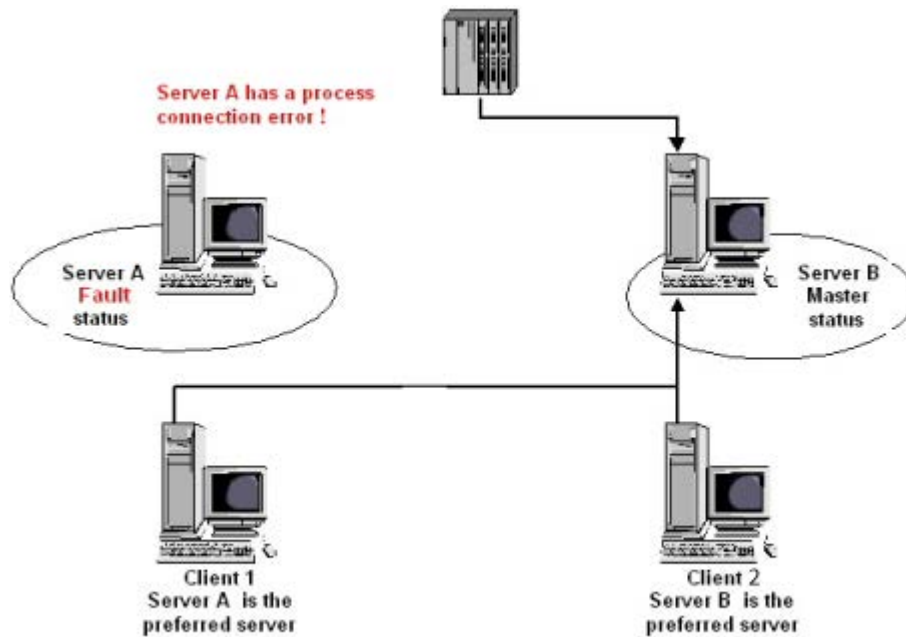
- 優先サーバー A があるクライアント 1
- 優先サーバー B のあるクライアント 2



4.6 WinCC 冗長性のシナリオ

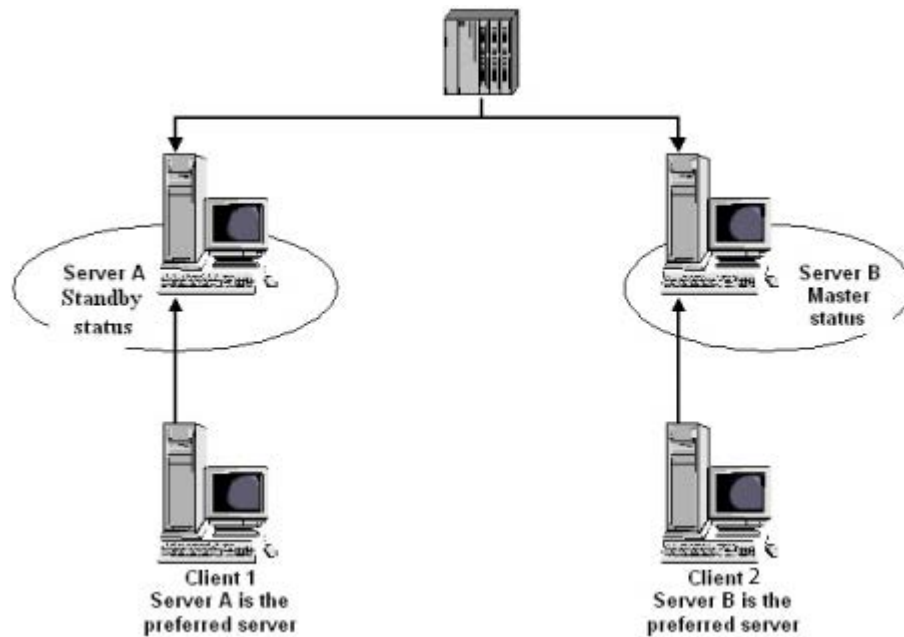
サーバー A のプロセス接続エラー

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。

この制限は、OPC UA には適用されません。

下記も参照

故障シナリオ (ページ 158)

冗長システムのセットアップ方法 (ページ 140)

WinCC の 2 重化 (ページ 130)

冗長サーバーを構成する方法 (ページ 143)

ユーザーアーカイブの同期の設定方法 (ページ 147)

故障シナリオ (ページ 158)

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 159)
2. シナリオ 2 パートナサーバーへの接続エラー (ページ 162)
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 164)
4. シナリオ 4 プロセス接続の不良 (ページ 165)
5. シナリオ 5 ソフトウェアエラー (ページ 166)

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 167)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。

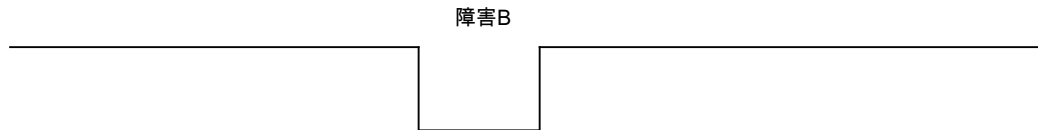
- サーバー 1 が、サーバー 2 の復帰時間(日付と時刻)を保存します。
- サーバー 1 は、サーバー 2 の復帰をシステムメッセージによってレポートします。
- サーバー 1 からの以下のアーカイブの冗長同期が、サーバー 2 で実行されます。
 - メッセージアーカイブ
 - プロセスデータアーカイブ
 - ユーザーアーカイブ
- サーバ 1 「@RM_MASTER」が設定されたまま、サーバ 2 「@RM_MASTER」がリセットされます。「@RM_MASTER_NAME」および「@RedundantServerState」はどちらのサーバーでも変更されません。
- 優先サーバーとしてサ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

下記も参照

WinCC リダンダントシステムタグ (ページ 167)

シナリオ 5 ソフトウェアエラー (ページ 166)

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 164)

シナリオ 2 パートナサーバーへの接続エラー (ページ 162)

故障シナリオ (ページ 158)

シナリオ 4 プロセス接続の不良 (ページ 165)

4.6 WinCC 冗長性のシナリオ

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、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 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 167)

シナリオ 5 ソフトウェアエラー (ページ 166)

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 164)

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 159)

故障シナリオ (ページ 158)

シナリオ 4 プロセス接続の不良 (ページ 165)

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。

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 167)

シナリオ 2 パートナサーバーへの接続エラー (ページ 162)

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 159)

故障シナリオ (ページ 158)

シナリオ 4 プロセス接続の不良 (ページ 165)

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。

4.6 WinCC 冗長性のシナリオ

- 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。

下記も参照

シナリオ 5 ソフトウェアエラー (ページ 166)

WinCC リダンダントシステムタグ (ページ 167)

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 164)

シナリオ 2 パートナサーバーへの接続エラー (ページ 162)

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 159)

故障シナリオ (ページ 158)

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。
- 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 167)

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 164)

シナリオ 2 パートナサーバーへの接続エラー (ページ 162)

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 159)

故障シナリオ (ページ 158)

シナリオ 4 プロセス接続の不良 (ページ 165)

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。

4.6 WinCC 冗長性のシナリオ

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。

詳細情報:

- [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

システムタグ	意味
@RM_UA_ONL_"Archiv name"	<p>診断に使用されます:</p> <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マッチングが正常に完了すると、スタンバイサーバーで設定されます。 プライマリサーバー(マスタ):スタンバイサーバーからのフィードバックメカニズムがないため、値は[1]のままです。 両方のサーバーで同じ値を得るためには、タグ同期を有効にしてください。 <p>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。</p>
@RM_Offline_UA_Name	<p>診断に使用。</p> <p>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。</p>

4.6 WinCC 冗長性のシナリオ

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	同期ステータス: <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。 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。

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、どちらのサーバーがプライマリサーバーとなるかを決定します。</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。接続ステータスまたはサーバーの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。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>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>

4.6 WinCC 冗長性のシナリオ

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ンまたは冗長化操作なし
@PRF_REDUNDANCY_PARTNER_CURRENT_STATE	<p>冗長化パートナーサーバーの冗長性ステータス</p>
@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。冗長性の切り換えはできません。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	<p>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。</p>
@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>

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。

詳細については、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 173)

ユーザーアーカイブの同期の設定方法 (ページ 147)

<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。

システムアラームを使用するには、システムメッセージの[アラームロギング]エディタの[使用]列を有効化します。

4.6 WinCC 冗長性のシナリオ

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。

番号	WinCC メッセージテキスト/説明
1012200	REDRT:パートナステーションで障害が発生しました パートナサーバーで WinCC が終了しました。
1012201	REDRT:パートナステーションがオンラインに戻りました パートナサーバーで WinCC が再起動しました。
1012202	REDRT:プロジェクトのファンクションが同一ではありません
1012203	REDRT:アーカイブの同期が失敗しました
1012204	REDRT:冗長性で内部エラーが発生しました
1012205	REDRT:パートナ接続に問題があります パートナサーバーへの接続に問題があります。
1012206	REDRT:パートナへの接続が再度確立されました パートナサーバーへの接続がリストアされました。
1012207	REDRT:パートナサーバー WinCC が起動されていません スタートアップ時に、WinCC が起動されていないことが特定されました。
1012208	REDRT:アーカイブの同期化が始動されました アーカイブの同期開始時にこのメッセージは出力されます。
1012209	REDRT:アーカイブの同期化が終了しました アーカイブの同期終了時にこのメッセージは出力されます。
1012210	REDRT:タグロギングが同期中です
1012211	REDRT:タグロギングの同期が完了しました
1012212	REDRT:アラームロギングが同期中です
1012213	REDRT:アラームロギングの同期が完了しました
1012214	REDRT:ユーザーアーカイブを同期中です。
1012215	REDRT:ユーザーアーカイブの同期が完了しました
1012216	REDRT:アーカイブの同期がキャンセルされました 別の障害により同期が中断されました。
1012217	REDRT:パートナサーバープロジェクトが有効化されていません スタートアップ時に、WinCC がパートナサーバーで実行されていないこと、またはランタイムでないことを、システムが検出しました。

番号	WinCC メッセージテキスト/説明
1012218	SWITCH:自動的にクライアントが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へ自動的に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へ手動で切り替えられました
1012220	UA:すべてのユーザーアーカイブに対して同期が有効になっています 冗長性同期がすべてのアーカイブに対して有効な場合、このメッセージはインポート中に出力されます。
1012221	UA:すべてのユーザーアーカイブに対して同期が有効になっていません 冗長性同期が少なくとも1つのアーカイブに対して無効な場合、このメッセージはインポート中に出力されます。
1012226	REDRT:パートナーサーバープロジェクトが有効化されています スタートアップ時に、WinCC がパートナーサーバーで有効化されていることを、システムが検出しました。
1012227	REDRT:エラー:パートナーサーバーがサーバーではありません スタートアップ時に、設定済みのパートナーサーバーがサーバーでないことを、システムが検出しました。
1012228	REDRT: CAS:アーカイブの同期化が[@2%#@]を始動しました
1012240	REDRT:<アプリケーション名>のエラー<エラーの説明>によりステータススイッチが生じます。 上記のアプリケーション内のエラーのために、アプリケーションのヘルスチェックにより、切り替えが行なわれました。
1012241	REDRT:ステータス<ステータスの指定>へ切り替えられました ステータス変更が通知されました。
1012244	REDRT:アラームロギングのオンライン更新中に過負荷が発生しました 同期するメッセージが多すぎます。
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:<メッセージが作成されたマシン名>:OS サーバー(マスタ)<マシン名>OS サーバー(スタンバイ)<マシン名>冗長化エラー。 故障シナリオによって、マスタサーバーとスタンバイサーバーの両方またはどちらか一方が、冗長化エラーを報告します。 冗長化が損なわれる可能性があります。

4.6 WinCC 冗長性のシナリオ

番号	WinCC メッセージテキスト/説明
1012248	REDRT:<メッセージが作成されたマシン名>: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>冗長化を復元済み
1012349	冗長制御:MAC アドレス<アドレス>のあるネットワークカードによる接続が失われました 冗長 LAN によるパートナーサーバーへの接続が中断されたか、切断されました。
1012350	冗長制御:MAC アドレス<アドレス>のあるネットワークカードによる接続が再度確立されました。 冗長 LAN によるパートナーサーバーへの接続がリストアされました。
1012351	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	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。
1012354	冗長制御:ステータスが FAULT に変更されました。しかし、サーバー分離が有効になっていません。
1012355	冗長制御:ステータスが FAULT に変更されました。しかし、サーバー分離が<名前>によりロックされています。理由:<理由>
1012356	冗長制御:故障状況が変化しました => サーバーが隔離されました
1012357	冗長制御:ステータスが FAULT に変更されました。しかし、自動再起動が有効になっていません。
1012358	冗長制御:ステータスが FAULT に変更されました。しかし、自動再起動がロックされていません。ネットワークアダプタが外され、DHCP がリリースされました。
1012359	冗長制御:コンピュータの再起動が<名前>により無効化されています。理由:<理由>
1012360	冗長制御:コンピュータの再起動が中断されています。最後の再起動は<秒数>秒前に行われました。
1012361	冗長制御:コンピュータの再起動が中断されています。<再起動の回数>回の再起動後の<秒数>秒間は、追加の再起動が許可されていません。
1012362	冗長制御:<秒数>秒以内にコンピュータを再起動します
1012700	自己診断:ステーション<ステーション名>の値<値>が無効です。
1012701	自己診断:ステーション<ステーション名>の値<値>がエラー上限に準拠していません。

番号	WinCC メッセージテキスト/説明
1012702	自己診断:ステーション<ステーション名>の値<値>がエラー下限に準拠していません。
1012703	自己診断:ステーション<ステーション名>の値<値>が警告上限に準拠していません。
1012704	自己診断:ステーション<ステーション名>の値<値>が警告下限に準拠していません。
1012705	自己診断:ステーション<ステーション名>の値<値>がエラー制限に準拠するようになりました。
1012706	自己診断:ステーション<ステーション名>の値<値>が適切です。
1012707	自己診断:ステーション<ステーション名>が@2%s@を生じさせます。
1012708	自己診断:ステーション<ステーション名>の値<値>が無効です。

下記も参照

WinCC リダンダントシステムタグ (ページ 167)

WinCC の証明書

5.1 WinCC の証明書の概要

証明書の目的

証明書は、設備、システム、ネットワークの通信をサイバー脅威から保護します。

証明書の目的:

- 認証
通信相手とのイニシャルハンドシェイク中に、デバイスは証明書を通信パートナーに送信します。通信パートナーが証明書を信頼する場合、接続が確立されます。

詳細情報:

- 通信パートナー (ページ 182)
- 信頼関係の作成: (ページ 183)

- 送信データの暗号化と復号化
接続が確立されると、通信パートナーはセッションキーを生成し、イニシャルハンドシェイク中にデバイスから送信された証明書の公開キーを使用してそれを暗号化します。暗号化されたセッションキーをデバイスに送信します。デバイスは、証明書の秘密キーを使用してセッションキーを復号化します。
デバイスは、証明書の秘密鍵を使用してセッションキーを復号化します。デバイスと通信パートナーは、このセッションキーを使用して通信を暗号化または復号化します。

証明書のタイプ

WinCC は、CA ベースの証明書と自己署名済みの証明書の使用をサポートしています。

注記

推奨事項:CA ベースの証明書

セキュリティ上の理由により、CA ベースの証明書の使用が推奨されます。

CA ベースの証明書

CA ベースの証明書は、認証機関(CA)によって署名されます。証明書に署名することにより、認証機関は証明書の信頼性と、その所有者(主体)の身元を保証します。

CA ベースの証明書は固有のものであり、システムに最高レベルの保護を提供します。

5.1 WinCC の証明書の概要

自己署名証明書

自己署名済みの証明書は、自身の署名によってその有効性を確認します。

自己署名済みの証明書は、CA ベースの証明書よりも安全性が低くなります。これらの使用には多くの制限があります。

WinCC PC 上のサードパーティの証明書

証明書で保護された通信では、次のデバイスがアプリケーション証明書を WinCC PC に送信します。

- TIA Portal で証明書が設定されている、ファームウェア 4.5 以降の S7-1200 シリーズおよびファームウェア 2.9 以降の S7-1500 の PC 接続 S7 PLC
- OPC UA サーバー設定で証明書ベースの通信のセキュリティポリシーが設定されている場合の PC の OPC UA 通信パートナー

これらのデバイスは、WinCC PC の通信パートナーです。

証明書は CA ベースまたは自己署名にすることができます。

WinCC 証明書

WinCC PC には、次の WinCC アプリケーション証明書(独自の証明書)を設定できます。デバイスはそれを通信パートナーに送信します。

証明書は CA ベースまたは自己署名にすることができます。

証明書	CA ベース	自己署名
WebUX WebNavigator 証明書	<p>「WinCC Certificate Manager」ツールを使用しています:</p> <ul style="list-style-type: none"> • 認証機関デバイスに証明書を作成するには • WinCC PC に証明書をインストールするには <p>詳細情報:</p>	<p>自己署名済みの証明書は、WinCC PC のインストール時に作成するか、その後 WebConfigurator または WebUX Configuration Manager で作成します。</p> <p>詳細情報:</p> <ul style="list-style-type: none"> • [オプション] > [WinCC/WebUX] > [WinCC Web コンフィギュレータ]
OPC UA サーバー証明書	<ul style="list-style-type: none"> • [WinCC Certificate Manager] > [証明書を使用可能にする (ページ 198)]。 	<p>自動的に生成された自己署名済みのデフォルト証明書を使用します。</p> <p>詳細情報:</p>
OPC UA クライアント証明書		<ul style="list-style-type: none"> • WinCC の証明書 > WinCC OPC UA 証明書の基礎 (ページ 238)
OPC UA クライアント証明書		
REST Service 証明書		-

下記も参照

WinCC Certificate Manager (ページ 186)

WinCC OPC UA 証明書の基礎 (ページ 238)

通信パートナー (ページ 182)

信頼関係の作成: (ページ 183)

証明書を使用可能にする (ページ 198)

S7 PLC との通信用証明書 (ページ 223)

5.2 基礎

5.2.1 通信パートナー

一方向認証と相互認証

証明書で保護された通信では、一方向認証と相互認証が区別されます。

一方向認証

証明書が必要なのは通信パートナー 1 名のみです。この通信相手は、接続が確立されると、その証明書を通信相手に送信します。通信パートナーが証明書を信頼する場合、接続が確立されます。

相互認証

両方の通信パートナーに証明書が必要です。接続を確立するときに証明書を交換します。通信パートナーが互いの証明書を信頼する場合、接続が確立されます。

WinCC PC の通信パートナー

WinCC PC は証明書を介して次のデバイスと通信を行います。

一方向認証

- WebUX サーバーまたは WebNavigator サーバーとして:クライアントの場合
PC は WebUX | WebNavigator 証明書をクライアントに送信します。
- 通信チャネル「SIMATIC S7-1200、S7-1500」経由:ファームウェア 4.5 以降の S7-1200 シリーズまたはファームウェア 2.9 以降の S7-1500 シリーズの接続された S7-PLC の場合
PLC は S7 証明書を PC に送信します。

相互認証

OPC UA 通信において:

- OPC UA サーバーとしての WinCC PC:OPC UA クライアント
WinCC PC は OPC UA サーバー証明書をクライアントに送信します。クライアントは、OPC UA クライアント証明書を PC に送信します。
- タグのインポート中の WinCC ES:WinCC OPC UA サーバーの場合
サーバーは、OPC UA サーバー証明書を WinCC ES に送信します。WinCC ES は、OPC UA タグインポートツールの証明書をサーバーに送信します。
- OPC UA クライアントとしての WinCC RT:WinCC OPC UA サーバーの場合
サーバーは OPC UA サーバー証明書を PC に送信し、PC は OPC UA クライアント証明書をサーバーに送信します。

必要条件:OPC UA サーバー設定では、証明書ベースの通信のセキュリティ ポリシーが設定されます。

下記も参照

信頼関係の作成: (ページ 183)

5.2.2 信頼関係の作成:

概要

証明書で保護された通信では、通信パートナーはイニシャルハンドシェイク中にアプリケーション証明書を交換します。

一方向認証では、1つのデバイスが証明書を送信します。双方向認証では、両方のデバイスが証明書を送信します。

アプリケーション証明書を受信したデバイスは、証明書によって通信が保護されているアプリケーションの証明書ストアをチェックします。

一方向認証	デバイス上のアプリケーションがアプリケーション証明書を信頼すると、接続が確立されます。 例:WebUX WebNavigator の通信
相互認証	接続は、一方のデバイス上のアプリケーションが他方のデバイスのアプリケーション証明書を信頼する場合に確立され、その逆も同様です。 例:OPC UA 通信:

CA ベースの証明書の使用

信頼関係のチェック

デバイスは、通信パートナーから送信されたアプリケーション証明書を発行した認証機関のルート証明書が、アプリケーションの証明書ストアで「trusted」ステータスになっているかどうかをチェックします。

はい デバイスがアプリケーション証明書を信頼しています。
接続が確立されます。

いいえ デバイスがアプリケーション証明書を信頼していません。
接続が拒否されています。
次回の試行で接続が成功するように、信頼関係を手動で確立します。

CA ベースの証明書による信頼関係の確立

証明機関のルート証明書を、アプリケーションの証明書ストアにあるデバイスに手動で一度インストールします。そうすると、デバイスが、認証機関が発行しているすべてのアプリケーション証明書を信頼します。

WinCC PC での手順:

- OPC UA 通信:
[WinCC の証明書] > [WinCC Certificate Manager] > OPC UA 通信における信頼関係の確立 (ページ 240)
- WebNavigator | WebUX の通信:
[WinCC の証明書] > [WinCC Certificate Manager] > WebUX/WebNavigator クライアントでルート証明書をインストール (ページ 242)
- S7 通信:
[WinCC の証明書] > [WinCC Certificate Manager] > S7 通信における信頼関係の確立 (ページ 232)

WinCC PC の通信パートナーでの手順:通信パートナーのユーザーヘルプで説明されているように進んでください。

自己署名済みの証明書を使用

信頼関係のチェック

デバイスは、アプリケーションの証明書ストア内のアプリケーション証明書が「trusted」ステータスであるかどうかをチェックします。

はい デバイスがアプリケーション証明書を信頼しています。
接続が確立されます。

いいえ デバイスがアプリケーション証明書を信頼していません。
接続が拒否されています。
次回の試行で接続が成功するように、信頼関係を手動で確立します。

自己署名済みの証明書での信頼関係の確立

デバイス上のアプリケーションの証明書ストアにアプリケーション証明書をインストールします。アプリケーションはアプリケーション証明書を信頼します。

WinCC PC での手順:

- OPC UA 通信:
[WinCC の証明書] > [WinCC Certificate Manager] > OPC UA 通信における信頼関係の確立 (ページ 240)
- WebNavigator | WebUX の通信:
[WinCC の証明書] > [WinCC Certificate Manager] > WebUX/WebNavigator クライアントでルート証明書をインストール (ページ 242)
- S7 通信:
[WinCC の証明書] > [WinCC Certificate Manager] > S7 通信における信頼関係の確立 (ページ 232)

WinCC PC の通信パートナーでの手順:通信パートナーのユーザーヘルプで説明されているように進んでください。

下記も参照

通信パートナー (ページ 182)

[セキュア通信]の設定 (ページ 223)

5.3 WinCC Certificate Manager

概要

「WinCC Certificate Manager」 ツールを使用して、WinCC PC 用の WinCC 証明書の作成、管理、インストールを行えます。

- 認証機関のルート証明書および CRL ファイル(CRL = 証明書失効リスト)
- アプリケーション証明書

また、Unified 認証機関の Unified 証明書を同時に管理するオプションもあります。管理オプションは、WinCC PC にインストールされている製品によって異なります。

- WinCC がインストールされていること
- WinCC と WinCC Unified Runtime がインストールされていること

注記

サードパーティのツールで証明書を作成

あるいは、次の証明書は、Certificate Manager の外部のサードパーティツールで作成することもできます。

証明書は自分で作成することも、外部の認証機関に依頼して作成してもらうこともできます。

- WinCC PC の CA ベースの OPC UA 証明書
これらの証明書を、PC の OPC UA 証明書ストア内の、独自の証明書があるフォルダにインポートします。
詳細情報: WinCC OPC UA 証明書の基礎 (ページ 238)
- REST 証明書
この証明書を PC の Microsoft Windows 証明書ストアにインポートします。次に、証明書を WinCC で選択したポートに接続します。
詳細情報: WinCC IT コネクタの設定方法 (ページ 514)

Certificate Manager を使用することを推奨します。

注記

外部 WinCC 認証機関なし

外部 WinCC 認証機関への接続および中間認証機関の使用は、Certificate Manager ではサポートされていません。

可用性

Certificate Manager は、WinCC がインストールされているすべてのデバイスで使用できます。

機能

Certificate Manager は、ネットワーク内で証明書を集中的に作成、管理、インストール、配布するために使用されます。

Certificate Manager を使用して、認証機関デバイスで次のタスクを実行できます。

WinCC 認証機関	WinCC Unified 認証機関
以下に基づく WinCC 認証機関の作成: <ul style="list-style-type: none"> • 秘密鍵 • 公開鍵(ルート証明書) • CRL ファイル(CRL = 証明書失効リスト) 	WinCC Unified Certificate Manager からエクスポートされた WinCC Unified 証明機関のインポート
PC 用の WinCC アプリケーション証明書の作成	Unified デバイス用アプリケーション証明書の作成
既存の証明書の再作成	既存の証明書の再作成
アプリケーション証明書とデバイスのルート証明書のインポートとインストール	アプリケーション証明書とデバイスのルート証明書のインポート アプリケーション証明書は、WinCC Unified Runtime が同時にインストールされている場合にのみインストールできます。 WinCC Unified Runtime がインストールされていない場合は、Unified Collaboration 証明書をインストールできます。
デバイスへ手動で配布するためのアプリケーションの証明書とルート証明書の暗号化されたエクスポート。	デバイスへ手動で配布するためのアプリケーションの証明書とルート証明書の暗号化されたエクスポート。
WinCC がインストールされている PC におけるデータのバックアップとリカバリのためのルート証明書、CRL ファイル、秘密鍵、ならびにすべてのデバイスの証明書の暗号化されたエクスポートとインポート	WinCC と WinCC Unified Runtime がインストールされている PC におけるデータのバックアップとリカバリのためのルート証明書、CRL ファイル、秘密鍵、ならびにすべてのデバイスの証明書の暗号化されたエクスポートとインポート。
PC の外部通信ピアへの配布のためのルート証明書と CRL ファイルのエクスポート	デバイスの外部通信ピアへの配布のためのルート証明書と CRL ファイルのエクスポート
PC と外部通信ピアへの配布のための更新済みの CRL ファイルのエクスポート	デバイスと外部通信ピアへの配布のための更新済みの CRL ファイルのエクスポート
アプリケーション証明書を公的証明書としてエクスポート	アプリケーション証明書を公的証明書としてエクスポート

5.3 WinCC Certificate Manager

WinCC アプリケーション証明書

Certificate Manager では、次の CA ベースの WinCC アプリケーション証明書を作成できます。

- WebUX | WebNavigator 証明書
- OPC UA サーバー証明書
- OPC UA クライアント証明書
- OPC UA Tags Importer 証明書
- REST Service 証明書

詳細情報

Certificate Manager の操作:

- 証明書を使用可能にする (ページ 198)

ユーザーインターフェースの構造とカスタマイズ:

- Certificate Manager のインターフェース (ページ 191)

下記も参照

信頼関係の作成: (ページ 183)

WinCC クラウドコネクタの MQTT 設定 (ページ 619)

5.3.1 基礎

5.3.1.1 認証機関

必要なアプリケーション証明書を発行するには、認証機関が必要です。[WinCC Certificate Manager] ツールで、認証機関を作成します。

注記

WinCC Unified Certificate Manager で作成された証明機関を WinCC Certificate Manager にインポートするオプションもあります。

認証機関とアプリケーション証明書を生成する PC は、認証機関デバイスです。

認証機関の設定全体

認証機関の設定全体には次のものが含まれています。

- 秘密鍵
認証機関は秘密鍵を使用してアプリケーション証明書に署名します。署名は、証明書が本物であることを保証します。
秘密鍵は、認証機関デバイスに残ります。
- 公開鍵/ルート証明書(CA 証明書)と CRL ファイル(証明書失効リスト)。
ルート証明書と CRL ファイルは、PC と外部通信ピアへ配布されます。

注記

このヘルプはランタイムルート証明書を使用しています。

- 認証機関とアプリケーション証明書に追加される PC。

CA コンテナ

CA コンテナ(認証機関コンテナ)には次のものが含まれます。

- ルート証明書
- CRL ファイル
- 認証機関に追加される PC のアプリケーション証明書

5.3.1.2 必要な証明書

WinCC Runtime で CA ベースの証明書を使用するには、次の証明書が必要です。

WinCC 証明書

- WinCC 認証機関のルート証明書および CRL ファイル
- 認証機関により PC に対して発行されている WinCC アプリケーション証明書

これらの証明書は、WinCC Certificate Manager により認証機関デバイスで作成し、配布します。

WinCC Certificate Manager でこれらの証明書を PC にインポートしてインストールします。

WinCC Certificate Manager は、PC と通信ピアの間の信頼関係を確立するのをサポートします。

5.3 WinCC Certificate Manager

WinCC ルート証明書と CRL ファイル

WinCC 認証機関を作成すると、WinCC ルート証明書とその CRL ファイルも自動的に作成されます。

WinCC アプリケーション証明書

PC に対して次のアプリケーション証明書が必要です。

- PC が Web サーバーの場合:WebUX | WebNavigator 証明書

注記

Web サーバー(IIS)も PC にインストールする必要があります。

- PC が OPC UA サーバーの場合:OPC UA サーバー証明書
- PC が OPC UA クライアントの場合:
 - OPC UA クライアント証明書
 - OPC UA サーバーの変数をインポートするには:OPC UA Tags Importer 証明書
- PC が REST インターフェースを使用している場合:REST Service 証明書

OPC UA 通信ピアの証明書

- 通信ピアの OPC UA アプリケーション証明書
- この証明書を発行した認証機関のルート証明書

通信ピアのユーザーヘルプで説明されているように、これらの証明書を作成してください。

下記も参照

証明書を使用可能にする (ページ 198)

OPC UA 通信における信頼関係の確立 (ページ 240)

5.3.1.3 パスワード要件

WinCC Certificate Manager のパスワードは次のような要件になっています。

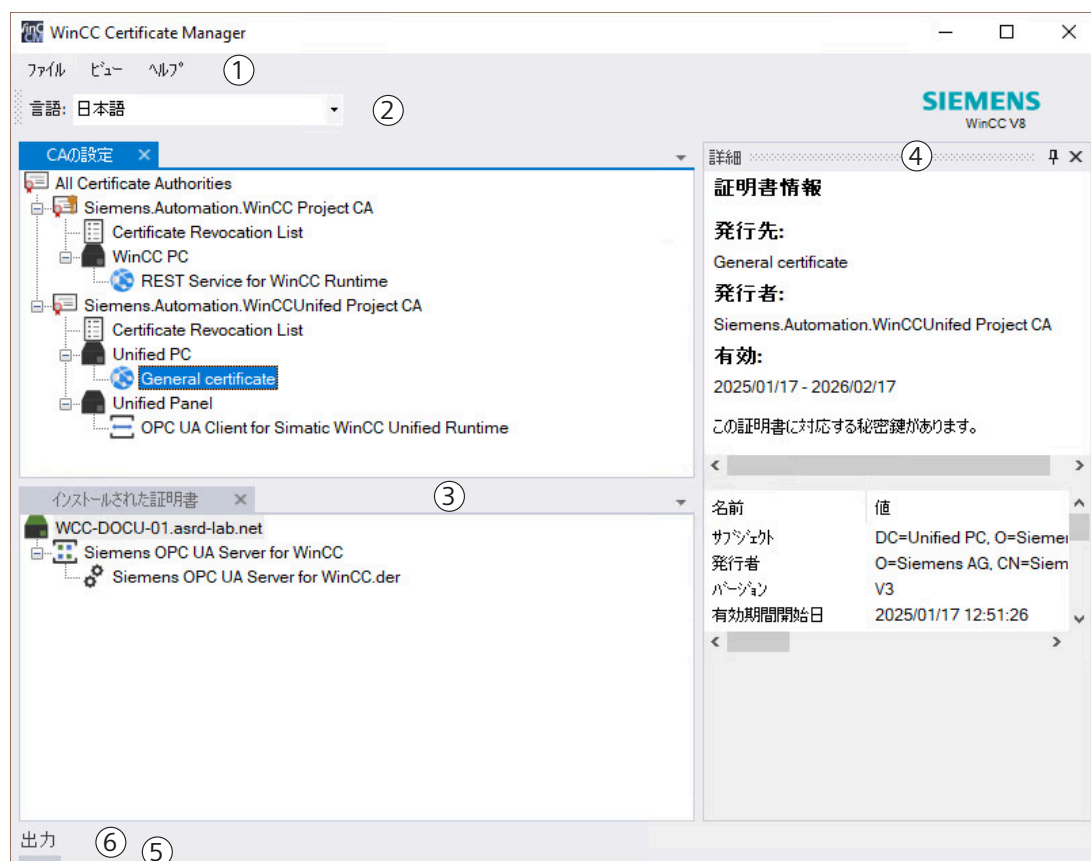
- 長さ:8 文字以上
- いずれの場合も、大文字、小文字、数字、特殊文字をそれぞれ 1 文字以上使用します

5.3.2 Certificate Manager のインターフェース

5.3.2.1 ユーザーインターフェースの構造

概要

WinCC Certificate Manager のユーザーインターフェースのレイアウトは次のようになっています。



- ① メニューバー
- ② ツールバー
- ③ [CA 設定]タブと[インストールされている証明書]タブの作業エリア
- ④ [詳細]エリア(固定)
[詳細]エリアは、作業エリアで選択した証明書の詳細を表示します。
- ⑤ 情報バー
- ⑥ [出力]エリア(非表示)
[出力]エリアは、オペレーターのコントロールアクションをログに残します。


5.3 WinCC Certificate Manager

インターフェースをカスタマイズして、要件に合わせることができます。

メニューバー

メニュー	説明
[ファイル]>[終了]	Certification Manager を閉じます。
[表示]	表示する Certification Manager のインターフェースエレメントを設定します。 次のユーザーインターフェースのエレメントを開くか、閉じます。 <ul style="list-style-type: none"> • [出力]エリア • [詳細]エリア • [CA 設定]タブ • [インストール済み証明書]タブ
[ヘルプ]	[Info Certificate Manager] インストール済みのソフトウェアのバージョンに関する情報を提供するダイアログが開かれます。

ツールバー

ボタン	
	ユーザーインターフェース言語を変更するには

作業エリアのタブ

[CA 設定]タブ (ページ 193)および[インストール済み証明書]タブ (ページ 194)を参照してください。

下記も参照

サーフェスのカスタマイズ (ページ 194)

5.3.2.2 [CA 設定]タブ

認証機関デバイスで

認証機関デバイスにおいて、[CA 設定]タブで認証機関を作成し、設定します。

- 認証機関とルート証明書を作成できます。
- PC を追加できます。
- PC に対してアプリケーション証明書を作成できます。
- 以下のエクスポートを実行できます。
 - 証明書の展開
 - データのバックアップ
- 証明書を再作成できます。
以下のオプションがあります。
 - ルート証明書の再作成
 - CRL ファイルの更新
 - PC の証明書設定の再作成
 - PC の個別のアプリケーション証明書の再作成
- 証明書機関デバイスが WinCC PC として使用されている場合:デバイスで PC のアプリケーション証明書をインストールします。

注記

タブの内容

Certification Manager の起動後に、証明書機関が Certificate Manager の最後の終了時に持っていたデータと同じデータが表示されます。

- データがまだ生成されていない場合、[設定を開く...]と[証明書機関を作成...]というノードが表示されます。
 - データがすでに生成されている場合は、次のように表示されます。
 - ルート証明書とその CRL ファイル
 - 設定された統合デバイスとそのアプリケーション証明書このデータを編集できます。
-

5.3 WinCC Certificate Manager

PC の場合

認証機関として使用されていない PC 上では、次のアクションを[CA 証明書]タブで実行します。

- PC に証明書設定をインポートします。
- 完全な証明書設定または個別のアプリケーション証明書をインストールします。
- インストールされた証明書を削除します。

注記

タブの内容

Certificate Manager の始動後に、[新しい設定を開く...]と[証明書機関を作成...]というノードが表示されます。

新しい証明書を開くと、次の画面が表示されます。

- 認証機関のルート証明書および CRL ファイル
- 証明機関で設定された PC とそのアプリケーション証明書

ローカル PC の証明書設定のみをインストールできます。他の PC の証明書設定は情報を提供する目的のみで表示されます。他の PC の証明書設定は変更できません。

Certificate Manager を閉じると、設定も閉じられます。

5.3.2.3 [インストール済み証明書]タブ

[インストール済み証明書]タブに、ローカル PC にインストールされているアプリケーション証明書が表示されます。

証明書をアンインストールするには、ローカル PC 上の証明書を削除します。

5.3.2.4 サーフエスのカスタマイズ

WinCC Certificate Manager のサーフェスの表示と手配は、次のように設定できます。

	閉じる/開く	移動	ドッキング切り離し/ドック	固定/固定解除	表示/非表示
[詳細]エリア	✓	✓	✓	✓	✓
[出力]エリア	✓	✓	✓	✓	✓
作業エリアのタブ	✓	✓	-	-	-

閉じる/開ける

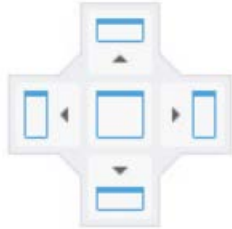
タブやエリアを閉じるには、[X]ボタンをクリックします。または、[表示]メニューでタブまたは領域のエントリを無効にします。

タブまたは領域を再度開くには、[表示]メニューでタブまたは領域のエントリを有効にします。

5.3 WinCC Certificate Manager

移動

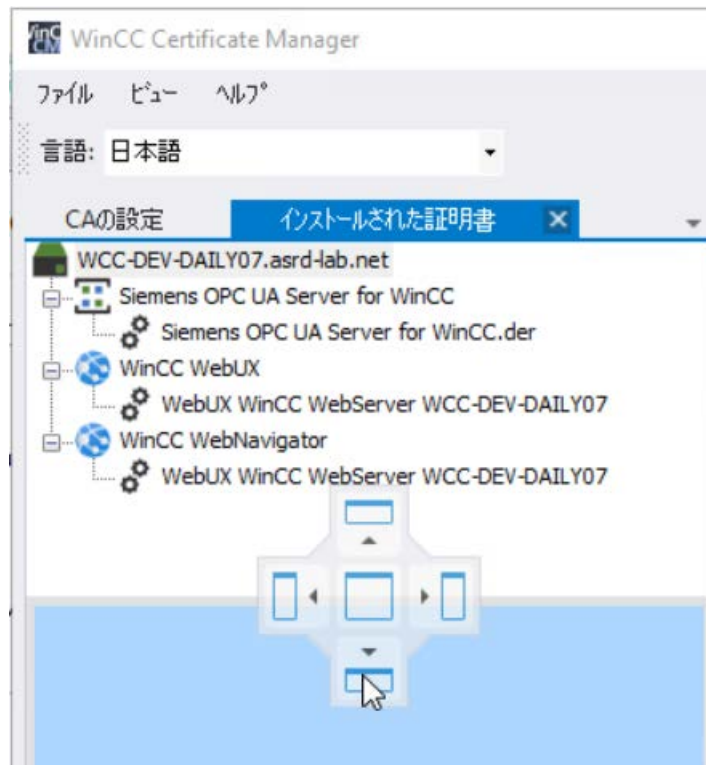
1. マウスの左ボタンを押した状態で、タブやエリアのタイトルバーをドラッグします。挿入できる場所はインターフェースに表示されます。



提供される挿入位置は次の条件によって異なります。

- タブまたは領域を移動する
- アプリケーションウィンドウにすでに表示されている要素

2. 新しい手配のプレビューを確認するには、マウスのカーソルをそのような場所のいずれかに移動して、マウスのカーソルを押した状態にします。



3. 希望の挿入場所でマウスのカーソルを開放します。タブやエリアが移動されます。



ドッキング切り離し/ドック

[詳細]または[出力]エリアのヘッダーを移動する場合、エリアのドッキング切り離しがアプリケーションウィンドウから行われ、スタンドアロンのウィンドウとして表示されます。ウィンドウを自由に移動できます。

エリアをアプリケーションウィンドウに再度ドックするには、提示されている挿入場所のいずれかに移動します。

固定/固定解除

次のボタンは、[詳細]エリアや[出力]エリアの固定や固定解除を行います。

ボタンの提示	ステータス	設定の変更
	固定 フォーカスされていない場合でも、エリアが表示されます。	ボタンをクリックして設定を切り替えます。
	フォーカスされなくなると、エリアが非表示になります。	

表示/非表示

必要条件

[詳細]エリアおよび[出力]エリアが固定されていません。

手順

エリアを表示するには、テキストをクリックします。エリアが表示されます。

エリアの外でマウスをクリックすると、自動的に非表示になります。

5.3.2.5 ユーザーインターフェース言語の変更

手順

1. ツールバーの矢印の付いたボタンをクリックします。



2. 希望する言語を選択します。

結果

ユーザーインターフェース言語が変更されます。

5.3.3 証明書を使用可能にする

WinCC Runtime の手順

WinCC Runtime で CA ベースの証明書を使用するには、次のステップを実行します。

1. どの PC を認証機関デバイスとして使用するか選択します。
2. この PC で認証機関を作成します (ページ 200)。
これにより、ルート証明書、CRL ファイル、秘密鍵を作成します。
3. 認証機関に PC を追加します (ページ 202)。
4. PC に必要なアプリケーション証明書を追加します (ページ 205)。
5. PC の証明書設定をエクスポートします (ページ 208)。
6. 各 PC に対して:
 - 証明書設定を PC にインポートします (ページ 210)。
 - PC の証明書設定、あるいは個別のアプリケーション証明書を、PC にインストールします (ページ 212)。

注記

ルート証明書のインストール

証明書設定、あるいは個別のアプリケーション証明書をインストールすると、ルート証明書と CRL ファイルが常に PC にインストールされ、信頼できるものとして分類されます。

7. PC と通信ピアの間の信頼関係を確立します。
詳細情報:
 - WebUX|WebNavigator の通信用 (ページ 242)
 - OPC UA 通信における信頼関係の確立 (ページ 240)
 - S7 の通信用 (ページ 232)

WinCC Unified Runtime の手順(WinCC Unified Runtime をインストールしない場合)

WinCC Certificate Manager を使用して CA ベースの統合証明書の証明書設定を管理するには、次のステップを実行します。

1. WinCC Certificate Manager で、WinCC Unified Certificate Manager を使用して作成された Unified 認証機関の設定(全体の設定)を開きます。(ページ 210)
2. 必要に応じて、Unified 証明機関の証明書構成を管理します。
WinCC Unified Runtime をインストールせずに Unified 証明書をインストールすることはできません。
例外:Unified Collaboration 証明書のインストールが可能です。

WinCC Unified Runtime の手順(WinCC Unified Runtime のインストールあり)

並行して Unified Runtime がインストールされている WinCC PC 上の WinCC Certificate Manager で PC 独自の CA ベースの Unified 証明書をインストールまたは削除するには、次のステップを実行します。

1. 必要に応じて、PC の更新された Unified 証明書設定を認証機関デバイスにエクスポートします。
2. PC で証明書の設定を開きます。
3. 必要な証明書をインストールまたはアンインストールします。

追加オプション

Certificate Manager は、次のオプションも提供します。

- 証明書の新規作成 (ページ 217)、例えば、有効期限が切れたため
- アプリケーション証明書を公的証明書としてエクスポート (ページ 216)
- 認証機関のデータバックアップの作成 (ページ 220)

下記も参照

WinCC 認証機関とルート証明書の作成 (ページ 200)

デバイスの追加 (ページ 202)

アプリケーション証明書の追加または削除 (ページ 205)

証明書設定のエクスポート (ページ 208)

証明書設定のインポート (ページ 210)

証明書設定または個別の証明書のインストール (ページ 212)

ルート証明書と CRL ファイルのエクスポート (ページ 214)

5.3 WinCC Certificate Manager

認証機関のバックアップ (ページ 220)

アプリケーション証明書を公的証明書としてエクスポート (ページ 216)

証明書の再作成 (ページ 217)

5.3.4 WinCC 認証機関とルート証明書の作成

必要条件

WinCC 認証機関をまだ作成していないこと。

手順

1. WinCC Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. 作業エリアで[すべての証明機関]を選択します。
4. コンテキストメニューを開き、[設定を作成]を選択します。
5. [新しい認証機関]ダイアログのルート証明書のプロパティを入力します。フィールドは自由に編集できます。
必須フィールド:
 - [名前]
 - 秘密鍵のパスワードフィールド必要に応じて、別の暗号鍵の長さとして証明書のランタイムを選択してください。
6. [作成]をクリックします。

結果

- 秘密鍵が生成されます。
- ルート証明書が生成されます。
- 空の CRL (証明書失効リスト) ファイルが生成されます。
- [CA 設定] タブで、ルート証明書のノードが作成され、その下に CRL ファイルのものが作成されます。

注記

秘密鍵は、認証機関デバイスでのみ利用可能です。認証機関は秘密鍵を使用して、PC のアプリケーション証明書を署名します。

ルート証明書と CRL ファイルは、PC の証明書設定に属します。証明書設定をエクスポートまたはインポートすると、ルート証明書と CRL ファイルもエクスポートまたはインポートされます。証明書設定が PC にインストールされると、ルート証明書と CRL ファイルは自動的にインストールされ、信頼できるものとして分類されます。

次のステップ

- 認証機関に PC を追加します。
- 外部通信ピアに対して行うときなど、証明書設定なしにルート証明書とその CRL ファイルを配布するには、ルート証明書と CRL ファイルをエクスポートします。

認証機関とルート証明書の削除**通知****データロスの防止**

認証機関とルート証明書は、次の場合のみ削除してください。

- 認証機関を保存した後。
- 認証機関とそのデータが必要なくなったとき。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. ルート証明書のコンテキストメニューで「削除」 エントリを選択します。

結果

認証機関とその設定全体はデバイスにより削除されます。

注記

証明書設定が PC にすでにインストールされている場合は、証明書はまだそこにインストールされています。PC で証明書をアンインストールします。

5.3 WinCC Certificate Manager

下記も参照

- デバイスの追加 (ページ 202)
- ルート証明書と CRL ファイルのエクスポート (ページ 214)
- パスワード要件 (ページ 190)
- アプリケーション証明書の追加または削除 (ページ 205)
- アプリケーション証明書のアンインストール (ページ 214)
- 認証機関のバックアップ (ページ 220)
- 証明書設定または個別の証明書のインストール (ページ 212)
- 証明書を使用可能にする (ページ 198)

5.3.5 デバイスの追加

必要条件

認証機関が、WinCC Certificate Manager の認証機関デバイスで作成されたかインポートされています。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
参照:
 - ルート証明書
 - CRL ファイル
 - すでに認証機関に追加されているすべてのデバイス
 - デバイスのアプリケーション証明書

3. ルート証明書で右クリックして、[デバイスを追加...]を選択します。
4. デバイスのデバイス名か IP アドレス、あるいはその両方を[新しいデバイス]ダイアログに入力します。
この情報は、デバイスのアプリケーション証明書が追加されたときに証明書に書き込まれます。これらは検証の際に使用されます。

注記

推奨される操作

デバイス名および IP アドレスを入力してください。

必要な入力

- ドメインのデバイスのデバイス名として完全修飾ドメイン名(FQDN)を使用してください。
これにより、Web ページにアクセスする際に検証エラーが防止されます。
- WebUX | WebNavigator 証明書のあるデバイスの場合は、アイデンティティプロバイダーのアドレス、および Web サーバーの Web ページを生成するために使用された情報を入力してください。
例:IP アドレスが使用されている場合は、IP アドレスの入力は必須です。デバイス名の入力はオプションですが、推奨されています。
- OPC UA サーバーや OPC UA クライアントとして使用されているデバイスの場合は、デバイス名を入力してください。
- ダイナミック IP アドレスを使用しているデバイスの場合は、デバイス名のみを入力してください。

注記

複数の IP アドレスの入力

[IP]フィールドに複数の IP アドレスを入力する場合は、[;]で区切ってください。デバイスの IP アドレスを最初の IP アドレス(固有の IP)として入力してください。

IP アドレスが、証明書のサブジェクト代替名に追加されます。

例:

HMI デバイスが OPC UA サーバーで NAT ルーターがあります。OPC UA クライアントは NAT ルーターを介してサーバーと通信を行っています。OPC UA サーバーの HMI デバイスのプライベート IP アドレス(固有の IP)、および Certificate Manager のパブリック IP アドレスを入力してください。

注記

許可されているデバイス名

FQDN のホスト名をデバイス名として使用できます。

「localhost」という名前を使用することはできません。Certificate Manager は、「localhost」をローカルデバイスのデバイス名に自動的に置き換えます。

注記

CA インフラストラクチャからデバイスを削除して、再度追加することによってのみ、この設定を後で変更することができます。この結果として、デバイスのアプリケーション証明書を再度追加し、配布し、インストールする必要があります。

5.3 WinCC Certificate Manager

結果

デバイスのノードが[CA 設定]タブで生成されます。

デバイスノードのアイコン:



ローカルマシン(追加されている場合)



その他のデバイス

次のステップ

必要なアプリケーション証明書をデバイスに追加します。

下記も参照

アプリケーション証明書の追加または削除 (ページ 205)

WinCC 認証機関とルート証明書の作成 (ページ 200)

デバイスの削除 (ページ 204)

IP アドレスのこの後の変更 (ページ 221)

証明書を使用可能にする (ページ 198)

5.3.6 デバイスの削除

必要条件

- 認証機関が、WinCC Certificate Manager の認証機関デバイスで作成された。
- デバイスが認証機関に追加された。

手順

1. デバイスで Certificate Manager を開きます。
2. [CA 設定]タブを選択します。
3. コンテキストメニューでデバイスの[削除]オプションを選択します。

結果

デバイスとアプリケーションの証明書は、認証機関から削除されます。

注記

削除しても、デバイスにインストールされている証明書設定には影響はありません。
必要に応じて、WinCC Certificate Manager でデバイスから証明書をアンインストールします。

下記も参照

デバイスの追加 (ページ 202)

アプリケーション証明書のアンインストール (ページ 214)

5.3.7 アプリケーション証明書の追加または削除

必要条件

デバイスが、WinCC Certificate Manager で認証機関に追加された。

5.3 WinCC Certificate Manager

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
3. 必要なアプリケーション証明書をデバイスに追加します。
各証明書に対して次のステップを実行します。
 - デバイスを右クリックして、[<証明書タイプ>を追加...]を選択します。
 - [新しい証明書]ダイアログの証明書のプロパティを入力します。
必要に応じて、別の暗号鍵の長さとして証明書のランタイムを選択してください。

注記

有効性

WebUX | WebNavigator 証明書の最大有効期間は 27 か月に制限されています。一部のブラウザでは、これより長い有効期間は受け入れられていません。

注記

WebUX | WebNavigator 証明書の名前として、[完全修飾ドメイン名]を使用してください。

-
- [作成]をクリックします。

結果

デバイスの証明書設定が完了しています。

次のステップ

証明書設定をエクスポートします。

注記

認定機関デバイスをランタイム PC として使用する場合は、PC の証明書設定、あるいは個別のアプリケーション証明書を直接インストールします。エクスポートは必要ありません。

アプリケーション証明書の削除

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
3. 目的のデバイスの下のアプリケーション証明書をクリックし、コンテキストメニューで「削除」エントリを選択します。

結果

アプリケーション証明書が削除されています。

注記

削除しても、デバイスにインストールされている証明書設定には影響はありません。
必要に応じて、デバイスで証明書をアンインストールします。

追加オプション

たとえば有効期限が切れるため、アプリケーション証明書を再作成できます。
アプリケーション証明書を公的証明書としてエクスポートできます。

下記も参照

証明書設定のエクスポート (ページ 208)

証明書設定または個別の証明書のインストール (ページ 212)

デバイスの追加 (ページ 202)

アプリケーション証明書の再作成 (ページ 217)

アプリケーション証明書を公的証明書としてエクスポート (ページ 216)

必要な証明書 (ページ 189)

アプリケーション証明書のアンインストール (ページ 214)

証明書を使用可能にする (ページ 198)

5.3 WinCC Certificate Manager

5.3.8 PC のためのエクスポート、インポート、インストール

5.3.8.1 証明書設定のエクスポート

PC の証明書設定は、アプリケーションの証明書、およびルート証明書と CRL ファイルで構成されています。

注記

証明書機関のデバイスをランタイム PC として使用し、このデバイスの変更された証明書設定を提供するだけの場合は、証明書設定のエクスポートとインポートは必要ありません。

この場合、証明書設定または個々のアプリケーション証明書を直接インストールします。

注記

Unified 証明書のエクスポート、インポート、およびインストール

WinCC Certificate Manager は以下をサポートします。

- WinCC Certificate Manager で認証機関が開かれている Unified 証明書のエクスポート。
- WinCC と WinCC Unified Runtime が PC に並行してインストールされている場合、Unified PC の証明書設定または PC の個々の Unified アプリケーション証明書をインポートしてインストール。

概要


デバイスの証明書設定のエクスポートは、次の場合に必要です。

- デバイスを追加し、アプリケーション証明書を初めて設定した後
- アプリケーション証明書を追加、削除、または再作成した後
- ルート証明書を再作成した後
- CRL ファイルを更新した後

WinCC PC のエクスポートオプション

以下のオプションがあります。

- CA コンテナのエクスポート
全 PC の証明書設定は、共有ファイルにエクスポートされます。
各 PC に同じファイルをインポートします。この後は、各 PC においてそれぞれのデバイスの証明書設定のみをインストールできます。
- PC のエクスポート
選択した PC の証明書設定のみをエクスポートします。

	効率的な手順のためのヒント
推奨手順:	
<ul style="list-style-type: none">• 複数の PC の証明書設定が変更された場合:CA コンテナのエクスポート。• 個別の PC の証明書設定が変更された場合:このデバイスの証明書設定をエクスポートします。	

Unified デバイスのエクスポートオプション

WinCC Certificate Manager で Unified 認証機関を開いた場合は、次のようになります。

- Unified PC:WinCC PC で説明したものと同一エクスポートオプションがあります。
- Unified パネル:デバイスをエクスポートします。

必要条件

Certificate Manager で希望する一台または複数台の PC の証明書設定を完了したこと。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
3. 次の手順を実行します。
すべての PC の証明書設定をエクスポートするには:
 - ルート証明書モジュールを右クリックします。
 - [エクスポート]>[CA コンテナ...]を選択します。個々のデバイスの証明書設定をエクスポートするには、デバイスのコンテキストメニューで [デバイスをエクスポート]>[PC 〜]エントリを選択します。
4. 次のダイアログでパスワードを入力して繰り返し、エクスポートファイルを保護します。

5.3 WinCC Certificate Manager

5. [エクスポート]をクリックします。
6. [保存]をクリックして、保存場所とファイル名を選択します。

結果

デバイスの証明書設定やすべてのデバイスの証明書設定は、指定されたパスワードで暗号化された安全な保管ファイルに保存されます。

次のステップ

デバイスの証明書設定がインポートされます。

WinCC PC および並列 WinCC インストールを備えた Unified PC の場合は、このヘルプの説明に従って進めてください。

並列 WinCC がインストールされていない Unified PC の場合は、「WinCC Unified Runtime の証明書」ヘルプの説明に従って進めます。

Unified パネルの場合は、「WinCC Unified Runtime の証明書」ヘルプの説明に従って進めます。

下記も参照

証明書設定のインポート (ページ 210)

パスワード要件 (ページ 190)

証明書を使用可能にする (ページ 198)

5.3.8.2 証明書設定のインポート

PC の証明書設定は、アプリケーションの証明書、およびルート証明書と CRL ファイルで構成されています。

注記

証明書機関のデバイスをランタイム PC として使用し、このデバイスの変更された証明書設定を提供するだけの場合は、証明書設定のエクスポートとインポートは省略されます。この場合、証明書設定または個々のアプリケーション証明書を直接インストールします。

必要条件

- 認証機関デバイスで、CA コンテナまたは PC の証明書設定がエクスポートされていること。
- インポートする証明書設定のある PC は、エクスポートファイルの保存場所へアクセスできること。

手順

1. PC で WinCC Certificate Manager を開きます。
2. [CA 設定]タブを選択します。
3. 作業エリアで[すべての証明機関]を選択します。
4. コンテキストメニューを開き、[設定を開く]を選択します。
5. エクスポートファイルを選択します。
6. エクスポート時に選択したパスワードを入力します。
7. 入力を確定します。

結果

設定ファイルが[CA 設定]タブにロードされます。タブの内容は、エクスポート中に選択したオプションにより異なります。

- エクスポートされた CA コンテナ:
認証機関のすべてのデバイスの証明書設定が表示されます。
インストールできるのは、ローカルデバイスの証明書だけです。他のデバイスの表示は、情報提供を目的としています。それらの設定は変更できません。
- エクスポートされたデバイスの証明書設定:
表示とインストールは、ローカルデバイスの証明書設定に限定されます。

注記

Certificate Manager を終了すると、ロードされた設定が閉じます。

次のステップ

デバイスの証明書設定、あるいは個別のアプリケーション証明書をインストールします。

5.3 WinCC Certificate Manager

下記も参照

証明書設定または個別の証明書のインストール (ページ 212)

証明書設定のエクスポート (ページ 208)

証明書を使用可能にする (ページ 198)

5.3.8.3 証明書設定または個別の証明書のインストール


PC の証明書設定全体、あるいは個別のアプリケーション証明書をインストールできます。

PC の証明書設定は、アプリケーションの証明書、およびルート証明書と CRL ファイルで構成されています。

必要条件

- WinCC Certificate Manager が、証明書をインストールする PC で開かれていること。
- PC の証明書設定が、Certificate Manager がインストールされているデバイスにインポートされていること。
- Unified 認証機関からの証明書をインストールする場合: WinCC Unified Runtime は PC にインストールされています。

インストール

1. [CA 設定] タブを選択します。
2. 以下のオプションの 1 つを選択します。
証明書設定全体をインストールするには:
 - ローカルデバイスのノードを右クリックします。
ローカルデバイスには次のアイコンがあります: 
 - [すべての証明書をインストール] を選択します。個別のアプリケーション証明書をインストールするには:
 - ローカルマシンのノードで、証明書を右クリックします。
 - [インストール] を選択します。

結果

デバイスの証明書設定全体、あるいは個別のアプリケーション証明書がインストールされます。

注記

デバイスと通信ピアの間の信頼関係がすでに確立されている場合、デバイスは通信パートナーと正常に通信を行うことができます。

インストールの際は、次のことが事細かに発生します。

- アプリケーション証明書が、それぞれのアプリケーションに対して定義されている証明書ストアにインストールされます。
- ルート証明書が、証明書ストアで信頼できるものとして分類されます。
- CRL ファイルが証明書ストアにインストールされます。
- WebUX | WebNavigator 証明書がインストールされていて、Web サーバーの Web ページがすでにセットアップされている場合、インストールにより、証明書は Web ページに自動的にバインドされます。証明書が、インストールの際に選択された証明書を置換します。この後、新しい証明書の使用を実行するために、Web ページが再起動されます。接続されている WebUX クライアントは切断され、再度ログインする必要があります。WebUX | WebNavigator 証明書がインストールされていて、Web ページがまだセットアップされていない場合、接続は正常に行われません。証明書は[インストール済み証明書]タブに表示されません。Certificate Manager は、これを[出力]エリアに入力してログを残します。

注記

OPC UA サーバー証明書は、ランタイムを再起動した後に反映されます。

下記も参照

OPC UA 通信における信頼関係の確立 (ページ 240)

アプリケーション証明書のアンインストール (ページ 214)

証明書設定のインポート (ページ 210)

証明書を使用可能にする (ページ 198)

5.3 WinCC Certificate Manager

5.3.8.4 アプリケーション証明書のアンインストール

PC にインストールされている WinCC アプリケーション証明書をアンインストールすることも選択できます。

必要条件

- WinCC Certificate Manager が PC で開かれていること。
- Certificate Manager で少なくとも 1 つのアプリケーション証明書が PC にインストールされていること。

手順

1. [インストール済み証明書]タブを選択します。
2. 開く証明書を右クリックします。
3. [削除]を選択します。

下記も参照

証明書設定または個別の証明書のインストール (ページ 212)

5.3.9 ルート証明書と CRL ファイルのエクスポート

概要

WinCC Certificate Manager を使用すると、ルート証明書および CRL ファイルを、証明書設定から別々に、公的証明書としてエクスポートおよび配布することができます。PC とその外部通信ピアとの間に信頼関係を確立するため、あるいは期限の切れた CRL ファイルを更新するために、これは必要です。

以下のオプションがあります。

- ルート証明書と CRL ファイルのエクスポート
- CRL ファイルのみのエクスポート

必要条件

WinCC 認証機関が作成されているか、Unified 認証機関が認証機関デバイスの Certificate Manager にインポートされています。

ルート証明書と CRL ファイルのエクスポート

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA の設定] タブで、ルート証明書をクリックします。
3. コンテキストメニューで[エクスポート]>[CA 証明書...] エントリを選択します。
4. ファイル形式を選択します。
5. 入力を確定します。
6. ターゲットフォルダを選択します。
7. 入力を確定します。

ルート証明書とその CRL ファイルはターゲットフォルダのそれぞれ別のファイルにエクスポートされます。

CRL ファイルのみをエクスポート

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. ルート証明書で、証明書失効リストを右クリックします。
4. [エクスポート] を選択します。
5. ファイル形式を選択します。
6. 入力を確定します。
7. ターゲットフォルダを選択します。
8. 入力を確定します。

CRL ファイルがターゲットフォルダにエクスポートされます。

次のステップ

ファイルを外部通信ピアにインポートします。デバイスのユーザーヘルプで説明されているステップを実行してください。

下記も参照

WinCC 認証機関とルート証明書の作成 (ページ 200)

証明書を使用可能にする (ページ 198)

5.3 WinCC Certificate Manager

5.3.10 アプリケーション証明書を公的証明書としてエクスポート

必要条件

アプリケーション証明書が、WinCC Certificate Manager のインストールされている PC に追加されていること。

認証機関デバイスへの証明書のエクスポート

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
3. デバイスの下にあるアプリケーション証明書を右クリックします。
4. [証明書をエクスポート...]を選択します。
5. ファイル形式を選択します。
6. 入力を確認します。
7. ターゲットフォルダを選択します。
8. 入力を確認します。

証明書を PC にエクスポート

追加の要件

- アプリケーション証明書が、WinCC Certificate Manager の搭載されている PC にインストールされていること。

手順

1. PC で Certificate Manager を開きます。
2. [インストール済み証明書]タブを選択します。
3. アプリケーション証明書を右クリックします。
4. [証明書をエクスポート...]を選択します。
5. ファイル形式を選択します。
6. 入力を確認します。
7. ターゲットフォルダを選択します。
8. 入力を確認します。

結果

アプリケーション証明書の公開鍵がエクスポートされます。アプリケーション証明書の公開鍵をデバイスの外部通信相手に配布します。

下記も参照

アプリケーション証明書の追加または削除 (ページ 205)

証明書設定または個別の証明書のインストール (ページ 212)

5.3.11 証明書の再作成

WinCC Certificate Manager を使用すると、既存の証明書を再作成できます。これは、次の場合に必要です。

- アプリケーション証明書の期限切れ:
アプリケーション証明書を再作成します。
- ルート証明書の期限切れ:
認証機関の設定全体を再作成します。
- CRL ファイルの期限切れ
CRL ファイルを更新します。
- Certificate Manager でデバイスを追加する際に IP アドレスが指定されたため生じた PC の IP アドレスの変更。
デバイスとアプリケーションの証明書を再度追加します。

下記も参照

アプリケーション証明書の再作成 (ページ 217)

設定全体の再作成 (ページ 218)

CRL ファイルの更新 (ページ 219)

IP アドレスのこの後の変更 (ページ 221)

証明書を使用可能にする (ページ 198)

5.3.11.1 アプリケーション証明書の再作成

次の場合は、アプリケーション証明書を再作成します:

- 証明書の有効期限が切れている。
- 有効な証明書のエントリを、たとえばエントリを修正するために編集する必要がある。

5.3 WinCC Certificate Manager

手順

1. WinCC Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
3. 目的のデバイスのアプリケーション証明書を右クリックして、[再作成]を選択します。
[証明書の再作成]ダイアログが開きます。既存の証明書のエントリがダイアログに読み込まれます。
4. 必要なプロパティを変更します。
5. [作成]をクリックします。

結果

証明書が再作成されます。デバイスの証明書設定がエクスポートされます。デバイスに証明書をインストールします。

下記も参照

PC のためのエクスポート、インポート、インストール (ページ 208)

証明書を使用可能にする (ページ 198)

5.3.11.2 設定全体の再作成

ルート証明書の有効期限が切れると、認証機関の設定全体を再作成する必要があります。WinCC Certificate Manager は、これに関するサポートを提供します。

必要条件

認証機関が、認証機関デバイスの Certificate Manager で作成され、設定されています。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定]タブを選択します。
認証機関の設定が表示されます。
3. ルート証明書をクリックし、コンテキストメニューで[すべて再作成...]エントリを選択します。
4. [認証機関の再作成]ダイアログが開きます。
以前の認証機関のプロパティがデフォルトとして引き継がれます。
5. 証明機関を作成するときと同じパスワードを入力します。証明書を確認します。
6. [作成]をクリックします。

結果

認証機関の設定全体が再作成されます。

- 秘密鍵
- ルート証明書
- CRL ファイル
- すべてのデバイスとそのアプリケーション証明書

次のステップ

- PC の証明書設定をエクスポートします。証明書設定を PC にインポートしてインストールします。
- ルート証明書と CRL ファイルを外部通信ピアに配布します。

下記も参照

PC のためのエクスポート、インポート、インストール (ページ 208)

証明書を使用可能にする (ページ 198)

5.3.11.3 CRL ファイルの更新

WinCC Certificate Manager でルート証明書が作成されると、CRL ファイルの有効期限が 24 ヶ月で設定されます。

この有効期限が切れる前に、CRL ファイルを更新する必要があります。

必要条件

認証機関が、Certificate Manager の認証機関デバイスで生成されていること。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. ルート証明書で、[証明書失効リスト] ノードをクリックします。
4. コンテキストメニューの[更新] コマンドを選択します。

5.3 WinCC Certificate Manager

結果

有効期間が 24 ヶ月の新しい CRL ファイルが作成される。

次のステップ

ファイルをエクスポートします。ファイルを配布します。

下記も参照

ルート証明書と CRL ファイルのエクスポート (ページ 214)

WinCC 認証機関とルート証明書の作成 (ページ 200)

証明書を使用可能にする (ページ 198)

5.3.12 認証機関のバックアップ

手順

認証機関のすべてのデータをバックアップするには、次のように進んでください。

1. WinCC Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. ルート証明書をクリックして、コンテキストメニューから[エクスポート]>[フルバックアップ]を選択します。
4. [エクスポート] ダイアログでパスワードを入力して繰り返し、バックアップファイルを保護します。
5. [エクスポート] をクリックします。
6. [保存] をクリックして、保存場所とファイル名を選択します。

結果

認証機関の設定全体がバックアップファイルに書き込まれる。

バックアップのロード

1. Certificate Manager を開きます。
2. [CA 設定] タブで、[設定を開く...] エントリをダブルクリックします。
3. バックアップファイルを選択し、[開く] をクリックして確定します。
4. バックアップを作成するときに設定したパスワードを入力し、[開く] で確認します。

下記も参照

パスワード要件 (ページ 190)

証明書を使用可能にする (ページ 198)

5.3.13 IP アドレスのこの後の変更

概要

WinCC Certificate Manager で PC を追加する際は、デバイス名または FQDN および/または IP アドレスを指定します。デバイスのアプリケーション証明書に、この情報は書き込まれます。

IP アドレスを入力して、この後それが変更されると、新しいアドレスと証明書の情報は一致しなくなります。以下に説明されている手順に従ってください。

Certificate Manager で IP アドレスを入力しなかった場合、追加のステップは必要ありません。

手順

1. Certificate Manager を認証機関デバイスで開きます。
2. [CA 設定] タブを選択します。
3. IP アドレスが変更されている PC を削除します。
4. 新しい IP アドレスで PC を再度追加します。
5. 必要なアプリケーション証明書を PC に追加します。
6. 証明書設定のエクスポート、配布、インストールを行います。

認証機関デバイスの IP アドレスの変更

変更された IP アドレスの PC も認証機関デバイスである場合、次のように進んでください。

1. Certificate Manager から PC を削除します。
2. PC を再度追加します。
3. 認証機関の設定全体を手動で再作成します。設定を配布してインストールします。

下記も参照

デバイスの削除 (ページ 204)

デバイスの追加 (ページ 202)

5.3 WinCC Certificate Manager

アプリケーション証明書の追加または削除 (ページ 205)

PC のためのエクスポート、インポート、インストール (ページ 208)

WinCC 認証機関とルート証明書の作成 (ページ 200)

証明書を使用可能にする (ページ 198)

5.4 S7 PLC との通信用証明書

5.4.1 [セキュア通信]の設定

[セキュア通信]

「Secure Communication」は、証明書によって PLC との通信を保護します。

V17 以降の TIA Portal では、「Secure Communication」は、以下のシリーズの S7 PLC で使用できます。

- ファームウェア 4.5 以降の S7-1200
- ファームウェア 2.9 以降の S7-1500

STEP 7 コンポーネントは、公開キーと秘密鍵を使用した非対称キーの手順を使用します。暗号化プロトコルとして TLS (Transport Layer Security)が採用されています。

注記

証明書の必須使用

TIA Portal の[保護とセキュリティ]>[接続メカニズム]で S7 PLC に対して[セキュアな PG/PC および HMI の通信のみを許可]オプションが有効になっている場合、PLC との通信には S7 証明書の使用が必須です。

WinCC の「Secure Communication」:

WinCC は TLS バージョン「TLS 1.3」の上記 PLC で「Secure Communication」をサポートします。

「セキュアな通信」を行うには、有効な S7 証明書を PLC にインストールする必要があります。WinCC PC はこの S7 証明書を信頼する必要があります。

ランタイムで「Secure Communication」を使用する場合は、通信チャンネルの接続を切り替えることもできます。

注記

推奨事項:接続エラーの表示

ランタイムに接続エラーを検出するには、WinCC で WinCC AlarmControl を設定します。

手順:初期設定

1. TIA Portal で PLC を設定します。
2. TIA Portal で PLC の S7 証明書を作成します。
あるいは、サードパーティプログラムで証明書を作成し、その証明書を TIA Portal にインポートすることもできます。
詳細情報: S7 証明書の基礎 (ページ 228)
3. PLC のプロパティ[保護とセキュリティ]>[接続メカニズム]で、TIA Portal の S7 証明書を PLC 通信証明書として選択します。
4. TIA Portal で PLC 設定を PLC にロードします。
ロードすると、PLC 通信証明書として設定された証明書が、PLC 証明書ストア内の独自の証明書にインストールされます。
5. WinCC PC と PLC の間に信頼関係を確立します。
詳細情報: S7 通信における信頼関係の確立 (ページ 232)
6. WinCC 設定システムとランタイムが異なる PC にインストールされている場合は、WinCC プロジェクトをランタイム PC にロードします。

手順:S7 証明書の新規作成

S7 証明書の有効期限が切れた場合、または PLC のアクセスポイントが変更された場合は、S7 証明書を再作成する必要があります。

手順:CA ベースの証明書の使用

1. TIA Portal で PLC の S7 証明書を再作成します。
あるいは、サードパーティプログラムで新しい証明書を作成し、その証明書を TIA Portal にインポートすることもできます。
詳細情報: S7 証明書の新規作成 (ページ 231)
2. PLC のプロパティ[保護とセキュリティ]>[接続メカニズム]で、TIA Portal に新しく作成された S7 証明書を PLC 通信証明書として選択します。
3. TIA Portal で PLC 設定を PLC にロードします。
ロードすると、PLC 通信証明書として設定された証明書が、PLC 証明書ストア内の独自の証明書にインストールされます。
WinCC PC が S7 証明書の派生元のルート証明書をすでに信頼している場合、PC は S7 証明書も自動的に信頼します。
これは、例えば次のような場合に適用されます:
 - 最初に設定した S7 証明書と新しく作成した S7 証明書を作成するときに、同じルート証明書を選択している場合。
 - WinCC PC が、S7 証明書が同じルート証明書を持っている別の S7-PLC を信頼している場合。
4. 次の要件に従って、WinCC-PC と PLC の間に信頼関係を確立する必要があります。
 - 最初に設定された S7 証明書と新しく作成された S7 証明書を作成するときに、異なるルート証明書を選択しています。
 - WinCC PC が、新しく作成された S7 証明書のルート証明書をまだ信頼していません。
詳細情報: S7 通信における信頼関係の確立 (ページ 232)

手順:自己署名済みの証明書の使用

1. TIA Portal で PLC の S7 証明書を再作成します。
あるいは、サードパーティプログラムで証明書を作成し、その証明書を TIA Portal にインポートすることもできます。
詳しくは: S7 証明書の基礎 (ページ 228) > 証明書の作成
2. PLC のプロパティ[保護とセキュリティ] > [接続メカニズム]で、TIA Portal に新しく作成された S7 証明書を PLC 通信証明書として選択します。
3. TIA Portal でハードウェア設定を PLC にロードします。
ロードすると、PLC 通信証明書として設定された証明書が、PLC 証明書ストア内の独自の証明書にインストールされます。
4. TIA Portal から WinCC に S7 証明書をダウンロードします。
ロードすると、WinCC PC 上の証明書との信頼関係が確立されます。
詳細情報: S7 通信における信頼関係の確立 (ページ 232)
5. WinCC 設定システムとランタイムが異なる PC にインストールされている場合は、WinCC プロジェクトをランタイム PC にロードします。

結果

- 次の接続要求時に、WinCC は S7 証明書を信頼します。
- 接続が確立されます。
- 証明書が接続が終了するまで使用されます。

接続の終了

次のイベントによりセッションが終了します:

はい	いいえ
ランタイムの停止または再起動	PLC の停止または再起動
コントローラからの接続の切断	PLC の証明書を含むハードウェア設定の更新
	証明書の有効期限切れ

WinCC プロジェクトとコントローラで証明書を互いに個別にインストールできます。

PLC の更新とランタイムの終了および再起動は、同時に実行する必要はありません。

ただし、システムのセキュリティを向上するため、期限日に達したらすぐに新しい証明書を作成してください。

接続要求の拒否

WinCC PC が S7 証明書を信頼しない場合は、接続要求を拒否します。

5.4 S7 PLC との通信用証明書

システムメッセージが生成されます。

再接続時間(ReconnectTime)が経過すると、WinCC PC が新しい接続試行を開始します。

システムメッセージ

次のシステムメッセージは WinCC PC で証明書のステータスをドキュメント化します。

番号	WinCC メッセージテキスト	注記
1000306	次のアドレスを使用した接続 @1% <i>s</i> @で一般証明書エラーが発生 しました: @2% <i>s</i> @。 エラーコード: @3% <i>s</i> @。	このシステムメッセージは、PLC と WinCC デバイス間で証明書が異なる場 合、または証明書がまだ WinCC デバイ スにインポートされていない場合に表 示されます。 対策:両側に同じ証明書があることを確 認してください。 WinCC で現在使用可能な証明書に関す る情報は、チャンネルのコンテキスト メニューの[ルート証明書の削除]で確認 できます。
1000307	@2% <i>s</i> @の PLC 証明書が期限切れ になりました。 エラーコード: @3% <i>s</i> @。	このシステムメッセージは、S7 証明書 またはルート証明書の有効期限が切れ たときに表示されます。 対策: 1. 影響を受ける証明書を更新します。 2. これらの証明書を PLC にダウンロー ドします。 3. 証明書を WinCC にインポートしま す。
1000308	@2% <i>s</i> @の PLC 証明書は信頼され ていません。手動で信頼するこ とができます。 エラーコード: @3% <i>s</i> @。	このシステムメッセージは、接続の変 更後に表示されます。 対策:証明書を「trusted」フォルダに移 動します。

番号	WinCC メッセージテキスト	注記
1000309	@2%s@の PLC 証明書は信頼されていません。手動で信頼することができません。 エラーコード: @3%s@。	このシステムメッセージは、PLC と WinCC デバイス間で証明書が異なる場合に表示されます。 対策:両側に同じ証明書があることを確認してください。
1000310	@2%s@の PLC 証明書が取り消されました。 エラーコード: @3%s@。	証明書が失効している場合、このシステムメッセージが表示されます。 対策:有効な証明書を提供してください。

詳細情報

WinCC 情報システム:

- TIA Portal で証明書を作成、再作成、またはインポート (ページ 228)
- 新しい証明書を作成するための基礎 (ページ 231)
- WinCC PC と S7-PLC 間の信頼関係を確立 (ページ 232)
- [通信] > [SIMATIC S7-1200、S7-1500 チャンネル] > [チャンネルの設定] > [接続の変更]

SiePortal では:

- SiePortal:STEP 7 (TIA Portal) - 「セキュア通信」ドキュメント(ID 109815056) (<https://support.industry.siemens.com/cs/ww/en/view/109815056/161011991691>)
- SiePortal: 「WinCC V7 - 安全な接続」 (ID 109798498) (<https://support.industry.siemens.com/cs/ww/en/view/109798498>)
- SiePortal: S7-1500 / S7-1200 / ET 200SP との WinCC 通信のための「SIMATIC SCADA Export」ドキュメント(ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)
- SiePortal: 「TIA Portal の SIMATIC SCADA Export」 (ID 109748955)をダウンロード (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)
- SiePortal: STEP 7 (TIA Portal) - ドキュメント:署名と証明書(ID 109798671) (<https://support.industry.siemens.com/cs/ww/en/view/109798671/143786688779>)

下記も参照

信頼関係の作成: (ページ 183)

5.4.2 S7 証明書の基礎

注記

証明書の必須使用

S7 PLC の設定で TIA Portal の[セキュアな PG/PC および HMI 通信のみを許可]オプションが選択されている場合は、S7 証明書の使用が必須です。

概要

証明書のタイプ

S7 証明書は自己署名または CA ベースにすることができます。

セキュリティ上の理由により、CA ベースの証明書の使用が推奨されます。

証明書の作成

S7 証明書は TIA Portal で作成または再作成されます。

あるいは、既存の証明書を TIA Portal にインポートすることもできます。

TIA Portal の Certificate Manager に関する詳細情報

SiePortal:STEP 7 (TIA Portal) - 「セキュア通信」ドキュメント(ID 109815056) (<https://support.industry.siemens.com/cs/ww/en/view/109815056/161011991691>):

- > [グローバルセキュリティ設定]
- > [セキュアな PG/HMI 通信]

CA ベースの S7 証明書の挿入

TIA Portal で S7 証明書を作成します。または、サードパーティプログラムで S7 証明書を作成し、TIA Portal にインポートします。

TIA Portal での作成

必要条件:[S7 PLC の設定] > [保護とセキュリティ] > [Certificate Manager] > [TIA Portal による証明書管理]:[Certificate Manager のグローバルセキュリティ設定を使用]オプションが選択されています。

以下のオプションがあります。

- デフォルトの認証機関の使用
[Certificate Manager のグローバルセキュリティ設定を使用]オプションを選択すると、グローバルセキュリティの Certificate Manager にデフォルトのルート証明書が提供されます。
次の手順を実行します。
 - グローバルセキュリティの Certificate Manager で S7 証明書を作成します。
[CA ベース]オプションを作成するときに、デフォルトのルート証明書を選択します。
S7 証明書は、ルート証明書に属する秘密鍵で署名されます。
 - PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。
- サードパーティプログラムで作成された認証機関または中間認証機関をインポートして使用
次の手順を実行します。
 - サードパーティプログラムで認証機関または中間認証機関を作成します。
 - グローバルセキュリティの Certificate Manager の TIA Portal に、認証機関または中間認証機関のルート証明書と秘密鍵をインポートします。
中間認証機関を使用する場合は、インポートコンテナに同じ証明書チェーンが含まれている必要があります。
インポートファイルは次の形式である必要があります: 「PKCS#12」(P12 ファイル/PFX ファイル)
 - グローバルセキュリティの Certificate Manager で S7 証明書を作成します。[CA ベース]オプションを作成するときに、インポートされたルート証明書を選択します。
S7 証明書は、ルート証明書に属する秘密鍵で署名されます。
 - PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。

サードパーティプログラムでの作成

1. サードパーティプログラムを使用して CA ベースの S7 証明書を作成します。
2. S7 証明書、その秘密鍵、および証明書チェーン全体を TIA Portal にインポートします。
インポートファイルは次の形式である必要があります: 「PKCS#12」(P12 ファイル/PFX ファイル)
3. PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。

自己署名済みの S7 証明書の使用

以下のオプションがあります。

- S7 PLC に対して[セキュアな PG/PC および HMI 通信のみを許可]オプションが選択されている場合は、TIA Portal は自己署名済みのデフォルト証明書を自動的に生成します。このデフォルトの証明書を使用できます。
PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。
- S7 PLC のローカルの Certificate Manager の TIA Portal で独自の自己署名済みの証明書を作成します。
PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。
- サードパーティプログラムを使用して自己署名済み証明書を作成します。
証明書を TIA Portal にアップロードします。
インポートファイルは次の形式である必要があります:「PKCS#12」(P12 ファイル/PFX ファイル)
PLC のローカルの Certificate Manager で S7 証明書を PLC に割り付けます。

PLC への S7 証明書のインストール

PLC に S7 証明書をインストールするには、TIA Portal でハードウェア設定を PLC にロードします。

ロードプロセスにより、証明書が PLC 独自の証明書にインストールされます。

詳細情報: SiePortal:STEP 7 (TIA Portal) - 「セキュア通信」ドキュメント(ID 109815056) (<https://support.industry.siemens.com/cs/ww/en/view/109815056/161011991691>)

WinCC PC の証明書の信頼

WinCC と PLC 間の通信を行うには、信頼できるサードパーティ証明書として S7 証明書を WinCC PC にインストールする必要があります。

詳細情報: S7 通信における信頼関係の確立 (ページ 232)

有効性

S7 証明書の有効期間は、証明書の作成時に特定されます。

有効性の表示

詳細情報: 証明書の有効期限をチェック (ページ 236)

有効期限切れ

PLC クライアントは、クライアントに設定されているタイムゾーンに基づいて証明書の有効性をチェックします。PLC に設定されているタイムゾーンは関係ありません。

有効期限が切れる前に早めに新しい証明書を作成してください。

新しい証明書を PLC にロードします。PLC クライアントの証明書との信頼関係を確立します。

下記も参照

[セキュア通信]の設定 (ページ 223)

S7 証明書の新規作成 (ページ 231)

SiePortal: 「WinCC V7 - 安全な接続」 (ID 109798498) (<https://support.industry.siemens.com/cs/ww/en/view/109798498>)

5.4.3 S7 証明書の新規作成

理由

次の場合には、S7 証明書を再作成する必要があります:

- 有効期限が切れている場合
- PLC アクセスポイントが変更されている場合

PLC アクセスポイントの変更の例:

- PLC または通信プロセッサの IP アドレスが、たとえば 192.168.178.1 から 192.168.178.2 に変更されました。
- PLC の CPU ハードウェアが変更されました:
デバイスが交換されたときに新しい IP アドレスが追加されました。例:
516-3AN01-0AB0 V2.9 から 518-4AP00-0AB0 V3.0 へのデバイスの交換
証明書にまだ含まれていない追加の IP アドレスのあるモジュールが追加されました。
- メモリカードの IP アドレスが変更されます。

レクリエーションは必要ありません

以下の場合にはレクリエーションは必要ありません:

- IP アドレスのあるモジュールの削除
- インターフェースアドレスを証明書にすでに含まれている IP アドレスに変更

5.4 S7 PLC との通信用証明書

- PLC の中央 IO モジュールおよび分散 IO モジュールの変更
- 新しい IP アドレスが追加されないデバイスの交換(例: 516-3AN01-0AB0 V2.9 から 516-3AP03-0AB0 V3.0)
- シンボル名の変更

新規作成

証明書の初期作成時と同じオプションがあります。

詳細情報: S7 証明書の基礎 (ページ 228)

自己署名済みの証明書の自動再作成

PLC のハードウェア設定をコンパイルするときに、TIA Portal は次の PLC 証明書をチェックします。

- 自己署名済みのデフォルト証明書
- TIA Portal で生成された自己署名済みの証明書で、作成時に「サブジェクト代替名」パラメータが変更されていないもの

これらの証明書のいずれかの有効期限が切れている場合、または PLC のアクセスポイントが変更されている場合、TIA Portal はコンパイル中に証明書を再作成します。

TIA Portal は、コンパイル情報にシステムメッセージとともにプロセスをドキュメント化します。

新しい証明書は、デバイスが次回ロードされたときにロードされます。

WinCC PC 上の新しい証明書との信頼関係を確立します。

下記も参照

S7 通信における信頼関係の確立 (ページ 232)

証明書の有効期限をチェック (ページ 236)

5.4.4 S7 通信における信頼関係の確立

概要

WinCC と S7 PLC 間の接続は、WinCC が S7 証明書を信頼している場合にのみ確立されます。

信頼関係を確立するには、PLC の S7 証明書を TIA Portal から WinCC にロードします。

ランタイムで接続を切り替えた後に信頼関係を確立するには([Change Connection])、証明書を手動で信頼します。

自動的な信頼関係

S7 証明書が CA ベースであり、WinCC PC が S7 証明書の派生元のルート証明書をすでに信頼している場合、PC は S7 証明書を自動的に信頼します。

接続の変更後に、TIA Portal から WinCC に S7 証明書をロードしたり、証明書を手動で信頼したりする必要はありません。

例

- WinCC PC の場合、CA ベースの S7 証明書を使用した S7 PLC への接続が設定されました。
- S7 証明書が TIA Portal から WinCC にインポートされました。
PC は S7 証明書のルート証明書と S7 証明書を信頼します。
- WinCC PC の場合、CA ベースの S7 証明書を使用して 2 番目の S7 PLC への接続が設定されました。
証明書には最初の PLC と同じルート証明書があります。
- 2 番目の PLC に接続しようとする、PC は証明書のルート証明書をすでに信頼しているため、PLC の証明書を自動的に信頼します。

必要条件

- 有効な S7 証明書が TIA Portal の PLC 用に作成されているか、TIA Portal にインポートされています。
- 証明書は、PLC のプロパティの[保護とセキュリティ]>[接続メカニズム]で PLC 通信証明書として設定されます。
- 証明書が PLC にロードされました。
- PLC 設定は TIA Portal でコンパイルされました。

ロードによる証明書の信頼

1. 「SIEMENS SIMATIC SCADA Export」 ツールを使用して、TIA Portal プロジェクトから PLC データをエクスポートします:
TIA Portal プロジェクトにおいて、PLC のコンテキストメニューで[SIMATIC SCADA にエクスポート]エントリを選択します。
2. 必要な場合、WinCC の[SIMATIC S7-1200, S7-1500 Channel]通信チャンネルで接続を作成します。
または、すでに作成された接続を選択します。
3. 接続のコンテキストメニューで[AS シンボル] > [ファイルからロード]を選択します。
4. ロード対象の希望するデータレコードを選択します。
5. 接続の証明書を受け入れることを確認します。
利用可能なコントローラデータがロードされます。
必要な証明書は、プロセスでも転送されます。
6. 必要な証明書をインポートするには、関連するプロンプトで[はい]をクリックして確定します。
すでにインポートされている証明書を削除するには、接続のコンテキストメニューで[証明書を削除する]エントリを選択します。

結果

- 証明書は信頼できるサードパーティの証明書で PC にインストールされます。
WinCC PC は S7 証明書を信頼します。
- CA ベースの S7 証明書を使用する場合は、次のことも適用されます:
ロード中に、S7 証明書の完全な証明書チェーンが WinCC にインストールされます。
WinCC PC は S7 証明書のルート証明書を信頼します。これは、PC が同じルート証明書を持つすべての S7 証明書も信頼することを意味します。

注記

証明書ファイルの可視性

証明書ファイルはデータベースの内部に保存されます。ファイルは WinCC 証明書ストアに表示されません。

接続変更後の証明書の信頼

ランタイムで通信チャンネルの接続を変更するには([Change Connection])、現在手動で接続されている S7 PLC の証明書に対して信頼関係を確立します。

手動で信頼する場合は、WinCC 証明書ストア内の証明書ファイルを必要なフォルダに移動します。([Manual Trust])

手動信頼は、通信チャンネルの接続がランタイムに切り替えられた場合にのみサポートされます。

詳細情報: AUTOHOTSPOT

信頼関係の終了

自己署名済みの S7 証明書

- S7 証明書の有効期限が切れた後の次の接続試行で。
S7 証明書を再作成します。
- 新しい S7 証明書が PLC にロードされた後の次の接続試行で。

CA ベースの S7 証明書

- S7 証明書またはそのルート証明書の有効期限が切れた後の次の接続試行で。
S7 証明書またはルート証明書、ならびに S7 証明書を再作成します。
- 新しい S7 証明書を PLC にロードした後の次の接続試行で。

有効期限切れの場合の動作

PLC と PC がセキュアな接続を介して通信している間に証明書の有効期限が切れても、セッションは有効なままになります。

次の接続要求時に、PC は期限切れの証明書を拒否します。接続を確立しません。

証明書の有効期限が切れたことを通知するシステムメッセージが PC に生成されます。

S7 証明書を再作成します。

注記

有効期限

有効期限は、PC に設定されているタイムゾーンに基づいて特定されます。PLC に設定されているタイムゾーンは関係ありません。

注記

有効期限のチェック

WinCC では、証明書の有効期間をチェックできます。

詳細情報: 証明書の有効期限をチェック (ページ 236)

下記も参照

S7 証明書の基礎 (ページ 228)

[セキュア通信]の設定 (ページ 223)

SiePortal: S7-1500 / S7-1200 / ET 200SP との WinCC 通信のための「SIMATIC SCADA Export」ドキュメント(ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

SiePortal: 「TIA Portal の SIMATIC SCADA Export」(ID 109748955)をダウンロード (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)

5.4.5 証明書の有効期限をチェック

必要条件

- 通信チャンネル「SIMATIC S7-1200、S7-1500 チャンネル」は、WinCC プロジェクトで設定されます。
- 通信チャンネルは S7-1200/S7-1500 PLC に接続されます。
- TIA Portal で PLC の S7 証明書が設定されました。
- S7 証明書が TIA Portal から WinCC にインポートされました。

S7 証明書の有効期限のチェック

1. 接続のコンテキストメニューで[証明書を削除する]を選択します。
[証明書を削除する]ダイアログには、証明書の有効期限が表示されます。
2. ダイアログで[いいえ]をクリックします。
証明書を削除せずにダイアログは閉じられます。

ルート証明書の有効期限のチェック

S7 証明書が CA ベースの場合、S7 証明書を発行した認証機関のルート証明書の有効期間をチェックできます。

1. チャンネルのコンテキストメニューで[ルート証明書を削除する]を選択します。
[ルート証明書を削除する]ウィンドウには、通信チャンネルの接続用にロードされたすべてのルート証明書が表示されます。
2. [名前]列と[有効期間終了日]列を使用して、目的のルート証明書の有効期間をチェックします。
3. [キャンセル]をクリックします。
証明書は削除されずにウィンドウが閉じます。

その他の手順

または、証明書の有効期限が切れたときに TIA Portal でチェックすることもできます。

デバイス固有の Certificate Manager	S7 PLC の S7 証明書の有効期限: <ul style="list-style-type: none">• 自己署名• CA ベース
グローバルな Certificate Manager	有効期限: <ul style="list-style-type: none">• ルート証明書• CA ベースの S7 証明書

5.5 OPC UA による通信のための証明書

5.5.1 WinCC OPC UA 証明書の基礎

OPC UA 用の WinCC 証明書ストア

OPC UA クライアント証明書ストア

- 保存先: C:\Program Files (x86)\Siemens\WinCC\opc\UAClient\PKI
- 構造:

フォルダ	
「own」	WinCC RT:独自の OPC UA クライアント証明書 WinCC ES:独自の OPC UA タグインポーター証明書
[証明書] > [信頼済み]	WinCC PC が信頼する OPC UA サーバーの証明書。
[証明書] > [信頼できない]	WinCC PC によって信頼されていない OPC UA サーバーの証明書。

OPC UA サーバー証明書ストア

- 保存先: C:\Program Files (x86)\Siemens\WinCC\opc\UAServer\PKI
WinCC/Connectivity Pack を WinCC Connectivity ステーションと共に使用する場
合: C:\Program Files
(x86)\Siemens\ConnectivityStation\OPC\UAServer\PKI\CA
- 構造:

フォルダ	
「own」	WinCC RT:独自の OPC UA サーバー証明書
[証明書] > [信頼済み]	WinCC PC によって信頼される OPC UA クライアントの証明書。
[証明書] > [信頼できない]	WinCC PC によって信頼されていない OPC UA クライアントの証明書。

WinCC PC の独自の CA ベースの OPC UA 証明書

WinCC PC には、次の OPC UA アプリケーション証明書(独自の証明書)を設定できます:

- OPC UA サーバー証明書
- WinCC RT:OPC UA クライアント証明書
- WinCC ES:OPC UA Tags Importer 証明書

CA ベースの証明書の使用

WinCC PC 用に独自の CA ベースの OPC UA アプリケーション証明書を生成してインストールするには、「WinCC Certificate Manager」ツールを使用します。

詳細情報:WinCC の証明書 > WinCC Certificate Manager > 証明書を使用可能にする (ページ 198)

注記

サードパーティツール

または、OpenSSL 3.0 を使用するサードパーティツールを使用して、WinCC PC 用の OPC UA アプリケーション証明書を作成することもできます。

それらの証明書を OPC UA クライアント証明書ストアまたは PC の OPC UA サーバー証明書ストアの「own」フォルダにコピーして、証明書を PC にインストールします。証明書は、この PC の Certificate Manager の[インストールされた証明書]タブに表示されます。[インストールされた証明書]タブで証明書を表示したりアンインストールしたりできます。

サードパーティツールを使用して、PC の OPC UA 通信パートナーにルート証明書を配布します。

有効期限が切れたら、サードパーティ ツールでアプリケーション証明書を再作成します。

自己署名済みの証明書を使用

WinCC PC に独自の自己署名済みの OPC UA アプリケーション証明書を使用するには、自動的に生成されたデフォルトの証明書を使用します。

このような証明書を使用するために、CA ベースの OPC UA サーバー証明書、OPC UA クライアント証明書、または OPC UA タグインポーター証明書を PC にインストールすることはできません。必要に応じてこれらの証明書をアンインストールしてください。詳細については、アプリケーション証明書のアンインストール (ページ 214)を参照してください。

5.5.2 OPC UA 通信における信頼関係の確立

CA ベースの OPC UA 証明書の手順

注記

自動的な信頼関係

次の場合、デバイスは通信パートナーの OPC UA アプリケーション証明書を自動的に信頼します。

- デバイスと通信パートナーはどちらも WinCC PC です。
同じ WinCC 認証機関が両方の PC に対して OPC UA アプリケーション証明書を発行しました。PC の証明書設定が、WinCC Certificate Manager を使用して PC にインストールされました。証明書設定をインストールすると、WinCC 認証機関のルート証明書が、信頼できる証明書を含むフォルダ内の OPC UA 証明書ストアの PC に自動的にインストールされます。
- デバイスには 2 つの OPC UA 通信パートナーがあります。
両方の通信パートナーの OPC UA 証明書は、同じ証明機関によって発行されました。認証機関のルート証明書は、この通信パートナーへの接続を可能にするために、デバイスの OPC UA 証明書ストアにすでにインストールされています。

他のステップは必要ありません

必要条件

- OPC UA サーバー設定では、証明書ベースの通信のセキュリティポリシーが設定されます。
- WinCC 認証機関は、WinCC PC に必要な OPC UA アプリケーション証明書を発行しました。
- 別の認証機関が、通信パートナーに必要な OPC UA アプリケーション証明書を発行しました。

手順

1. WinCC 認証機関デバイスの WinCC ルート証明書と CRL ファイルを、外部記憶媒体にエクスポートします。
詳細情報:WinCC の証明書 > ルート証明書と CRL ファイルのエクスポート (ページ 214)
2. WinCC PC の OPC UA 通信パートナーのデバイスで:
 - 外部記憶媒体を接続します。
 - ルート証明書とその CRL ファイルをデバイスにインストールして、ルート証明書を信頼します。
OPC UA 通信ピアのユーザーヘルプで説明されているように進んでください。
3. OPC UA 通信ピアのルート証明書とその CRL ファイルを外部記憶媒体にエクスポートします。
通信ピアのユーザーヘルプで説明されているように進んでください。

4. 外部記憶媒体を PC へ接続します。
5. OPC UA 通信ピアのルート証明書とその CRL ファイルを次のフォルダーにコピーします。
 - OPC UA サーバーとしての WinCC PC:
ファイルを次のフォルダにコピーします。
<サーバー設定ファイルで定義された保存場所>\trusted\certs
デフォルト設定での保存場所:
C:\Program Files
(x86)\Siemens\WinCC\opc\UAServer\PKI\trusted\certs
 - WinCC/Connectivity Pack を WinCC Connectivity Station と併用する場合の OPC UA サーバーとしての WinCC PC:
C:\Program Files
(x86)\Siemens\ConnectivityStation\OPC\UAServer\PKI\CA
 - OPC UA クライアントとしての WinCC PC:
C:\Program Files (x86)\Siemens\WinCC\opc\UAClient\PKI

自己署名済みの OPC UA 証明書の手順

下記も参照

WebUX/WebNavigator クライアントでルート証明書をインストール (ページ 242)

信頼関係の作成: (ページ 183)

OPC UA のセキュリティ概念 (ページ 465)

証明書を使用可能にする (ページ 198)

5.6 WebUX|WebNavigator の通信における証明書

5.6.1 WebUX/WebNavigator クライアントでルート証明書をインストール

概要

Web サーバーに対する信頼できる接続を確立するために、WebUX クライアントと WebNavigator クライアントは WebUX WebNavigator 証明書のルート証明書を信頼する必要があります。これを行うには、ルート証明書を Web クライアントの Trusted Certificate Authority のあるフォルダにインストールする必要があります。

信頼関係の自動作成

Web クライアントは、次のような場合、Web サーバーのルート証明書を自動的に信頼します。

- Web クライアントが WebNavigator クライアントであるか、Microsoft Edge か Chrome を使っている Web クライアントである場合。
- Web クライアントのデバイスが、WinCC のインストールされている PC である場合。
- Web サーバーの PC と Web クライアントの PC が同じ認証機関である場合。
次のアプリケーション証明書が、WinCC Certificate Manager の認証機関デバイスで作成されている場合。
 - Web サーバーの PC の場合:WebUX | WebNavigator 証明書
 - Web クライアントの PC の場合:少なくとも 1 つのアプリケーション証明書
- 次の証明書が Certificate Manager の搭載されている PC にインストールされている場合:
 - Web サーバーの PC の場合:WebUX | WebNavigator 証明書
 - Web クライアントの PC の場合:少なくとも 1 つのアプリケーション証明書

信頼関係を手動で作成

Web クライアントがルート証明書をまだ信頼していない場合は、ルート証明書を手動でインストールします。

プロシージャは、使用している Web クライアントにより異なります。

注記

Web クライアントが同じ認証機関の複数の Web サーバーと通信を行っている場合、Web クライアントにルート証明書を 1 回インストールすれば大丈夫です。

必要条件

- WebUX | WebNavigator 証明書のルート証明書が、認証機関デバイス(たとえば、外部記憶媒体など)にエクスポートされていること。
- WebUX クライアントや WebNavigator クライアントがユーザーが保管場所にアクセスできること。

WinCCViewerRT や Internet Explorer (WebNavigator クライアント)のプロシージャ

WebNavigator クライアントは、Microsoft Windows システムの証明書保管を使用しています。

信頼できるルート証明書機関のフォルダに Web サーバーのルート証明書をインストールするには、次の手順を実行してください。

1. WebNavigator クライアントデバイスで、ルート証明書ファイルをダブルクリックします。ルート証明書は、Microsoft Windows 標準の[証明書]ダイアログに表示されます。
2. [証明書をインストール...]を選択します。証明書のインポートウィザードが開きます。
3. [ローカルコンピュータ]を保管場所として選択し、[信頼されるルート証明書機関]を証明書ストアとして選択します。
4. インポートを開始します。

その他の手順

1. Microsoft Windows システム証明書ストアを直接起動します。
2. [信頼されるルート認証機関]フォルダをクリックして、[すべてのタスク]>[インポート]を選択します。証明書のインポートウィザードが開きます。
3. ステップ 3 から上記の通りに進めます。

Web サーバーで WinCCViewerRT から初めて接続を確立するときに、ルート証明書をインストールすることもできます。

1. WinCCViewerRT が Web サーバーを信頼していない場合、セキュリティの警告が表示されます。[証明書を表示]をクリックします。
2. [証明書をインストール...]をクリックします。
3. ステップ 3 から上記の通りに進めます。

WebUX クライアントとしての Microsoft Edge と Chrome のプロシージャ

Microsoft Edge と Chrome は、Microsoft Windows システムの証明書ストアを使用しています。

5.6 WebUX|WebNavigator の通信における証明書

信頼できるルート証明書機関のフォルダに Web サーバーのルート証明書をインストールするには、WebNavigator クライアントの上記の手順を実行してください。

固有の証明書ストアのあるブラウザのプロシージャ

WebNavigator クライアントのブラウザに固有の証明書ストアがあり、証明書ストアのルート証明書が信頼できるルート証明書機関のフォルダにまだない場合は、手動でルート証明書をインストールします。

ブラウザのユーザーヘルプで説明されているステップを実行してください。

たとえば、Firefox の場合は、次の手順を実行します。

1. Firefox で、[証明書]の[設定]>[プライバシーとセキュリティ]で、[証明書を表示]をクリックします。
2. [証明書管理]ウィンドウで、[認証機関]タブを選択します。
3. [インポート]をクリックして、ルート証明書ファイルを選択します。
4. 開かれたウィンドウで、[この証明書は Web サイトを特定可能]オプションを選択し、選択した内容を確定します。

下記も参照

ルート証明書と CRL ファイルのエクスポート (ページ 214)

証明書を使用可能にする (ページ 198)

プロセス通信

6.1 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

6.2 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が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。

6.3 WinCC プロセス通信

6.3.1 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ながあります。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。

6.3 WinCC プロセス通信

下記も参照

BinWrite メカニズムの原則 (ページ 283)

外部タグ (ページ 255)

WinCC 通信の原則 (ページ 248)

6.3.2 WinCC 通信の原則

概要

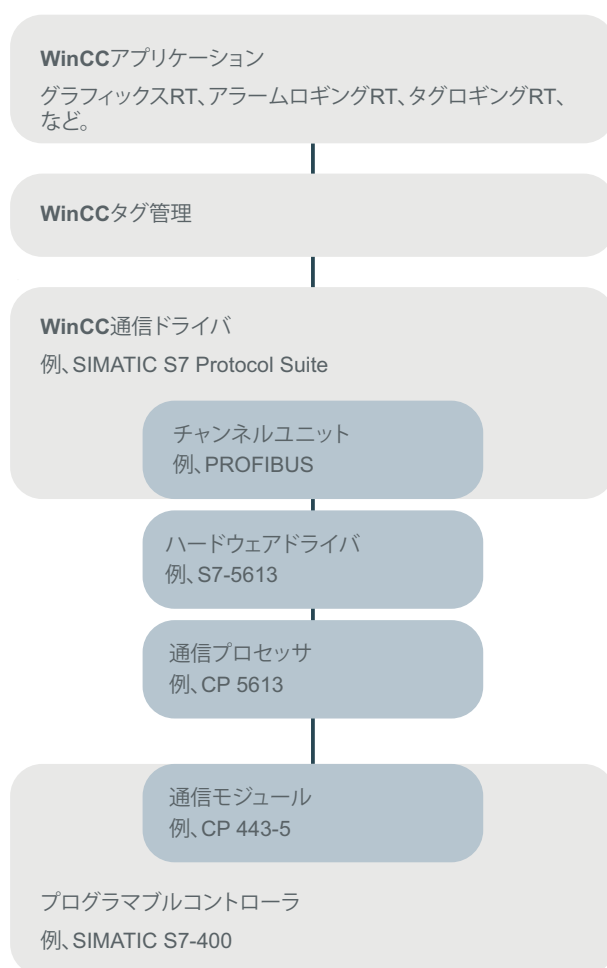
WinCC は、「タグ管理」を使ってタグを一元的に管理します。プロジェクトで作成され、プロジェクトデータベースに保存されたすべてのデータとタグは、WinCC Runtime で取得および管理されます。

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を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 では、さまざまなバスシステムを使用してさまざまな AS を接続するための通信ドライバが多数用意されています。

複数の PLC と通信する際のパフォーマンスを向上させるために、各通信ドライバを WinCC プロジェクトに複数回統合させることができます。複数の統合により、それぞれのチャンネルの名前は接尾辞によって明確に識別されます。

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。

6.3 WinCC プロセス通信

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。そのため、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られる必要があります。

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つの通信パートナーで設定された論理的な割り付けです。

すべての接続には、通信パートナー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。

さまざまなチャネルユニット機能

チャネルユニットはさまざまな機能をサポートしており、これらの機能はチャネルユニットのプロパティ内のプロパティビットによって記述されます。チェックボックスは、チャネルユニットにプロパティビットが設定されているかどうかを示します。

機能	意味
バイトアクセス	チャネルユニットはこのプロパティビットを使用して、通信相手のバイトアドレスに書き込むことができるかどうかを指定します。バイトアドレスへの書き込みは、上位バイトを変更せずにワードの下位バイトを書き込むことができ、その逆も同様であることを意味します。
ビットアクセス	チャネルユニットはこのプロパティビットを使用して、ビットを書き込むことができるかどうかを指定します。ビットの書き込みとは、他のビットを変更せずに、通信相手のメモリ領域にビットを書き込むことができることを意味します。
ウォッチドッグ	このプロパティビットを使用して、チャネルユニットは独自のライフビート監視を実行するかどうかを決定します。チャネルユニットは独自のライフビート監視機能を備えており、通信チャネルと通信相手自体の両方が通信の準備ができていることを確認します。
起動信号	チャネルユニットはこのプロパティビットを使用して、通信相手に再起動を自動的に通知するかどうかを指定します。自動起動ディスプレイにより、チャネルユニットは通信相手に再起動を通知します。
再入力	チャネルユニットはこのプロパティビットを使用して、API 関数が再入可能かどうかを指定します。再入可能チャネルユニットにより、WinCC データマネージャの複数のスレッドから同時に API 関数を呼び出すことができるようになります。オペレーティングシステムリソース(セマフォ、イベントなど)によるスレッドの必要な同期は、チャネルユニットを介して行われます。チャネルユニットが再入力可能でない場合、WinCC データマネージャはそのスレッドを同期します。
リモートタグ	このプロパティビットを使用して、チャネルユニットがランタイムモードで起動された場合に、さらに論理接続を登録できるかどうかを指定します。チャネルユニットがランタイムモードで起動され、プロパティビットが設定されている場合、新しく設定された論理接続は WinCC データマネージャによってチャネルユニットにすぐに登録されます。チャネルユニットは対応する特定の接続を作成し、論理接続がすぐに操作可能になるようにします。
オンライン接続	このプロパティビットを使用して、チャネルユニットがランタイムモードで起動された場合に、さらに論理接続を登録できるかどうかを指定します。
オンラインタグ	チャネルユニットはこのプロパティビットを使用して、タグのオンライン設定をサポートするかどうかを指定します。

機能	意味
タグの登録	チャンネルユニットはこのプロパティビットを使用して、初期化フェーズ中にユニットの論理接続に割り当てられた WinCC タグを登録するかどうかを指定します。
独自のプロパティ	このプロパティビットを使用して、チャンネルユニットは独自の内部設定値の設定が含まれているかどうかを判断します。
Intel バイトオーダー	チャンネルユニットはこのプロパティビットを使用して、プロセスデータが INTEL バイトシーケンスで通信相手に存在するかどうかを判断します。
有効/無効	このプロパティビットを使用して、チャンネルユニットは、システムタグ「@<...>@ForceConnectionStateEx」を介して接続をアクティブ化または非アクティブ化できるかどうかを指定します。

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 252)

6.3.3 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 291)」

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	用途	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。
@<...>@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

1. 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。
2. 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。
3. 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 295)

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 385)

パフォーマンスタグによる接続の確認 (ページ 291)

6.3.4 外部タグ

6.3.4.1 外部タグ

概要

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:たとえば、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エリア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と AS フォーマットを合わ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。

6.3 WinCC プロセス通信

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(たとえば、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。たとえば、値範囲[0~127]での[ByteToUnsignedDword]および[ByteToUnsignedWord]。
この場合、AS データの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**不適切な設定による通信の中断**

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

この表は、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
符号付き 16 ビット値	はい
符号なし 32 ビット値	はい

データタイプ	タイプ変換
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

タイプ変換は解釈可能である必要があります

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。

たとえば、この値がユーザーによって WinCC に入力される時 (mbar 単位)、プロセスがユニットバーでのセットポイントの設定を求めることがあります。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。

6.3 WinCC プロセス通信

これらは、特定の方法で 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、すでにインポートされたタグに影響しません。

下記も参照

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 (ページ 269)

WinCC データタイプでソートしたフォーマット調整 (ページ 260)

外部タグを以下のように設定します (ページ 259)

6.3.4.2 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ブが、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。

6.3.4.3 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. [線形スケール]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。

6.3 WinCC プロセス通信

下記も参照

BinWrite メカニズムの原則 (ページ 283)

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 (ページ 269)

WinCC データタイプでソートしたフォーマット調整 (ページ 260)

新規接続の作成方法 (ページ 258)

6.3.4.4 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 6-1 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

フォーマット調整"符号付き 8 ビット値"	値の範囲
CharToBCDWord	0...127
CharToBCDDword	0...127
CharToSignedBCDByte	-9...+9
CharToSignedBCDWord	-128...+127
CharToSignedBCDDword	-128...+127
CharToExtSignedBCDByte	-79...+79
CharToExtSignedBCDWord	-128...+127
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 6-2 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

6.3 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
ByteToBCDWord	0...255
ByteToBCDDword	0...255
ByteToAikenByte	0...99
ByteToAikenWord	0...255
ByteToAikenDword	0...255
ByteToExcessByte	0...99
ByteToExcessWord	0...255
ByteToExcessDword	0...255

表 6-3 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

フォーマット調整"符号付き 16 ビット値"	値の範囲
ShortToAikenWord	0...9999
ShortToAikenDword	0...32767
ShortToSignedAikenByte	-9...+9
ShortToSignedAikenWord	-999...+999
ShortToSignedAikenDword	-32768...+32767
ShortToExcessByte	0...99
ShortToExcessWord	0...9999
ShortToExcessDword	0...32767
ShortToSignedExcessByte	-9...+9
ShortToSignedExcessWord	-999...+999
ShortToSignedExcessDword	-32768...+32767

表 6-4 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

6.3 WinCC プロセス通信

フォーマット調整"符号なし 16 ビット値"	値の範囲
WordToSimaticBCDCounter	0...999
WordToSimaticCounter	0...999

表 6-5 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

フォーマット調整"符号付き 32 ビット値"	値の範囲
LongToSignedAikenByte	-9...+9
LongToSignedAikenWord	-999...+999
LongToSignedAikenDword	-9999999...+99999999
LongToSignedAikenQword	-999999999...+9999999999
LongToExcessByte	0...99
LongToExcessWord	0...9999
LongToExcessDword	0...99999999
LongToSignedExcessByte	-9...+9
LongToSignedExcessWord	-999...+999
LongToSignedExcessDword	-9999999...+99999999
LongToSignedExcessQword	-999999999...+9999999999
LongToSimaticBCDTimer	10...9990000
LongToSimaticTimer	10...9990000
LongToSimaticLTime	00:00:00.000...596.31.23.647

表 6-6 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

6.3 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
DwordToAikenDword	0...999999999
DwordToAikenQword	0...9999999999
DwordToExcessByte	0...99
DwordToExcessWord	0...9999
DwordToExcessDword	0...999999999
DwordToExcessQword	0...9999999999
DwordToSimaticBCDTimer	10...9990000
DwordToSimaticTimer	10...9990000
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

表 6-7 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

フォーマット調整"浮動小数点数 32 ビット IEEE 754"	値の範囲
FloatToExtSignedBCDWord	-7999...+7999
FloatToExtSignedBCDDword	-7.999999e+07...+7.999999e+07
FloatToAikenByte	0...99
FloatToAikenWord	0...9999
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 6-8 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

6.3 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
DoubleToFloat	+3.402823e+38
DoubleToMSBByte	-127...+127
DoubleToMSBWord	-32767...+32767
DoubleToMSBDword	-2147483647...+2147483647
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
DoubleToBCDByte	0...99
DoubleToBCDWord	0...9999
DoubleToBCDDword	0...99999999
DoubleToBCDQword	0...9999999999999999
DoubleToSignedBCDByte	-9...+9
DoubleToSignedBCDWord	-999...+999
DoubleToSignedBCDDword	-9999999...+9999999
DoubleToSignedBCDQword	-9999999999999999...+9999999999999999
DoubleToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDWord	-7999...+7999
DoubleToExtSignedBCDDword	-79999999...+79999999
DoubleToExtSignedBCDQword	-7999999999999999...+7999999999999999
DoubleToAikenByte	0...99
DoubleToAikenWord	0...9999
DoubleToAikenDword	0...99999999
DoubleToAikenQword	0...9999999999999999
DoubleToSignedAikenByte	-9...+9
DoubleToSignedAikenWord	-999...+999
DoubleToSignedAikenDword	-9999999...+9999999
DoubleToSignedAikenQword	-9999999999999999...+9999999999999999
DoubleToExcessByte	0...99
DoubleToExcessWord	0...9999
DoubleToExcessDword	0...99999999
DoubleToExcessQword	0...9999999999999999
DoubleToSignedExcessByte	-9...+9

フォーマット調整"浮動小数点数 64 ビット IEEE 754"	値の範囲
DoubleToSignedExcessWord	-999...+999
DoubleToSignedExcessDword	-9999999...+9999999
DoubleToSignedExcessQword	-999999999999999...+999999999999999
DoubleToSimaticBCDTimer	10...9990000
DoubleToS5Float	+/-1.701411e+38
DoubleToSimaticTimer	10...9990000

6.3.4.5 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 6-9 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

6.3 WinCC プロセス通信

表 6-10 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 6-11 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...9,9999999e+07
LongToAikenDword	0...999999999
ShortToAikenDword	0...32767
WordToAikenDword	0...65535

表 6-12 AikenQWord

タイプ変換"AikenQWord"	値の範囲
DoubleToAikenQword	0...9999999999999999
DwordToAikenQword	0...9999999999

表 6-13 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 6-14 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 6-15 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

6.3 WinCC プロセス通信

タイプ変換"BCDDWord"	値の範囲
ShortToBCDDword	0...32767
WordToBCDDword	0...65535

表 6-16 BCDQWord

タイプ変換"BCDQWord"	値の範囲
DoubleToBCDQword	0...9999999999999999
DwordToBCDQword	0...999999999

表 6-17 Double

タイプ変換"Double"	値の範囲
DoubleToDouble	+/-1.79769313486231e+308 (変換なし)
FloatToDouble	+/-3.402823e+38

表 6-18 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 6-19 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 6-20 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 6-21 ExcessQWord

タイプ変換"ExcessQWord"	値の範囲
DoubleToExcessQword	0...999999999999999999
DwordToExcessQword	0...9999999999

表 6-22 ExtSignedBCDByte

タイプ変換"ExtSignedBCDByte"	値の範囲
CharToExtSignedBCDByte	-79...+79
DoubleToExtSignedBCDByte	-79...+79
FloatToExtSignedBCDByte	-79...+79

6.3 WinCC プロセス通信

タイプ変換"ExtSignedBCDByte"	値の範囲
LongToExtSignedBCDByte	-79..+79
ShortToExtSignedBCDByte	-79...+79

表 6-23 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 6-24 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 6-25 ExtSignedBCDQWord

タイプ変換"ExtSignedBCDQWord"	値の範囲
DoubleToExtSignedBCDQword	-7999999999999999...+7999999999999999
LongToExtSignedBCDQword	-79999999...+79999999

表 6-26 Float

タイプ変換"Float"	値の範囲
DoubleToFloat	+/-3.402823e+38
FloatToFloat	+/-3.402823e+38 (変換なし)

表 6-27 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 6-28 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 6-29 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 6-30 MSBQWord

タイプ変換"MSBQWord"	値の範囲
DoubleToMSBQword	-9223372036854775808...+9223372036854775808
LongToMSBQword	-2147483648...+2147483647

6.3 WinCC プロセス通信

表 6-31 S5Float

タイプ変換"S5Float"	値の範囲
DoubleToS5Float	+/-1.701411e+38
FloatToS5Float	+/-1.701411e+38

表 6-32 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 6-33 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 6-34 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 6-35 SignedQWord

タイプ変換"SignedQWord"	値の範囲
DoubleToSignedQword	-9223372036854775808...+9223372036854775808
LongToSignedQword	-2147483648...+2147483647

表 6-36 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 6-37 SignedAikenWord

タイプ変換"SignedAikenWord"	値の範囲
CharToSignedAikenWord	-128...+127
DoubleToSignedAikenWord	-999...+999

6.3 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 6-38 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 6-39 SignedAikenQWord

タイプ変換" SignedAikenQWord "	値の範囲
DoubleToSignedAikenQword	-999999999999999...+999999999999999
LongToSignedAikenQword	-999999999...+999999999

表 6-40 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 6-41 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 6-42 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 6-43 SignedBCDQWord

タイプ変換" SignedBCDQWord "	値の範囲
DoubleToSignedBCDQword	-999999999999999...+999999999999999
LongToSignedBCDQword	-2147483648...+2147483647

表 6-44 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 6-45 SignedExcessWord

タイプ変換" SignedExcessWord "	値の範囲
CharToSignedExcessWord	-128...+127
DoubleToSignedExcessWord	-999...+999

6.3 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 6-46 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 6-47 SignedExcessQWord

タイプ変換" SignedExcessQWord "	値の範囲
DoubleToSignedExcessQword	-999999999999999...+999999999999999
LongToSignedExcessQword	-999999999...+999999999

表 6-48 SimaticCounter

タイプ変換" SimaticCounter "	値の範囲
WordToSimaticCounter	0...999

表 6-49 SimaticBCDCounter

タイプ変換" SimaticBCDCounter "	値の範囲
WordToSimaticBCDCounter	0...999

表 6-50 SimaticTimer

タイプ変換"SimaticTimer"	値の範囲
DoubleToSimaticTimer	10...9990000
DwordToSimaticTimer	10...9990000
FloatToSimaticTimer	10...9990000
LongToSimaticTimer	10...9990000

表 6-51 SimaticBCDTimer

タイプ変換"SimaticBCDTimer"	値の範囲
DoubleToSimaticBCDTimer	10...9990000
DwordToSimaticBCDTimer	10...9990000
FloatToSimaticBCDTimer	10...9990000
LongToSimaticBCDTimer	10...9990000

表 6-52 SimaticLTime

タイプ変換"SimaticLTime"	値の範囲
LongToSimaticLTime	00:00:00.000...596.31.23.647

表 6-53 SimaticLTimeOfDay

タイプ変換"SimaticLTimeOfDay"	値の範囲
DwordToSimaticLTimeOfDay	00:00:00.000...23.59.59.999

表 6-54 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

6.3 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 6-55 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 6-56 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 6-57 UnsignedQWord

タイプ変換"UnsignedQWord"	値の範囲
DoubleToUnsignedQword	0...18446744073709551616
DwordToUnsignedQword	0...4294967295

6.3.4.6 BinWrite メカニズムの原則

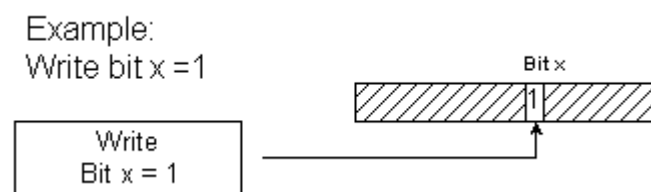
はじめに

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

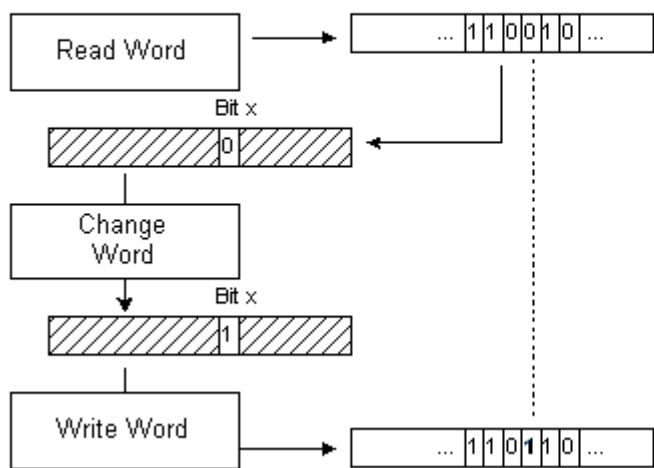
6.3 WinCC プロセス通信

- SIMATIC TI Ethernet 第 4 層
- SIMATIC TI Serial

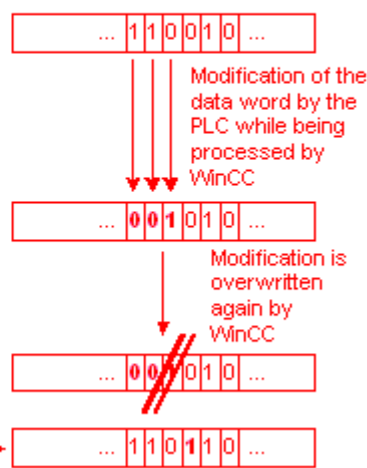
ビットまたはバイトを書き込むには、チャンネルユニットは、まず BinWrite メカニズムでデータワード全体を読み取ります。アドレス指定対象のデータが、読み込まれたワードで変更されます。次に、変更されたビットまたはバイトではなく、ワード全体(!)が書き戻されます。

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。

Example:
Write bit x = 1



Problem Case



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。

下記も参照

BinWrite 付きタグの設定方法 (ページ 285)

6.3.4.7 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

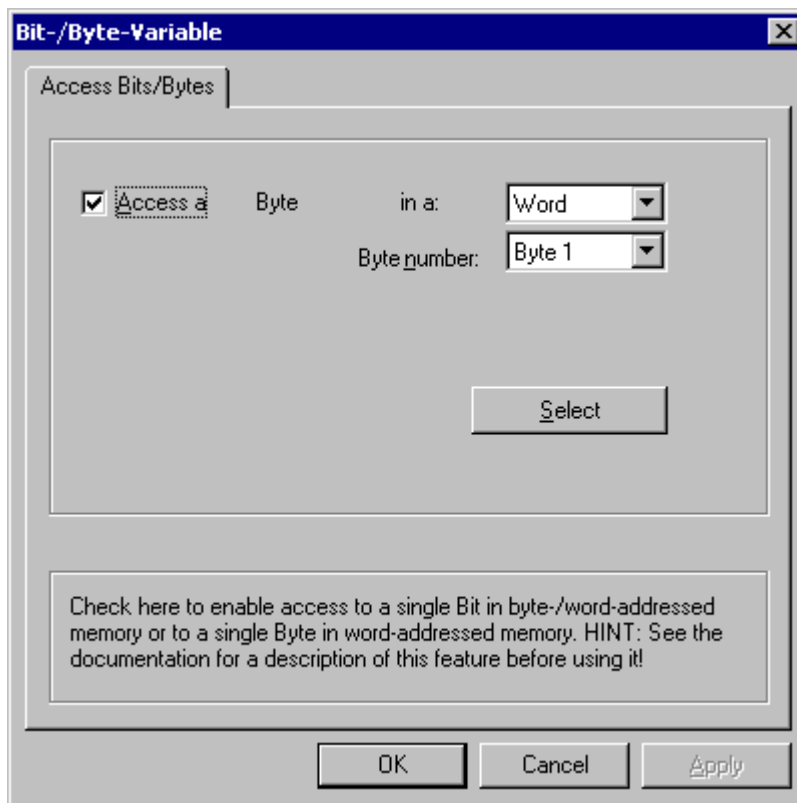
- 必要な通信プロセッサとハードウェアドライバがインストールされている。
- 例えば「SIMATIC S5 Ethernet Layer 4」のような、必要な通信ドライバがインストールされている。ただし、これはビットアクセス/バイトアクセスをサポートしません。
- そのチャンネルユニットに基いて、接続が既に作成されている。

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」

6.3 WinCC プロセス通信

- [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。



- [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。
- [OK]をクリックしてダイアログを閉じます。

下記も参照


外部タグを以下のように設定します (ページ 259)

新規接続の作成方法 (ページ 258)

外部タグ (ページ 255)

6.3.5 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	

6.3.6 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。

6.3 WinCC プロセス通信

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－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 342)

タグの品質コード (ページ 374)

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 387)

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 385)

内部タグの作成方法 (ページ 388)

チャンネル診断 (ページ 291)

エラー検出に関する一般情報 (ページ 290)

"OPC"チャンネルの診断のための機能 (ページ 353)

"SIMATIC S7 Protocol Suite"チャンネル - 診断オプション (ページ 329)

"システム情報"チャンネル - 診断オプション (ページ 314)

診断チャンネル「SIMATIC S7-1200/S7-1500」 (ページ 319)

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 295)

チャンネル診断の原理
(ページ 297)

パフォーマンスタグによる接続の確認 (ページ 291)

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特定します。詳細情報:

- 「[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 295)」
- 「ランタイム時の接続ステータスタグの設定 (ページ 252)」

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。

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。

7.3 チャンネル診断

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。 リセットタグはこれらの値に影響を与えません。 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 リセットタグを使用して、値を「0」にリセットできます。 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

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。

7.3 チャンネル診断

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。

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 252)

[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 295)

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。

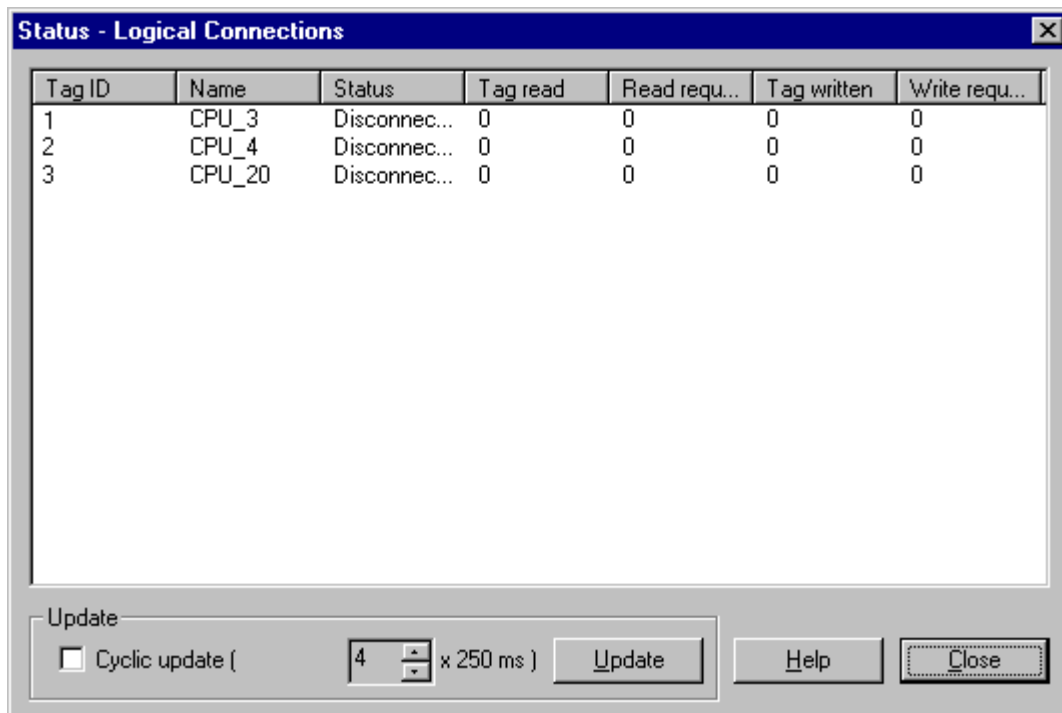
特定の接続が確立または終了しているかどうかだけを照会する場合は、システムタグ「@<Connectionname>@ConnectionStateEx」を使用します。

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 252)

パフォーマンススタグによる接続の確認 (ページ 291)

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

概要

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/統計情報の出力
- 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 Microsoft 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>という名前と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できるだけ多く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 298)

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 306)

[チャンネル診断]を使用したチャンネルの診断 (ページ 309)

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 310)

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 311)

ランタイムの開始方法 (ページ 313)

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。

ActiveX コントロールはグラフィックデザイナーの[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 310)

[チャンネル診断]を使用したチャンネルの診断 (ページ 309)

7.3.4.3 「チャンネル診断」による「SIMATIC S7-1200/S7-1500」診断チャネルの評価

概要

このセクションでは、「チャンネル診断」で「SIMATIC S7-1200/S7-1500」チャネルを評価するためのカウンタと値について説明します。

説明

Status	Name	Counters	Value
✓	SIMATIC S7-1200, S7-1500 Channel	Connection State	ready
✓	PLC41	Plc Address	190.99.180.141
		EntryPoint	AS_Network
		Plc Attributes (free/max)	39936 / 40000
		Plc Subscriptions (free/max)	742 / 750
		Subscription Memory (free/max)	2097152 / 2097152
		Max tags per request (read/write)	100 / 100
		Plc Operating State	Run
		Plc Tag Subscriptions	500 ms (8)
		Plc Tag Polling	5 s (5), 10 s (1)
		Connection Aborts	0
		Plc Protection-Level	no protection, full access
		Hmi Protection-Level	no protection, full access
		Read Duration of Plc Tags (min/last/max)	1 ms (5) / 2 ms (1) / 134 ms (5)
		Write Duration of Plc Tags (min/last/max)	0 ms (0) / 0 ms (0) / 0 ms (0)
		Active services	DataAccess
		Force Connection State	enable
		Fcs Connections	1
		Fcs Disconnections	0
		Security State	TLS not active

1 second 3:28:05 PM

7.3 チャンネル診断

ステータスメッセージの概要

シンボル	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。

カウンタと値の説明

カウンタ	説明	値(例:)
Connection State	チャンネルの接続状態	disconnected, init, ready
Plc Address	設定されたアドレス(例:IP アドレス)	192.168.100.100
EntryPoint	エントリポイント	CP-TCPIP, S7ONLINE, AS_Network
Plc Attributes (free/max)	<ul style="list-style-type: none"> 属性参照の無料サブスクリプション数 属性参照のサブスクリプションの最大数 [Plc Attributes (max)]は、5 秒未満のサイクルでサブスクリプションの形式でリクエストできるタグの最大数です。	39941 / 40000
Plc Subscriptions (free/max)	<ul style="list-style-type: none"> 無料サブスクリプション サポートされるサブスクリプションの最大数 	742 / 750
Subscriptions Memory (free/max)	<ul style="list-style-type: none"> サブスクリプション用の無料ストレージスペース サブスクリプションの最大ストレージ容量 	2097152 / 2097152
Max tags per request (read/write)	読み取り/書き込み要求あたりのタグの最大数	100 / 100
Plc Operation State	PLC 操作モード	Run, Stop, Startup, ...
Plc Tag Subscriptions	PLC タグサブスクリプション <ul style="list-style-type: none"> リクエストサイクルはミリ秒単位で指定されます。 タグの数は括弧内に表示されます。 	500 ms (8)

カウンタ	説明	値(例:)
Plc Tag Polling	<p>サイクリック PLC タグ要求</p> <ul style="list-style-type: none"> リクエストサイクルは秒単位で指定されます。 タグの数は括弧内に表示されます。 <p>説明: 「チャンネル診断」では、次の5つのタグが5秒周期で要求されます。</p> <ul style="list-style-type: none"> 無料属性 無料サブスクリプション 動作状態 保護レベル サブスクリプションの現在のストレージ容量 <p>タグは10秒周期で要求されます。このシステム診断値は内部使用を目的としています。</p>	5 s (5), 10 s (1)
Connection Aborts	チャンネルの運用開始以来の切断累計数	0
Plc Protection-Level	<p>設定された PLC 保護レベル</p> <p>TIA Portal/STEP 7 のコントロールを設定する際は、[Plc Protection-Level]設定を指定できます。</p>	no protection, full access
HMI Protection-Level	設定された(合法化された) HMI 保護レベル	no protection, full access
Read Duration of Plc Tags (min/last/max)	<p>PLC タグの読み取り期間</p> <ul style="list-style-type: none"> 最小期間 最終読み取り 最大期間 	1 ms (5) / 1 ms (5) / 946 ms (5)
Write Duration of Plc Tags (min/last/max)	<p>PLC タグの書き込み期間</p> <ul style="list-style-type: none"> 最小期間 最近の読み取り 最大期間 	0 ms (0) / 0 ms (0) / 0 ms (0)
Active Services	<p>アクティブサービス</p> <p>この値は、アクティブな証明書が利用可能かどうかを示します。</p>	DataAccess, Alarming, Diagnosis
Force Connection State	優先接続ステータス	enable
Fcs Connections	[Force Connection State]接続の確立	1

7.3 チャンネル診断

カウンタ	説明	値(例:)
Fcs Disconnections	[Force Connection State]接続の切断	0
Security State	セキュリティステータス <ul style="list-style-type: none"> • TLS は有効化または無効化されている 	TLS not active

下記も参照

[チャンネル診断]によるチャンネルの診断
(ページ 297)

7.3.4.4 「チャンネル診断」による「OPC」診断チャネルの評価

概要

このセクションでは、「チャンネル診断」で「OPC」チャネルを評価するためのカウンタと値について説明します。

説明

Status	Name	Counters	Value
	OPC	Requested Tags	0
	NewConnection_2	Data Change Notifications	0
		Data Change Notification Tags	0
		Synchronous Reads	0
		Synchronous Writes	0
		Synchronous Read Tags	0
		Synchronous Write Tags	0
		AddItem failures	0
		OPC Groups	0
		Items added	0
		Server uses DataAccess	
		Server Status	
		ForcedConnectionStateEx	ENABLED
		Current Activity	
		Last Error Time	2024-11-21 16:11:26,228
		Last Error	0x800401f3
		Last Error Name	CO_E_CLASSSTRING
		Last Error in Call	GetServerClassID
		Server Info	
		Locale-ID	
		Maximum Time between CBs	
		Last Time between CBs	

1 second 4:11:27 PM

ステータスメッセージの概要

シンボル	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。

カウンタと値の説明

カウンタ	説明	値
Requested Tags	読み取り操作で受信した要求タグ値の数	0
Data Change Notifications	受信したデータ変更通知の数	0

7.3 チャンネル診断

カウンタ	説明	値
Data Change Notification Tags	データ変更通知のタグの数	0
Synchronous Reads	受信した同期読み取り要求の数	0
Synchronous Writes	受信した同期書き込み要求の数	0
Synchronous Read Tags	同期読み取りから受信したタグの合計数	0
Synchronous Write Tags	同期書き込みから受信したタグの合計数	0
AddItem failures	OPC グループ内の失敗したエントリの数	0
OPC Groups	作成された OPC グループの数	0
Items added	すべての OPC グループのエントリの総数	0
Server uses DataAccess	サーバー上の OPC DA バージョン	V1.0(A), V2.0X, V3.0
Server Status	サーバーステータス	-
ForcedConnectionStateEx	接続ステータス:接続がアクティブ化されているか無効化されているかを示します。	ENABLED, DISABLED
Current Activity	現在の活動	-
Last Error Time	最後のエラーのタイムスタンプ	2024-11-21 16:11:26,228
Last Error	最後のエラー	0x800401f3
Last Error Name	最後のエラーの名前	CO_E_CLASSTRING
Last Error in Call	最後にエラーが発生した場所を要求	GetServerClassID
Server Info	サーバー情報	-
Maximum Time between CBs	コールバック間の最大時間	-
Last Time between CBs	コールバック間の最後の期間	-

7.3.4.5 「チャンネル診断」による「OPC UA」診断チャンネルの評価

概要

このセクションでは、「チャンネル診断」で「OPC UA」チャンネルを評価するためのカウンタと値について説明します。

説明

Status	Name	Counters	Value
	OPC UA WinCC Channel	Server Status	
	WCC-INT01-VM05	ForcedConnectionStateEx	ENABLED
		Data Change Notifications	0
		Data Change Notification Tags	0
		Synchronous Reads	0
		Asynchronous Writes	0
		Synchronous Read Tags	0
		Asynchronous Write Tags	0
		Asynchronous Reads	0
		Asynchronous Read Tags	0
		Last Activity	Get Connection Status
		Last Error Time	2024-11-21 16:07:05,017
		Last Error	0x80004005
		Last Error Name	Unspecified error
		Last Error in Call	RegisterVariables
		LastDataChangeCallBack time	
		Last Time between CBs	

1 second 4:18:13 PM

ステータスメッセージの概要

シンボル	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。

カウンタと値の説明

カウンタ	説明	値
Server Status	接続ステータス	RUNNING, FAIL, NOCONFIG, SUSPENDED, COMM_FAULT
ForcedConnectionStateEx	接続ステータス	ENABLED, DISABLED
Data Change Notifications	受信したデータ変更通知の数	0

7.3 チャンネル診断

カウンタ	説明	値
Data Change Notification Tags	データ変更通知からのタグの数	0
Synchronous Reads	受信した周期的な読み取り数(サブスクリプション)	0
Asynchronous Writes	受信した非同期書き込み操作の数	0
Synchronous Read Tags	すべてのサブスクリプションの周期的に受信された読み取りタグの合計数	0
Asynchronous Write Tags	非同期書き込みから受信したタグの合計数	0
Asynchronous Reads	受信した非同期読み取り操作の数	0
Asynchronous Read Tags	非同期読み取りから受信したタグの合計数	0
Last Activity	最後に受信したリクエスト	StartRead, Cyclic Read, Read Async, Get Connection Status, Stop Read
Last Error Time	最後のエラーのタイムスタンプ	2024-11-21 16:07:05,017
Last Error	最後のエラー	0x80004005
Last Error Name	最後のエラーの名前	Unspecified error
Last Error in Call	最後にエラーが発生した場所を要求	RegisterVariables
LastDataChangeCall Back time	データ変更の最後のコールバックのタイムスタンプ	-
Last Time between CBs	コールバック間の最終時間	-

7.3.4.6 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

概要

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。

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. [グラフィックデザイナー]を起動し、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伸ばすことにより実行できます。
3. 画面を保存します。
4. グラフィックデザイナーツールバーから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。
チャンネルのステータス情報が、[チャンネル診断]アプリケーションウィンドウに表示されます。

Status	Name	Counters	Value
✓	SIMATIC S7-1200, S7-1500 Channel	Connection State	ready
✓	PLC41	Plc Address	190.99.180.141
		EntryPoint	AS_Network
		Plc Attributes (free/max)	39936 / 40000
		Plc Subscriptions (free/max)	742 / 750
		Subscription Memory (free/max)	2097152 / 2097152
		Max tags per request (read/write)	100 / 100
		Plc Operating State	Run
		Plc Tag Subscriptions	500 ms (8)
		Plc Tag Polling	5 s (5), 10 s (1)
		Connection Aborts	0
		Plc Protection-Level	no protection, full access
		Hmi Protection-Level	no protection, full access
		Read Duration of Plc Tags (min/last/max)	1 ms (5) / 2 ms (1) / 134 ms (5)
		Write Duration of Plc Tags (min/last/max)	0 ms (0) / 0 ms (0) / 0 ms (0)
		Active services	DataAccess
		Force Connection State	enable
		Fcs Connections	1
		Fcs Disconnections	0
		Security State	TLS not active

1 second 3:28:05 PM

6. コントロールの歯車アイコンをクリックします。
[設定]ダイアログが現れます。
7. ドロップダウンリストから[チャンネル名]を選択します。
8. ログファイルの名前を指定します。
9. 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。

7.3 チャンネル診断

トレースファンクションの有効化に関する詳細情報は「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 311)」を参照してください。

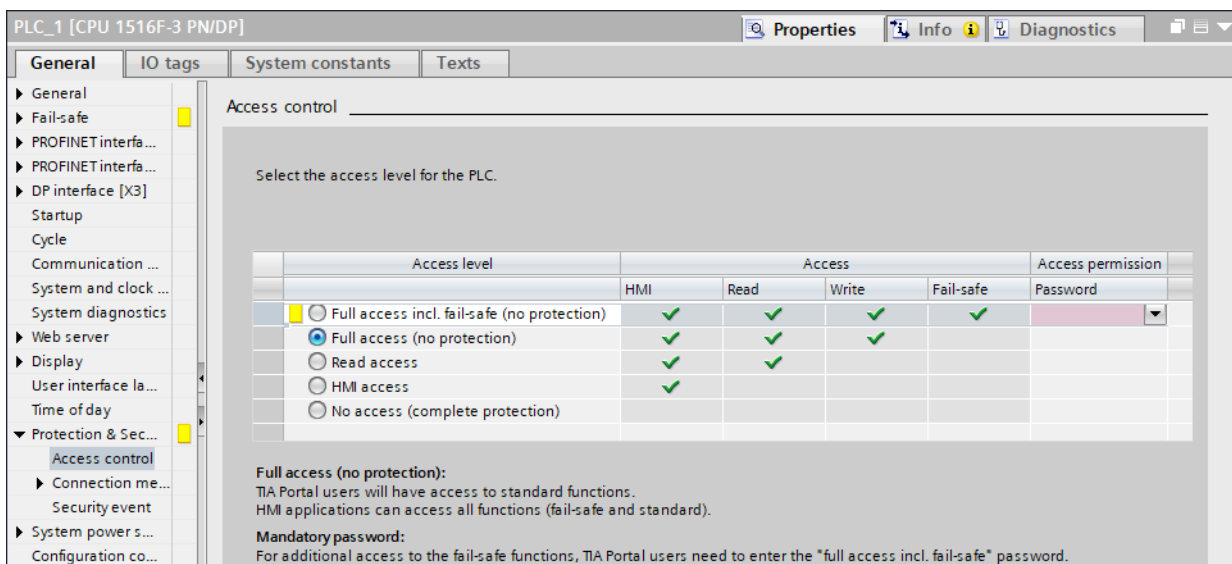
注記

[チャンネル診断] - 範囲

[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。

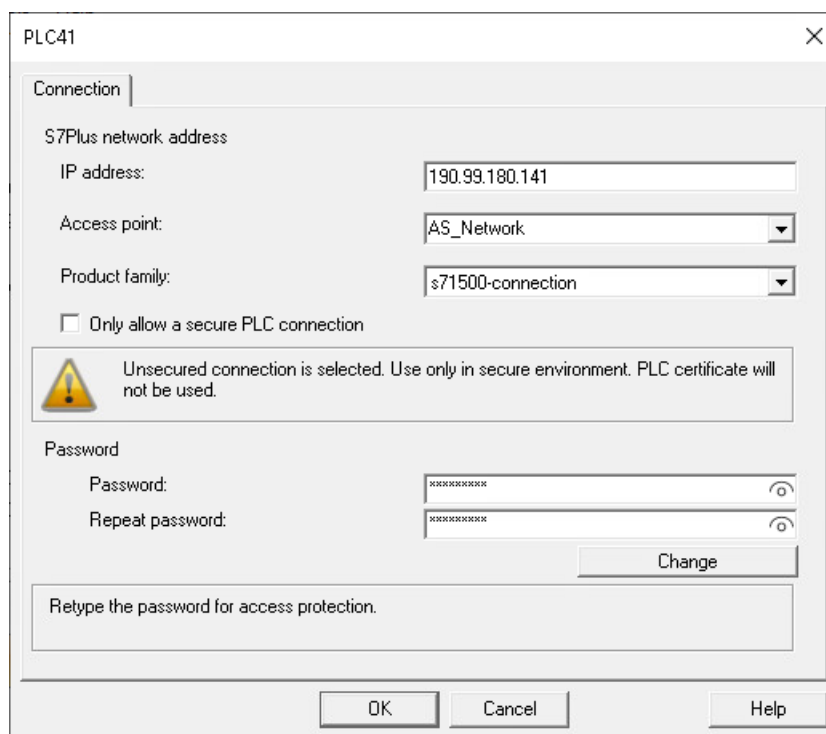
保護とセキュリティ

TIA Portal/STEP 7 のコントロールを設定する際は、[Plc Protection-Level]設定を指定できません。



接続の保護のためにパスワードを割り付ける際の追加情報:

- [通信] > SIMATIC S7-1200, S7-1500 Channel > [チャンネル設定] > [AUTOHOTSPOT]



パスワードを割り当てないと、診断に[パスワードが無効か保護が完了しています]というメッセージが表示されます。

PLC の保護レベルが使用されています。

両方の通信パートナーに同じパスワードを割り当てると、状態[保護がありません。フルアクセスです]というステータスが表示されます。これは、HMI デバイスは PLC にフルアクセスできるものの、PLC への接続がパスワードで保護されていることを意味します。

この状態は表示されているチャンネルとそのパートナーに対してのみ表示されます。これは、HMI デバイスの一般保護等級について言及しません。

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 311)

7.3.4.7 [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。

7.3 チャンネル診断

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せん。

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 310)

ActiveX コントロールによるチャンネル診断
(ページ 298)

7.3.4.8 チャンネル診断によるチャンネルのチェック方法

概要

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。
チャンネルのステータス情報が、[チャンネル診断]アプリケーションウィンドウに表示されます。
4. [チャンネル診断]アプリケーションウィンドウの歯車アイコンをクリックします。
[設定]ダイアログが現れます。

5. ドロップダウンリストから[チャンネル名]を選択します。
6. ログファイルの名前を指定します。

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 311)

ランタイムの開始方法 (ページ 313)

ActiveX コントロールによるチャンネル診断
(ページ 298)

[チャンネル診断]を使用したチャンネルの診断 (ページ 309)

7.3.4.9 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

概要

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し、有効にする方法を説明します。

通信ステータスとエラーに関する詳細情報は、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

詳細については、下記の FAQ を参照してください。

- 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)
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. [チャンネル診断]アプリケーションウィンドウの歯車アイコンをクリックします。
[設定]ダイアログが現れます。
3. 適切なチャンネルを選択します。
4. [フラグの設定]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。
5. [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。
6. 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。
7. [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。
8. [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。
9. [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。
10. (ファイルの最大数およびファイルサイズに達すると、古いものから始め)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。
11. [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 358)

チャンネルおよび接続のチェック方法 (ページ 360)

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>)

7.3.4.10 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 310)

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 318)

チャンネルおよび接続のチェック方法 (ページ 316)

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 318)

チャンネルおよび接続のチェック方法 (ページ 316)

7.4.3.2 チャンネルおよび接続のチェック方法





概要

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。

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

シンボル	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。

Status	Name	Counters	Value
✓	System Info	State	ready
✓	NewConnection_1	Invalid Tag	none
		Address Error	none
		Size Error	none
		Type Error	none
		Error Count	0
		PrinterName	none
		PrinterUserName	none
		Print State	0x0
		Print Jobs Count	0
		Print Job State	0x0
		Print Job Total Pages	0
		Print Job Pages Printed	0
		Print PRTOUTDirectory	none
		Print SpoolDirectory	none

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 311)

ログファイルのエントリの説明 (ページ 314)

タグのチェック方法 (ページ 318)

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 374)

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。

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。

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	

注記

マルチクライアントシステムでの表示

クライアントがサーバーからサーバーパッケージをダウンロードし、ランタイムで起動すると、SysDiagControl が PLC の情報だけでなく、使用されているサーバーパッケージに関する情報も表示します。

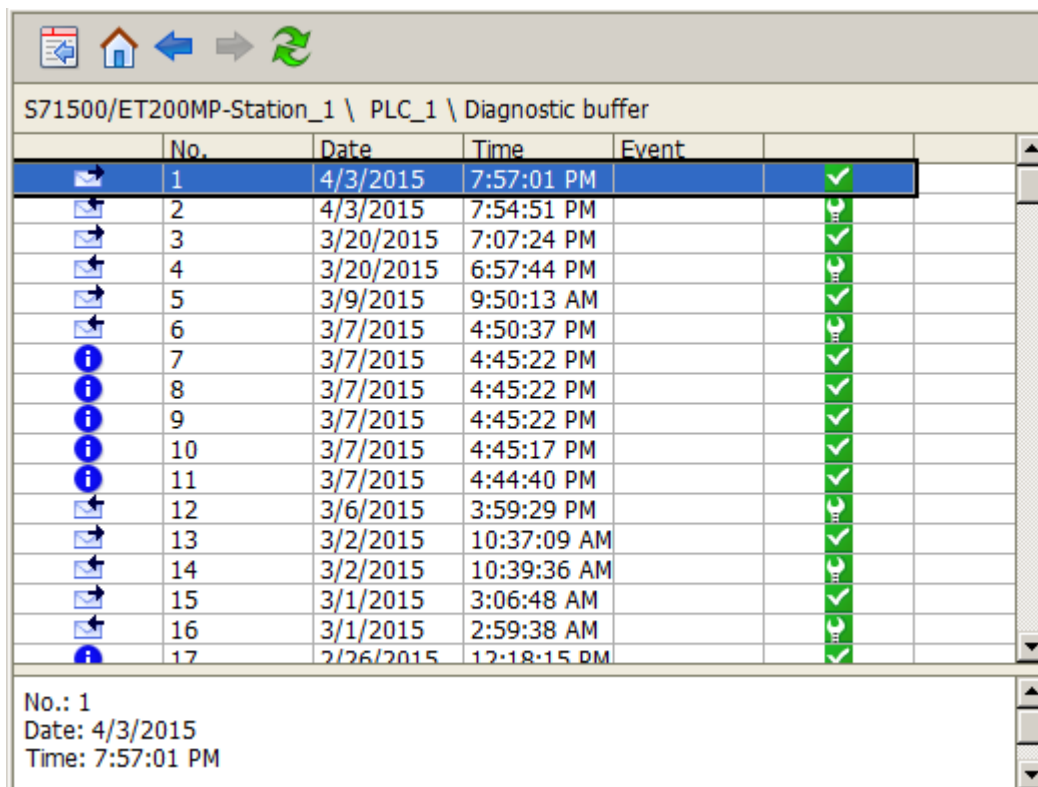
診断バッファ表示

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。

7.5 診断チャンネル「SIMATIC S7-1200/S7-1500」



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。



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。

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

システム診断の構成方法 (ページ 323)

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せん。

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. 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 319)

7.5.3 SysDiagControl の設定の情報

コントロールはこの情報をどこで取得するのでしょうか?

SysDiagControl で表示された情報は、コントローラの設定に基づいています。

ダイアログ[WinCC SysDiagControl Properties]の設定が作成されます。



このダイアログは画像へコントロールを挿入する際に表示されます。

コントロールがすでに挿入されている場合、コントロールをダブルクリックして、ダイアログを開きます。

詳細情報: 「システム診断の構成方法 (ページ 323)」。

コントローラは、データを対応する PLC から取得するために、どの接続が考慮されているかを把握する必要があります。

詳細は、「システム診断を設定する方法」セクションの「必要条件」を参照してください。

ランタイムが有効なときには、必要な情報は設定データから取得されます。PLC との通信を確立するために、設定データが使用されます。PLC との接続が確立されるとすぐに、SysDiagControl が必要な情報を PLC から取得するようにリクエストし、コントロールでそれを表示します。

PLC との通信がランタイムのアクティベーション時に失敗すると、設定で利用可能なデータに対応する個別のデータが表示されます。

PLC への接続が確立されるまで、ステーション名などの情報が設定データから取得されません。接続が確立されるとすぐに、PLC から新しいデータを利用することができるようになり、コントロールで自動的に表示されるようになります。

ランタイムでは、接続が確立されるとすぐに、コントロールで表示するデータが PLC から取得されます。コントロールで表示される情報のほとんどは、PLC から直接取得されます。例外は、TextList から抽出する必要のあるデータです。

既存のテストリストは再ロードするのでしょうか?あるいはランタイムは再起動するのでしょうか?

AS テキストリストは、手動、あるいは[自動更新]設定で更新する必要があります。AS テキストリストがプロジェクトで更新されるとすぐに、SysDiagControl のある画像を再ロードする必要があります。

以下に WinCC 情報の基本設定の詳細情報が記載されています:

- [WinCC での作業]>[メッセージシステムの設定]>[メッセージシステムの設定]>[AS メッセージでの作業]
- [WinCC での作業]>[プロジェクトでの作業]>プロジェクトの作成および編集>[オンライン変更のロード]>[オンライン変更のロードの使用と制限]
- [設定]>[マルチユーザーシステム]:
 - [クライアント設定]>[インポートパッケージの設定 (ページ 50)]
 - [サーバーの設定]>[パッケージエクスポートの設定方法 (ページ 37)]

ランタイムが有効な際に、接続を再度作成するために、どのようなステップが必要でしょうか?

ランタイムが有効化された際、[CCSystemDiagnosticsHost]サービスが設定ファイルを開始時に 1 回のみ読み取ります。

新しい接続を作成するとき、設定データが変更され、[ダウンロード済みのデルタ]として見なされます。システム診断は現在、[ダウンロード済みのデルタ]をサポートしていません。AS メッセージのダウンロードのための設定は、準拠する必要があります。

AS メッセージ/テキストリンクの更新のためには、要件に従って、ステップに従います。新しい接続を SysDiagControl に反映させるためには、ランタイムを再起動する必要があります。

ランタイムのロード後に接続を変更した場合は、何をする必要がありますでしょうか？

シナリオ

- ファイルのロードのある新しい接続が作成されました。
- すべての AS テキストリストが、ランタイムで有効化され、ロードされています。
- PLC はまだ利用可能ではありません。
- この接続は、PLC 1516 から PLC 1516F に変更します。

[PLC 1516 から PLC 1516F に変更中]は何を意味しているのでしょうか？

接続が作成され、PLC 名が PLC 1516F に変更された場合は、この変更を反映するために、PLC ヘダダウンロードする必要があります。正常なダウンロードの後に、接続がすでに確立されている場合は、PLC におけるそのような変更は SysDiagControl に報告されます。

接続がまだ確立されていない場合、まだ古い名前のある設定データの情報が考慮されます。設定データが開始時に取得され、追加の変更が排除されるためです。接続が今確立されると、SysDiagControl が PLC からデータを直接取得し、最新データを取得します。

接続が再確立されていない場合でも、ランタイムの再起動では変更が反映されます。変更が設定データにすでに存在するためです。

下記も参照

システム診断の構成方法 (ページ 323)

インポートパッケージのコンフィグレーション (ページ 50)

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 37)

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 340)

チャンネルおよび接続のチェック方法 (ページ 338)

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 336)

SIMATIC NET での通信プロセッサのチェック (ページ 335)

通信プロセッサの設定確認方法 (ページ 333)

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

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 340)

チャンネルおよび接続のチェック方法 (ページ 338)

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 336)

SIMATIC NET での通信プロセッサのチェック (ページ 335)

通信プロセッサの設定確認方法 (ページ 333)

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 336)

SIMATIC NET での通信プロセッサのチェック (ページ 335)

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 336)

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

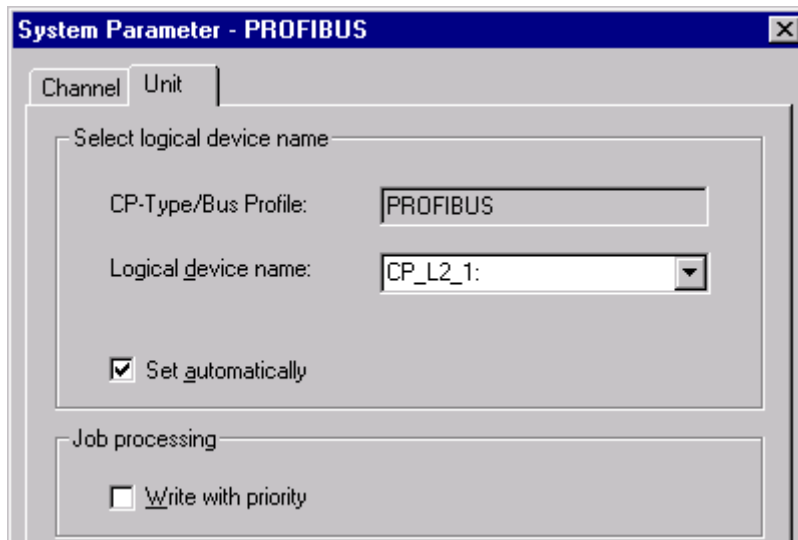
このセクションでは、システムパラメータと、接続およびタグの設定をチェックする方法について説明します。この例では、PROFIBUS 通信に「CP 5613 A3」通信プロセッサを使用します。

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 338)

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





概要

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

シンボル	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。

Status	Name	Counters	Value
<input checked="" type="checkbox"/>	SIMATIC S7 Protocol Suite	State	ready
<input checked="" type="checkbox"/>	PLC60	First Error Code	none
		Last Error Code	none
		Error count	0
		Unit	TCP/IP
		Device	AS_Network
		PLC Cycle Management	on
		Change Driven Transfer	on
		Monitoring CPU Stop	on
		Lifebeat Monitoring	60 s / 30 s
		Connection Reference	0401
		PDU size	480
		PLC Flags	0607
		Request Queue Size	0
		Request Counter	8
		Response counter	306
		Own cycles	0
		AS cycles	1
		Max. PLC Cycles	32
		Cycle Overflow	0
		VMD State	run
		ForcedConnectionState	enabled
		ConnectionState	ready
		ConnectionEstablishMode	automatic
		ForceConnectionState	up
		ForceConnectionAddress	projected
		ProjectedConnectionAddress	IP,190.99.180.160,,0,3,02
		AlternateConnectionAddress	IP,190.99.180.160,,0,3,02
		Write Response	no measurement
		Read Response	no measurement
		Cycle Create Response	no measurement
		Cycle Delete Response	no measurement
		Cycle 500ms	var=6, req=1, owl=0
1 second			3:20:35 PM

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、[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。

7.6 "SIMATIC S7 Protocol Suite"チャンネルの診断

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ーを確認します。詳細については、「ログファイルのエントリーの説明」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。

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 311)

ログファイルのエントリーの説明 (ページ 330)

タグのチェック方法 (ページ 340)

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。
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 374)

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 351)

チャンネルおよび接続のチェック方法 (ページ 349)

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 347)

SIMATIC NET での通信プロセッサのチェック (ページ 346)

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 344)

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 347)

タグのチェック方法 (ページ 351)

チャンネルおよび接続のチェック方法 (ページ 349)

SIMATIC NET での通信プロセッサのチェック (ページ 346)

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 344)

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。

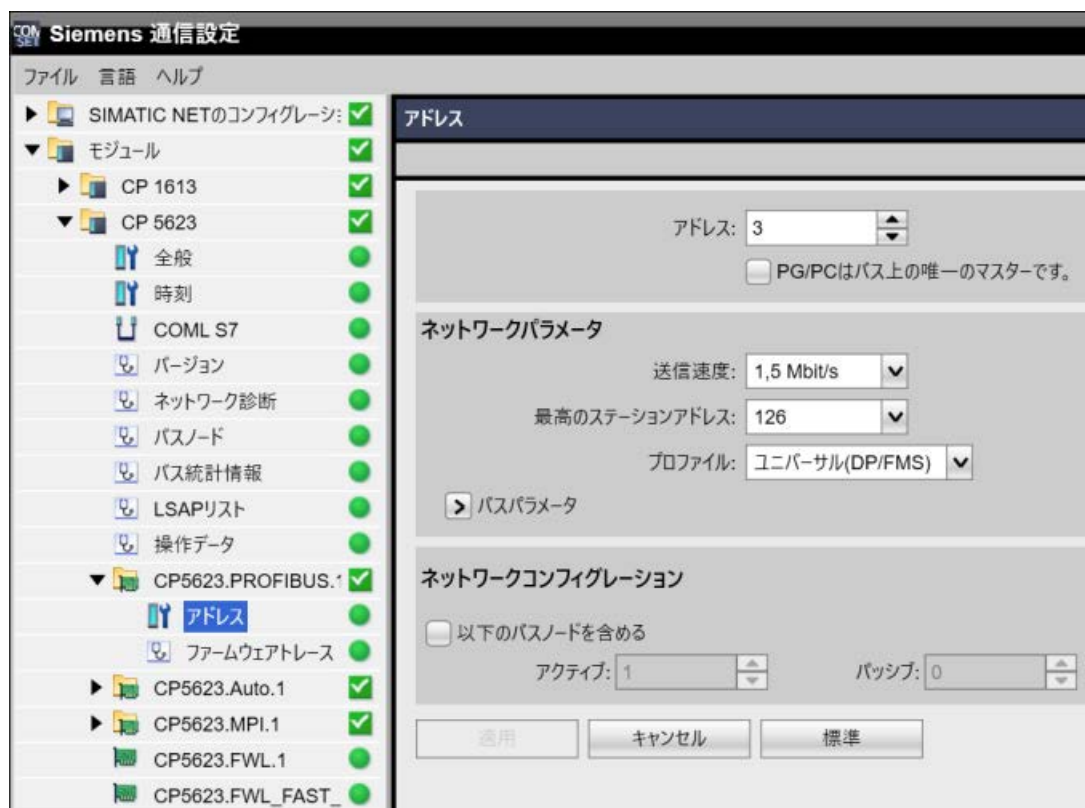


図 7-2 設定例:CP 5623

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 347)

SIMATIC NET での通信プロセッサのチェック (ページ 346)

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 347)

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

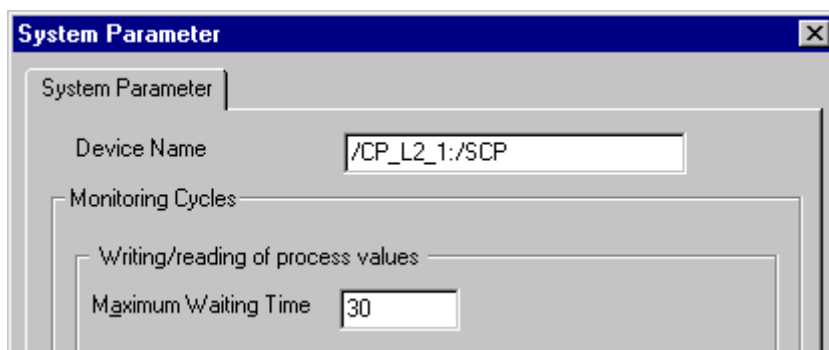
このセクションでは、システムパラメータと、接続およびタグの設定をチェックする方法について説明します。

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

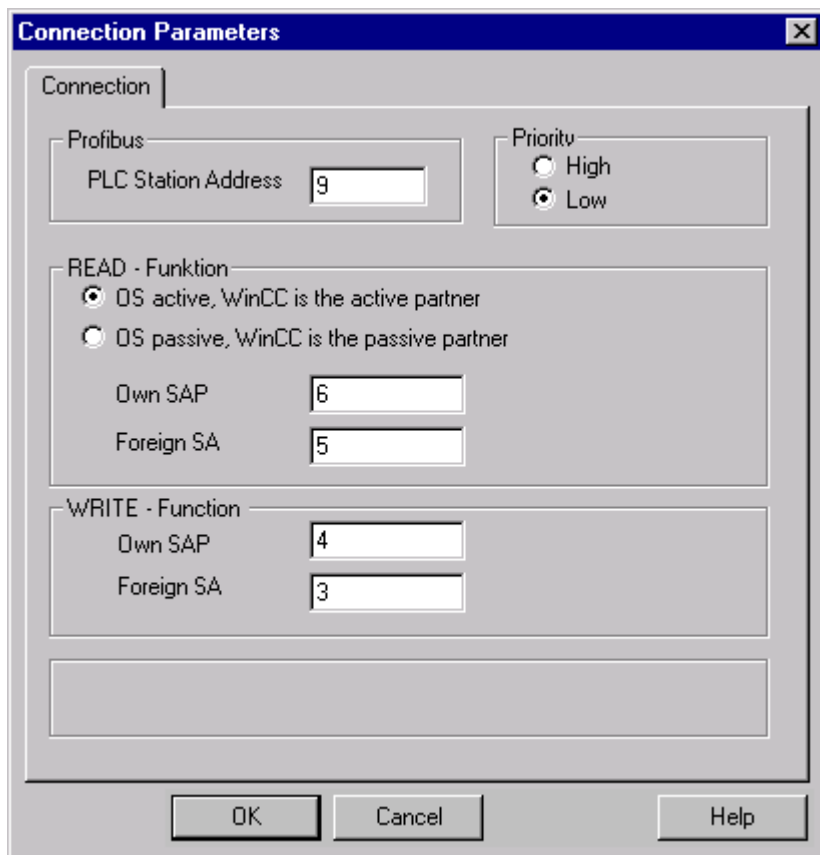
1. WinCC エクスプローラのナビゲーションウィンドウで、[SIMATIC S5 PROFIBUS FDL]アイコンの前にある"プラス符号"をクリックします。[FDL(CP5412/A2-1)]アイコンのショートカットメニューで、[システムパラメータ]をクリックします。[システムパラメータ]ダイアログが開きます。



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 349)

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。

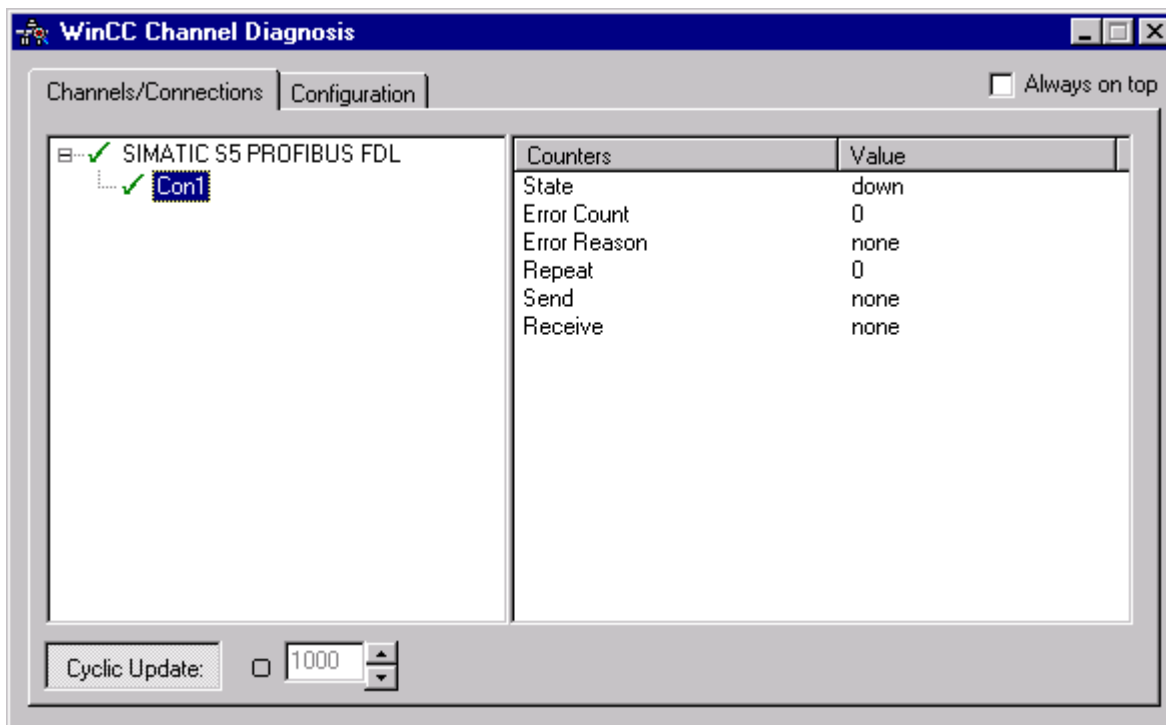
ステータスメッセージの概要

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 311)

ログファイルのエントリの説明 (ページ 343)

タグのチェック方法 (ページ 351)

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 374)

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 362)

チャンネルおよび接続のチェック方法 (ページ 360)

コンフィグレーションデータのチェック方法 (ページ 358)

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 2000-03-2410:46:18,746UTC
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 356)

"INFO"フラグのエントリ (ページ 355)

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 2000-03-2410:46:18,746UTC
```

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。

7.8.3 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 362)

チャンネルおよび接続のチェック方法 (ページ 360)

コンフィグレーションデータのチェック方法 (ページ 358)

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 354)」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 311)」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 311)

ログファイルのエントリの説明 (ページ 354)

チャンネルおよび接続のチェック方法 (ページ 360)

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





概要

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

シンボル	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。

Status	Name	Counters	Value
✖	OPC	Requested Tags	0
✖	NewConnection_2	Data Change Notifications	0
		Data Change Notification Tags	0
		Synchronous Reads	0
		Synchronous Writes	0
		Synchronous Read Tags	0
		Synchronous Write Tags	0
		AddItem failures	0
		OPC Groups	0
		Items added	0
		Server uses DataAccess	
		Server Status	
		ForcedConnectionStateEx	ENABLED
		Current Activity	
		Last Error Time	2024-11-21 16:11:26.228
		Last Error	0x800401f3
		Last Error Name	CO_E_CLASSSTRING
		Last Error in Call	GetServerClassID
		Server Info	
		Locale-ID	
		Maximum Time between CBs	
		Last Time between CBs	
1 second		4:11:27 PM	

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。

7.8 "OPC"チャンネルの診断

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 354)」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 311)」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 311)
ログファイルのエントリの説明 (ページ 354)
タグのチェック方法 (ページ 362)

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 374)

7.9 [OPC UA]チャンネルの診断

7.9.1 OPC UA 通信のエラー処理

OPC UA 経由の通信中にエラーが発生した場合、WinCC は診断と修正のサポートを提供します。

通信テストの手順は、WinCC の使用方法に関係ありません。

エラー処理

初めに、WinCC OPC UA クライアント上でチャンネル診断を実行し、OPC UA サーバーへの接続設定が可能かを確認します。

詳細情報:

- 「チャンネルとタグの診断 (ページ 289)」

OPC UA 通信の詳細については、FAQ 109763315 を参照してください。

- 「SiePortal:SIMATIC/SITOP との OPC UA 通信 (<https://support.industry.siemens.com/cs/ww/en/view/109763315>)」

トレース機能

さらなる診断にはトレース機能を使用できます。

- 「OPC UA クライアントのトレース機能の設定方法 (ページ 364)」
- 「OPC UA サーバーのトレース機能の設定方法 (ページ 367)」

7.9.2 OPC UA クライアントのトレース機能の設定方法

概要

WinCC は、OPC UA サーバーの OPC UA クライアントとしても使用できます。この場合、[OPC UA WinCC Channel]チャンネルを設定する必要があります。

OPC UA クライアントの診断には次のファンクションが使用できます。

- WinCC "チャンネル診断"
詳細情報:[チャンネル診断]によるチャンネル診断 (ページ 297)
- OPC UA クライアントの設定クライアントとして機能する OPC UA Tag Importer のトレースレコード。
- OPC UA クライアントのランタイムクライアントとして機能する OPC UA Data Acquisition Server のトレースレコード。

必要条件

- WinCC プロジェクトを使用して、PC が WinCC OPC UA クライアントとしてセットアップされていること。
- WinCC プロジェクトを使用して、PC が WinCC OPC UA サーバーとしてセットアップされていること。
- [OPC UA WinCC Channel]が、OPC UA クライアントの WinCC プロジェクトに統合されていること。
- 作成した OPC UA クライアントと OPC UA サーバー間の接続が設定されていること。
- OPC UA クライアントと OPC UA サーバーで WinCC プロジェクトが有効にされていること。

手順

1. OPC UA Tag Importer のトレースレコードを有効にするには、次のフォルダに移動します:
 - 「<WinCC インストールパス>\opc\UAClient\UaConfigServer\」
トレース設定はファイル「CCOpUaImporter.xml」を介して行われます。
2. OPC UA Data Acquisition Server のトレースレコードを有効にするには、次のフォルダに移動します:
 - 「<WinCC インストールパス>\opc\UAClient\UaDAS\」
トレース設定はファイル「CcUaDAS.xml」を介して行われます。

7.9 [OPC UA]チャンネルの診断

3. XML ファイルを開き、次のパラメータをそれぞれ設定します。

名前	値	意味
UaStackTraceEnabled	True/False	スタックトレースのレコードを有効にします。
UaStackTraceLevel	<ul style="list-style-type: none"> • None: トレースレコード • Error: 重大なエラー • Warning: 重大でないエラー • システム: めったに起こらない重大なイベント • 情報: 定期的なイベント • Debug: デバッグモード • Content: 追加コンテンツ(例: メッセージ本文全体) • すべて: すべて 	スタックログの情報レベルを指定します。
UaAppTraceEnabled	True/False	アプリのトレースレコードを有効にします。
UaAppTraceLevel	<ul style="list-style-type: none"> • None: アプリのトレースレコードなし • Errors: 予期しないエラー • Warning: エラーではない予期しない動作 • 情報: 重要なアクティビティに関する情報 • InterfaceCall: モジュールインターフェースの呼び出し • CtorDtor: オブジェクトの作成と削除 • ProgramFlow: 内部プログラムシーケンス • Data: データ 	アプリレコードの情報レベルを指定します。
UaAppTraceMaxEntries	100000	ファイル内のエントリの最大数を指定します。
UaAppTraceMaxBackup	5	最も古いトレースファイルが再度上書きされる前に作成されるトレースファイルの数を指定します。

4. 変更を保存します。

結果

トレースレコードを有効にしました。

OPC UA Tag Importer のトレースレコードファイルは、
「\WinCCProjects\la\OPC\UaClient\Trace\CCOpcUaImporter.log」にあります。

OPC UA Data Acquisition Server のトレースレコードファイルは、
「\WinCCProjects\la\OPC\UaClient\Trace\CcUaDas.log」にあります。

下記も参照

OPC UA サーバーのトレース機能の設定方法 (ページ 367)

7.9.3 OPC UA サーバーのトレース機能の設定方法

概要

WinCC は OPC UA サーバーとして使用できます。

通信に問題が発生した場合、または設定された OPC UA 接続が期待どおりに動作しない場合は、WinCC OPC UA サーバーと使用されている OPC UA クライアント間の OPC UA 通信のトレースレコードを作成できます。

OPC UA サーバーのトレース設定は、2 つの設定ファイル「OPCUAServerWinCC.xml」によって行われます。

設定ファイル	保存パス
サーバー固有の設定ファイル: <ul style="list-style-type: none"> 管理者権限が必要です。 スタックトレースのレコードを有効にすることができます。 	<WinCC installation path>\opc\UAServer\OPCUAServerWinCC.xml
プロジェクト固有の設定ファイル: <ul style="list-style-type: none"> 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。 	<WinCC project folder>\OPC\UAServer\OPCUAServerWinCC.xml

必要条件

- WinCCプロジェクトを使用して、PCがWinCC OPC UAクライアントとしてセットアップされていること。
- WinCCプロジェクトを使用して、PCがWinCC OPC UAサーバーとしてセットアップされていること。
- [OPC UA WinCC Channel]がOPC UAクライアントのWinCCプロジェクトで設定されていること。
- 作成したOPC UAクライアントとOPC UAサーバー間の接続が設定されていること。
- OPC UAクライアントとOPC UAサーバーでWinCCプロジェクトが有効にされていること。

サーバー固有のトレースレコードの有効化

1. 以下のファイルに移動します:
 - <WinCC インストールパス>\opc\UAServer\OPCUAServerWinCC.xml
2. ファイルを開き、以下のパラメータを設定します。

名前	値	意味
TraceFile	[CommonDocuments] \Siemens\WinCC\OPC\UAServer\ [ModuleName].txt	トレースレコードファイルのパスを定義します。
TraceEnable	True/False	トレースレコードは、このパラメータの値が「True」である場合のみ実行されます。
TraceLevel	<ul style="list-style-type: none"> • Error:予期しないエラー • Warning:エラーではない予期しない動作 • 情報:重要なアクティビティに関する情報 • InterfaceCall:モジュールインターフェースの呼び出し • CtorDtor:オブジェクトの作成と削除 • Content:追加コンテンツ(例:メッセージ本文全体) 	レコードの情報レベルを指定します。
TraceMaxBackFiles	10	最も古いトレースファイルが再度上書きされる前に作成されるトレースファイルの数を指定します。
maxRecords	100000	ファイル内のエントリの数を指定します。

7.9 [OPC UA]チャンネルの診断

- スタックトレースのレコードを有効にするには、ファイル内の<StackConfiguration>セクションに移動し、次のパラメータを設定します:

名前	値	意味
TraceEnable	True/False	トレースレコードは、このパラメータの値が「True」である場合のみ実行されます。
TraceLevel	Error:重大なエラー	レコードの情報レベルを指定します。
	Warning:重大でないエラー	
	システム:めったに起こらない重大なイベント	
	情報:定期的なイベント	
	Debug:デバッグモード	
	Content:追加コンテンツ(例:メッセージ本文全体)	

- 変更を保存します。

結果

プロジェクト固有のトレースレコードを有効にしました。

トレースレコードファイルは、定義された保存場所に保存されます。

プロジェクト固有のトレースレコードを有効化

1. 以下のファイルに移動します:
 - <WinCC project folder>\OPC\UAServer\OPCUAServerWinCC.xml
2. ファイルを開き、以下のパラメータを設定します。

名前	値	意味
TraceEnable	True/False	トレースレコードは、このパラメータの値が「True」である場合のみ実行されます。
TraceLevel	Error:予期しないエラー	レコードの情報レベルを指定します。
	Warning:エラーではない予期しない動作	
	情報:重要なアクティビティに関する情報	
	InterfaceCall:モジュールインターフェースの呼び出し	
	CtorDtor:オブジェクトの作成と削除	
	Content:追加コンテンツ(例:メッセージ本文全体)	
TraceMaxBackFiles	10	最も古いトレースファイルが再度上書きされる前に作成されるトレースファイルの数を指定します。
maxRecords	100000	ファイル内のエントリの数を指定します。
TraceFile	[WinCCProjectFolder] \OPCUAServer\Trace\ [ModuleName].txt	トレースレコードファイルのパスを定義します。

3. 変更を保存します。

結果

サーバー固有のトレースレコードを有効にしました。

トレースレコードファイルは、定義された保存場所に保存されます。

下記も参照

OPC UA クライアントのトレース機能の設定方法 (ページ 364)

7.10 タグの品質

7.10.1 タグの品質

はじめに

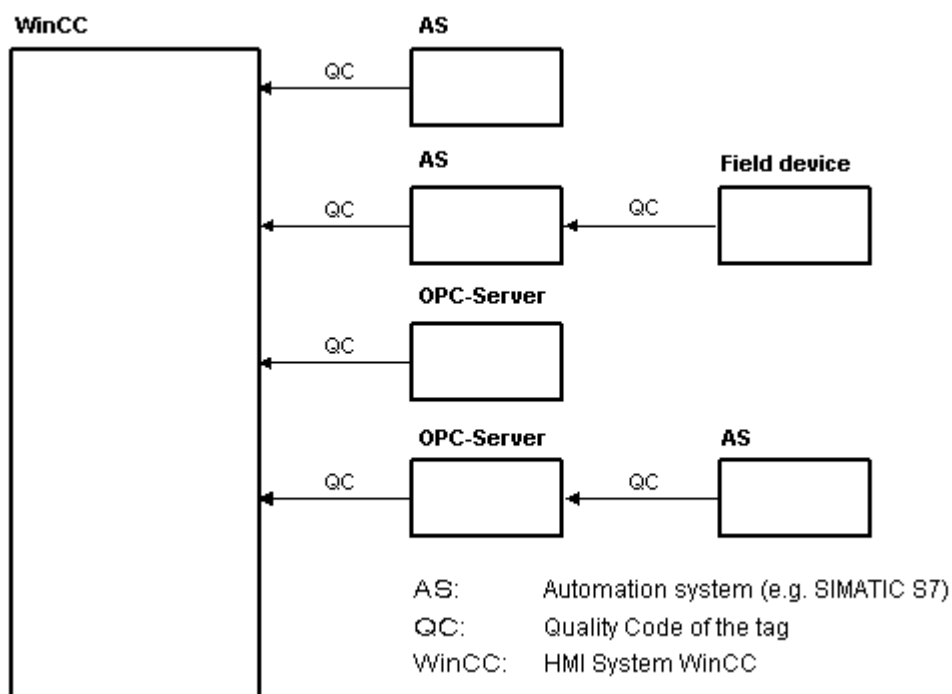
WinCCには2つの品質インジケータがあります。このインジケータによってタグの品質を評価できます。この2つのインジケータは、タグステータスと品質コードです。

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。

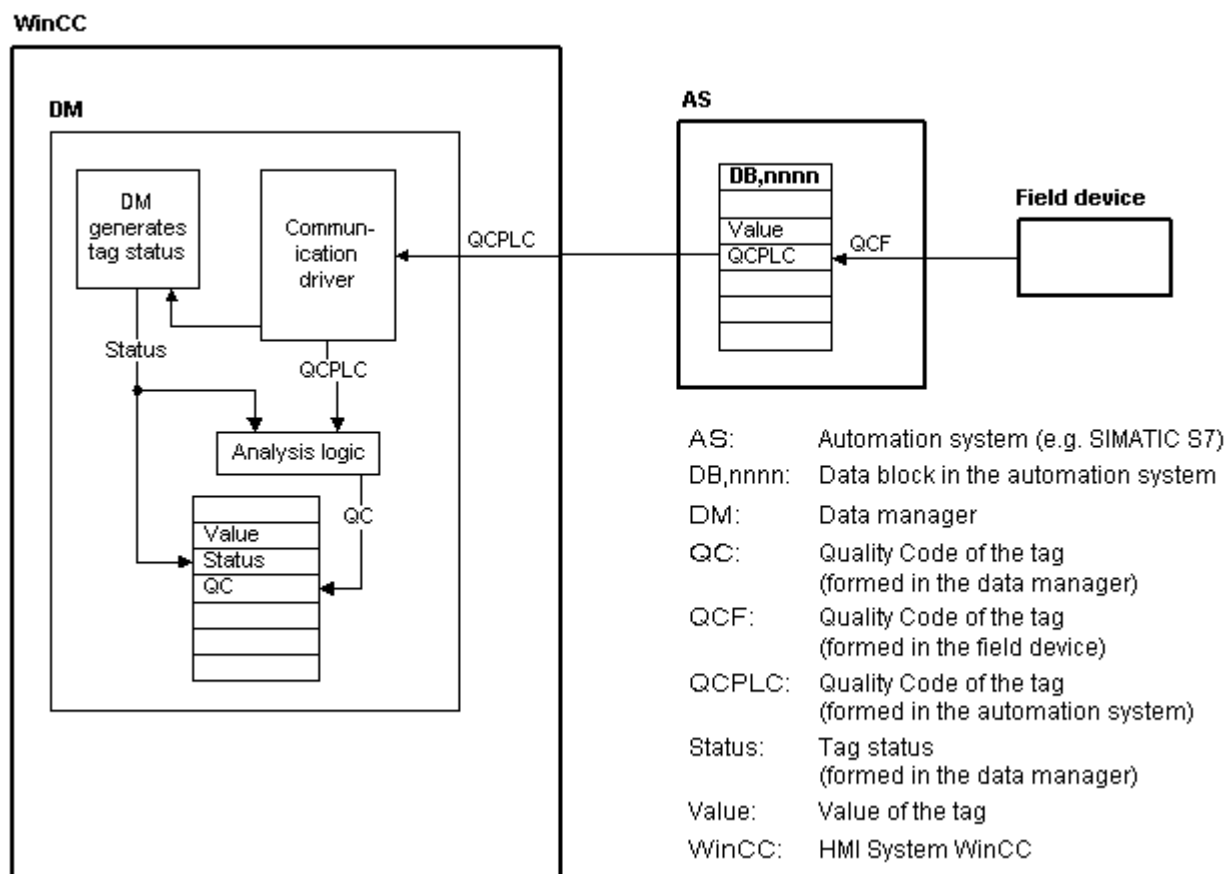
品質コードの重ね表示

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。

7.10 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。

7.10.2 タグの品質コード

概要

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 スクリプトを使用
- C スクリプトを使用
- ダイナミックダイアログ経由
- I/O フィールドの[品質コード - 変更タグ]イベントの評価

注記

プロセスタグ:メモリ要件

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 でタグを設定するときは、品質コードに必要なメモリに注意してください。

例:

- 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、データブロックの末尾など、タグを設定するときにこのバイトを考慮してください。
-

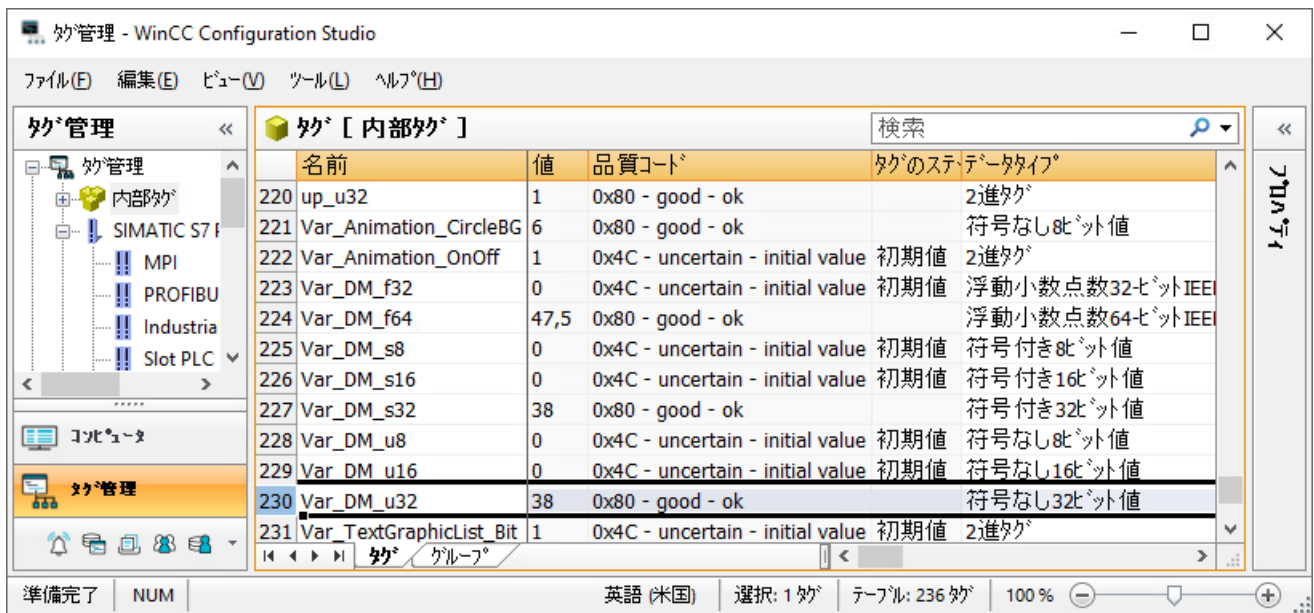
品質コードのタグ管理での表示

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。

7.10 タグの品質



品質コードがプロセス画像表示に与える影響

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。

- 品質コードの値が 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。

QQSSSLL
 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
Bad The value is not useful	0	0	-	-	-	-	-	-
Uncertain The quality of the value is less than normal, but the value may still be useful.	0	1	-	-	-	-	-	-
Good (Non-Cascade) The quality of the value is good. Possible alarm conditions may be indicated by the sub-status.	1	0	-	-	-	-	-	-
Good (Cascade) The value may be used in control.	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。

7.10 タグの品質

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	Bad	Device passivated - Diagnostic alerts inhibited	0	0	1	0	0	0	1	1
0x3F	Bad	Function check - Local override	0	0	1	1	1	1	1	1
0x1C	Bad	Out of Service The value is not reliable because the block is not being evaluated, and may be under construction by a configurer. Set if the block mode is O/S.	0	0	0	1	1	1	-	-
0x73	Uncertain	Simulated value - Start	0	1	1	1	0	0	1	1
0x74	Uncertain	Simulated value - End	0	1	1	1	0	1	-	-
0x84	Good (Non- Cascade)	Active Update event Set if the value is good and the block has an active Update event.	1	0	0	0	0	1	-	-
0x24	Bad	Maintenance alarm - More diagnostics available.	0	0	1	0	0	1	-	-
0x18	Bad	No Communication, with no usable value Set if there has never been any communication with this value since it was last "Out of Service".	0	0	0	1	1	0	-	-
0x14	Bad	No Communication, with last usable value Set if this value had been set by communication, which has now failed.	0	0	0	1	0	1	-	-
0x0C	Bad	Device Failure Set if the source of the value is affected by a device failure.	0	0	0	0	1	1	-	-
0x10	Bad	Sensor failure	0	0	0	1	0	0	-	-
0x08	Bad	Not Connected Set if this input is required to be connected and is not connected.	0	0	0	0	1	0	-	-

コード (16進 数)	品質		Q	Q	S	S	S	S	L	L
0x04	Bad	Configuration Error Set if the value is not useful because there is some inconsistency regarding the parameterization or configuration, depending on what a specific manufacturer can detect.	0	0	0	0	0	1	-	-
0x00	Bad	non-specific There is no specific reason why the value is bad. Used for propagation.	0	0	0	0	0	0	-	-
0x28	Bad	Process related - Substitute value	0	0	1	0	1	0	-	-
0x2B	Bad	Process related - No maintenance	0	0	1	0	1	0	1	1
0x68	Uncertain	Maintenance demanded	0	1	1	0	1	0	-	-
0x60	Uncertain	Simulated value Set when the process value is written by the operator while the block is in manual mode.	0	1	1	0	0	0	-	-
0x64	Uncertain	Sensor calibration	0	1	1	0	0	1	-	-
0x5C	Uncertain	Configuration error	0	1	0	1	1	1	-	-
0x58	Uncertain	Sub-normal	0	1	0	1	1	0	-	-
0x54	Uncertain	Engineering Unit Range Violation Set if the value lies outside of the set of values defined for this parameter. The Limits define which direction has been exceeded.	0	1	0	1	0	1	-	-
0x50	Uncertain	Sensor conversion not accurate	0	1	0	1	0	0	-	-
0x4B	Uncertain	Substitute (constant)	0	1	0	0	1	0	1	1
0x78	Uncertain	Process related - No maintenance	0	1	1	1	1	0	-	-
0x4C	Uncertain	Initial Value Value of volatile parameters during and after reset of the device or of a parameter.	0	1	0	0	1	1	-	-
0x48	Uncertain	Substitute value Predefined value is used instead of the calculated one. This is used for fail safe handling.	0	1	0	0	1	0	-	-

7.10 タグの品質

コード (16進 数)	品質		Q	Q	S	S	S	S	L	L	
0x44	Uncertain	Last Usable Value Whatever was writing this value has stopped doing so. This is used for fail safe handling.	0	1	0	0	0	0	1	-	-
0x40	Uncertain	Non-specific There is no specific reason why the value is uncertain.	0	1	0	0	0	0	0	-	-
0xE0	Good (Cascade)	Initiate Fail Safe (IFS) The value is from a block that wants its downstream output block (e.g. AO) to go to Fail Safe.	1	1	1	0	0	0	0	-	-
0xD8	Good (Cascade)	Local Override (LO) The value is from a block that has been locked out by a local key switch or is a Complex AO/DO with interlock logic active. The failure of normal control must be propagated to a function running in a host system for alarm and display purposes. This also implies "Not Invited".	1	1	0	1	1	0	0	-	-
0xD4	Good (Cascade)	Do Not Select (DNS) The value is from a block which should not be selected, due to conditions in or above the block.	1	1	0	1	0	1	0	-	-
0xCC	Good (Cascade)	Not Invited (NI) The value is from a block which does not have a target mode that would use this input.	1	1	0	0	1	1	0	-	-
0xC8	Good (Cascade)	Initialization Request (IR) The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or the mode is wrong.	1	1	0	0	1	0	0	-	-
0xC4	Good (Cascade)	Initialization Acknowledge (IA) The value is an initialized value from a source (cascade input, remote-cascade in, and remote-output in parameters).	1	1	0	0	0	1	0	-	-
0xC0	Good (Cascade)	OK No error or special condition is associated with this value.	1	1	0	0	0	0	0	-	-
0xA0	Good (Non-Cascade)	Initiate fail safe	1	0	1	0	0	0	0	-	-

コード (16進 数)	品質		Q	Q	S	S	S	S	L	L
0x98	Good (Non- Cascade)	Unacknowledged Critical Alarm Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8.	1	0	0	1	1	0	-	-
0x94	Good (Non- Cascade)	Unacknowledged Advisory Alarm Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8.	1	0	0	1	0	1	-	-
0x90	Good (Non- Cascade)	Unacknowledged Update event Set if the value is good and the block has an unacknowledged Update event.	1	0	0	1	0	0	-	-
0x8C	Good (Non- Cascade)	Active Critical Alarm Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8.	1	0	0	0	1	1	-	-
0x88	Good (Non- Cascade)	Active Advisory Alarm Set if the value is good and the block has an active Alarm with a priority less than 8.	1	0	0	0	1	0	-	-
0xA8	Good (Non- Cascade)	Maintenance demanded	1	0	1	0	1	0	-	-
0xA4	Good (Non- Cascade)	Maintenance required	1	0	1	0	0	1	-	-
0xBC	Good (Non- Cascade)	Function check - Local override	1	0	1	1	1	1	-	-
0x80	Good (Non- Cascade)	OK No error or special condition is associated with this value.	1	0	0	0	0	0	-	-

7.10 タグの品質

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
O.K. The value is free to move.	-	-	-	-	-	-	0	0
Low limited The value has acceded its low limits.	-	-	-	-	-	-	0	1
High limited The value has acceded its high limits.	-	-	-	-	-	-	1	0
Constant (high and low limited) The value cannot move, no matter what the process does.	-	-	-	-	-	-	1	1

OPC との通信時の品質コード

OPC UA の品質コード

OPC UA ステータスコードは、WinCC 品質コードとは異なります。

WinCC OPC UA サーバーから外部 OPC UA クライアントにタグを送信する場合、WinCC 品質コードを OPC UA ステータスコードに変換する必要があります。

逆方向の変換も行われます。

次の表は、WinCC 品質コードと OPC UA ステータスコードの関係を示しています。

WinCC 品質コード	OPC UA ステータスコード
0x0	Opc_Ua_Bad
0x40	Opc_Ua_Uncertain
0x80	Opc_Ua_Good
0xc0	Opc_Ua_Good
0x4	Opc_Ua_BadConfigurationError
0x8	Opc_Ua_BadNotConnected
0xc	Opc_Ua_BadDeviceFailure
0x10	Opc_Ua_BadSensorFailure
0x14	Opc_Ua_UncertainNoCommunicationLastUsableValue
0x18	Opc_Ua_BadNoCommunication

WinCC 品質コード	OPC UA ステータスコード
0x1c	Opc_Ua_BadOutOfService
0x20	Opc_Ua_BadWaitingForInitialData
0x44	Opc_Ua_UncertainLastUsableValue
0x50	Opc_Ua_UncertainSensorNotAccurate
0x54	Opc_Ua_UncertainEngineeringUnitsExceeded
0x58	Opc_Ua_UncertainSubNormal
0xd8	Opc_ua_GoodLocalOverride

OPC 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

7.10.3 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。

7.10 タグの品質

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
DM_VARSTATE_HARDWARE_ERROR	0x00 04	ネットワークモジュールに欠陥があります。
DM_VARSTATE_MAX_LIMIT	0x00 08	設定された上限値を超えています。
DM_VARSTATE_MIN_LIMIT	0x00 10	設定された下限値を下回っています。
DM_VARSTATE_MAX_RANGE	0x00 20	フォーマット上限値を超えています。
DM_VARSTATE_MIN_RANGE	0x00 40	フォーマット下限値を下回っています。
DM_VARSTATE_CONVERSION_ERROR	0x00 80	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)

フラグ名	値	説明
DM_VARSTATE_STARTUP_VALUE	0x01 00	タグの初期値
DM_VARSTATE_DEFAULT_VALUE	0x02 00	タグ置換値
DM_VARSTATE_ADDRESS_ERROR	0x04 00	チャンネルアドレス指定エラー
DM_VARSTATE_INVALID_KEY	0x08 00	タグが見つかりません/使用できません
DM_VARSTATE_ACCESS_FAULT	0x10 00	タグアクセスが許可されていません
DM_VARSTATE_TIMEOUT	0x20 00	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	0x40 00	サーバーが使用できない

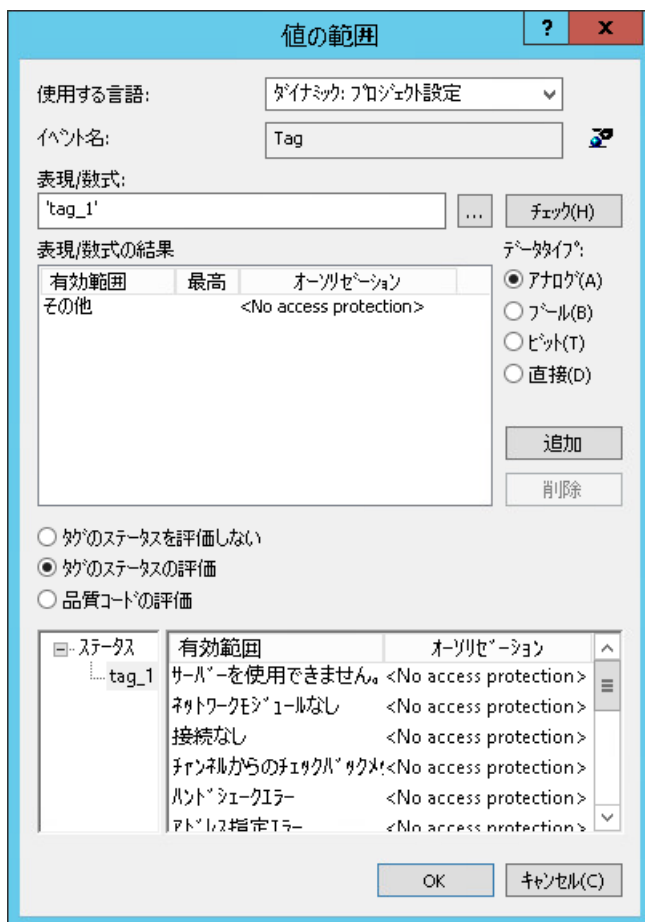
7.10.4 モニタ接続ステータスへのタグステータスの使用

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 252)

7.10.5 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。

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。

7.10.6 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。

WinCC Configuration Studio - タグ管理

ファイル(F) 編集(E) ビュー(V) ツール(L) ヘルプ(H)

タグ管理 << 検索

名前	値	品質コード	タグのステータスタイプ
220 up_u32	1	0x80 - good - ok	2進タグ
221 Var_Animation_CircleBG	6	0x80 - good - ok	符号なし8ビット値
222 Var_Animation_OnOff	1	0x4C - uncertain - initial value	初期値 2進タグ
223 Var_DM_f32	0	0x4C - uncertain - initial value	初期値 浮動小数点数32ビットIEEE
224 Var_DM_f64	47,5	0x80 - good - ok	浮動小数点数64ビットIEEE
225 Var_DM_s8	0	0x4C - uncertain - initial value	初期値 符号付き8ビット値
226 Var_DM_s16	0	0x4C - uncertain - initial value	初期値 符号付き16ビット値
227 Var_DM_s32	38	0x80 - good - ok	符号付き32ビット値
228 Var_DM_u8	0	0x4C - uncertain - initial value	初期値 符号なし8ビット値
229 Var_DM_u16	0	0x4C - uncertain - initial value	初期値 符号なし16ビット値
230 Var_DM_u32	38	0x80 - good - ok	符号なし32ビット値
231 Var_TextGraphicList_Bit	1	0x4C - uncertain - initial value	初期値 2進タグ

準備完了 NUM 英語 (米国) 選択: 1 タグ テーブル: 236 タグ 100%

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 374)

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 1.05.03)
- アーカイブされたプロセス値(OPC Historical Data Access 1.20; OPC UA Historical Access 1.05.03)
- 履歴メッセージ(OPC Historical Alarms and Events 1.10)
- メッセージ(OPC Alarms and Events 1.10; OPC UA Alarms and Conditions 1.05.03)

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。

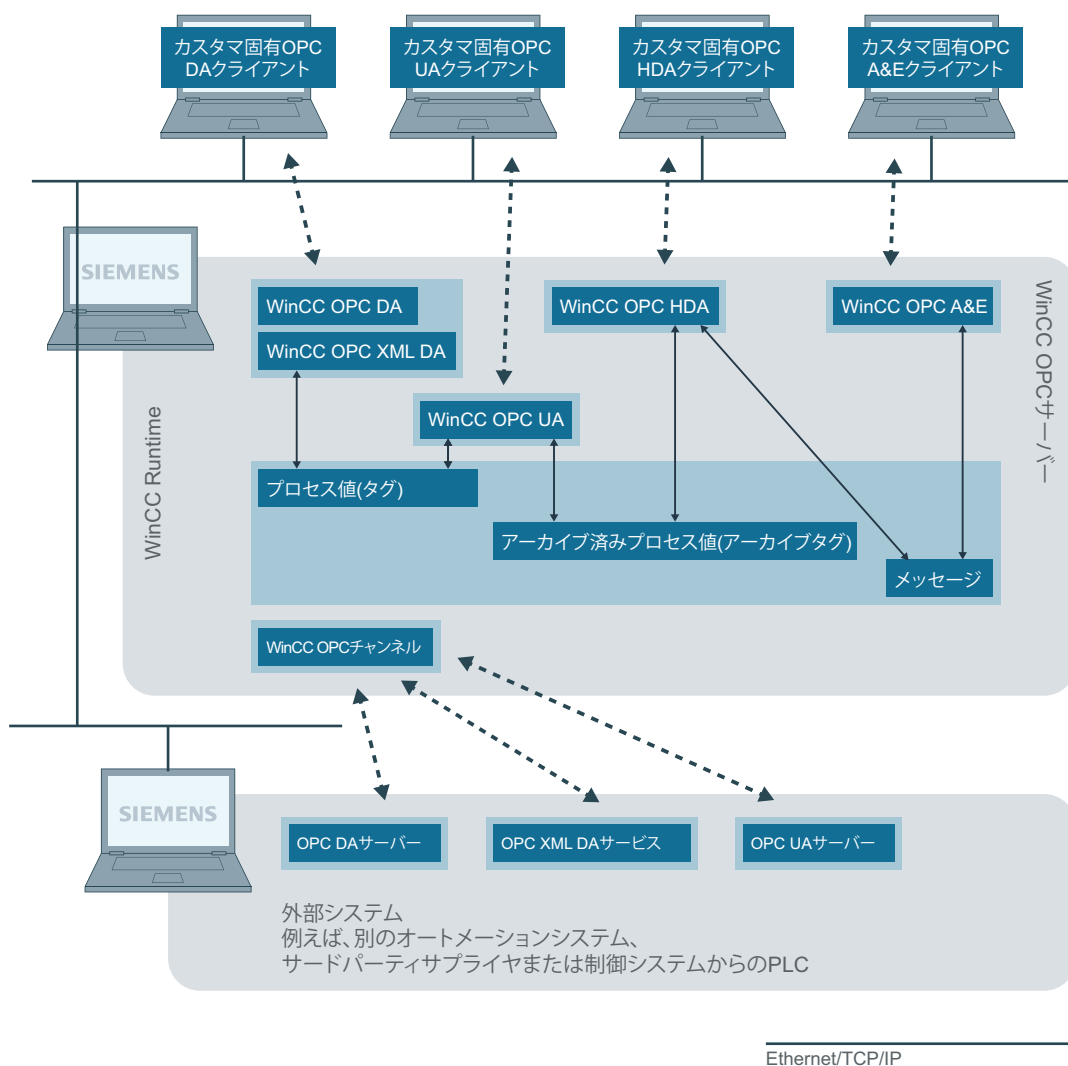
WinCC OPC 通信の概念

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、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、通信パートナー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 DA サーバー

8.6.1 WinCC OPC DA サーバーの機能

概要

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 OPC DA サーバーは、WinCC のインストール中に選択できます。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-OPC DA サーバーが使用されている場合は、アプリケーション「OPC-DA サーバー、OPC-A&E サーバー、OPC-HDA サーバー」を有効にする必要があります。
WinCC Configuration Studio の「コンピュータ」エディタの[WinCC Runtime の起動時に処理]タブで、アプリケーションを有効にできます。
- 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 400)

複数の OPC DA サーバーの使用 (ページ 399)

WinCC 間の接続例 (ページ 401)

WinCC と SIMATIC NET FMS OPC サーバー間の接続例 (ページ 405)

WinCC と SIMATIC NET S7 OPC サーバー間の接続例 (ページ 407)

WinCC と Microsoft Excel 間の接続例 (ページ 413)

www.opcfoundation.org (<http://www.opcfoundation.org>)

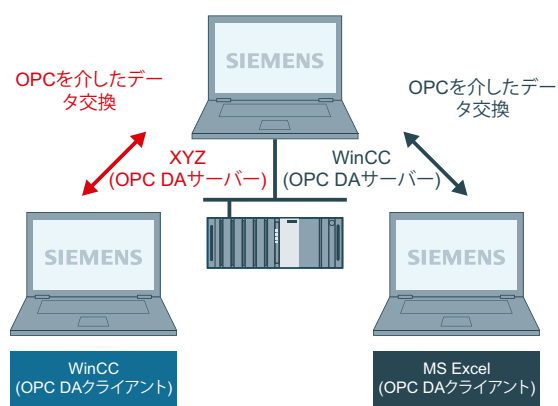
8.6.2 複数の OPC DA サーバーの使用

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 400)

8.6.3 OPC DA サーバー名の問い合わせ

はじめに

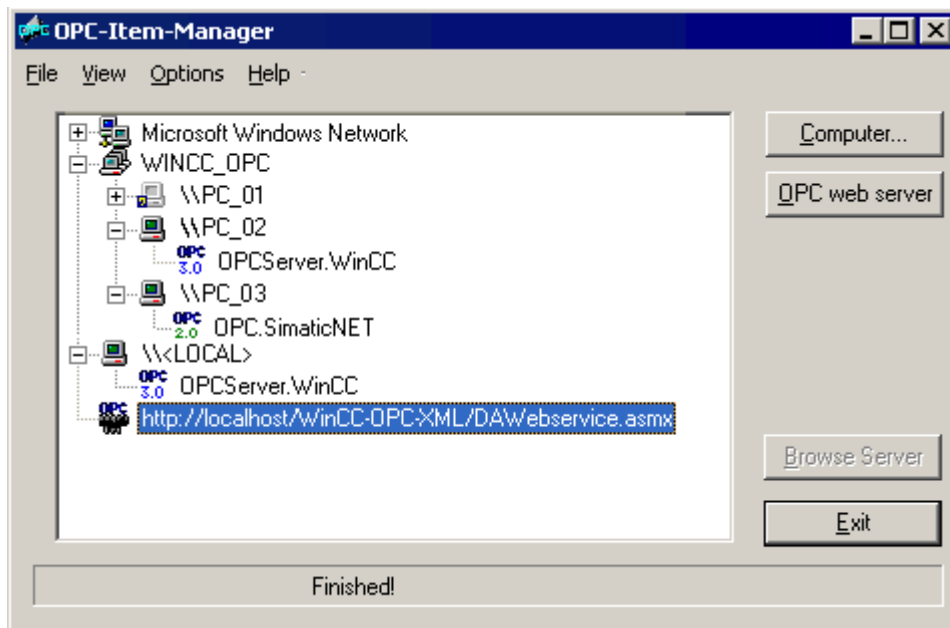
1 台のコンピュータに複数の OPC DA サーバーをインストールできます。 OPC 項目マネージャを使用すると、ワークステーションが使用できる OPC DA サーバーの名前が選択ウィンドウに表示されます。 これらの OPC DA サーバーは、同じコンピュータ上、またはネットワーク環境内の各コンピュータ上で実行できます。

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。



8.6.4 OPC DA 接続の例

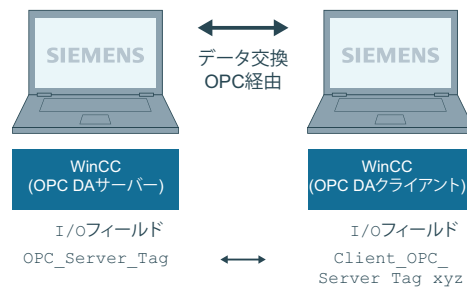
8.6.4.1 WinCC 間の接続

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 402)

WinCC OPC DA クライアント上の WinCC プロジェクトの設定 (ページ 403)

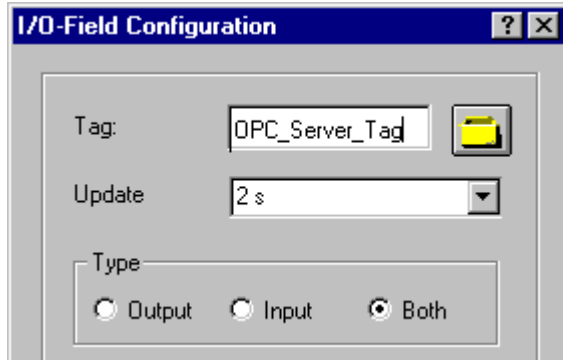
WinCC OPC DA サーバー上の WinCC プロジェクトの設定方法

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 403)

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

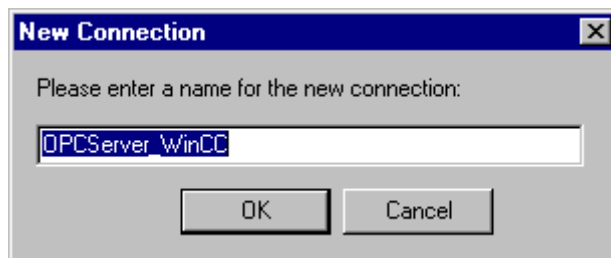
このセクションでは、WinCC OPC DA サーバー上の WinCC タグを読み取るために、WinCC OPC DA クライアント上で WinCC タグを作成します。このタグ値は I/O フィールドに表示されます。

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。

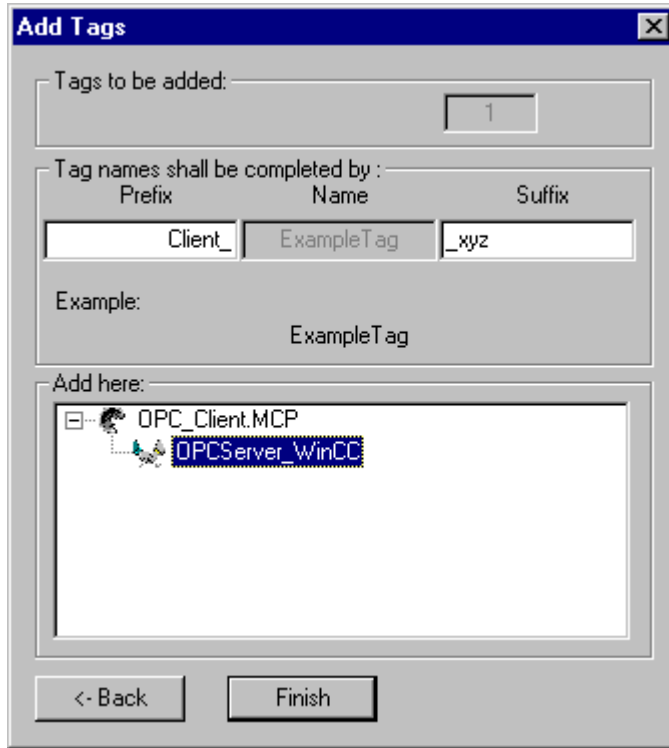
手順

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。

- [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。



- [OPCServer.Wi nCC ...]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。
- 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。
- [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。
- 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 402)

8.6.4.2 WinCC と SIMATIC NET FMS OPC サーバー間の接続

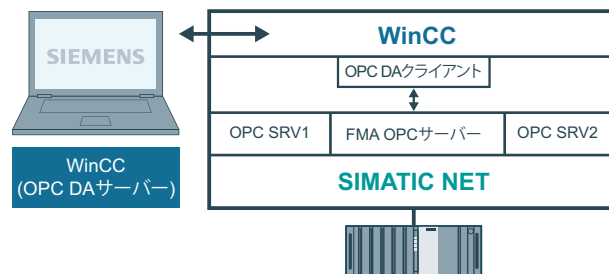
WinCC と SIMATIC NET FMS OPC サーバー間の接続例

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

下記も参照

WinCC と SIMATIC NET FMS OPC サーバー間の接続の設定方法 (ページ 406)

<http://support.automation.siemens.com/> (<http://support.automation.siemens.com/>)

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_SlimaticNET"を入力します。[OK]をクリックします。

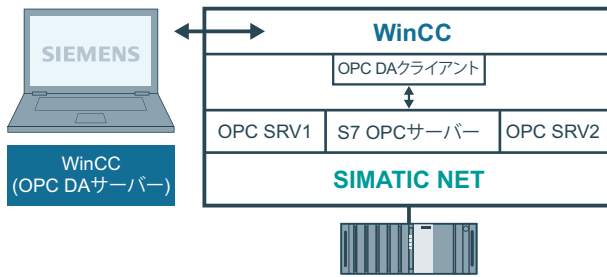
5. [タグの追加]ダイアログを開きます。
[接頭語]フィールドに"Client_"を入力し、[接尾語]フィールドには"_xyz"を入力します。接続"OPC_SimaticNET"を選択します。[終了]をクリックします。
6. [OPC.SIMATICNet..]ダイアログで[戻る]ボタンをクリックします。"OPC 項目マネージャ"で[終了]をクリックして OPC 項目マネージャを閉じます。
7. グラフィックデザイナを起動し、画面を新たに開きます。画面内に I/O フィールドを追加します。[スマートオブジェクト]下のオブジェクトリストで[I/O フィールド]オブジェクトを選択します。[I/O 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。

8.6.4.3 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

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 412)

SIMATIC NET S7 OPC サーバーへのタグの追加 (ページ 409)

<http://support.automation.siemens.com/> (<http://support.automation.siemens.com/>)

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

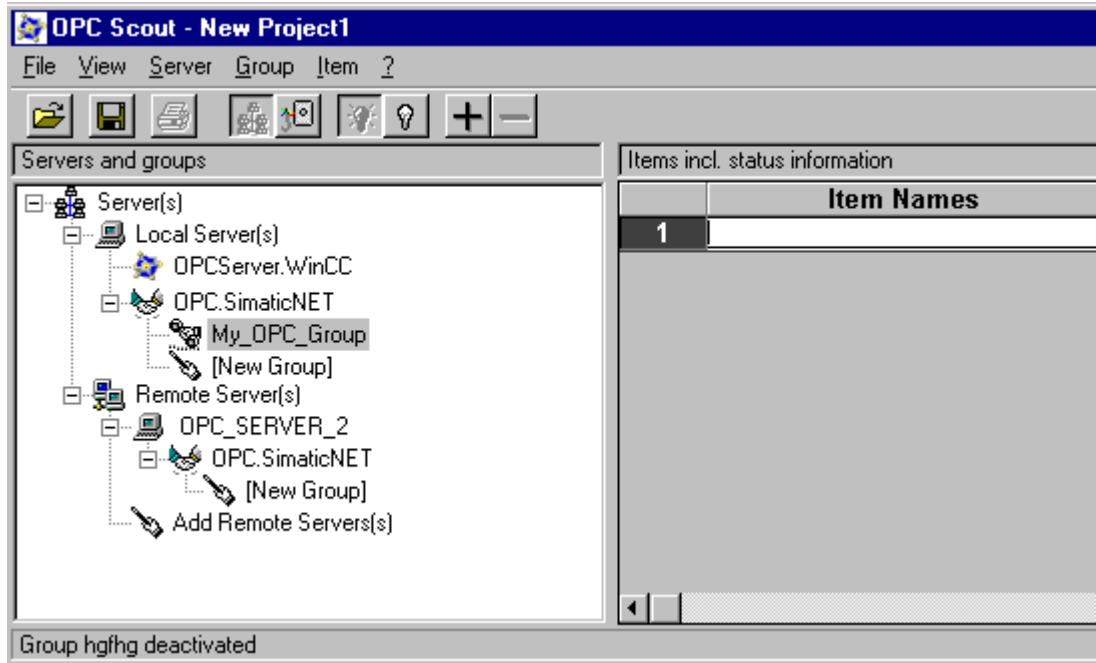
パラメータ	値
データタイプ	W
範囲バイト	0
番号値	1
項目別名	MW0

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。

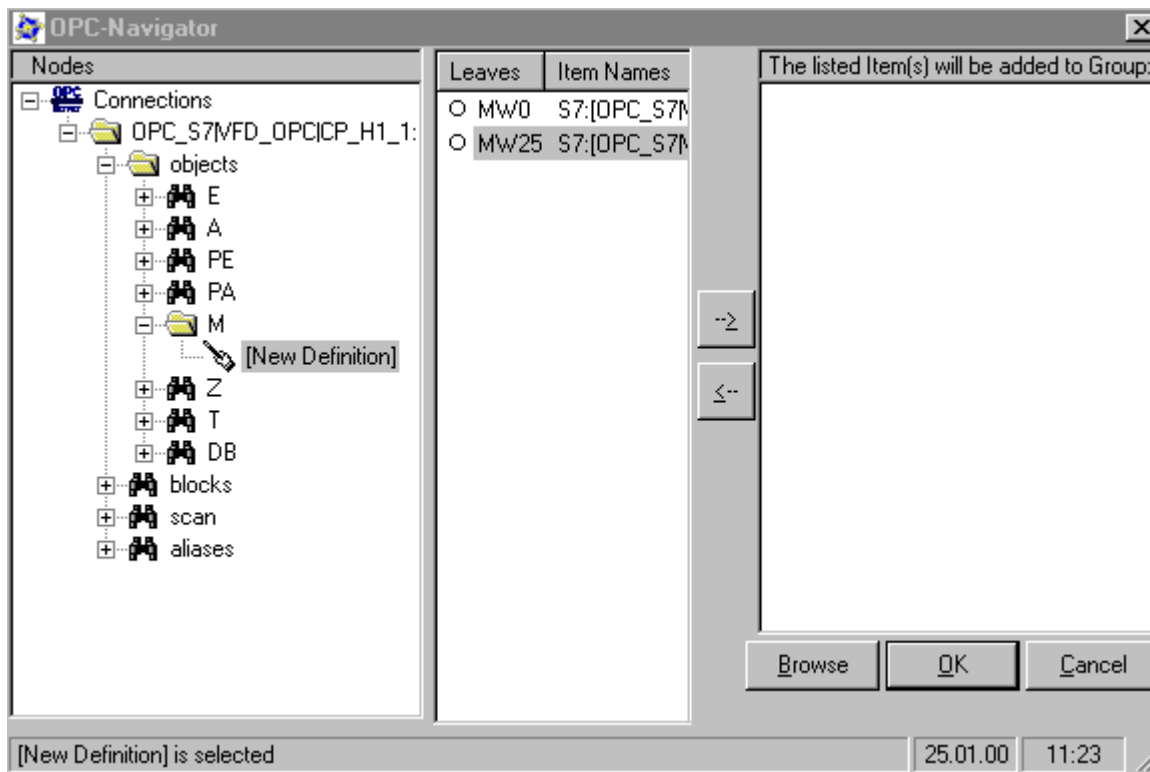
手順

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。

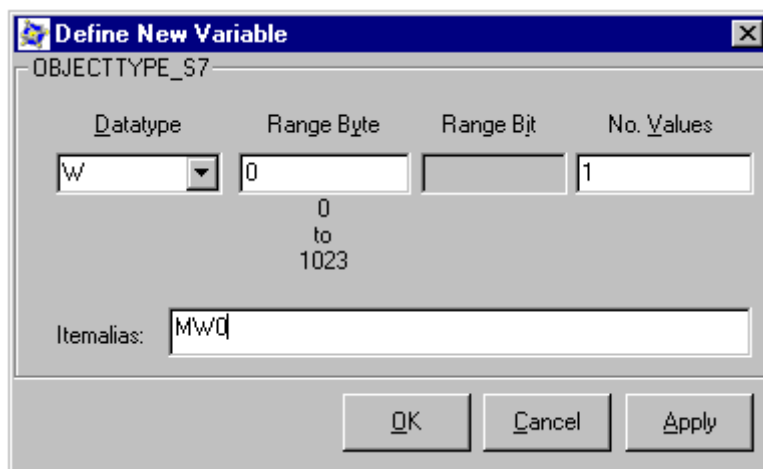


2. [ローカルサーバー]下の[OPC.SimaticNet]を選択します。同じコンピュータ上で SIMATIC S7 OPC サーバーが実行されていない場合ショートカットメニュー[サーバー]で[リモートサーバーの追加]を選択します。[リモートサーバーの追加]ダイアログで、 OPC サーバーとして使用するコンピュータの名前を入力した後、 [OK]をクリックしてこのダイアログを閉じます。
3. ショートカットメニュー[OPC.SimaticNet]で[接続]を選択します。 [グループの追加]ダイアログが表示されます。グループの名前を入力します。 [OK]をクリックしてダイアログを閉じます。

- 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。



- "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。
- [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。

- 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 412)

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 409)

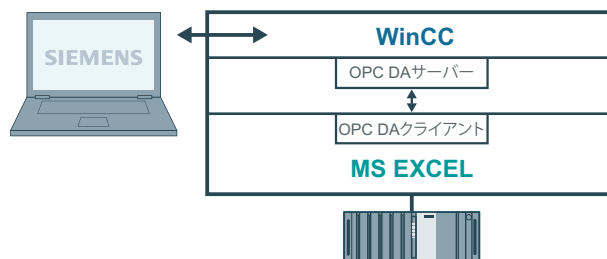
8.6.4.4 WinCC と Microsoft Excel 間の接続

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 417)

Microsoft Excel での OPC DA クライアントの作成 (ページ 414)

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 = "MyGroupName"
    '----- 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 が返されず
    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

```

8.6 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 402)

Microsoft Excel での WinCC タグへのアクセスの設定方法

はじめに

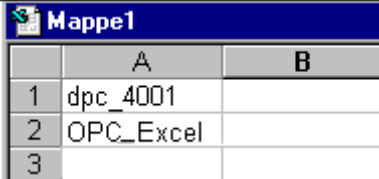
Excel OPC DA クライアントは、WinCC OPC DA サーバーの WinCC タグを読み取って、このタグの値をセル内に書き込みます。WinCC OPC DA サーバーの WinCC プロジェクトでは、I/O フィールドにこのタグの値が表示されます。セル内のタグ値が変更されると、それに応じて WinCC OPC DA サーバーの I/O フィールド内の値が変更されます。

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。

8.7 WinCC OPC HDA サーバー

8.7.1 WinCC OPC HDA サーバーの機能

概要

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 OPC HDA サーバーは、WinCC のインストール中に選択できます。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。

ライセンスング

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

設定に関する注意

WinCC OPC HDA サーバーが使用されている場合は、アプリケーション「OPC-DA サーバー、OPC-A&E サーバー、OPC-HDA サーバー」を有効にする必要があります。

WinCC Configuration Studio の「コンピュータ」エディタの[WinCC Runtime の起動時に処理]タブで、アプリケーションを有効にできます。

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 426)
- WinCC OPC HDA サーバーのデータ構造 (ページ 420)
- www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2 WinCC OPC HDA サーバーのデータ構造

8.7.2.1 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。

	説明
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ファンクションの概要 (ページ 423)

WinCC OPC HDA サーバーの時間形式 (ページ 424)

サポートされている属性の概要 (ページ 421)

サポートされている集約の概要 (ページ 422)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2.2 サポートされている属性の概要

概要

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 420)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2.3 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返します。
OPCHDA_START	時間間隔の先頭で、未処理データの初期値を返します。
OPCHDA_END	時間間隔の終わりに、未処理データの最終値を返します。
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返します。
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返します。
OPCHDA_TOTAL	指定の時間間隔に対応した合計値を返します。
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返します。
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返します。
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返します。
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返します。
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返します。
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返します。

集約	説明
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返します。
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返します。
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返します。
OPCHDA_VARIANCE	指定の時間間隔に対応した未処理データの分散を返します。
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 の差を返します。
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 420)

WinCC OPC HDA サーバーの機能 (ページ 418)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2.4 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戻します。
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戻します。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戻します。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戻します。

下記も参照

WinCC OPC HDA サーバーの機能 (ページ 418)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.2.5 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

HOUR = 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

8.7 WinCC OPC HDA サーバー

+/-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 418)

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.3 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	OPCHDA_RAW	未処理データ送信の品質を示します。	GOOD BAD UNCERTAIN
0x00080000	OPCHDA_CALCULATED	計算したデータ送信の品質を示します。	GOOD BAD UNCERTAIN
0x00100000	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	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	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>)



8.7.4 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。


	説明
	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。
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

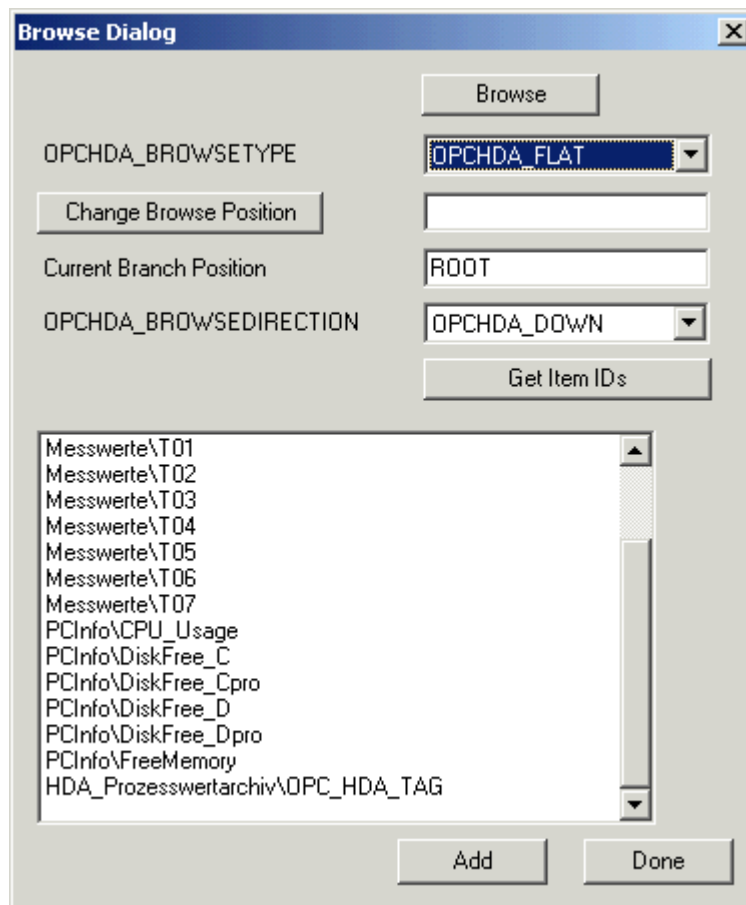
サイクリックアーカイブ	スワップ後のサイクリックアーカイブ	WinCCでサポートされています。	説明
可能	不可	☺	サイクリックアーカイブに時間周期が指定されている場合、プロセス値を編集できます。
可能	可能	☹	対応する時間周期のデータバッファが、アーカイブバックアップにスワップされます。アーカイブバックアップではプロセス値を編集できません。

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。

8.7.5 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

www.opcfoundation.org (<http://www.opcfoundation.org>)

8.7.6 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

概要

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 HDA OPC 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、OPC HDA 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

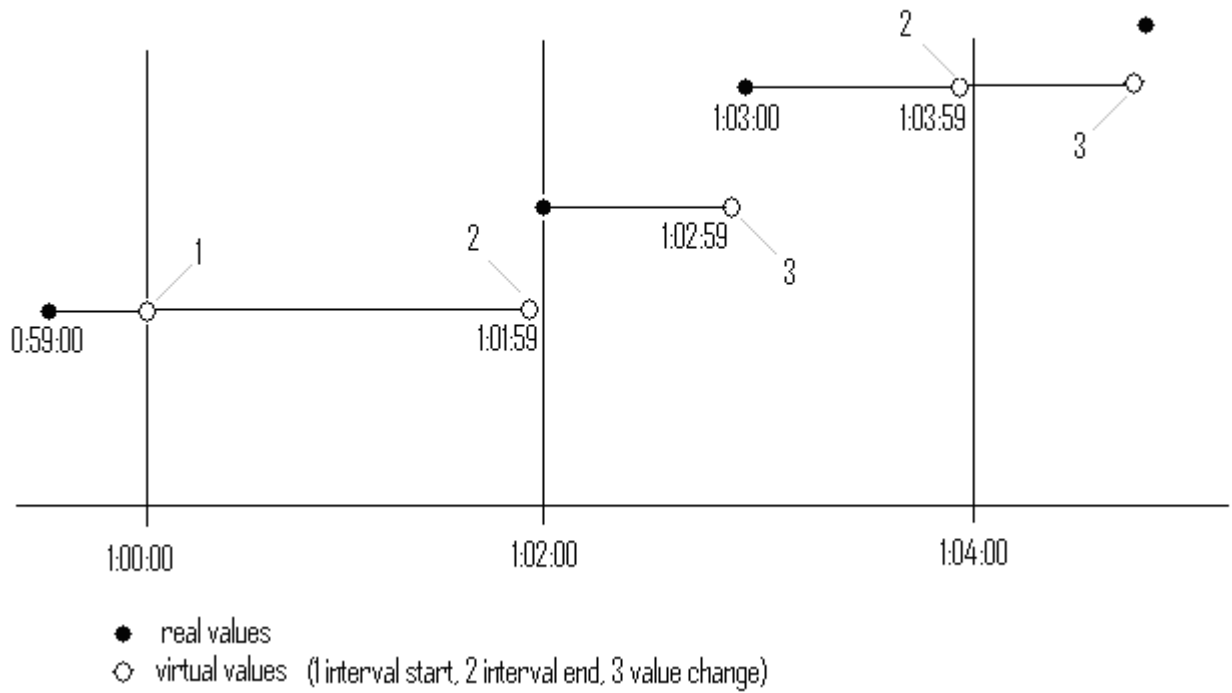
注記

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。

番号	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



8.8 WinCC OPC A&E サーバー

8.8.1 WinCC OPC A&E サーバーの機能

概要

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 OPC A&E サーバーは、WinCC のインストール中に選択できます。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

設定に関する注意

WinCC OPC HDA サーバーが使用されている場合は、アプリケーション「OPC-DA サーバー、OPC-A&E サーバー、OPC-HDA サーバー」を有効にする必要があります。

WinCC Configuration Studio の「コンピュータ」エディタの[WinCC Runtime の起動時に処理]タブで、アプリケーションを有効にできます。

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 445)
- 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 438)
- www.opcfoundation.org (<http://www.opcfoundation.org>)

8.8.2 OPC A&E を使用した WinCC メッセージシステムのマッピング

8.8.2.1 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 POC 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。

OPC	WinCC メッセージシステム
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。
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 443)

WinCC メッセージシステムの属性 (ページ 440)

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 439)

8.8.2.2 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

8.8 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。

8.8.2.3 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。

8.8.2.4 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 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
TYPE NAME	メッセージタイプ名を出力します。	VT_BSTR
FOREGROUND COLOR	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	メッセージに確認が必要かどうかを示します。	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
TEXT08	UserTextBlock08 の内容を出力します。	VT_BSTR
TEXT09	UserTextBlock09 の内容を出力します。	VT_BSTR
TEXT10	UserTextBlock10 の内容を出力します。	VT_BSTR
PROCESSVALUE 01	ProcessValueBlock01 の内容を出力します。	VT_VARIANT
PROCESSVALUE 02	ProcessValueBlock02 の内容を出力します。	VT_VARIANT
PROCESSVALUE 03	ProcessValueBlock03 の内容を出力します。	VT_VARIANT
PROCESSVALUE 04	ProcessValueBlock04 の内容を出力します。	VT_VARIANT
PROCESSVALUE 05	ProcessValueBlock05 の内容を出力します。	VT_VARIANT
PROCESSVALUE 06	ProcessValueBlock06 の内容を出力します。	VT_VARIANT

8.8 WinCC OPC A&E サーバー

OPC 属性	WinCC メッセージシステム	データタイプ
PROCESSVALUE 07	ProcessValueBlock07 の内容を出力します。	VT_VARIANT
PROCESSVALUE 08	ProcessValueBlock08 の内容を出力します。	VT_VARIANT
PROCESSVALUE 09	ProcessValueBlock09 の内容を出力します。	VT_VARIANT
PROCESSVALUE 10	ProcessValueBlock10 の内容を出力します。	VT_VARIANT
STATETEXT	ステータスメッセージを出力します。	VT_BSTR
INFO TEXT	メッセージの情報テキストを出力します。	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 を出力します。	VT_I4
TYPEID	メッセージタイプ ID を出力します。	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	メッセージを生成した AS の番号を出力します。	VT_I2
CPUNR	メッセージを生成した CPU の番号を出力します。	VT_I2
DURATION	メッセージの有効化、無効化、確認の間隔を出力します。	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	メッセージのパラメータを出力します(メッセージ設定のイメージ)。	VT_BSTR
BLOCKINFO	メッセージブロックの現在の内容を出力します。	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 PC の名前を出力します。	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

8.8.2.5 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
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

WinCC	NewState	ChangeState
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>)

8.8.3 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。

8.8.4 階層的アクセス機能付き OPC A&E サーバー

8.8.4.1 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

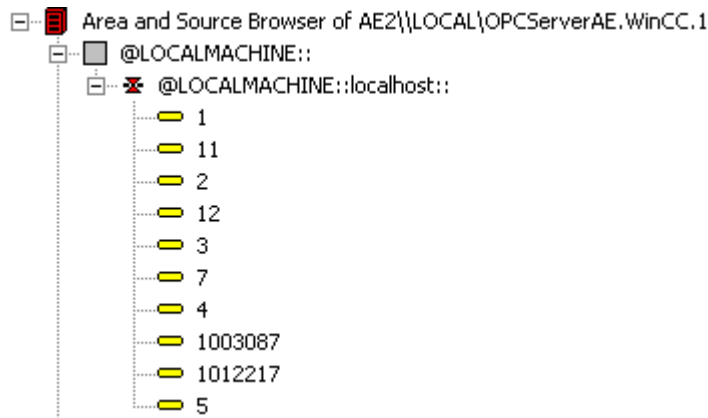
詳細は、「インターフェース」の章の「プロセスコントロールのための OPC - OLE」を参照してください。

8.8.4.2 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。



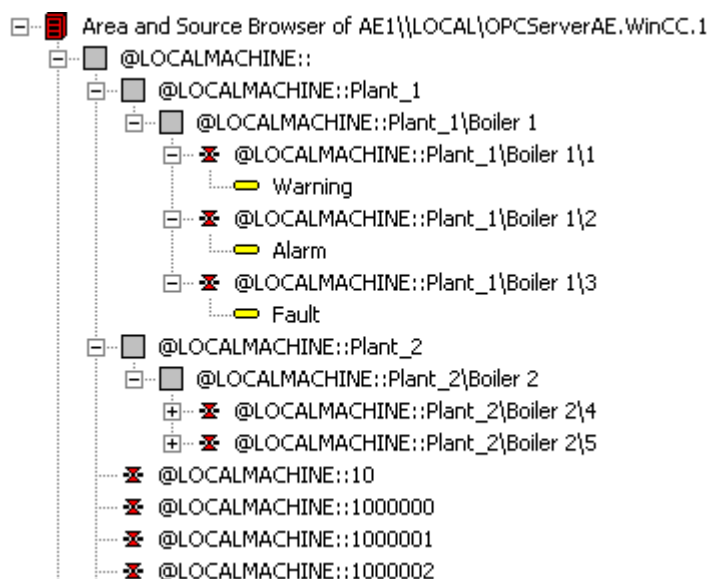
階層アクセス権限を持つ OPC A&E でのメッセージの表示

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、障害箇所に相当します。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。

8.8.4.3 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>::Area\user 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かを示します。	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部 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。

8.8 WinCC OPC A&E サーバー

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

概要

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
TYPE NAME	メッセージタイプ名を出力します。	VT_BSTR
FOREGROUND COLOR	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	メッセージに確認が必要かどうかを示します。	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

OPC 属性	WinCC メッセージシステム	データタイプ
TEXT07	UserTextBlock07 の内容を出力します。	VT_BSTR
TEXT08	UserTextBlock08 の内容を出力します。	VT_BSTR
TEXT09	UserTextBlock09 の内容を出力します。	VT_BSTR
TEXT10	UserTextBlock10 の内容を出力します。	VT_BSTR
PROCESSVALUE 01	ProcessValueBlock01 の内容を出力します。	VT_VARIANT
PROCESSVALUE 02	ProcessValueBlock02 の内容を出力します。	VT_VARIANT
PROCESSVALUE 03	ProcessValueBlock03 の内容を出力します。	VT_VARIANT
PROCESSVALUE 04	ProcessValueBlock04 の内容を出力します。	VT_VARIANT
PROCESSVALUE 05	ProcessValueBlock05 の内容を出力します。	VT_VARIANT
PROCESSVALUE 06	ProcessValueBlock06 の内容を出力します。	VT_VARIANT
PROCESSVALUE 07	ProcessValueBlock07 の内容を出力します。	VT_VARIANT
PROCESSVALUE 08	ProcessValueBlock08 の内容を出力します。	VT_VARIANT
PROCESSVALUE 09	ProcessValueBlock09 の内容を出力します。	VT_VARIANT
PROCESSVALUE 10	ProcessValueBlock10 の内容を出力します。	VT_VARIANT
STATETEXT	ステータスメッセージを出力します。	VT_BSTR
INFO TEXT	メッセージの情報テキストを出力します。	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 を出力します。	VT_I4
TYPEID	メッセージタイプ ID を出力します。	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	メッセージを生成した AS の番号を出力します。	VT_I2
CPUNR	メッセージを生成した CPU の番号を出力します。	VT_I2

OPC 属性	WinCC メッセージシステム	データタイプ
DURATION	メッセージの有効化、無効化、確認の間隔を出力します。	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	メッセージのパラメータを出力します(メッセージ設定のイメージ)。	VT_BSTR
BLOCKINFO	メッセージブロックの現在の内容を出力します。	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 PC の名前を出力します。	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番号を出力します。	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进行处理するには、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
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

WinCC	NewState	ChangeState
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。

8.8.4.4 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。

8.8.5 アーカイブメッセージの読取り

8.8.5.1 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"をサポートしている必要があります。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られます。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。

8.8.5.2 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。

```
OPCHAEsServer 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)。

```
OPCHAEsServer 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 >= htStartTime
- 最後に転送されたメッセージのタイムスタンプ >= htEndTime

bBounds=TRUE:

- 最初に転送されたメッセージのタイムスタンプ <= htStartTime
- 最後に転送されたメッセージのタイムスタンプ >= hEndTime

デフォルト設定は FALSE です。

8.8.5.3 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"OPCHAEServer"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」モードで使用されます。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。

8.8.5.4 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

名前	EventType	EventType (アーカイブメッセージ)
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
```

8.9 WinCC OPC UA サーバー

8.9.1 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
- 設定された C/VB スクリプトの実行

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 1.05.03 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 472)」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://[NodeName]:[Port]」

パラメータ	説明
NodeName	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

8.9.2 OPC UA のセキュリティ概念

概要

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。

8.9 WinCC OPC UA サーバー

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 493)」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。

ストレージパス	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	opclUAServer\PKI\CA\certs
	秘密鍵	opclUAServer\PKI\CA\private

設定ファイルの保存場所を変更できます。

```
<ServerCertificate>[ApplicationPath]/PKI/CA/certs/</ServerCertificate>
```

注記

サーバー証明書を提供するために、Windows ベースの証明機関がサポートされています。

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のみが必要となります。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。

```
<DefaultApplicationCertificateStore>
  <!--The maximum size of the TrustList in bytes. 0 means no limit.-->
  <MaxTrustListSize>0</MaxTrustListSize>
  <!--For CA signed certificates this flag controls if server shall send the complete certificate chain instead of just sending the certificate.
  <SendCertificateChain>true</SendCertificateChain>
  <!--File based certificate store used with OpenSSL
  <OpenSSLStore>
    <CertificateTrustListLocation>[ApplicationPath]/PKI/CA/certs/</CertificateTrustListLocation>
    <CertificateRevocationListLocation>[ApplicationPath]/PKI/CA/crl/</CertificateRevocationListLocation>
    <IssuersCertificatesLocation>[ApplicationPath]/PKI/CA/certs/</IssuersCertificatesLocation>
    <IssuersRevocationListLocation>[ApplicationPath]/PKI/CA/crl/</IssuersRevocationListLocation>
  </OpenSSLStore>

  <!--Windows based certificate store.
  <!--Application instance certificate for the Server.-->
  <ServerCertificate>
</DefaultApplicationCertificateStore>
```

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。

注記

サーバー証明書を提供するために、Windows ベースの証明機関がサポートされています。

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。

```
<!--Folder used to store rejected client certificates. Administrators can copy files from here to the trust list.
[ApplicationPath] can be used as placeholder for the application path.-->
<RejectedCertificatesDirectory>[ApplicationPath]/PKI/CA/rejected/certs</RejectedCertificatesDirectory>
```

注記

拒否された証明書の保管は[File based]です。

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

WinCC OPC UA サーバーの設定ファイル (ページ 493)

OPC UA 通信における信頼関係の確立 (ページ 240)

8.9.3 セキュリティメカニズムの構成

概要

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、[UaEndpoint] > [SecuritySetting]にあるサーバー設定ファイルを使用して設定されます。

EndpointConfiguration

[UaEndpoint] > [SecuritySetting]で、サーバーで使用可能なすべての[Security Profile]と[Message Security Mode]の組み合わせの一覧を参照できます。

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)

Security Profile	Message Security Mode	説明
http://opcfoundation.org/UA/ SecurityPolicy#Aes128_Sha256_RsaOaep 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)
http://opcfoundation.org/UA/ SecurityPolicy#Aes256_Sha256_RsaPss 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"、"Aes128_Sha256_RsaOaep"、"Aes256_Sha256_RsaPss"の使用の必要条件: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

```
<UaEndpoint>
  <SerializerType>Binary</SerializerType>
  <!-- URL of the Endpoint
  <Url>opc.tcp://[NodeName]:4862</Url>
  <!-- Optional URL that allows to define a specific address the stack should use to bind to.
  <!-- It is up to an administrator to configure the actual exposed SecurityPolicies.
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#None</SecurityPolicy>
    <MessageSecurityMode>None</MessageSecurityMode>
  </SecuritySetting>
  <!-- The SecurityPolicy Basic128Rsa15 is deprecated and is no longer considered secure.
  <!-- <SecuritySetting>
  <!-- The SecurityPolicy Basic256 is deprecated and is no longer considered secure.
  <!-- <SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <!-- Flag indicating if the endpoint is provided in GetEndpoints and is therefore visible to a client. Default is true. -->
  <IsVisible>true</IsVisible>
  <!-- Flag indicating if the endpoint URL is provided as discovery URL. Default is true. -->
  <IsDiscoveryUrl>true</IsDiscoveryUrl>
  <!-- This option can be activated if certificates are used only for message security but not for application authentication.
  <AutomaticallyTrustAllClientCertificates>false</AutomaticallyTrustAllClientCertificates>
  <!-- For calculating the server signature, the server needs to append the client certificate to the client nonce.
  <CreateSignatureWithChain>false</CreateSignatureWithChain>
  <!-- Some of the OPC UA security checks are optional in OPC UA or cause interoperability issues with older OPC UA clients
  <SecurityCheckOverrides>
  <!-- Optional list of clients operating with a reverse connect from the server.-->
  <ReverseConnect>
  <!-- Optional list of alternative endpoint URLs known to the server.
  <AlternativeEndpointUrls>
  <!-- Windows based certificate store.
  <!-- Flag indicating if the endpoint is returned in discovery depending on EndpointUrl match. -->
  <ReturnOnlyOnEndpointUrlMatch>false</ReturnOnlyOnEndpointUrlMatch>
</UaEndpoint>
```

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 [UserIdentityTokens] 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。

- 匿名ログイン
- ユーザー名とパスワードによる認証:
通信を確立するときに、有効なユーザー名とパスワードの組み合わせが指定されます。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。
- ユーザー証明書による認証:
通信を確立するときに、有効なユーザー証明書と秘密鍵の組み合わせが指定されます。WinCC OPC UA サーバーは、組み合わせを確認して、ユーザー証明書が信頼できるものであると通信を許可します。

UserIdentityTokens は、WinCC OPC UA サーバーの設定ファイルで設定されます。

```
<UserIdentityTokens>
  <!--Enable anonymous login true/false-->
  <EnableAnonymous>true</EnableAnonymous>
  <!--Enable user/password login true/false-->
  <EnableUserPw>true</EnableUserPw>
  <!--Enable certificate based user login true/false-->
  <EnableCertificate>true</EnableCertificate>
  <!--Enable issued token based user login true/false-->
  <EnableIssuedToken>false</EnableIssuedToken>
  <!--The security policy to use when encrypting or signing the UserIdentityToken when it is passed to the server.
  <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</SecurityPolicy>
  <!--Configuration for file based certificate store to handle user certificates.-->
  <DefaultUserCertificateStore>
  <!--Folder used to store rejected user certificates.-->
  <RejectedUserCertificatesDirectory>[ApplicationPath]/PKI/CA/rejected/certs</RejectedUserCertificatesDirectory>
  <!--Maximum number of certificates stored in the rejected directory.-->
  <RejectedUserCertificatesCount>100</RejectedUserCertificatesCount>
  <IssuedTokenSettings>
</UserIdentityTokens>
<!--User identity token configuration end-->
```

この設定では、WinCC OPC UA サーバーは匿名ユーザーと Policy の [UserName] と [Certificate] の両方をサポートします。

8.9.4 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 1.0.10 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
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 設定された C-VB スクリプト
*):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 427)」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 1.04 をサポートします。

- 6.6.6 ベースサーバー動作ファセット
- 6.6.24 標準イベントサブスクリプションサーバーファセット

8.9 WinCC OPC UA サーバー

- 6.6.26 A & C ベース条件サーバーファセット
- 6.6.39 メソッドサーバーファセット
- 6.6.48 履歴未処理データサーバーファセット
- 6.6.53 履歴データ挿入サーバーファセット
- 6.6.54 履歴データ更新サーバーファセット
- 6.6.56 履歴データ削除サーバーファセット
- 6.6.143 UA-TCP UA-SC UA バイナリ
- 6.6.161 セキュリティポリシー - なし
- 6.6.162 セキュリティポリシー - Basic128Rsa15
- 6.6.163 セキュリティポリシー - Basic256 #
- 6.6.164 セキュリティポリシー[A] - Aes128-Sha256-RsaOaep
- 6.5.165 セキュリティポリシー[B] - Basic256Sha256
- 6.6.166 セキュリティポリシー - Aes256-Sha256-RsaPss
- 6.6.167 ユーザートークン - 匿名ファセット
- 6.6.168 ユーザートークン - ユーザー名パスワードサーバーファセット
- 6.6.169 ユーザートークン - X509 証明書サーバーファセット

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"
6.6.2 Core Server Facet	Attribute Services	Attribute Write Index
6.6.16 Standard DataChange Subscription Server Facet	Monitored Item Services	DeadBand Filter
6.6.18 Enhanced DataChange Subscription Server Facet	Monitored Item Services	-

Profile	"Group"	サポートされていませ ん"Conformance Unit"
6.6.21 Data Access Server Facet	Data Access	Data Access Analog Data Access Multistate Data Access PercentDeadBand Data Access Semantic Changes Data Access Two State
6.6.70 Standard UA Server Profile	Attribute Services	Attribute Write StatusCode & Timestamp

8.9.5 WinCC OPC UA サーバーの名前領域

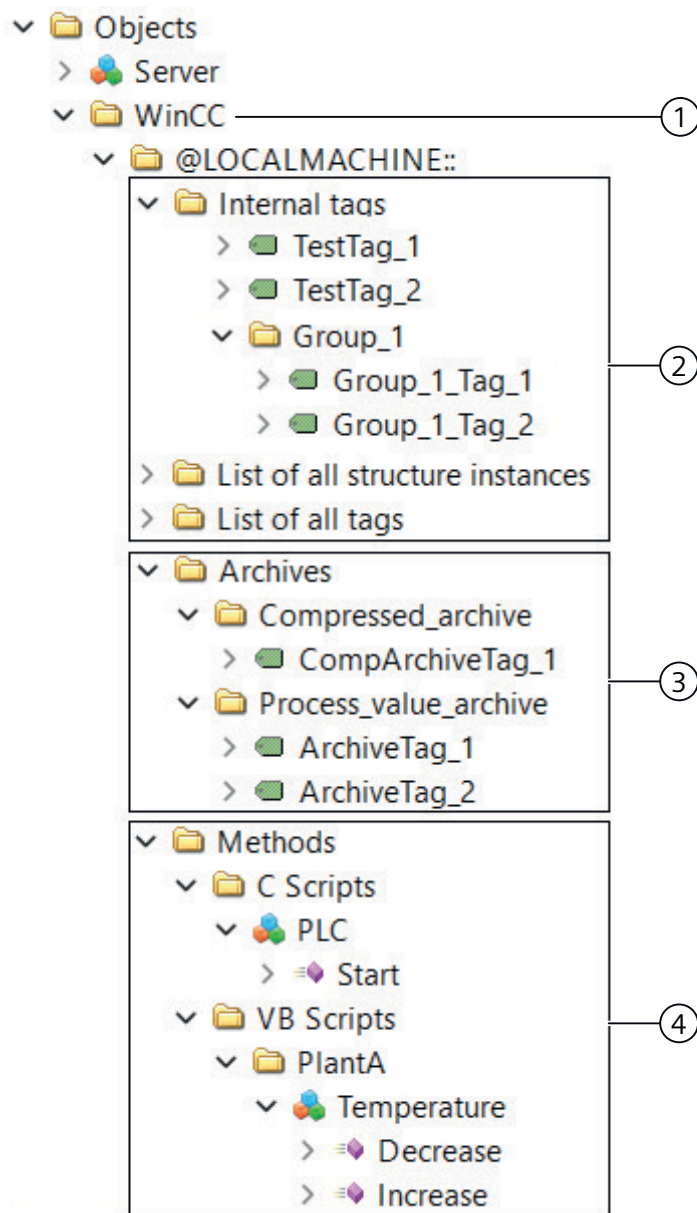
概要

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)
- アーカイブタグを含むデータログ
- WinCC メッセージ
- 設定された C-VB スクリプト

WinCC OPC UA サーバーの名前空間は「Objects」デフォルトフォルダにあります。

以下の図には、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ています。



- ① WinCC の特定の名前空間の開始ノード。
- 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
- ④ 設定された C/VB スクリプト

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 タグがアーカイブされる場合、表示される OPC UA タグには、アーカイブ設定のための「HasHistoricalConfiguration」タイプの参照が追加で含まれます。「Historizing」属性および「AccessLevel」属性がそれに応じて設定されます。

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 1.04 Specification』文書の「§5.6」に記載されています。

属性	説明	コメント
Nodetid	WinCC タグの一意的指定	-
BrowseName	WinCC タグ名	-
DisplayName	WinCC タグ名	-
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 データ型 <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。

ネストされた構造のサポート(UDT 内の UDT)

WinCC はタグの階層構造をサポートしており、タグには他のタグがメンバーとして含まれ、異なる構造を形成することができます。

このような構造内タグは、「AS からロード」によりインポートするか、OPC サーバー上で参照することによってインポートします。

アーカイブタグの表示

プロセス値および圧縮アーカイブが「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、関連するアーカイブタグに対する「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アーカイブタグが「BaseDateVariableType」タイプの OPC UA タグによって表示されます。アーカイブタグには常に、アーカイブ設定の「HasHistoricalConfiguration」タイプの参照が含まれます。

以下の表は、WinCC アーカイブタグを表す 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 1.04 Specification』文書の「§5.6」に記載されています。

属性	説明	コメント
NodeId	アーカイブタグの一意の指定	-
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	なし	アーカイブタグの場合、この属性は読み取ることも変更することもできません。
DataType	たとえば以下のような、WinCC タグタイプに対応する OPC UA データ型 <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。

8.9.6 OPC UA データアクセス

内部および外部の WinCC タグが「DataItem」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。

8.9.7 OPC UA Historical Access

概要

[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は、「OPC UA Part 11 - Historical Access 1.04 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。追加情報は、『OPC UA Part 11 - Historical Access 1.04 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

8.9.8 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へ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 WinCC-Event-Notifications と Subscriptions 経由で、Monitored Event Items 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 名前空間の開始ノードです。

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 1.05.03」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。詳細については、Part 9: Alarms and Conditions 1.05.03 Specification の仕様で確認できます。

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せん。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」の優先度を設定できます。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
ConditionClassName	「ProcessConditionClassType」
Retain	保留メッセージの場合、「TRUE」
Nodeld	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
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

メッセージステータス	EnabledState/Id	ActiveState/Id	AckedState/Id
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

メッセージグループ

WinCC 7.3 では、WinCC アラームグループは名前空間に表示されません。

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。

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 486)

8.9.9 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	メッセージに確認が必要かどうかを示します。	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
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	ランタイムの起動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を示します。	String
QUITCOUNT	未処理で未確認のメッセージの数	Int32
PARAMETER	メッセージの設定パラメータ	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

WinCC メッセージ属性	意味	データタイプ
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	メッセージのコメント	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	ランタイムの起動後のメッセージの数	Int64

特別な WinCC メッセージ属性の設定

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。3つの異なるマッピングモード[Mapping Mode 1]を含む設定ファイルが、デフォルトで有効にされます。[Mapping Mode 3]では、特別なメッセージ属性である BIG_COUNTER、HIDDEN_COUNT、OS_EVENTID、OS_HIDDEN がさらに有効になります。

以下の手順を実行し、[Mapping Mode 3]を有効にします。

1. テキストエディタを使用して「OPC\AlarmEvent\bin」フォルダの WinCC インストールパスの「CcAeProvider.ini」ファイルを開きます。
2. [Mapping Mode 3]セクションで、[[OpcMapping]], [OpcSource ...], [OpcMessage ...]のラインの始めにあるコメントマークを削除します。

```

;=====
; Mapping Mode 3 "OPC Mapping"
;=====
[OpcMapping]
OpcSource = TEXT02
; EVENTID
; TEXT01...10
; INFOTEXT
OpcMessage = TEXT01
; TEXT01...10
; STATETEXT
; INFOTEXT
; COMMENT

```

3. 必要に応じて、セクションのメッセージ属性のマッピングを変更します。
4. ランタイムを再起動します。

特別なメッセージ属性のある WinCC OPC UA サーバーの[Mapping Mode 3]が有効にされます。

下記も参照

OPC UA アラームと条件 (ページ 481)

8.9.10 OPC UA メソッド

8.9.10.1 [OPC UA メソッド]エディタ

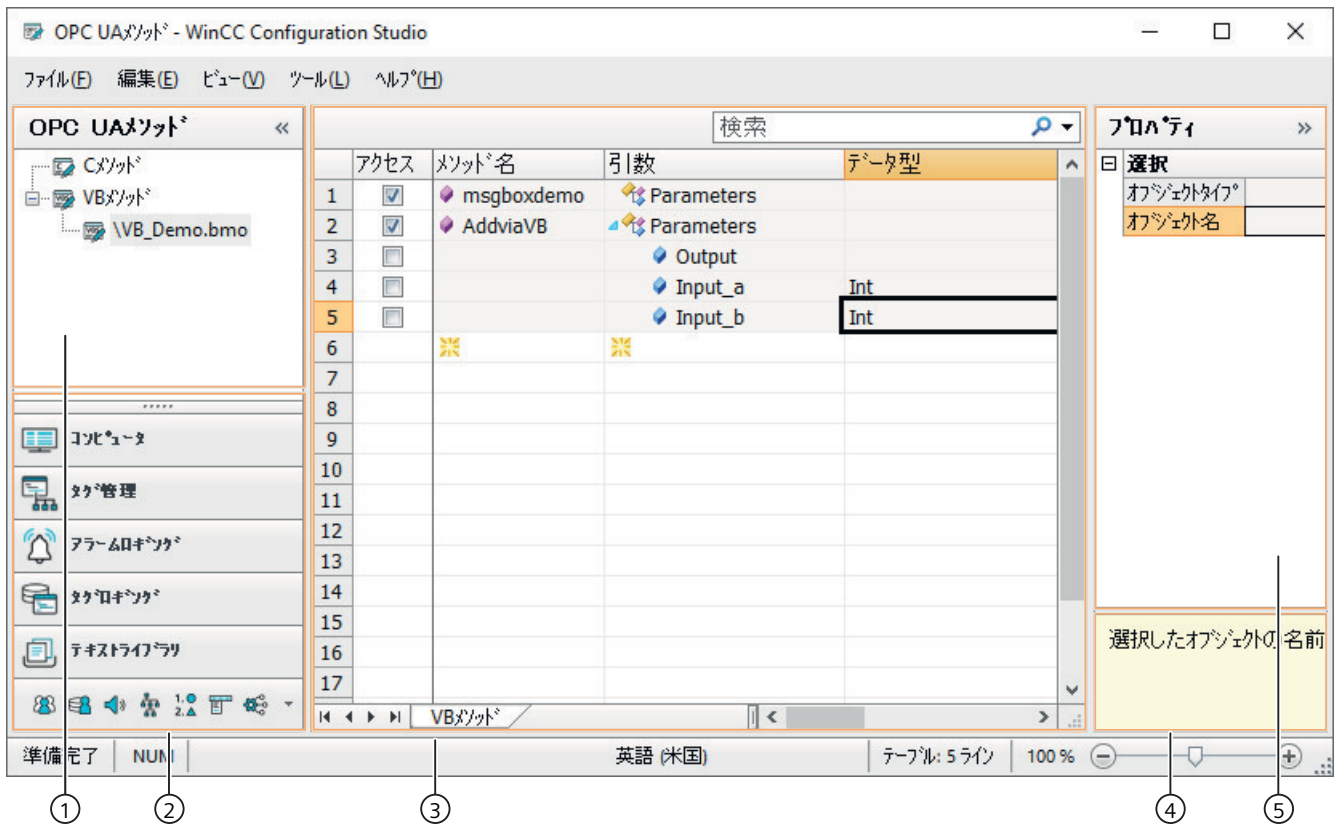
[OPC UA メソッド]エディタでは、OPC UA 経由で呼び出す作成したスクリプトを管理します。

WinCC エクスプローラで[グローバルスクリプト]>[OPC UA メソッド]エントリをダブルクリックしてエディタを起動します。

[OPC UA メソッド]エディタの構造

[OPC UA メソッド]エディタは、3つの作業エリアに分かれています。

- ナビゲーションエリア
 - オブジェクトをフォルダとして表示するツリー表示
 - エディタ間で切り替えるためのナビゲーションバー
- テーブルエリア
 - オブジェクトの選択と設定
- プロパティエリア
 - 選択したオブジェクトのプロパティ
 - 選択したプロパティの[ポップアップヒント]



① ナビゲーションエリア

プロジェクト機能とプロジェクトモジュールは、ナビゲーション領域にツリービューとして表示されます。



スクリプトは次の順序でソートされます:

- C メソッド
- VB メソッド

選択したフォルダでは、サポートされているすべてのメソッドがテーブルエリアに表示されます。

アイコン

[OPC UA メソッド]エディタでは、次のシンボルが使用されます。

アイコン	意味
	C メソッド
	VB メソッド/VB モジュール

② エディタの選択

ナビゲーションバーは、ツリー表示下のエリアに表示され、別の WinCC エディタへのアクセスを有効にします。

③ テーブルエリア

表には、ツリービューで選択されたスクリプト言語に割り当てられている、OPC UA でサポートされているすべてのメソッドが表示されます。VB メソッドの場合、特定のプロジェクトモジュールを選択して、そのモジュールに含まれるメソッドのみを表示することもできます。

テーブル領域では、メソッドへの OPC UA アクセスを設定し、VB メソッドに転送されるパラメータのデータ型を指定します。

テーブル領域で新しい要素を編集、削除、または追加することはできません。メソッドは、[グローバル スクリプト C]または[グローバルスクリプト VB]エディタで作成します。

④ ポップアップヒント

選択したプロパティの説明を表示します。

⑤ プロパティ

選択したオブジェクトのプロパティが表示されます。

[OPC UA メソッド]エディタには編集可能なプロパティはありません。

8.9.10.2 OPC UA 経由でメソッドへのアクセスを設定する方法

概要

作成されたメソッドについては、[OPC UA メソッド]エディタで OPC UA 経由のアクセスを設定します。

プロジェクト関数(グローバルスクリプト C)とプロジェクトモジュール(グローバルスクリプト VB)がサポートされています。

次の情報に注意してください:

- 内部関数、標準関数、標準モジュールへの OPC UA アクセスはできません。ただし、これらの関数はプロジェクト関数内で使用できます。
- OPC UA を使用して VB スクリプトをアドレス指定する場合、戻りパラメータのデータ型は常に「変数」になります。OPC UA クライアントは、受信した変数から戻り値の正しいデータ型を抽出する必要があります。
- データ構造と配列は、入力パラメータおよび戻りパラメータとしてサポートされていません。
- 参照経由のタグ転送(ByRef)は関数に対してのみサポートされ、プロシージャに対してはサポートされません(Sub)。参照を介してタグ転送を使用する手順を作成した場合、[OPC UA メソッド]エディタには表示されません。
- ユーザー認証はサポートされていません。
- OPC UA クライアントは一度に 1 つのメソッドのみを実行できます。

必要条件

- プロジェクトが作成されていること。
- メソッドは、プロジェクト関数またはプロジェクトモジュールとして作成されます。

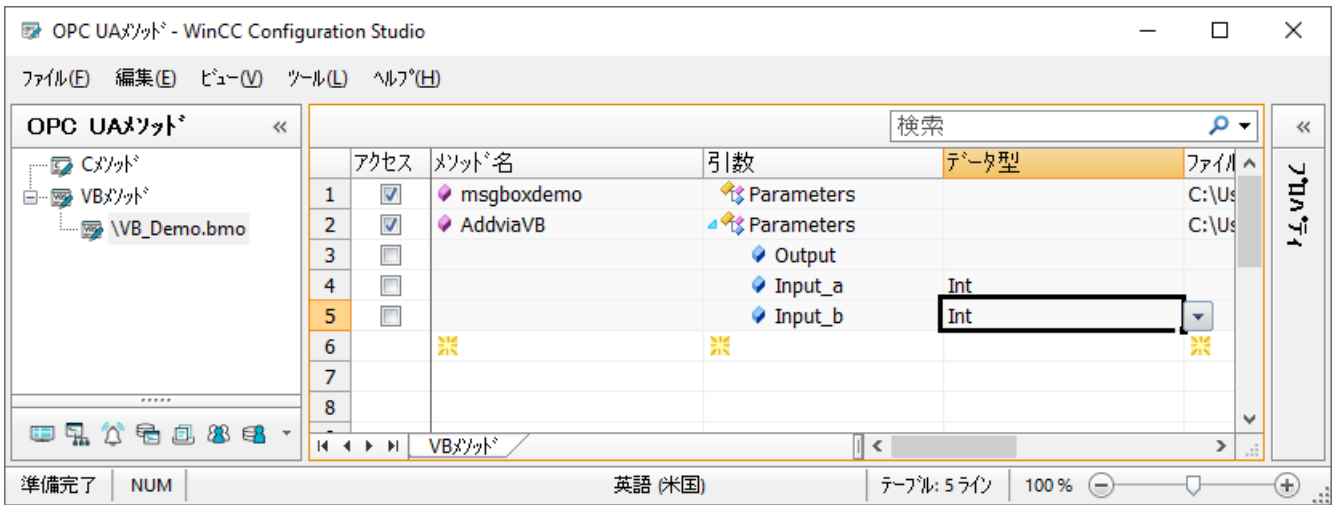
手順

1. WinCC エクスプローラで[グローバルスクリプト]>[OPC UA メソッド]エントリをダブルクリックして、[OPC UA メソッド]エディタを開きます。
[OPC UA メソッド]エディタが開きます。ナビゲーション領域には、C メソッドと VB メソッドのフォルダが表示されます。
2. ナビゲーションエリアでフォルダを選択します。
対応するメソッドは、テーブルエリアに表示されます。

3. OPC UA 経由で利用できるようにする各メソッドについて、[アクセス]オプションボタンを選択します。
4. VB メソッドの場合:[引数]列でメソッドのパラメータを展開し、[データ型]列で各入力パラメータに適したデータ型を選択します。

注記

デフォルトでは、常に[文字列]データ型が指定されます。



結果

メソッドに対して OPC UA 経由のアクセスを設定しました。

OPC UA クライアントでは、「Call」関数を使用してメソッドを呼び出すことができるようになりました。名前、説明、入力/戻りパラメータなどの既存のメタデータを表示することもできます。

8.9.11 WinCC OPC UA サーバーの設定

8.9.11.1 WinCC OPC UA サーバーの設定ファイル

概要

WinCC OPC UA サーバーは、設定ファイル「OPCUAServerWinCC.xml」を使用して設定します。

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。

8.9 WinCC OPC UA サーバー

「OPC UA サーバーのコンフィグレーション方法 (ページ 499)」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:<UaEndpoint>セクション

「OPC UA のセキュリティ概念 (ページ 465)」の URL で追加情報を参照できます。

<UaEndpoint>	
<Url> <...></...> </Url>	WinCC OPC UA サーバーの URL の設定

<pre><SecuritySetting> <SecurityPolicy> <...></...> </SecurityPolicy> <MessageSecurityMode> <...></...> </MessageSecurityMode> ... </SecuritySetting></pre>	<p>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。</p>
<pre><DefaultApplicationCertificateStore> <TrustedCertificateStore> > <TrustedCertificates> <...></pre>	<p>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。</p>
<pre></UaEndpoint></pre>	

例:OPC UA アプリケーションセキュリティ

```

<UaEndpoint>
  <SerializerType>Binary</SerializerType>
  <!-- URL of the Endpoint
  <Url>opc.tcp://[NodeName]:4862</Url>
  <!-- Optional URL that allows to define a specific address the stack should use to bind to.
  <!-- It is up to an administrator to configure the actual exposed SecurityPolicies.
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#None</SecurityPolicy>
    <MessageSecurityMode>None</MessageSecurityMode>
  </SecuritySetting>
  <!-- The SecurityPolicy Basic128Rsa15 is deprecated and is no longer considered secure.
  <!-- <SecuritySetting>
  <!-- The SecurityPolicy Basic256 is deprecated and is no longer considered secure.
  <!-- <SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <SecuritySetting>
    <SecurityPolicy>http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss</SecurityPolicy>
    <MessageSecurityMode>Sign</MessageSecurityMode>
    <MessageSecurityMode>SignAndEncrypt</MessageSecurityMode>
  </SecuritySetting>
  <!-- Flag indicating if the endpoint is provided in GetEndpoints and is therefore visible to a client. Default is true. -->
  <IsVisible>true</IsVisible>
  <!-- Flag indicating if the endpoint URL is provided as discovery URL. Default is true. -->
  <IsDiscoveryUrl>true</IsDiscoveryUrl>
  <!-- This option can be activated if certificates are used only for message security but not for application authentication.
  <AutomaticallyTrustAllClientCertificates>false</AutomaticallyTrustAllClientCertificates>
  <!-- For calculating the server signature, the server needs to append the client certificate to the client nonce.
  <CreateSignatureWithChain>false</CreateSignatureWithChain>
  <!-- Some of the OPC UA security checks are optional in OPC UA or cause interoperability issues with older OPC UA clients
  <SecurityCheckOverrides>
  <!-- Optional list of clients operating with a reverse connect from the server.-->
  <ReverseConnect>
  <!-- Optional list of alternative endpoint URLs known to the server.
  <AlternativeEndpointUrls>
  <!-- Windows based certificate store.
  <!-- Flag indicating if the endpoint is returned in discovery depending on EndpointUrl match. -->
  <ReturnOnlyOnEndpointUrlMatch>false</ReturnOnlyOnEndpointUrlMatch>
</UaEndpoint>
    
```

構造:<UaServerConfig>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 465)」を参照してください。

<UaServerConfig>	
<UserIdentityTokens>	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。
</UserIdentityTokens>	

<pre><FastInsert> <Users> <...></...> </Users> <Clients> <...></...> </Clients> </FastInsert></pre>	<p>最適化した WinCC アーカイブ書き込みアクセスの設定</p>
<pre></UaServerConfig></pre>	

構造:<ServerCertificate>セクション

<ServerCertificate>セクションの<UaServerConfig>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 465)」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。

<pre><ServerCertificate> <CertificateSettings></pre>	
<pre><OrganizationUnit>...</...> <Organization>...</...> <Country>...</...></pre>	<p>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。</p>
<pre><KeyLength>...</...></pre>	<p>証明書を作成するときに使われた秘密鍵の長さ 長さは CertificateType により異なります。</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 が使用されるとき最短の長さ¹⁾
<pre><CertificateType>...</...></pre>	<p>証明書に署名するために使用される証明書タイプ</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 Sha256¹⁾

<pre><YearsValidFor>...</...></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:5
<pre></CertificateSettings> </ServerCertificate></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
- CertificateType:Sha256

例:証明書の制御のためのパラメータ

```
<ServerCertificate>
  <!--File based server certificate store used with OpenSSL
  [ApplicationPath] can be used as placeholder for the application path.-->
  <OpenSSLStore>
  </OpenSSLStore>
  <!--Windows based server certificate store.
  </Windows based server certificate store.
  <!--SecurityCheckOverrides for the own server certificate. -->
  <SecurityCheckOverrides>
  </SecurityCheckOverrides>
  <!--Enable server certificate creation if certificate is not available; true/false-->
  <GenerateCertificate>true</GenerateCertificate>
  <!--Settings for a certificate that is generated by the server-->
  <CertificateSettings>
    <!--Name of the application - [ServerName] is the default value to use the configured server name-->
    <CommonName>[ServerName]</CommonName>
    <!--DomainComponent - [NodeName] is the default value to use the hostname of the machine -->
    <DomainComponent>[NodeName]</DomainComponent>
    <!--Name of the organization using the OPC UA server-->
    <Organization>Siemens AG</Organization>
    <!--Name of the organization unit using the OPC UA server-->
    <OrganizationUnit>DI FA HMI</OrganizationUnit>
    <!--Name of the location where the OPC UA server is running -->
    <!--<Locality>LocationName</Locality> -->
    <!--State where the OPC UA server is running -->
    <State>EW</State>
    <!--Two letter code for country where the OPC UA server is running, e.g. DE or US-->
    <Country>DE</Country>
    <!--The number of years the certificate is valid for. The maximum accepted number is 20,
    <YearsValidFor>5</YearsValidFor>
    <!--Key length of the certificate to create. Valid values are 1024, 2048 for RsaMin and 2048, 3072 and 4096 for RsaSha256-->
    <KeyLength>2048</KeyLength>
    <!--Defines the algorithm used to sign the certificate. Valid values are RsaMin and RsaSha256.
    <CertificateType>RsaSha256</CertificateType>
    <!-- An application instance certificate needs to provide one or more DNSNames and/or IPAddresses at which the Endpoint can be reached.
    This information is added to the SubjectAlternativeName of the certificate.
    If this parameter is not set, the [NodeName] is used by default.-->
    <!-- Example for a list with 2 DNSNames plus 2 IPAddresses
    <IPAddress>2a00:1158:400:407:0:0:0:1b2</IPAddress>
    <IPAddress>213.95.4.190</IPAddress>
    <DNSName>demo.sagautomation.com</DNSName>
    <DNSName>[NodeName]</DNSName> -->
    <DNSName>[NodeName]</DNSName>
  </CertificateSettings>
</ServerCertificate>
```

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。

例:サーバー証明書のストレージパス

```

<ServerCertificate>
  <!--File based server certificate store used with OpenSSL
    [ApplicationPath] can be used as placeholder for the application path.-->
  <OpenSSLStore>
    <ServerCertificate>[ApplicationPath]/PKI/CA/certs/</ServerCertificate>
    <ServerPrivateKey>[ApplicationPath]/PKI/CA/private/</ServerPrivateKey>
  </OpenSSLStore>

  <!--Windows based server certificate store.
    - <StoreLocation> location of the store. Valid values are LocalMachine and CurrentUser
    - <StoreName> is the name of the certificate store in the local computer
    - <ServerCertificateThumbprint> is the thumbprint of the server certificate used to load from store
  <WindowsStore>
    <StoreLocation>LocalMachine</StoreLocation>
    <StoreName>UA Applications</StoreName>
    <ServerCertificateThumbprint></ServerCertificateThumbprint>
  </WindowsStore> -->

  <!--SecurityCheckOverrides for the own server certificate. -->
  <SecurityCheckOverrides>

  <!--Enable server certificate creation if certificate is not available; true/false-->
  <GenerateCertificate>true</GenerateCertificate>
  <!--Settings for a certificate that is generated by the server-->
  <CertificateSettings>
</ServerCertificate>

```

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。

8.9.11.2 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。

設定ファイルを開く

1. Windows エクスプローラを開きます。
2. 「<WinCC プロジェクトフォルダー>\OPCUAServer」 ディレクトリに移動します。
3. 設定ファイル「OPCUAServerWinCC.xml」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 493)」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、<Url>でポート番号 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[NodeName]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:

```
- <Url>  
  opc.tcp://[NodeName]:4862  
</Url>
```

セキュリティ設定の指定

1. 通信のためのセキュリティ設定を定義します。
詳細情報については、「OPC UA のセキュリティ概念 (ページ 465)」を参照してください。
2. <UaEndpoint>で、対応する「メッセージセキュリティモード」を設定します。
3. 設定を無効にするには、<SecuritySetting>... </SecuritySetting> エントリ全体を削除します。

例:

- <SecuritySetting>
 <SecurityPolicy>http://opcfoundation.org/UA/
 SecurityPolicy#None</SecurityPolicy>
 <MessageSecurityMode>None</MessageSecurityMode>
</SecuritySetting>

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。

生産性の高い動作を行っている際にクライアントとサーバー間の安全な通信を実現するために、少なくとも以下の設定を使用します。

- SecurityPolicy: Basic128Rsa15
 Message Security Mode: Sign
-

ユーザー ID の指定

1. <UserIdentityTokens>で、接続を確立するためのユーザー 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 465)」を参照してください。
2. 設定を有効または無効にするには、値を[True"/][False]に従って設定します。

例:

- <UserIdentityTokens>
 <EnableAnonymous>>true</EnableAnonymous>
 <EnableUserPw>>true</EnableUserPw>
 <EnableCertificate>>true</EnableCertificate>
 </UserIdentityTokens>

最適 WinCC アーカイブ書き込みアクセスの設定

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設定します。

- [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>
```

8.9.12 UaExpert

8.9.12.1 UaExpert に関する一般情報

概要

UaExpert は、Unified Automation (<https://www.unified-automation.com/products/development-tools/uaexpert.html>)からテスト目的で提供されている無料の OPC UA クライアントです。

UaExpert には、次のプラグインがあります:

- OPC UA Data Access View
- OPC UA Alarms & Conditions View
- OPC UA Historical Trend View
- OPC UA Performance Plugin
- Image Viewer
- Server Diagnostics View
- Simple Datalogger CSV Plugin
- GDS Push-Model Plugin

UaExpert は、Windows と Linux で利用できます。

動作原理

OPC UA クライアント「UaExpert」は、OPC UA サーバーと直接通信します。

クライアントとサーバーは Ethernet 経由で接続され、TCP/IP を使用して OPC UA 経由で通信します。

必要条件

UaExpert が特定の PLC タグへの読み取りおよび書き込みアクセスを承認するには、これらのタグを OPC UA に対して有効にする必要があります。

下記も参照

WinCC OPC UA サーバーの動作原理 (ページ 464)

8.9.12.2 サポートされている OPC UA 機能の概要

OPC UA クライアント「UaExpert」は、OPC UA の次の機能をサポートしています。

- Browse
- Read/Write
- Subscription
- Methods
- Historical Access
- Alarms and Conditions

Browse および Read/Write

それぞれのノードとタグを UaExpert の[Address Space]ビューで検索できます。

タグの機能と値は、対応するタグをクリックするとすぐに[Attributes]ビューに表示されます。

現在の値を取得するには、[Refresh]をクリックします。このプロセスでは、[Read]機能をトリガしています。

Subscription

UaExpert は、[EventNotifier]属性のあるすべてのノードにサブスクライブできます。

これらのノードは、サブスクリプションでサブスクライブしてイベントを受信できます。

UaExpert でサブスクリプションを作成するには、次の手順に従います。

1. 個々の変数を[Data Access View]ビューにドラッグアンドドロップします。
その後、UaExpert はタグへのサブスクリプションを自動的に作成します。
2. [Data Access View]ビューでは、サブスクライブされたタグのショートカットメニューで [Monitored Item Settings] ウィンドウを開きます。
3. 次の設定を指定します。
 - Sampling Intervall
 - Queue Size
 - Discard Oldset
4. サブスクライブされたタグのショートカットメニューから、[Publishing Mode]を有効にします。
5. サブスクライブされたタグのショートカットメニューを開くには、[Subscription Settings] ウィンドウを開き、サブスクリプションの設定を行います。

Methods

OPC UA サーバーの利用可能なメソッドは、[Address Space]にと一緒に表示されます。

UaExpert でメソッドを呼び出すには、次の手順に従います。

1. それぞれのメソッドを右クリックして、[Call]を選択します。
2. ポップアップウィンドウでは、入力引数を指定して[Call]をクリックします。

このメソッドが実行されます。OutputArguments が表示されます。

Historical Access

選択したタグの[属性]ビューでは、タグの履歴データが収集されているかどうか、および OPC UA クライアントがデータにアクセスできるかどうかを確認できます。

[履歴トレンドビュー]ビューでは、データを周期的に、または 1 回だけクエリできます。

[履歴トレンドビュー]ビューは、2 つの領域に分かれています。

- [Configuration]:
 - 複数のデータソースが存在する場合は、異なる表示色を選択します。
 - Single Update を選択した場合、表示されるデータは開始と終了の時間によって定義されます。
 - Cyclic Update を選択した場合、データは設定可能な間隔で呼び出されます。
- [History Data]:データは図として表示されます。

Alarms and Conditions

SubscribeToEvents 状態のために EventNotifier 属性のあるノードを[Event View]ビューにドラッグして、このビューで現在のイベントとアラームを監視および編集できます。

[Event View]ビューは、3 つの領域に分かれています:

- [Configuration]:リストから目的のイベントフィールドを選択できます。
- [Events"/ "Alarms]:イベントビューとアラームビューを切り替えることで、受信イベントの連続リストと保留中のアラームのステータスを確認できます。
- [Details]:選択したイベントの詳細を表示できます。

下記も参照

サポートされる OPC UA サービスとプロファイル (ページ 472)

8.9.12.3 UaExpert でデータを読み取る方法

必要条件

- UaExpert が WinCC マシンにインストールされていること。
- OPC UA サーバーが WinCC マシンからアクセスできること。

手順

OPC UA クライアント「UaExpert」を介してデータを読み取るには、次の手順に従います。

1. UaExpert を起動し、[Add Server]をクリックします。
2. ポップアップウィンドウで、リスト領域[Custom Discovery]にある[< Double click to Add Server... >]をダブルクリックします。
3. OPC UA サーバーの URL とポート(opc.tcp://[IP-Adresse]:4840)を入力し、[OK]をクリックします。
4. 接続を確立する OPC UA サーバーのエンドポイントを選択します。
5. 認証モードを選択します。
6. [Connect Automatically]チェックボックスをオンにして、[OK]で確定します。
7. [Accept the server certificate temporarily for this session]チェックボックスをオンにしてサーバー証明書を受け入れます。
証明書は、UaExpert の信頼リストに保存されていません。
証明書を信頼リストに永続的に追加するには、[Trust Server Certificate]を選択する必要があります。
8. 次に、[Continue]をクリックします。
WinCC の OPC UA サーバーに接続されます。
9. OPC UA サーバーの[Address Space]で、[Root > Objects] > [ServerInterfaces]から、選択されたタグに移動します。
ここでは、OPC UA サーバーインターフェースのプロジェクトエンジニアリングで設定した OPC UA ノードのみが表示されます。
10. 選択したタグを[Data Access View]領域にドラッグアンドドロップします。
[Data Access View]領域内で、変化するデータを観察したり、データを編集したりできます。

8.10 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お問い合わせください。

WinCC REST 通信

9.1 WinCC の REST インターフェースと MQTT プロトコル

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。

- WinCC は、HTTPS などの柔軟な認証メカニズムを介して安全な REST 通信をサポートします。
- **WinCC REST サービス**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。
- **WinCC IT Connector**を使用すると、ランタイムデータをタグ値やメッセージテキストの形式で外部 REST インターフェースに送信できるようになります。
REST 呼び出しの本文は、外部インターフェースの要件に合わせて適宜設定できます。
セキュリティ設定を使用すると、異なる認証手順を使用できます。
REST インターフェースに加えて、WinCC IT Connector では MQTT プロトコルを使用します。
- ランタイムデータを、REST プロトコルを介して、**WinCC/Cloud Connector** のタグ値の形式でクラウドへ送信できます。
REST 呼び出しは、基本認証と事前定義されている本文構造を使用します。
詳細情報:
 - [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 615)]

注記**WinCC REST サービス:最大接続数**

接続数は、10 秒間隔で 1000 件の接続に制限されます。

制約:サポートされるデータ型

タグにアクセスするとき、次のデータ型はサポートされません:

- 未処理データタグ
- テキスト参照

WinCC と STEP 7 間の REST インターフェースの使用**注記**

STEP 7 プロジェクトで設定された REST 設定は、プロジェクトをロードするときにリセットされます。

[全体リセットによって OS を完了]モードで変換する場合、タグの REST 設定は SIMATIC Manager を介してリセットされます。

タグをコンパイルする前にエクスポートするには、次のオプションがあります。

- タグ管理メニューバーで、[編集]>[エクスポート]を選択し、ファイルを保存します。
- ドライバまたはチャンネルのショートカットメニューで、[エクスポート]オプションを選択し、ファイルを保存します。

REST 設定でタグをインポートするには、プロジェクトをコンパイルした後、タグ管理メニューバーの[編集]>[インポート]オプションを使用します。

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。

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。

言語設定

WinCC REST インターフェースは、メッセージシステムなどの言語依存テキストに対する HTTP 要求の一部として言語設定の指定をサポートします。

言語設定はヘッダー内のキー値のペアとして転送され、次の 2 つのヘッダーが関連します。

- **Content-Language**

[Content-Language]ヘッダーは、URL またはクエリで使用される言語(メッセージクラスの名前やメッセージタイプなど)を指定します。

言語は、ISO 639-1 に従って 2 文字で指定されます(例: [Content-Language: en])。

- **Accept-Language**

[Accept-Language]ヘッダーには、要求されたデータが返される言語が表示されます。複数の言語コードが可能です(例: [Accept-Language: fr-CH, en;q=0.8, de;q=0.7, *;q=0.5])。

WinCC プロジェクトに適切な言語が作成されていない場合、またはヘッダーが指定されていない場合は、[en-US]が使用されます。

URL コーディング

HTTP 呼び出しの URL は、指定された構造を持っています。

URL の構造とファンクションは、式で以下の予約されている文字により識別されます:

! # \$ % & ' () * + , / : ; = ? @ []

9.1 WinCC の REST インターフェースと MQTT プロトコル

これらの予約されている文字が、URL 内で特別な意味を持たされていないが、名前などの一部である場合、これらの文字のエンコーディングを行ってください。

例

構造タグエレメント「MyTag.[NewElement]」のランタイム値を読み取りたいとします。これを実行するには、URL で以下のように文字 "[" を "%5B"、 "]" を "%5D" にエンコードする必要があります。

```
https://<ホスト>:<ポート>/WinCCRestService/tagManagement/Value/MyTag.%5BNewElement%5D
```

詳細情報やコード表については、Wikipedia:URL エンコーディング (https://en.wikipedia.org/wiki/URL_encoding) で参照できます。

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	[WinCC/Connectivity Pack] オプションに対して有効なライセンスが見つかりませんでした。
404	Not Found	リソースが見つかりませんでした。 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。
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。

レスポンスコード	説明	コメント
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にリクエストをもう一度送信してください。
500	InternalServerError	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。

MQTT プロトコル

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。

下記も参照

[REST を経由したクラウドへのデータ転送 \(ページ 615\)](#)

[メソッドの概要 \(ページ 526\)](#)

[方法のフィルタ \(ページ 601\)](#)

[WinCC クラウドコネクタの REST 設定 \(ページ 626\)](#)

[WinCC REST サービスの設定方法 \(ページ 512\)](#)

9.2 WinCC REST サービスの設定方法

[コンピュータ]エディタで、プロジェクトプロパティの 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。

ライセンスニング

WinCC REST サービスを操作するには、REST ライセンスまたは[WinCC/Connectivity Pack] オプションのライセンスが必要です。

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。

[REST API]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

アーカイブシステム:アーカイブタグアクセスの権限

[タグロギング]エディタでは、アーカイブタグごとに値の読み取りを特定の権限に制限できます。

[REST API]グループのタグプロパティの[読み取り権限]フィールドで、適切な権限を選択します。

メッセージシステム:ランタイムメッセージへのアクセス権限

「アラームロギング」エディタで、必要な権限を使用して REST フィルタを設定できます。

REST フィルタ経由でメッセージを要求するには、WinCC ユーザーに REST フィルタで定義されている権限を付与します。

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、ユーザー管理で WinCC ユーザーを作成できます。

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。

手順

1. WinCC Configuration Studio で[コンピュータ]エディタを開きます。
2. ナビゲーションエリアでの[プロジェクト]エントリを選択します。
3. WinCC REST サービスを開始するには、[プロパティ-プロジェクト]エリアの[REST 設定]で[スイッチオン]オプションを有効にします。
4. [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。
5. ナビゲーションエリアで[ローカル設定]エントリを選択します。
6. [プロパティ-ローカル設定]エリアの[REST 設定]で接続データを指定します。
 - ホスト名:WinCC サーバーの名前
 - ポート:アクセスに使用されるポート番号サーバーへのパスが、[サービス URL]フィールドに表示されます。

注記

設定[ホスト名]と[ポート]はローカル Windows レジストリに保存され、プロジェクトがランタイムで実行されているコンピュータで作成される必要があります。

7. セキュア接続をセットアップするには、[証明書]フィールドで[...]ボタンをクリックします。使用可能な証明書が表示されます。
8. 証明書を選択し、[OK]で確定します。

注記

WinCC Certificate Manager (Windows プログラムグループ[Siemens Automation] > [WinCC Certificate Manager])を使用して、システム全体で証明書を作成し、インストールします。

証明書は、後で[コンピュータ]エディタで選択できるようになります。

下記も参照

- WinCC の REST インターフェースと MQTT プロトコル (ページ 507)
- メソッドの概要 (ページ 526)

9.3 WinCC IT コネクタの設定方法

WinCC IT Connector は、REST 呼び出しまたは MQTT トピックを介してランタイムデータを任意のインターフェースに送信します。

WinCC/Cloud Connector とは異なり、呼び出しの本文の設定は変更できます。

基本認証に加えて、他の認証メソッドがサポートされています。

[IT Connector]エディタでは、REST サーバーとそのエンドポイント、および MQTT ブローカとそのトピックを設定できます。

外部 REST インターフェースで認証する場合は、サーバーのサポートされている認証メソッドを設定します。

外部 REST サーバの設定

1. コンピュータのスタートアップリストで、[IT Connector Service]アプリケーションを有効にします。
2. WinCC Configuration Studio で[IT Connector]エディタを開きます。
3. ナビゲーションエリアで[REST サーバー]エントリを右クリックし、ショートカットメニューで[新しい REST サーバー]を選択します。
REST サーバーが作成されます。
4. [プロパティ - REST サーバー]エリアで、外部 REST サーバーの接続データを設定します。
 - アドレス:IP アドレスかコンピュータ名
 - ポート:アクセスに使用されるポート番号
 - パス:BaseURL
 - プロキシ:IT Connector 用の HTTP プロキシ。
プロキシサーバーが設定されていない場合は、ランタイム時に「SYSTEM」ユーザーの Windows プロキシ設定が使用されます。
 - タイムアウト:タイムアウト(秒)
 - セキュリティ:使用される認証方法
[URL]フィールドがサーバーへのパスを表示します。
5. 認証接続をセットアップするには、[セキュリティ]フィールドで外部サーバーの認証方法を選択します。
認証設定のそれぞれのフィールドが有効にされます。
6. 認証方法を設定します。

認証方法の設定

外部 REST インターフェースのさまざまな認証要件を柔軟に設定するために、WinCC IT コネクタは次の認証方法をサポートしています。

認証なし

外部 REST インターフェースは自由に使用でき、認証は必要ありません。認証ヘッダが REST 呼び出しに追加されています。

注記

TLS の使用

REST サーバーが暗号化プロトコルとして TLS を使用する場合、接続を確立できるように、WinCC コンピュータ上の関連付けられたサーバー証明書を信頼する必要があります。

基本認証

外部 REST インターフェースは、ユーザー名とパスワードでの認証をサポートします。

アクセスデータは、外部 REST サーバーで指定されます。

1. [プロパティ - REST サーバー]エリアの[セキュリティ]フィールドで[基本認証]オプションを選択します。
2. [ユーザー名]フィールドと[パスワード]フィールドにアクセスデータを入力してください。
[基本認証]を使用して外部 REST サーバーに対して認証できます。

Bearer トークン

外部 REST インターフェースは、Bearer トークンでの認証(Bearer 認証)をサポートします。

Bearer トークンを取得するには、REST インターフェースプロバイダが提供する指示に従ってください。

1. [プロパティ - REST サーバー]エリアの[セキュリティ]フィールドで[Bearer トークン]オプションを選択します。
2. トークンを[Bearer トークン]フィールドにコピーするか、[...]ボタンをクリックしてトークンをファイルとしてインポートします。

Bearer トークンを使用して外部 REST サーバーで認証できます。

JSON Web トークン

REST インターフェースで JSON Web Token (JWT)を使って認証するには、まずトークンを作成する必要があります:

1. [プロパティ - REST サーバー]エリアの[セキュリティ]フィールドで[JSON Web Token]オプションを選択します。
2. [JSON Web Token]フィールドの[...]ボタンをクリックします。
JWT を設定し、作成するためのウィンドウが表示されます。

9.3 WinCC IT コネクタの設定方法

3. REST インターフェースプロバイダの仕様に従って、JWT の次の設定を指定します。
 - トークンタイプ:認証ヘッダの形式:<トークンタイプ><トークン>
 - 耐用年数:アクセストークンの有効性(秒単位)
 - ペイロード:JWT クレームをキー/値のペアとして説明します。
 - エクスポートされている証明書:アクセストークンを.pfx または.p12 のファイル形式で署名するための証明書
 - パスワード:証明書のパスワード

4. [トークンを作成]をクリックします。
JWT が設定データから作成され、[JWT アクセストークン]フィールドに表示されます。

5. [OK]をクリックし、作成された JWT を適用します。

JWT を使用して外部 REST サーバーで認証できます。ランタイムが開始されるたび、耐用年数の期限が切れた後に、JWT は自動的に再作成されます。

認証コード

[認証コード]メソッドを使用して、REST インターフェースで OAuth2 認証フロー[認証コード]を介してトークンをリクエストします。

プロセスでは、Web ページが表示されます。そこで、ユーザー名とパスワードを使用して認証できます。

準備のために、外部 REST インターフェースで新規ユーザーを作成します。

さらに、クライアント ID、クライアントシークレット、アクセストークンの耐用年数、トークンを転送するための復帰アドレスのある OAuth2 クライアントが必要です。

ユーザーと OAuth2 クライアントを作成するには、REST インターフェースプロバイダが提供する指示に従ってください。

さらに、エンドポイントの仕様、および REST インターフェースの認証フローの適用範囲が必要です。

1. [プロパティ - REST サーバー]エリアの[セキュリティ]フィールドで[認証コード]オプションを選択します。
2. [認証コード]フィールドで、[...]ボタンをクリックします。
認証の詳細を入力するウィンドウが表示されます。

3. REST インターフェースプロバイダの仕様に従って、以下の設定を指定します。
 - **エンドポイント認証**:表示された Web ページにユーザー名とパスワードを入力することにより、認証のエンドポイントを設定します。
 - **エンドポイントトークン**:認証のための REST インターフェースのエンドポイント。必要なアクセストークンを戻します。
 - **リダイレクト URL**:アクセストークンに受け渡される復帰アドレス。「localhost」を含める必要があります。
 - **適用範囲**:REST サービスが認証に求める有効範囲。
 - **クライアントキー**:OAuth2 クライアントのクライアント ID
 - **クライアントシークレット**:OAuth2 クライアントのクライアントシークレット
 - **PKCE あり**:SHA256 ハッシュ関数を使用して追加の安全メカニズムを有効にします。
 - **インプリシットグラント**:トークンエンドポイントを使用せずに認証エンドポイントから直接トークンを戻します。
 - **基本認証**:トークンエンドポイントに対して認証コードの代わりに基本認証を使用します。
 - **Bearer 認証**:アクセストークンが呼び出しではなくヘッダで受け渡されます。
4. [トークンをリクエスト]をクリックします。
標準的な Web ブラウザが、認証のページを開きます。
5. 新しいユーザーのアクセスデータを指定します。
認証エンドポイントは、認証コードや OAuth2 クライアント認証情報をトークンエンドポイントに転送し、トークンをリクエストします。
トークンエンドポイントがアクセストークンを戻します。[アクセストークン]、[リフレッシュトークン]、[トークンタイプ]、[有効期限]のフィールドは、トークンの値を表示します。

注記

すべての REST サービスがリフレッシュトークンを生成するわけではありません。
リフレッシュトークンは、ユーザーのインタラクションなしで、アクセストークンを自動的にリクエストします。

6. [OK]をクリックして、生成されたトークンを受け入れます。

アクセストークンを使用して外部 REST サーバーで認証できます。

新しいアクセストークンは、リフレッシュトークンが REST サービスにより割り付けられている場合のみ、ランタイムで自動的に作成されます。リフレッシュトークンなしの場合、トークンは耐用年数が切れた後にエディタを介して再度リクエストされる必要があります。

クライアント認証情報

[クライアント認証情報]メソッドを使用して、REST インターフェースで OAuth2 認証フロー[クライアント認証情報]を介してトークンをリクエストします。

準備を行うために、クライアント ID、クライアントシークレット、アクセストークンの耐用年数のある OAuth2 クライアントを外部 REST インターフェースで作成します。

9.3 WinCC IT コネクタの設定方法

OAuth2 クライアントを作成するには、REST インターフェースプロバイダが提供する指示に従ってください。

さらに、エンドポイントの仕様のデータ、および REST インターフェースの認証フローの適用範囲が必要です。

1. [プロパティ - REST サーバー]エリアの[セキュリティ]フィールドで[クライアント認証情報]オプションを選択します。
2. [クライアント認証情報]フィールドの[...]ボタンをクリックします。
クライアント認証情報を入力するウィンドウが表示されます。
3. REST インターフェースプロバイダの仕様に従って、以下の設定を指定します。
 - **エンドポイント認証:**認証のための REST インターフェースのエンドポイント。必要なアクセストークンを戻します。
 - **適用範囲:**REST サービスが認証に求める有効範囲。
 - **クライアントキー:**OAuth2 クライアントのクライアント ID
 - **クライアントシークレット:**OAuth2 クライアントのクライアントシークレット
4. [トークンをリクエスト]をクリックします。
認証エンドポイントがアクセストークンを戻します。
アクセストークンの値は、[アクセストークン]、[トークンタイプ]、[有効期限]のフィールドに表示されます。
5. [OK]をクリックし、作成されたアクセストークンを適用します。
アクセストークンを介して外部 REST サーバーで認証できます。
アクセストークンは、耐用年数が切れた後にランタイムで自動的に再作成されます。

REST サーバーのエンドポイントの設定

ベースで外部 REST サーバーを設定した後に、REST インターフェースの個別のエンドポイントとそのメッセージ形式を指定する必要があります。

エンドポイントを作成するには、以下の手順に従います。

1. ナビゲーションエリアで[設定されているサーバー]を右クリックし、ショートカットメニューで[新しいエンドポイント]を選択します。
REST サーバーのエンドポイントが作成されます。
プレースホルダーを使用してエンドポイントに名前を付けることができます。
2. [プロパティ-エンドポイント]エリアで、以下の全般設定を指定します。
 - **モード**:HTTP メソッドのタイプ
 - **トリガ**:REST 呼び出しをトリガするトリガタイプを指定します。
 - **本文**:REST 呼び出しの本文
 - **本文タイプ**:本文のタイプ。JSON ボディの構造とプレースホルダーがエディタを介してモデル化されます。[ユーザー定義]の設定では、本文は[本文]フィールドを介して手動で指定されます。
 - **内容タイプ**:REST 呼び出しのユーザーデータ形式
 - **ステータス**:REST サーバー応答のステータスコードが書き込まれるタグ。
 - **送信先**:REST サーバー応答の本文を編集する方法を示します。
タグ:REST サーバー応答の本文が書き込まれるタグ。
VBS ファンクション:REST サーバー応答の本文を使用して VB スクリプトを呼び出します。
VBS コード:REST サーバー応答の本文を使用して VB コードを実行します。
3. REST 呼び出しのトリガを指定するには、以下の手順に従います。
 - **[タグトリガ]**の場合:[タグトリガ]タブでタグとサイクルを指定します。サイクルがタグなしで定義されている場合、REST 呼び出しは指定の時間間隔で周期的にトリガされます。
 - **[アラームトリガ]**の場合:[アラームトリガ]タブで、REST 呼び出しをトリガするアラームのフィルタを指定します。
4. JSON ボディの値に対してプレースホルダーを作成するには、[プレースホルダー]タブを開きます。

9.3 WinCC IT コネクタの設定方法

5. 以下の設定で、必要なプレースホルダーを作成します。
 - **タイプ:**タグの[値]、[品質]、[タイムスタンプ]、[テキストリスト]、[日付/時刻]、または[カウンタ]。アラームトリガでは、[メッセージテキスト]も選択可能です。
 - **タグ名:**プレースホルダーに使用されるタグ。
 - **メッセージブロック:**[メッセージテキスト]タイプのみ:メッセージテキストが抽出されるアラームトリガのメッセージブロック。
 - **テキストリスト:**[テキストリスト]タイプのみ:タグ値のテキストのあるテキストリストの名前。
 - **ビット番号:**テキストリストからのビットの数。
 - **日付/時刻:**時間基準と時間変位を入力します。時間基準と時間変位は、REST 呼び出しが送信されるタイミングを定義します。
 - **カウンタ:**開始値、終了値、ステップを定義します。
REST サービスを開始するには、カウンタは開始値から始まり、呼び出しごとに徐々に増加します。終了値に達すると、カウンタは開始値にリセットされます。
ランタイム時に設定が保存されるか、冗長サーバーがマスタ/スタンバイに変更されると、カウンタは開始値にリセットされます。
6. JSON ボディの構造を指定するには、[JSON ボディ]タブを開きます。
7. 希望のシーケンスで JSON ボディの個別エレメントを追加します。
[データ型]フィールドで、エレメントに対して、次のいずれかのタイプを指定します。
 - オブジェクト
 - 配列
 - ブール値
 - 数字
 - 文字列
 - Null
8. [値]フィールドのデータ型[ブール値]、[数字]、[文字列]の全エレメントに対して、プレースホルダか統計値を割り付けます。
9. 最後に、[プロパティ-エンドポイント]の[本文]フィールドで、JSON ボディの結果を確認します。
10. クエリを定義するには、[クエリ]タブを開きます。
11. 各クエリのキー、演算子、値を定義します。
クエリはエンドポイントの URL に手動でのみ追加できます。
クエリの値のプレースホルダを指定できます。
12. ヘッダーを定義するには、[ヘッダー]タブを開きます。

13. 各ヘッダーのキーと値を定義します。
ヘッダー値のプレースホルダを指定できます。
14. [IT Connector]エディタの設定を保存します。

注記

変更された設定が保存されると、IT Connector はランタイムで再初期化されます。
タグトリガのディスパッチと初期化フェーズ中のトリガは繰り返されないことに注意してください。

これでエンドポイント設定は完了です。

エンドポイントの REST 呼び出しがトリガにより起動されます。

MQTT ブローカの設定

1. コンピュータのスタートアップリストで、[IT Connector Service]アプリケーションを有効にします。
2. WinCC Configuration Studio で[IT Connector]エディタを開きます。
3. ナビゲーション領域で、[MQTT ブローカ]エントリを右クリックし、ショートカットメニューから[新しい MQTT ブローカ]を選択します。
MQTT ブローカが作成されます。
4. [プロパティ - MQTT ブローカ]領域で、MQTT ブローカのプロパティを設定します。
 - 一般 > ID: MQTT ブローカの一意の ID
 - 一般 > ブローカ: MQTT ブローカのアドレス
 - 一般 > ポート: アクセスに使用されるポート番号
 - 一般 > ステーション: クライアントの一意の名前
 - MQTT > CA 証明書: 認証機関のルート証明書
 - MQTT > クライアント証明書
 - MQTT > クライアントキー
 - MQTT > ユーザー: MQTT ユーザー
 - MQTT > パスワード: MQTT ユーザーのパスワード
 - MQTT > QoS: サービス品質の取得頻度
 - MQTT > 「遺言」 - トピック: 接続が失われた場合にクライアントに転送されるトピック。
 - MQTT > 「遺言」 - ペイロード: 接続が失われた場合に転送されるペイロード。
 - MQTT > ステータス: 応答のステータスコードが書き込まれるタグ。

MQTT ブローカのトピックの設定

MQTT ブローカを設定した後、個々のトピック、そのトリガー、およびペイロードを定義する必要があります。

9.3 WinCC IT コネクタの設定方法

新しいトピックを作成するには、以下のステップに従ってください。

1. ナビゲーションエリアで、設定されている MQTT ブローカを右クリックし、ショートカットメニューで[新しいトピック]を選択します。
MQTT ブローカトピックが作成されます。
テーマに名前を付けるにはプレースホルダを使用できます。
2. [プロパティ-トピック]領域で、以下の一般設定を指定します。
 - **ID:**トピックの一意の ID
 - **トピック:**トピックの名前
 - **トリガ:**トピックをトリガするトリガタイプを指定します。
 - **ペイロード:**リクエストに対して MQTT ブローカに送信されるペイロード。
 - **ペイロードタイプ:**トピックのタイプ。JSON ペイロードの構造とプレースホルダはエディタを介してモデル化されます。[ユーザー定義]設定では、ペイロードは[ペイロード]フィールドを介して手動で指定されます。
 - **ブローカ:**関連する MQTT ブローカのアドレス(読み取り専用)
 - **ID:**関連する MQTT ブローカの一意の ID (読み取り専用)
3. トピックトリガを指定するには、次の手順に従います。
 - **[タグトリガ]**の場合:[タグトリガ]タブでタグとサイクルを指定します。サイクルがタグなしで定義されている場合、トピック呼び出しは指定の時間間隔で周期的にトリガされます。
 - **[アラームトリガ]**の場合:[アラームトリガ]タブで、トピック呼び出しをトリガするアラームのフィルタを指定します。
4. JSON ボディの値に対してプレースホルダーを作成するには、[プレースホルダー]タブを開きます。

5. 以下の設定で、必要なプレースホルダーを作成します。
 - **タイプ:**タグの[値]、[品質]、[タイムスタンプ]、[テキストリスト]、[日付/時刻]、および[カウンタ]。アラームトリガでは、[メッセージテキスト]も選択可能です。
 - **タグ名:**プレースホルダーに使用されるタグ。
 - **メッセージブロック:**[メッセージテキスト]タイプのみ:メッセージテキストが抽出されるアラームトリガのメッセージブロック。
 - **テキストリスト:**[テキストリスト]タイプのみ:タグ値のテキストのあるテキストリストの名前。
 - **ビット番号:**テキストリストからのビットの数。
 - **日付/時刻:**時間基準と時間変位を入力します。時間基準と時間変位は、トピックが送信されるタイミングを定義します。
 - **カウンタ:**開始値、終了値、ステップを定義します。サービスを開始するには、カウンタは開始値から始まり、呼び出しごとに徐々に増加します。終了値に達すると、カウンタは開始値にリセットされます。ランタイム時に設定が保存されるか、冗長サーバーがマスタ/スタンバイに変更されると、カウンタは開始値にリセットされます。
カウンタは負の値を受け入れることができます。

[日付/時刻]および[カウンタ]タイプのプレースホルダは、WinCC ペイロードの外部にあり、ペイロードのテーマの一部になる場合があります。
6. [IT Connector]エディタの設定を保存します。

注記

変更された設定が保存されると、IT Connector はランタイムで再初期化されます。タグトリガのディスパッチと初期化フェーズ中のトリガは繰り返されないことに注意してください。

トピックの設定が完了しました。

エンドポイントのトピック呼び出しがトリガにより起動されます。

下記も参照

WinCC Certificate Manager (ページ 186)

9.4 メッセージシステムで REST フィルタを設定するには

概要

「アラームログ」エディタでは、ランタイムメッセージを読み取るメソッドに使用する REST フィルタを設定します。

必要条件

- WinCC REST サービスが設定され、有効にされています。

サポートされているメッセージブロック

次のメッセージブロックをフィルタとして使用できます。コンテンツ言語のメッセージブロックの名前をクエリとして使用します。

メッセージブロック	説明
日付、時刻	日付または時刻のメッセージブロックの名前。 サポートされている演算子は、「=」、「!=」、「<」、「>」、「<=」、「>=」です。
番号、優先度、すべてのプロセス値ブロック	数値。 サポートされている演算子は、「=」、「!=」、「<」、「>」、「<=」、「>=」です。
コンピュータ名、ユーザー名、すべてのテキストブロック	文字または文字列。 サポートされている演算子は「=」と「!=」です。
ステータス	数値。 サポートされている演算子は、「=」、「!=」、「<」、「>」、「<=」、「>=」です。
クラス	メッセージクラスの名前。 サポートされている演算子は「=」と「!=」です。
タイプ	メッセージタイプの名前。 サポートされている演算子は「=」と「!=」です。

SQL フィルタでは、すべてのフィルタを「AND」で組み合わせることができます。

「OR」または「IN」によるフィルタの組み合わせはサポートされていません。

戻り値をさらにフィルタすることもできます。方法のフィルタ (ページ 601)を参照してください。

手順

1. WinCC Configuration Studio で「アラームロギング」エディタを開きます。
2. ナビゲーションエリアで「REST フィルタ」エントリを選択します。
「フィルタ」エディタが開きます。
3. 「フィルタ」エディタで新しいエントリを作成します。
4. フィルタに名前を付けます。
5. フィルタのアクセス認証を選択します。
6. 選択のフィルタ基準とオペランドを選択します。これには[選択]ダイアログを使用します。

9.5 メソッドの概要

タグ管理メソッド

基本 URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/

エンドポイント

- Connection/<ConnectionName>
接続の設定データの読み取り (ページ 530)
- Connection
すべての接続の設定データの読み取り (ページ 530)
- Group/<GroupName>
タググループの設定データの読み取り (ページ 532)
- Group
すべてのタググループの設定データの読み取り (ページ 532)
- StructureType/<StructureName>
構造タイプの設定データの読み取り (ページ 534)
- StructureTypes
すべての構造タイプの設定データの読み取り (ページ 534)
- StructureTag/<StructureTypeName>
構造タイプのインスタンスの読み取り (ページ 537)
- StructureTags
複数の構造タイプのインスタンスの読み取り (ページ 538)
- Value/<TagName>
タグのランタイム値の読み取り (ページ 539)
値のタグへの書き込み (ページ 540)
- Values
すべてのタグのランタイム値の読み取り (ページ 541)
値の複数タグへの書き込み (ページ 542)
- tag/<TagName>
タグの設定データの読み取り (ページ 542)
- tags
すべてのタグの設定データの読み取り (ページ 543)

システムアーカイブの方法

基本 URL

https://<ホスト>:<ポート>/WinCCRestService/tagLogging/

エンドポイント

- Archives
すべてのプロセス値アーカイブの設定データの読み取り (ページ 545)
- Archive/
 - <ArchiveName>
単一のプロセス値アーカイブの設定データの読み取り (ページ 549)
 - <ArchiveName>/Variable/<TagName>
単一のプロセス値アーカイブタグの設定データの読み取り (ページ 550)
 - <ArchiveName>/Variables
プロセス値アーカイブのすべてのタグの設定データの読み取り (ページ 551)
 - <ArchiveName>/Value/<TagName>
プロセス値アーカイブのタグのランタイム値の読み取り (ページ 554)
 - <ArchiveName>/Values
単一のプロセス値アーカイブの複数タグランタイムデータの読み取り (ページ 555)
異なるプロセス値アーカイブの複数タグのランタイム値を読み取る (ページ 558)
- Timer/<TimerName>
アーカイブシステムの単一時間の設定データの読み取り (ページ 561)
- Timers
アーカイブシステムの全時間の設定データの読み取り (ページ 562)
- Tag/<TagName>
アーカイブシステムタグの設定データの読み取り (ページ 565)
- Tags
全アーカイブシステムタグの設定データの読み取り (ページ 567)

メッセージシステムのメソッド

基本 URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/

9.5 メソッドの概要

エンドポイント

- Message/<MessageNumber>
メッセージの設定データの読み取り (ページ 569)
- Messages
すべてのメッセージの設定データの読み取り (ページ 570)
- MessageClass/<MessageClassName>
メッセージクラスの設定データの読み取り (ページ 572)
- MessageClasses
すべてのメッセージクラスの設定データの読み取り (ページ 574)
- MessageType/<MessageType Name>
メッセージタイプの設定データの読み取り (ページ 576)
- MessageTypes
すべてのメッセージタイプの設定データの読み取り (ページ 577)
- MessageBlock/<MessageBlockName>
メッセージブロックの設定データの読み取り (ページ 580)
- MessageBlocks
すべてのメッセージブロックの設定データの読み取り (ページ 580)
- MessageGroup/<MessageGroupName>
メッセージグループの設定データの読み取り (ページ 582)
- MessageGroups
すべてのメッセージグループの設定データの読み取り (ページ 583)
- LimitValue/<Tagname>
タグ制限の設定データの読み取り (ページ 585)
- LimitValues
すべてのタグ制限の設定データの読み取り (ページ 586)
- RestFilter/<Filtername>
REST フィルタの設定データの読み取り (ページ 592)
- RestFilters
すべての REST フィルタの設定データの読み取り (ページ 593)
- MessageList/<Filtername>
メッセージリストのランタイムメッセージの読み取り (ページ 593)
- ShortTermArchive/ <Filtername>
短期アーカイブからのランタイムメッセージの読み取り (ページ 594)

- LongTermArchive/<Filtername>
長期アーカイブからのランタイムメッセージの読み取り (ページ 595)
- HitList/<Filtername>
ヒットリストからのランタイムメッセージの読み取り (ページ 596)
- LockList/<フィルタ名>
ロックリストからのランタイムメッセージの読み取り (ページ 597)
- Hidelist/<Filtername>
非表示にするメッセージリストからのランタイムメッセージの読み取り (ページ 598)
- HiddenMessageList/<Filtername>
非表示のメッセージリストからのランタイムメッセージの読み取り (ページ 599)

下記も参照

WinCC の REST インターフェースと MQTT プロトコル (ページ 507)

方法のフィルタ (ページ 601)

WinCC REST サービスの設定方法 (ページ 512)

メッセージシステムで REST フィルタを設定するには (ページ 524)

9.6 タグ管理の方法

9.6.1 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"  
}
```

9.6.2 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"}
  ]
}
```

9.6 タグ管理の方法

9.6.3 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"
}
```

9.6.4 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"
    }
  ]
}
```

9.6.5 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/<StructureName>

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" }  
  ] }
```

9.6.6 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。

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"}
      ]
    }
  ]
}
```

9.6.7 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/<StructureTypeName>

9.6 タグ管理の方法

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" }  
  ]  
}
```

9.6.8 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"}
]
```

9.6.9 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

9.6 タグ管理の方法

URL

https://<ホスト>:<ポート>/WinCCRestService/tagManagement/Value/
<VariableName>

本文

なし

戻り値:例

```
{ "variableName": "Tag2",  
  "dataType": 2,  
  "value": "0",  
  "timestamp": "2020-03-20T09:12:38.5830Z",  
  "qualitycode": 76,  
  "errorCode": 0 }
```

9.6.10 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/
<VariableName>

本文

```
{ "value": "テキスト値" }
```

戻り値:例

```
{ "variableName": "Tag11",  
  "errorCode": 0 }
```

9.6.11 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": 76,
    "errorcode": 0,
  },
  {
    "variableName": "Tag2",
    "dataType": 2,
    "value": "0",
    "timestamp": "2020-03-20T09:12:38.5830Z",
    "qualitycode": 76,
    "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": 76,
    "errorcode": 0,
  },
  {
    "variableName": "Tag",
    "error": "見つかりませんでした"
  }
]
```

9.6.12 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

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}
]
```

9.6.13 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/
<VariableName>`**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" }
```

9.6.14 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"}  
  ]
```


9.7 システムアーカイブの方法

9.7.1 利用可能なメソッドのリスト

説明

不完全な呼び出し

ステータスコード 416 「Requested range not satisfiable」は「...tagLogging」のためにサポートされているメソッドを一覧表示します

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/tagLogging

本文

なし

戻り値

説明付きのサポートされているメソッドのリスト(Endpoints)

9.7.2 すべてのプロセス値アーカイブの設定データの読み取り

説明

すべてのプロセス値アーカイブ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/tagLogging/Archives

9.7 システムアーカイブの方法

本文

なし

戻り値:例

```
{
  "archives": [
    {
      "archiveName": "Archive1",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:31.000"
    },
    {
      "archiveName": "Archive2",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:39.000"
    },
    {
      "archiveName": "Archive3",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:44.000"
    },
    {
      "archiveName": "Archive4",
      "archiveType": "process value archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:11:57.000"
    },
    {
      "archiveName": "CompArchive1",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
      "lastChange": "2022-12-12 07:10:50.000"
    },
    {
      "archiveName": "CompArchive2",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled": false,
      "manualInput": true,
    }
  ]
}
```

9.7 システムアーカイブの方法

```
    "lastChange": "2022-12-12 07:11:03.000"
  },
  {
    "archiveName": "CompArchive3",
    "archiveType": "compressed archive",
    "comment": "",
    "disabled": false,
    "manualInput": true,
    "lastChange": "2022-12-12 07:11:03.000"
  },
  {
    "archiveName": "CompArchive4",
    "archiveType": "compressed archive",
    "comment": "",
    "disabled": false,
    "manualInput": true,
    "lastChange": "2022-12-12 07:12:20.000"
  }
]
}
```

代替アドレス指定

URL

https://<ホスト>:<ポート>/WinCCRestService/taglogging/Archives?
itemLimit=3&continuationPoint=2

戻り値:例

```
{
  "continuationPoint":5,
  "moreAvailable":true,
  "archives": [
    {
      "archiveName":"Archive3",
      "archiveType": "process value archive",
      "comment": "",
      "disabled":false,
      "manualInput":true,
      "lastChange":"2022-12-12 07:10:44.000"
    },
    {
      "archiveName":"Archive4",
      "archiveType": "process value archive",
      "comment": "",
      "disabled":false,
      "manualInput":true,
      "lastChange":"2022-12-12 07:11:57.000"
    },
    {
      "archiveName":"CompArchive1",
      "archiveType": "compressed archive",
      "comment": "",
      "disabled":false,
      "manualInput":true,
      "lastChange":"2022-12-12 07:10:50.000"
    }
  ]
}
```

9.7.3 単一のプロセス値アーカイブの設定データの読み取り

説明

単一のプロセス値アーカイブの設定データを読み取ります。

呼び出し

GET / POST

9.7 システムアーカイブの方法

URL

https://<ホスト>:<ポート>/WinCCRestService/tagLogging/Archive/
<ArchiveName>

本文

なし

戻り値:例

```
{  
  "archiveName": "Archive2",  
  "archiveType": "process value archive",  
  "comment": "",  
  "disabled": false,  
  "manualInput": true,  
  "lastChange": "2022-12-12 07:10:39.000"  
}
```

9.7.4 単一のプロセス値アーカイブタグの設定データの読み取り

説明

プロセス値アーカイブの単一の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/tagLogging/Archive/
<ArchiveName>/Variable/<VariableName>

本文

なし

戻り値:例

```
{
  "variableName": "Tag_14",
  "unit": "ha",
  "archiveName": "Archive3",
  "dataType": "analog",
  "comment": "",
  "processVariableName": "Tag_14",
  "disabled": false,
  "manualInput": true,
  "relevantLongTerm": true,
  "acquisitionCycle": "500 ms",
  "acquisitionType": "cyclicContinuous",
  "archivingCycle": "500 ms",
  "archivingFactor": 1,
  "canRead": false,
  "canWrite": false,
  "lastChange": "2022-12-12 09:45:48.000"
}
```

9.7.5 プロセス値アーカイブのすべてのタグの設定データの読み取り

説明

プロセス値アーカイブのすべての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/tagLogging/Archive/
<ArchiveName>/Variables

本文

なし

戻り値:例

```
{
  "variables": [
    {
      "variableName": "Tag_11",
      "unit": "Pound",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_11",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:33.000"
    },
    {
      "variableName": "Tag_15",
      "unit": "lbs",
      "archiveName": "Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_15",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:45:56.000"
    }
  ]
}
```

代替アドレス指定

URL

<https://<ホスト>:<ポート>/WinCCRestService/taglogging/Archive/<ArchiveName>/Variables?itemLimit=2&continuationPoint=12>

戻り値:例

```
{
  "continuationPoint":14,
  "moreAvailable":true,
  "variables": [
    {
      "variableName":"Tag_13",
      "unit": "ar",
      "archiveName":"Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName":"Tag_13",
      "disabled":false,
      "manualInput":true,
      "relevantLongTerm":true,
      "acquisitionCycle":"500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle":"500 ms",
      "archivingFactor":1,
      "canRead":false,
      "canWrite":false,
      "lastChange":"2022-12-12 09:45:44.000"
    },
    {
      "variableName":"Tag_14",
      "unit": "ha",
      "archiveName":"Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName":"Tag_14",
      "disabled":false,
      "manualInput":true,
      "relevantLongTerm":true,
      "acquisitionCycle":"500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle":"500 ms",
      "archivingFactor":1,
      "canRead":false,
      "canWrite":false,
      "lastChange":"2022-12-12 09:45:48.000"
    }
  ]
}
```

9.7 システムアーカイブの方法

9.7.6 プロセス値アーカイブのタグのランタイム値の読み取り

説明

ランタイムにおけるプロセス値アーカイブタグの値を読み取ります。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/TagLogging/Archive/
<ArchiveName>/Value/<VariableName>

本文

なし

戻り値:例

```
{
  "archive": "Archive3",
  "variableName": "Tag_13",
  "unit": "ar",
  "errorcode": 0,
  "values": [
    {
      "value": "79",
      "timestamp": "2022-12-12T13:57:18.5490Z",
      "qualitycode": 128
    },
    {
      "value": "68",
      "timestamp": "2022-12-12T13:57:19.490Z",
      "qualitycode": 128
    },
    {
      "value": "68",
      "timestamp": "2022-12-12T13:57:19.5490Z",
      "qualitycode": 128
    },
    {
      "value": "56",
      "timestamp": "2022-12-12T13:57:20.490Z",
      "qualitycode": 128
    },
    {
      "value": "56",
      "timestamp": "2022-12-12T13:57:20.5490Z",
      "qualitycode": 128
    }
  ]
}
```

9.7.7 単一のプロセス値アーカイブの複数タグランタイムデータの読み取り

説明

ランタイムにおける単一のプロセス値アーカイブの複数タグ値を読み取ります。

呼び出し

GET / POST

9.7 システムアーカイブの方法

URL

https://<ホスト>:<ポート>/WinCCRestService/tagLogging/Values

本文

```
{  
  "variableNames" : [ "Tag1", "Tag2", "Tag10", "Tag" ]  
}
```

戻り値:例

```
[
  {
    "archive": "Archive3",
    "variableName": "Tag_11",
    "unit": "Pound",
    "errorcode": 0,
    "values": [
      {
        "value": "44",
        "timestamp": "2022-12-12T14:01:56.490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      }
    ]
  },
  {
    "archiveName": "Archive3",
    "variableName": "Tag_12",
    "error": "Forbidden"
  },
  {
    "archive": "Archive3",
    "variableName": "Tag_13",
    "unit": "ar",
    "errorcode": 0,
    "values": [
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      },
      {
        "value": "21",
        "timestamp": "2022-12-12T14:01:57.5490Z",

```

9.7 システムアーカイブの方法

```
        "qualitycode":128
      }
    ]
  }
]
```

9.7.8 異なるプロセス値アーカイブの複数タグのランタイム値を読み取る

説明

ランタイムにおける異なるプロセス値アーカイブの複数タグ値を読み取ります。

呼び出し

GET / POST

URL

<https://<Host>:<Port>/WinCCRestService/tagLogging/Values>

本文

```
{
  "archives": [
    {
      "name": "Archive1",
      "variables": [
        {
          "name": "Variable1",
          "timeFrom": "2022-11-25T17:19:18.8980Z",
          "timeTo": "2022-11-25T17:20:18.8980Z",
          "maxValues": 100
        },
        {
          "name": "Variable2",
          "timeFrom": "2022-11-25T17:19:18.8980Z",
          "range": 10,
          "maxValues": 50
        }
      ]
    },
    {
      "name": "Archive2",
      "variables": [
        {
          "name": "Variable1",
          "timeTo": "2022-11-25T17:20:38.8980Z",
          "maxValues": 300
        },
        {
          "name": "Variable2",
          "range": 60,
          "maxValues": 400
        }
      ]
    }
  ]
}
```

戻り値:例

```
[
  {
    "archive": "Archive1",
    "variableName": "Variable1",
    "unit": "Pound",
    "errorcode": 0,
    "values": [
      {
        "value": "44",
        "timestamp": "2022-12-12T14:01:56.490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      }
    ]
  },
  {
    "archiveName": "Archive1",
    "variableName": "Variable2",
    "error": "Forbidden"
  },
  {
    "archive": "Archive2",
    "variableName": "Variable1",
    "unit": "ar",
    "errorcode": 0,
    "values": [
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:56.5490Z",
        "qualitycode": 128
      },
      {
        "value": "32",
        "timestamp": "2022-12-12T14:01:57.490Z",
        "qualitycode": 128
      },
      {
        "value": "21",
        "timestamp": "2022-12-12T14:01:57.5490Z",

```



```
        "qualitycode":128
      }
    ],
  },
  {
    "archiveName":"Archive2",
    "variableName":"Variable2",
    "error":"Forbidden"
  }
]
```

9.7.9 アーカイブシステムの単一時間の設定データの読み取り

説明

アーカイブシステムの単一時間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/tagLogging/Timer/
<TimerName>

本文

なし

戻り値:例

```
{
  "timerName":"3 seconds",
  "type": "cycleTime",
  "base":"1 second",
  "factor":"3",
  "lastChange":"2022-12-12 07:50:45.000"
}
```

9.7 システムアーカイブの方法

9.7.10 アーカイブシステムの全時間の設定データの読み取り

説明

アーカイブシステムの全時間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/tagLogging/Timers

本文

なし

戻り値:例

```
{
  "timers": [
    {
      "timerName": "1 day",
      "type": "cycleTime",
      "base": "1 day",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    },
    {
      "timerName": "1 hour",
      "type": "cycleTime",
      "base": "1 hour",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.480"
    },
    {
      "timerName": "1 minute",
      "type": "cycleTime",
      "base": "1 minute",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.477"
    },
    {
      "timerName": "1 second",
      "type": "cycleTime",
      "base": "1 second",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.473"
    },
    {
      "timerName": "3 seconds",
      "type": "cycleTime",
      "base": "1 second",
      "factor": "3",
      "lastChange": "2022-12-12 07:50:45.000"
    },
    {
      "timerName": "500 ms",
      "type": "cycleTime",
      "base": "500 ms",
      "factor": 1,
      "lastChange": "2022-12-12 07:08:42.470"
    },
    {
      "timerName": "TimeSeries Daily",
      "type": "timeSeries",
      "base": "daily",

```

9.7 システムアーカイブの方法

```
        "factor":1,
        "lastChange":"2022-12-12 07:51:14.000"
    },
    {
        "timerName":"TimeSeries Monthly",
        "type": "timeSeries",
        "base": "monthly",
        "factor":1,
        "lastChange":"2022-12-12 07:51:38.000"
    },
    {
        "timerName":"TimeSeries Weekly",
        "type": "timeSeries",
        "base": "weekly",
        "factor":1,
        "lastChange":"2022-12-12 07:51:26.000"
    },
    {
        "timerName":"TimeSeries Yearly",
        "type": "timeSeries",
        "base": "yearly",
        "factor":1,
        "lastChange":"2022-12-12 07:51:55.000"
    }
]
}
```

代替アドレス指定

URL

[https://<ホスト>:<ポート>/WinCCRestService/taglogging/Timers?
itemLimit=2&continuationPoint=0](https://<ホスト>:<ポート>/WinCCRestService/taglogging/Timers?itemLimit=2&continuationPoint=0)

戻り値:例

```
{
  "continuationPoint":2,
  "moreAvailable":true,
  "timers": [
    {
      "timerName":"1 day",
      "type": "cycleTime",
      "base":"1 day",
      "factor":1,
      "lastChange":"2022-12-12 07:08:42.480"
    },
    {
      "timerName":"1 hour",
      "type": "cycleTime",
      "base":"1 hour",
      "factor":1,
      "lastChange":"2022-12-12 07:08:42.480"
    }
  ]
}
```

9.7.11 アーカイブシステムタグの設定データの読み取り

説明

アーカイブシステムの単一タグの設定データを読み取ります。

タグ名は、個別アーカイブ内でのみ一意です。複数のタグを戻すことができます。

呼び出し

GET / POST

URL

https://<ホスト>:<ポート>/WinCCRestService/tagLogging/Variable/
<VariableName>

本文

なし

戻り値:例

```
{
  "variables": [
    {
      "variableName": "Tag_4",
      "unit": "1",
      "archiveName": "Archive1",
      "dataType": "analog",
      "comment": "",
      "processVariableName": "Tag_4",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "500 ms",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 09:44:14.000"
    },
    {
      "variableName": "Tag_4",
      "unit": "",
      "archiveName": "CompArchive1",
      "dataType": "compress",
      "comment": "",
      "processVariableName": "",
      "disabled": false,
      "manualInput": true,
      "relevantLongTerm": true,
      "acquisitionCycle": "",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle": "",
      "archivingFactor": 1,
      "canRead": false,
      "canWrite": false,
      "lastChange": "2022-12-12 07:12:32.000"
    }
  ]
}
```

9.7.12 全アーカイブシステムタグの設定データの読み取り

説明

アーカイブシステムの全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/tagLogging/Variables

本文

なし

戻り値:ページングのある例

URL

https://<ホスト>:<ポート>/WinCCRestService/taglogging/Variables?
itemLimit=2&continuationPoint=12

9.7 システムアーカイブの方法

```
{
  "continuationPoint":14,
  "moreAvailable":true,
  "variables": [
    {
      "variableName":"Tag_13",
      "unit": "ar",
      "archiveName":"Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName":"Tag_13",
      "disabled":false,
      "manualInput":true,
      "relevantLongTerm":true,
      "acquisitionCycle":"500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle":"500 ms",
      "archivingFactor":1,
      "canRead":false,
      "canWrite":false,
      "lastChange":"2022-12-12 09:45:44.000"
    },
    {
      "variableName":"Tag_14",
      "unit": "ha",
      "archiveName":"Archive3",
      "dataType": "analog",
      "comment": "",
      "processVariableName":"Tag_14",
      "disabled":false,
      "manualInput":true,
      "relevantLongTerm":true,
      "acquisitionCycle":"500 ms",
      "acquisitionType": "cyclicContinuous",
      "archivingCycle":"500 ms",
      "archivingFactor":1,
      "canRead":false,
      "canWrite":false,
      "lastChange":"2022-12-12 09:45:48.000"
    }
  ]
}
```


9.8 メッセージシステムのメソッド

9.8.1 メッセージの設定データの読み取り

説明

メッセージの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/Message/
<MessageNumber>

本文

なし

戻り値:例

```
{
  "messageNumber":6
  "MessageClass"Error",
  "messageType":"Alarm",
  "messageGroup":"Group",
  "priority":0,
  "messageTag":"MessageTag",
  "messageBit":0,
  "statusTag":"StatusTag",
  "statusBit":0,
  "acknowledgmentTag":"QuitTag",
  "acknowledgmentBit":0,
  "singleAcknowledgment":false,
  "centralSignalingDevice":false,
  "archived":true,
  "hasHelp":false
}
```

9.8.2 すべてのメッセージの設定データの読み取り

説明

すべてのメッセージの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/Messages

本文

なし

戻り値:例

```
{
  "messages":[
    {
      "messageNumber":1,
      "MessageClass"Error",
      "messageType":Alarm",
      "messageGroup": "",
      "priority":0,
      "messageTag": "",
      "messageBit":0,
      "statusTag": "",
      "statusBit":0,
      "acknowledgmentTag": "",
      "acknowledgmentBit":0,
      "singleAcknowledgment":false,
      "centralSignalingDevice":false,
      "archived":true,
      "hasHelp":false
    },
    ...
  ]
}
```

戻り値:ペー징のある例

URL

```
https://<Host>:<Port>/WinCCRestService/alarmLogging/Messages?  
itemLimit=2&continuationPoint=6
```

```
{  
  "continuationPoint":8,  
  "moreAvailable":true,  
  "messages":[  
    {  
      "messageNumber":7,  
      "MessageClass":"System, requires acknowledgment",  
      "messageType":"Process control system",  
      "messageGroup":"Group_2",  
      "priority":0,  
      "messageTag":"MessageTag",  
      "messageBit":1,  
      "statusTag":"StatusTag",  
      "statusBit":1,  
      "acknowledgmentTag":"QuitTag",  
      "acknowledgmentBit":1,  
      "singleAcknowledgment":false,  
      "centralSignalingDevice":false,  
      "archived":true,  
      "hasHelp":false  
    },  
    {  
      "messageNumber":8,  
      "MessageClass":"Error",  
      "messageType":"Warning",  
      "messageGroup":"Group_3",  
      "priority":0,  
      "messageTag":"MessageTag",  
      "messageBit":2,  
      "statusTag":"StatusTag",  
      "statusBit":2,  
      "acknowledgmentTag":"QuitTag",  
      "acknowledgmentBit":2,  
      "singleAcknowledgment":false,  
      "centralSignalingDevice":false,  
      "archived":true,  
      "hasHelp":false  
    }  
  ]  
}
```

9.8 メッセージシステムのメソッド

9.8.3 メッセージクラスの設定データの読み取り

説明

メッセージクラスの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageClass/
<MessageClassName>

本文

なし

戻り値:例

```
{
  "classId":1,
  "className":"Error",
  "messageTypes":[
    {
      "typeId":1,
      "typeName":"Alarm",
      "acknowledgmentCameIn":true,
      "acknowledgmentWentOut":false,
      "flashOn":false,
      "onlyForInitialValue":false,
      "withoutWentOut":false,
      "uniqueUser":false,
      "cameInComment":false,
      "commentRequired":false,
      "signatureRequired":false,
      "textCameIn": "+",
      "textWentOut": "-",
      "textAcknowledged": "*",
      "textCameInAndWentOut": "+/-",
      "fontColorCameIn":255,
      "backgroudColorCameIn":12632256,
      "fontColorWentOut":255,
      "backgroudColorWentOut":12632256,
      "fontColorAcknowledged":255,
      "backgroudAcknowledged":12632256
    },
    {
      "typeId":2,
      "typeName":"Warning",
      "acknowledgmentCameIn":true,
      "acknowledgmentWentOut":false,
      "flashOn":false,
      "onlyForInitialValue":false,
      "withoutWentOut":false,
      "uniqueUser":false,
      "cameInComment":false,
      "commentRequired":false,
      "signatureRequired":false,
      "textCameIn": "+",
      "textWentOut": "-",
      "textAcknowledged": "*",
      "textCameInAndWentOut": "+/-",
      "fontColorCameIn":16711680,
      "backgroudColorCameIn":12632256,
      "fontColorWentOut":16711680,
      "backgroudColorWentOut":12632256,
      "fontColorAcknowledged":16711680,
    }
  ]
}
```

9.8 メッセージシステムのメソッド

```
        "backgroudAcknowledged":12632256
    },
    {
        "typeId":3,
        "typeName":"Failure",
        "acknowledgmentCameIn":true,
        "acknowledgmentWentOut":false,
        "flashOn":false,
        "onlyForInitialValue":false,
        "withoutWentOut":false,
        "uniqueUser":false,
        "cameInComment":false,
        "commentRequired":false,
        "signatureRequired":false,
        "textCameIn": "+",
        "textWentOut": "-",
        "textAcknowledged": "*",
        "textCameInAndWentOut": "+/-",
        "fontColorCameIn":65535,
        "backgroudColorCameIn":12632256,
        "fontColorWentOut":65535,
        "backgroudColorWentOut":12632256,
        "fontColorAcknowledged":65535,
        "backgroudAcknowledged":12632256
    }
]
}
```

9.8.4 すべてのメッセージクラスの設定データの読み取り

説明

すべてのメッセージクラスの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/
MessageClasses

本文

なし

戻り値:例

```
{
  "messageClasses": [
    {
      "classId":1,
      "className":"Error",
      "messagesTypes":[...]
    },
    {
      "classId":17,
      "className":"System, requires acknowledgment",
      "messageTypes":[...]
    },
    {
      "classId":18,
      "className":"System, without acknowledgment",
      "messageTypes":[...]
    }
  ]
}
```

戻り値:ページングのある例

URL

```
https://<Host>:<Port>/WinCCRestService/alarmLogging/
MessageClasses?itemLimit=2&continuationPoint=3
```

```
{
  "continuationPoint":18,
  "moreAvailable":false,
  "messageClasses": [
    {
      "classId":17,
      ...
    },
  ]
}
```

9.8.5 メッセージタイプの設定データの読み取り

説明

メッセージタイプの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageType/
<MessageType>

本文

なし

戻り値:例

```
{
  "typeId":3,
  "typeName":"Failure",
  "acknowledgmentCameIn":true,
  "acknowledgmentWentOut":false,
  "flashOn":false,
  "onlyForInitialValue":false,
  "withoutWentOut":false,
  "uniqueUser":false,
  "cameInComment":false,
  "commentRequired":false,
  "signatureRequired":false,
  "textCameIn": "+",
  "textWentOut": "-",
  "textAcknowledged": "*",
  "textCameInAndWentOut": "+/-",
  "fontColorCameIn":65535,
  "backgroudColorCameIn":12632256,
  "fontColorWentOut":65535,
  "backgroudColorWentOut":12632256,
  "fontColorAcknowledged":65535,
  "backgroudAcknowledged":12632256
}
```


9.8.6 すべてのメッセージタイプの設定データの読み取り

説明

すべてのメッセージタイプの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageTypes

本文

なし

戻り値:例

```
{
  "messageTypes": [
    {
      "typeId": 1,
      "typeName": "Alarm",
      "acknowledgmentCameIn": true,
      "acknowledgmentWentOut": false,
      "flashOn": false,
      "onlyForInitialValue": false,
      "withoutWentOut": false,
      "uniqueUser": false,
      "cameInComment": false,
      "commentRequired": false,
      "signatureRequired": false,
      "textCameIn": "+",
      "textWentOut": "-",
      "textAcknowledged": "*",
      "textCameInAndWentOut": "+/-",
      "fontColorCameIn": 255,
      "backgroudColorCameIn": 12632256,
      "fontColorWentOut": 255,
      "backgroudColorWentOut": 12632256,
      "fontColorAcknowledged": 255,
      "backgroudAcknowledged": 12632256
    },
    {
      "typeId": 2,
      ...
    },
    {
      ...
    }
  ]
}
```

戻り値:ペー징ングのある例

URL

```
https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageTypes?
itemLimit=2&continuationPoint=257
```

```
{
  "continuationPoint":273,
  "moreAvailable":true,
  "messageTypes":[
    {
      "typeId":258,
      "typeName":"System messages",
      "acknowledgmentCameIn":true,
      "acknowledgmentWentOut":false,
      "flashOn":false,
      "onlyForInitialValue":false,
      "withoutWentOut":true,
      "uniqueUser":false,
      "cameInComment":false,
      "commentRequired":false,
      "signatureRequired":false,
      "textCameIn": "+",
      "textWentOut": "-",
      "textAcknowledged": "*",
      "textCameInAndWentOut": "+/-",
      "fontColorCameIn":0,
      "backgroudColorCameIn":12632256,
      "fontColorWentOut":0,
      "backgroudColorWentOut":12632256,
      "fontColorAcknowledged":0,
      "backgroudAcknowledged":12632256
    },
    {
      "typeId":273,
      "typeName":"Process control system",
      "acknowledgmentCameIn":false,
      "acknowledgmentWentOut":false,
      "flashOn":false,
      "onlyForInitialValue":false,
      "withoutWentOut":true,
      "uniqueUser":false,
      "cameInComment":false,
      "commentRequired":false,
      "signatureRequired":false,
      "textCameIn": "+",
      "textWentOut": "-",
      "textAcknowledged": "*",
      "textCameInAndWentOut": "+/-",
      "fontColorCameIn":0,
      "backgroudColorCameIn":12632256,
      "fontColorWentOut":0,
      "backgroudColorWentOut":12632256,
      "fontColorAcknowledged":0,
      "backgroudAcknowledged":12632256
    }
  ]
}
```

9.8 メッセージシステムのメソッド

```
    }  
  ]  
}
```

9.8.7 メッセージブロックの設定データの読み取り

説明

メッセージブロックの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageBlock/
<MessageBlockName>

本文

なし

戻り値:例

```
{  
  "blockNumber":17,  
  "blockName":"Loop Alarm",  
  "used":false,  
  "blockType": "systemBlocks"  
}
```

9.8.8 すべてのメッセージブロックの設定データの読み取り

説明

すべてのメッセージブロックの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageBlocks

本文

なし

戻り値:例

```
{
  "messageBlocks": [
    {
      "blockNumber": 1,
      "blockName": "Date",
      "used": true,
      "blockType": "systemBlocks"
    },
    {
      "blockNumber": 2,
      "blockName": "Time",
      "used": true,
      "blockType": "systemBlocks"
    },
    {
      "blockNumber": 1,
      "blockName": "Message text",
      "used": true,
      "blockType": "userTextBlocks"
    },
    {
      "blockNumber": 1,
      "blockName": "Process value:1",
      "used": false,
      "blockType": "processValueBlocks"
    },
    {
      "blockNumber": 2,
      "blockName": "Process value:2",
      "used": false,
      "blockType": "processValueBlocks"
    }
  ]
}
```

9.8 メッセージシステムのメソッド

戻り値:ページングのある例

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageTypes?
itemLimit=2&continuationPoint=3

```
{
  "continuationPoint":65538,
  "moreAvailable":true,
  "messageBlocks":[
    {
      "blockNumber":1,
      "blockName":"Date",
      "used":true,
      "blockType": "systemBlocks"
    },
    {
      "blockNumber":2,
      "blockName":"Time",
      "used":true,
      "blockType": "systemBlocks"
    }
  ]
}
```

9.8.9 メッセージグループの設定データの読み取り

説明

すべてのメッセージグループの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageGroup/
<MessageGroupName>

本文

なし

戻り値:例

```
{
  "groupName": "Group_7",
  "parent": "Group_3",
  "statusTag": "GroupStatusTag",
  "statusBit": 6
  "lockTag": "GroupMessageTag",
  "lockBit": 6
  "acknowledgmentTag": "GroupQuitTag",
  "acknowledgmentBit": 6
  "hideTag": "HideTag1"
}
```

9.8.10 すべてのメッセージグループの設定データの読み取り

説明

すべてのメッセージグループの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageGroups

本文

なし

戻り値:例

```
{
  "messageGroups": [
    {
      "groupName": "Alarm",
      "parent": "Error",
      "statusTag": "",
      "statusBit": 0,
      "lockTag": "",
      "lockBit": 0,
      "acknowledgmentTag": "",
      "acknowledgmentBit": 0,
      "hideTag": ""
    },
    {
      "groupName": "Group_2",
      "parent": "Group_1",
      "statusTag": "GroupStatusTag",
      "statusBit": 1,
      "lockTag": "GroupMessageTag",
      "lockBit": 1,
      "acknowledgmentTag": "GroupQuitTag",
      "acknowledgmentBit": 1,
      "hideTag": "HideTag1"
    }
  ]
}
```


戻り値:ページングのある例

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageTypes?

itemLimit=2&continuationPoint=3

```
{
  "continuationPoint":317,
  "moreAvailable":true,
  "messageGroups":[
    {
      "groupName":"Group_10",
      "parent":"Group_5",
      "statusTag":"GroupStatusTag",
      "statusBit":9,
      "lockTag":"GroupMessageTag",
      "lockBit":9,
      "acknowledgmentTag":"GroupQuitTag",
      "acknowledgmentBit":9,
      "hideTag":"HideTag1"
    },
    {
      "groupName":"Group_11",
      "parent":"Group_5",
      "statusTag":"GroupStatusTag",
      "statusBit":10,
      "lockTag":"GroupMessageTag",
      "lockBit":10,
      "acknowledgmentTag":"GroupQuitTag",
      "acknowledgmentBit":10,
      "hideTag":"HideTag1"
    }
  ]
}
```

9.8.11 タグ制限の設定データの読み取り

説明

タグの制限の設定データを読み取ります。

呼び出し

GET / POST

9.8 メッセージシステムのメソッド

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/LimitValue/
<Tagname>

本文

なし

戻り値:例

```
{
  "limitTag": "LimitTag",
  "messageNumber": 400,
  "sharedMessage": true,
  "limits": [
    {
      "comparison": "highLimit",
      "messageNumber": 1,
      "indirect": true,
      "comparsionValue": 0,
      "comparsionValueTag": "LimitCompare"
    },
    {
      "comparison": "lowLimit",
      "messageNumber": 2,
      "indirect": false,
      "comparsionValue": 1,
      "comparsionValueTag": ""
    }
  ]
}
```

9.8.12 すべてのタグ制限の設定データの読み取り

説明

全タグの制限の設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/LimitValues

本文

なし

戻り値:例

```
{
  "limitValues": [
    {
      "limitTag": "LimitTag",
      "messageNumber": 400,
      "sharedMessage": true,
      "limits": [
        {
          "comparison": "highLimit",
          "messageNumber": 1,
          "indirect": true,
          "comparsionValue": 0,
          "comparsionValueTag": "LimitCompare"
        },
        {
          "comparison": "lowLimit",
          "messageNumber": 2,
          "indirect": false,
          "comparsionValue": 1,
          "comparsionValueTag": ""
        }
      ]
    },
    {
      "limitTag": "LimitTag_1",
      "messageNumber": 0,
      "sharedMessage": false,
      "limits": [
        {
          "comparison": "highLimit",
          "messageNumber": 300,
          "indirect": false,
          "comparsionValue": 0,
          "comparsionValueTag": ""
        },
        {
          "comparison": "lowLimit",
          "messageNumber": 301,
          "indirect": false,
          "comparsionValue": 1,
          "comparsionValueTag": ""
        }
      ]
    },
    {
      "limitTag": "LimitTag_2",
      "messageNumber": 0,
      "sharedMessage": false,
```

```
"limits":[
  {
    "comparison": "valueIsSame",
    "messageNumber":304,
    "indirect":false,
    "comparsionValue":0,
    "comparsionValueTag": ""
  },
  {
    "comparison": "highLimit",
    "messageNumber":305,
    "indirect":false,
    "comparsionValue":1,
    "comparsionValueTag": ""
  }
]
},
{
  "limitTag":"LimitTag_3",
  "messageNumber":0,
  "sharedMessage":false,
  "limits":[
    {
      "comparison": "highLimit",
      "messageNumber":308,
      "indirect":false,
      "comparsionValue":0,
      "comparsionValueTag": ""
    },
    {
      "comparison": "valueIsDifferent",
      "messageNumber":309,
      "indirect":false,
      "comparsionValue":1,
      "comparsionValueTag": ""
    }
  ]
},
{
  "limitTag":"LimitTag_4",
  "messageNumber":0,
  "sharedMessage":false,
  "limits":[
    {
      "comparison": "highLimit",
      "messageNumber":312,
      "indirect":false,
      "comparsionValue":0,
      "comparsionValueTag": ""
    },
    {
```

9.8 メッセージシステムのメソッド

```
        "comparison": "lowLimit",
        "messageNumber": 313,
        "indirect": false,
        "comparsionValue": 1,
        "comparsionValueTag": ""
    },
    {
        "comparison": "valueIsSame",
        "messageNumber": 314,
        "indirect": false,
        "comparsionValue": 2,
        "comparsionValueTag": ""
    }
]
}
```

戻り値: ページングのある例

URL

```
https://<Host>:<Port>/WinCCRestService/alarmLogging/  
MessageClasses?itemLimit=2&continuationPoint=3
```

```
{
  "continuationPoint":5,
  "moreAvailable":false,
  "limitValues":[
    {
      "limitTag":"LimitTag_3",
      "messageNumber":0,
      "sharedMessage":false,
      "limits":[
        {
          "comparison": "highLimit",
          "messageNumber":308,
          "indirect":false,
          "comparsionValue":0,
          "comparsionValueTag": ""
        },
        {
          "comparison": "valueIsDifferent",
          "messageNumber":309,
          "indirect":false,
          "comparsionValue":1,
          "comparsionValueTag": ""
        }
      ]
    },
    {
      "limitTag":"LimitTag_4",
      "messageNumber":0,
      "sharedMessage":false,
      "limits":[
        {
          "comparison": "highLimit",
          "messageNumber":312,
          "indirect":false,
          "comparsionValue":0,
          "comparsionValueTag": ""
        },
        {
          "comparison": "lowLimit",
          "messageNumber":313,
          "indirect":false,
          "comparsionValue":1,
          "comparsionValueTag": ""
        },
        {
          "comparison": "valueIsSame",
          "messageNumber":314,
          "indirect":false,
          "comparsionValue":2,

```

9.8 メッセージシステムのメソッド

```
        "comparsionValueTag": ""
    }
  ]
}
]
```

9.8.13 REST フィルタの設定データの読み取り

説明

REST フィルタの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/RestFilter/
<Filtername>

本文

なし

戻り値:例

```
{
  "filterName":"Filter1",
  "authorization":"Process controlling",
  "selection":"MSGNR > 1"
}
```


9.8.14 すべての REST フィルタの設定データの読み取り

説明

すべての REST フィルタの設定データを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/RestFilters

本文

なし

戻り値:例

```
{
  "restFilters":[
    {
      "filterName":"Filter1",
      "authorization":"Process controlling",
      "selection":"MSGNR > 1"
    },
    {
      "filterName":"Filter2",
      "authorization":"Process controlling",
      "selection":"MSGNR = 3"
    }
  ]
}
```

9.8.15 メッセージリストのランタイムメッセージの読み取り

説明

メッセージウィンドウのメッセージリストに相当するランタイムメッセージを読み取ります。

9.8 メッセージシステムのメソッド

呼び出し

GET / POST

URL

`https://<Host>:<Port>/WinCCRestService/alarmLogging/MessageList/
<Filtername>`

本文

なし

戻り値:例

```
{  
  "messages": [  
    {  
      "messageNumber":733,  
      "timestamp":"2023-07-24T13:59:27.1410Z",  
      "Class":"Class5",  
      "Type":"Art_5_4",  
      "Foreground color":0,  
      "backcolor":16777215,  
      "flashcolor":16777215,  
      "Message text":"AlarmTag24",  
      ...  
    },  
    ...  
  ]  
}
```

9.8.16 短期アーカイブからのランタイムメッセージの読み取り

説明

メッセージウィンドウの短期アーカイブリストに相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/
ShortTermArchive/<Filtername>

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber": 64,
      "timestamp": "2023-07-24T13:44:29.6440Z",
      "Class": "Class1",
      "Type": "Art_1_3",
      "Foreground color": 0,
      "backcolor": 16777215,
      "flashcolor": 16777215,
      "Message text": "AlarmTag3",
      ...
    },
    ...
  ]
}
```

9.8.17 長期アーカイブからのランタイムメッセージの読み取り**説明**

メッセージウィンドウの長期アーカイブリストに相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/
LongTermArchive/<Filtername>

9.8 メッセージシステムのメソッド

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber": 509,
      "timestamp": "2023-07-24T13:45:37.5150Z",
      "Class": "Class4",
      "Type": "Art_4_2",
      "Foreground color": 0,
      "backcolor": 16777215,
      "flashcolor": 16777215,
      "Message text": "AlarmTag17",
      ...
    },
    ...
  ]
}
```

9.8.18 ヒットリストからのランタイムメッセージの読み取り

説明

メッセージウィンドウのヒットリストに相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/HitList/
<Filtername>

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber": 224,
      "timestamp": "2023-07-24T13:45:32.540Z",
      "frequency": 5,
      "sum +/-": 5,
      "average +/-": 1.25,
      "sum +/*1": 0,
      "average +/*1": 0,
      "sum +/*2": 0,
      "average +/*2": 0,
      "sum +/+": 10,
      "average +/+": 2.5
    },
    {
      "messageNumber": 256,
      "timestamp": "2023-07-24T13:45:30.9530Z",
      "frequency": 4,
      "sum +/-": 5,
      "average +/-": 1.6666666666666667,
      "sum +/*1": 0,
      "average +/*1": 0,
      "sum +/*2": 0,
      "average +/*2": 0,
      "sum +/+": 8,
      "average +/+": 2.6666666666666667
    },
    ...
  ]
}
```

9.8.19 ロックリストからのランタイムメッセージの読み取り

説明

メッセージウィンドウのロックリストに相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

9.8 メッセージシステムのメソッド

URL

https://<ホスト>:<ポート>/WinCCRestService/alarmLogging/LockList/<フィルタ名>

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber":1434,
      "timestamp":"2023-07-24T08:04:16.1120z",
      "Class":"Class10",
      "Type":"Art_10_1",
      "Message text":"AlarmTag46",
      "Point of error":"0",
      ...
    },
    ...
  ]
}
```

9.8.20 非表示にするメッセージリストからのランタイムメッセージの読み取り

説明

メッセージウィンドウの「非表示にするメッセージのリスト」に相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/Hidelist/
<Filtername>

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber": 156,
      "timestamp": "2023-07-24T08:04:16.1120Z",
      "Class": "Class2",
      "Type": "Art_2_1",
      "Message text": "AlarmTag6",
      "Point of error": "0",
      ...
    },
    ...
  ]
}
```

9.8.21 非表示のメッセージリストからのランタイムメッセージの読み取り

説明

メッセージウィンドウのメッセージリスト内の非表示メッセージに相当するランタイムメッセージを読み取ります。

呼び出し

GET / POST

URL

https://<Host>:<Port>/WinCCRestService/alarmLogging/
HiddenMessageList/<Filtername>

本文

なし

戻り値:例

```
{
  "messages": [
    {
      "messageNumber": 227,
      "timestamp": "2023-07-24T15:28:12.4280Z",
      "Class": "Class2",
      "Type": "Art_2_3",
      "Foreground color": 0,
      "backcolor": 16777215,
      "flashcolor": 16777215,
      "Message text": "AlarmTag8",
      "Point of error": "7",
      ...
    },
    ...
  ]
}
```


9.9 方法のフィルタ

フィルタを使用して戻り値を制限することができます。

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「?」や「*」を使用できます。

9.9 方法のフィルタ

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。
archiveName	プロセス値のアーカイブ	検索済みの文字列を含んでいるすべてのプロセス値のアーカイブを戻します。 プレースホルダ「?」や「*」を使用できます。
begin	プロセス値のアーカイブ	プロセス値のアーカイブからのタグ値が戻される時間範囲の開始部分。 日付フォーマット:YYYY-MM-DD hh:mm:ss:ms 例:2015-12-31 16:30:00:000
end	プロセス値のアーカイブ	プロセス値のアーカイブからのタグ値が戻される時間範囲の終了部分。 日付フォーマット:YYYY-MM-DD hh:mm:ss:ms 例:2015-12-31 16:30:00:000
range	プロセス値のアーカイブ	プロセス値のアーカイブからのタグ値が戻される秒単位の時間範囲。 [開始部分]と[終了部分]のパラメータへのリンクなしに使用された場合、プロセス値のアーカイブの最終秒が時間スパンとして設定されます。
maxValues	プロセス値のアーカイブ	戻されるタグ値の最大数。 他に別途指定がない限り最大で 1000 のタグ値。
Name	メッセージシステム	検索する名前を含むメッセージクラス、メッセージタイプ、メッセージブロック、メッセージグループ、および制限のすべてのエントリを返します。 プレースホルダ「?」や「*」を使用できます。

パラメータ	オブジェクト	説明
MessageClass	メッセージシステム	検索されたメッセージクラスのあるすべてのメッセージを返します。
MessageType	メッセージシステム	検索されたメッセージタイプのあるすべてのメッセージを返します。
MessageGroup	メッセージシステム	検索されたメッセージグループのあるすべてのメッセージ/グループを返します。
MessageTag	メッセージシステム	検索されたメッセージタグのあるすべてのメッセージを返します。
StatusTag	メッセージシステム	検索されたステータスタグのあるすべてのメッセージ/グループを返します。
AcknowledgmentTag	メッセージシステム	検索された確認タグのあるすべてのメッセージ/グループを返します。
GroupLockTag	メッセージシステム	検索されたロックタグのあるすべてのメッセージ/グループを返します。
GroupHideTag	メッセージシステム	検索された非表示タグのあるすべてのメッセージ/グループを返します。
maxValues	メッセージシステム	短期アーカイブ、長期アーカイブ、ヒットリストに返されるメッセージの最大数。 別途指定がない限り、最大 1,000 件のメッセージ。
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。メッセージシステムのメソッドには適用されません。 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。

9.9 方法のフィルタ

パラメータ	オブジェクト	説明
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

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_*

例: begin/end/range

プロセス値のアーカイブからのタグ値が戻される時間範囲を定義するには、パラメータ [begin]、[end]、[range]を使用します。

開始部分と終了部分の間のタグ値をリクエストしたい場合:

- /Values?begin=2015-11-26 10:30:02.000&end=2015-12-01 10:30:02.000

開始時間後に時間でタグ値をリクエストしたい場合:

- /Values?begin=2015-11-26 10:30:02.000&range=3600

9.9 方法のフィルタ

終了時間前に時間でタグ値をリクエストしたい場合:

- /Values?end=2015-12-01 10:30:02.000&range=3600

下記も参照

WinCC の REST インターフェースと MQTT プロトコル (ページ 507)

メソッドの概要 (ページ 526)

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
- Siemens MindSphere - MindConnect EU1
- Amazon: AWS
- Microsoft: Azure
- Alibaba Cloud の Siemens MindSphere(中国)
- 汎用 MQTT
- REST プロトコル

Cloud Connector の動作方法

Cloud Connector 経由で WinCC サーバーからクラウドにデータを送信します。

[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。

制限事項

- データは、WinCC では受信されません。
- WinCC クライアントからクラウドにデータを送信することはできません。

Message Queue Telemetry Transport (MQTT)

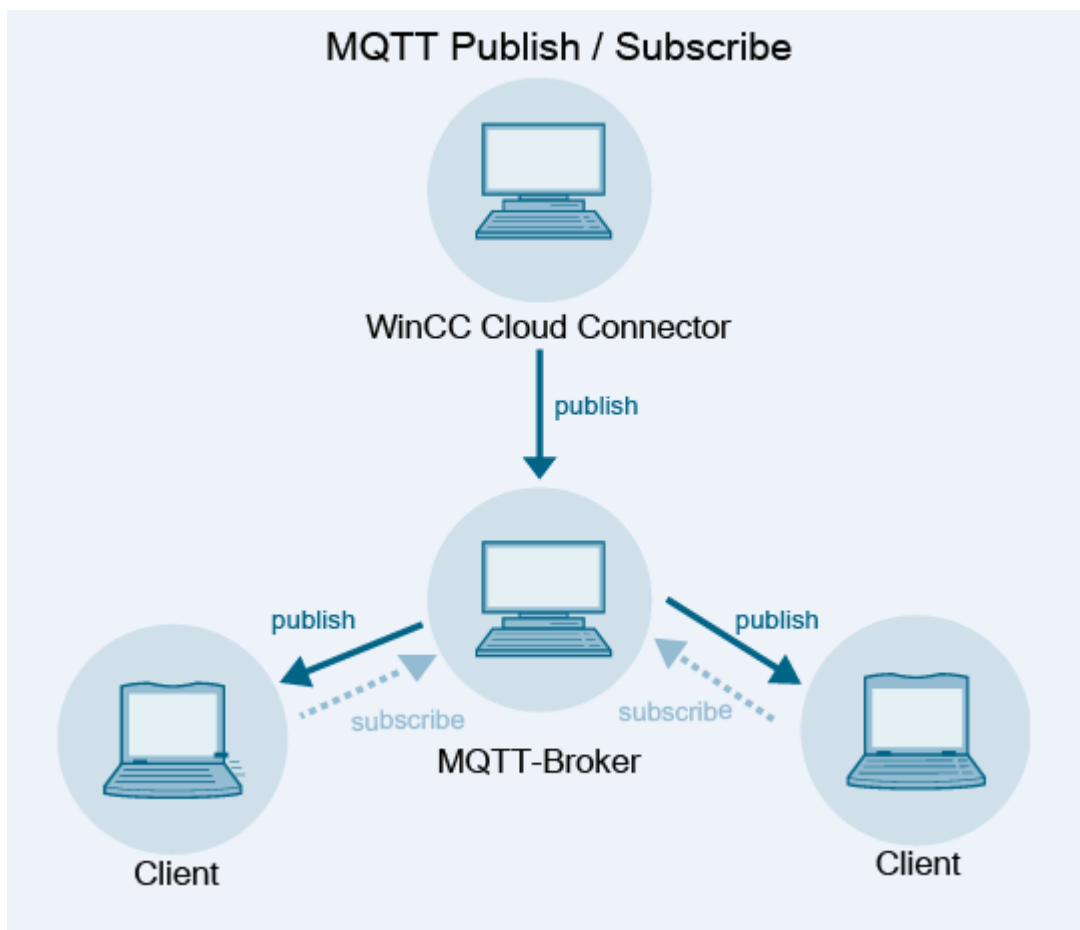
MQTT プロトコルを使用すると、中央サーバーである MQTT ブローカがデータ転送に使用されます。

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 507)]

下記も参照

WinCC クラウドコネクタの REST 設定 (ページ 626)

WinCC の REST インターフェースと MQTT プロトコル (ページ 507)

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)
- Siemens MindSphere - MindConnect EU1 (MQTT)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud (MQTT)の Siemens MindSphere:

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。

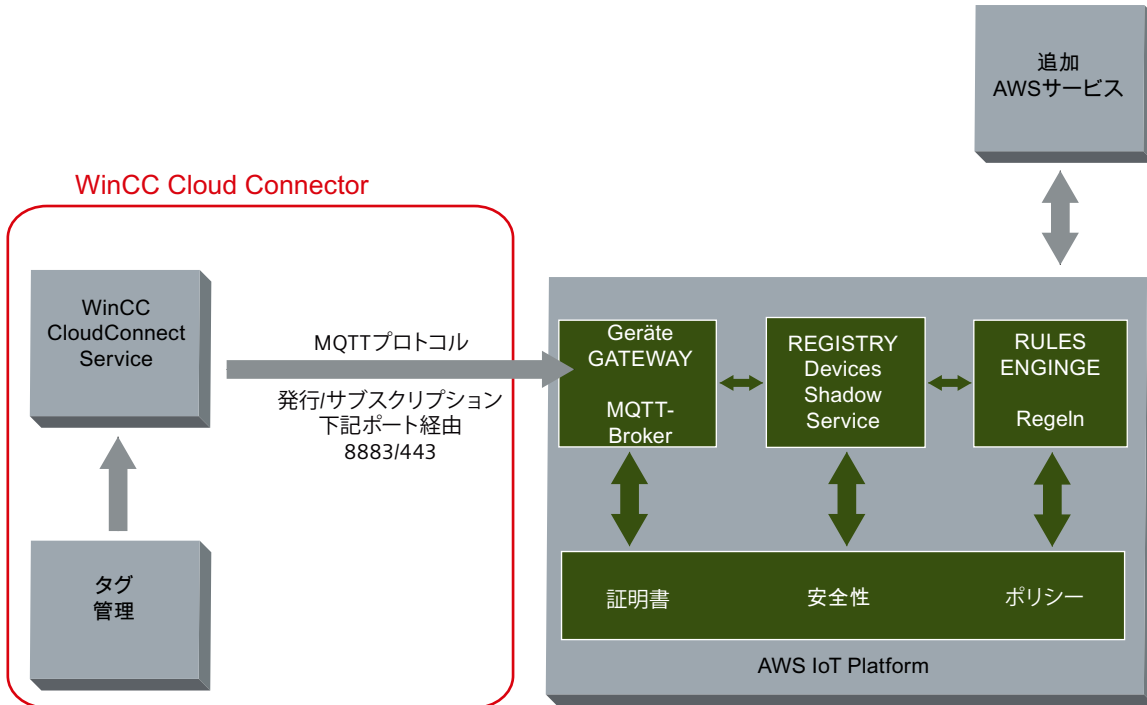
CCCloudConnect サービスは、標準ポート 8883 または 443 からデータを送信するために、クラウドの MQTT ブローカに接続する MQTT クライアントです。

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。

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

[MindSphere - MindConnect IoT Extension]や[MindSphere - MindConnect EU1]プロバイダを選択すると、WinCC からタグ名が適用されます。

[MindSphere - MindConnect EU1]プロバイダを選択すると、WinCC からタグコメントもユニットとして適用されます。

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せん。

最初に追加されるメッセージが最初に送信されます。

データ型

ほとんどのデータ型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型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。

10.3 MQTT を経由したクラウドへのデータ転送

MindSphere のタグタイプ

MindSphere では、以下のデータ型は 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

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。

- データバッファの標準サイズ:2 000 000 タグ値
システム設定およびクラウド接続の品質に応じて、より多くのタグ値が可能です。

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 619)

クラウド接続の診断 (ページ 645)

WinCC タグ管理での設定 (ページ 617)

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。

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 インターフェースと MQTT プロトコル (ページ 507)

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。

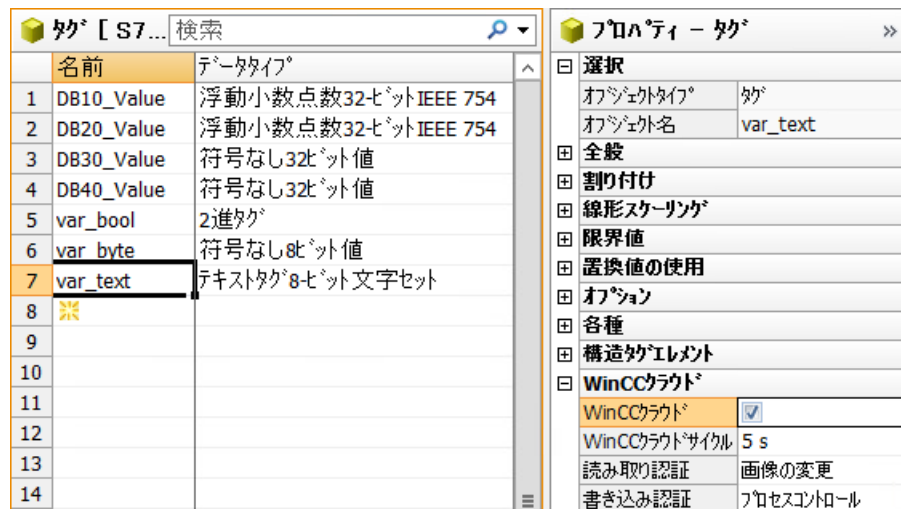
クラウドプロバイダは、それぞれ異なるタグタイプをサポートします。制限事項に関する情報は、「MQTT を経由したクラウドへのデータ転送 (ページ 611)」を参照してください。

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も関係します。



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 \(ページ 611\)](#)

[WinCC クラウドコネクタの MQTT 設定 \(ページ 619\)](#)

[クラウド接続の診断 \(ページ 645\)](#)

[IOT エクステンションにより MindSphere 接続を設定する方法 \(ページ 635\)](#)

[REST 経由のクラウド接続の設定方法 \(ページ 642\)](#)

[MQTT 経由のクラウド接続の設定方法 \(ページ 628\)](#)

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指定します。

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 screenshot shows the 'WinCC Cloud Connector構成' dialog box with the 'MQTT設定' tab selected. The 'MQTT設定' tab is active, and the 'REST設定' tab is also visible. The 'MQTT設定' section includes a checked checkbox for 'タグ値をMQTT経由でクラウドに送信'. Below this, the 'クラウドプロバイダ設定' section has a dropdown menu for 'クラウドプロバイダ' set to 'Generic (MQTT)'. The 'ブローカーアドレス(R):' field is empty, and the 'ブローカーポート(P):' field is set to '8883'. The 'デバイス設定' section has the 'ステーション名(S):' field set to 'WinCC'. There is an unchecked checkbox for '変更された値のみを送信'. The 'セキュリティ' section has three fields: '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 with an empty text box and a browse button (...). At the bottom right, there is a '接続のテスト' button.

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。

- MindSphere (MindConnect IoT Extension)
- MindSphere (MindConnect EU1)
- Amazon Web Services (MQTT)
- Microsoft Azure (MQTT)
- Alibaba Cloud (MQTT)の Siemens MindSphere:

アプリケーションの例

エントリ 109760955 では、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:

- 「MQTT を経由したクラウドへのデータ転送 (ページ 611)」

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)]または[MindSphere (MindConnect EU1)]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。

詳細については、AWS のオンライン文書で確認できます。「AWS IoT クライアント証明書の作成」。

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。

プロバイダ[MindSphere (MindConnect IoT Extension)]または[MindSphere (MindConnect EU1)]を選択すると、[セキュリティ]エリアの代わりに[MindSphere]エリアが表示され、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

注記

フォーマットの推奨

証明書を pem フォーマットでダウンロードします。

または、証明書を crt フォーマットでダウンロードし、それらを手動で.pem フォーマットに変換します。

変換に OpenSSL を使用します:

- `openssl x509 -in .\DigiCertGlobalRootG2.crt -out DigiCertGlobalRootG2.pem -outform PEM`

MindSphere クライアント証明書

プロバイダ[MindSphere (MindConnect IoT Extension)]または[MindSphere (MindConnect EU1)]を選択すると、[セキュリティ]エリアの代わりに[MindSphere]エリアが表示されます。

クライアント証明書は、プロバイダ[MindSphere (MindConnect EU1)]に対してのみ設定されます。

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 クライアント証明書

プロバイダ[MindSphere (MindConnect IoT Extension)]または[MindSphere (MindConnect EU)]を選択すると、[セキュリティ]エリアの代わりに[MindSphere]エリアが表示されます。

クライアントキーは、プロバイダ[MindSphere (MindConnect EU1)]に対してのみ設定されます。

登録/登録解除

このボタンは、プロバイダ[MindSphere (MindConnect IoT Extension)]を選択したときのみ表示されます。

[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)]を選択したときのみ表示されます。

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。

テナント ID

このフィールドは、プロバイダ[MindSphere (MindConnect EU1)]を選択したときのみ表示されます。

入力フィールドに MindSphere テナント ID を入力します。

アセットモデル/アセットインスタンス

このフィールドは、プロバイダ[MindSphere (MindConnect EU1)]を選択したときのみ表示されます。

アセットモデルとアセットインスタンスは JSON ファイルとして保存されます。

10.6 WinCC クラウドコネクタのMQTT 設定

入力フィールドに、アセットモデルとアセットインスタンスへのパスを入力します。

使用可能なアセットモデルやアセットインスタンスがない場合、WinCC タグ設定から[設定のエクスポート]ボタンを使用して生成できます。あるいは、テキストエディタで手動で作成することもできます。

設定のロード

このボタンは、プロバイダ[MindSphere (MindConnect EU1)]を選択したときにのみ表示されます。

[設定のロード]ボタンを使用して、アセットモデルとアセットインスタンスを MindSphere にロードできます。

設定のエクスポート

このボタンは、プロバイダ[MindSphere (MindConnect EU1)]を選択したときにのみ表示されます。

[設定のエクスポート]ボタンを使用して、アセットモデルとアセットインスタンスを WinCC タグ設定から生成できます。

ダイアログで次の MindSphere 特有の値を入力します:

- アスペクトタイプ名
- アスペクトタイプ説明
- アスペクト名
- アセットタイプ名
- アセットタイプ説明
- アセット名
- アセット説明
- アセットモデルのファイル:アセットモデルのファイル名とパス
- アセットインスタンスのファイル:アセットインスタンスのファイル名とパス

入力が確認されると、アセットモデルとアセットインスタンスが生成され、[アセットモデル]と[アセットインスタンス]に自動的に入力されます。

アセットモデルとアセットインスタンスがすでに存在している場合は、データの上書きをさらに確定する必要があります。

接続のチェック

[接続のチェック]ボタンを使用して、接続設定をリセットできます。

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 611)

MQTT 経由のクラウド接続の設定方法 (ページ 628)

クラウド接続の診断 (ページ 645)

WinCC タグ管理での設定 (ページ 617)

IOT エクステンションにより MindSphere 接続を設定する方法 (ページ 635)

REST 経由のクラウド接続の設定方法 (ページ 642)

WinCC クラウドコネクタの REST 設定 (ページ 626)

EU1 ブローカにより MindSphere 接続を設定する方法 (ページ 638)

SiePortal アプリケーション例:クラウドの 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。

The screenshot shows the 'WinCC Cloud Connector構成' dialog box with the 'REST設定' tab selected. The 'MQTT設定' tab is also visible. The 'REST設定' section includes a checked checkbox for 'タグ値をREST経由でクラウドに送信'. Below this, there are input fields for 'サービスアドレス(S):', 'サービスポート(P):' (set to 8080), and 'サービスパス(P):'. A dropdown menu for '送信方式:' is set to 'すべてのタグ値を"POST"コメントを使用して送信します。'. There is also an unchecked checkbox for '変更された値のみを送信する'. The '基本認証' section has input fields for 'ユーザー名(U):' and '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 619)

WinCC/Cloud Connector (ページ 607)

REST 経由のクラウド接続の設定方法 (ページ 642)

WinCC の REST インターフェースと MQTT プロトコル (ページ 507)

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される、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)]または["MindSphere (MindConnect EU1)]をクラウドプロバイダとして選択した場合は、次のそれぞれの指示に従ってください:

- IOT エクステンションにより MindSphere 接続を設定する方法 (ページ 635)
- EU1 ブローカにより MindSphere 接続を設定する方法 (ページ 638)

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:

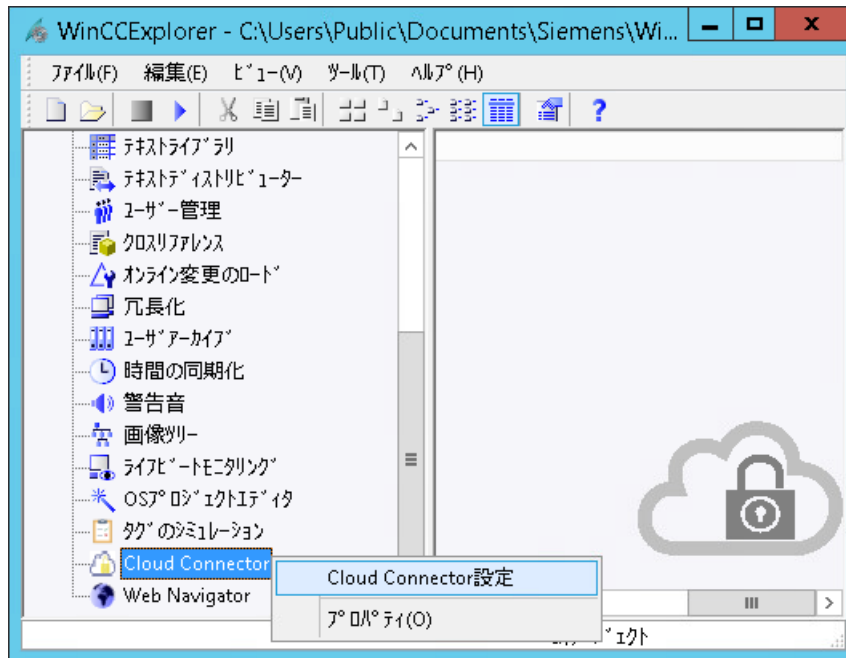
- REST 経由のクラウド接続の設定方法 (ページ 642)

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 screenshot shows the 'WinCC Cloud Connector構成' dialog box with the 'MQTT設定' tab selected. The 'MQTT設定' section has the checkbox 'タグ値をMQTT経由でクラウドに送信' checked. Under 'プロバイダ設定', 'クラウドプロバイダ' is set to 'Generic (MQTT)'. The 'ブローカーアドレス(R)' field is empty, and 'ブローカーポート(P)' is set to '8883'. Under 'デバイス設定', 'ステーション名(S)' is set to 'WinCC'. The checkbox '変更された値のみを送信' is unchecked. Under 'セキュリティ', there are three empty 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 with a browse button (...). A '接続のテスト' button is located at the bottom right of the dialog.

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 619)

MQTT を経由したクラウドへのデータ転送 (ページ 611)

IOT エクステンションにより MindSphere 接続を設定する方法 (ページ 635)

REST 経由のクラウド接続の設定方法 (ページ 642)

WinCC タグ管理での設定 (ページ 617)

EU1 ブローカにより MindSphere 接続を設定する方法 (ページ 638)

SiePortal アプリケーション例:クラウドの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)

10.9 MindConnect IoT Extension (Mindsphere / Insights Hub)から MindConnect MQTT にアセットを移行する方法

説明

MindConnect MQTT 経由で MindConnect IOT Extension において作成および設定されたアセットを接続することもできます。

このためには、既存のアセット設定を MindConnect IOT Extension から MindConnect MQTT へ移行します。

詳細については、Insights Hub の説明書で確認できます。

- 「MindConnect MQTT への移行 (<https://documentation.mindsphere.io/MindSphere/howto/how-to-migrate-mqtt-from-iotextension.html>)」

10.9 MindConnect IoT Extension (Mindsphere / Insights Hub)から MindConnect MQTT にアセットを移行する方法

手順

1. ブラウザ経由で MindConnect IOT Extension マッピングをエクスポートします。
このためには、次の原則に従って構造化された URL を呼び出します：
 - `https://{Tenant}-mciotextension.{Region}.mindsphere.io/service/oc2-map-config/api/v1/mcmqtt/{DeviceId}/assetmodeller+timeseries`
次のプレースホルダーに正しい値を入力します：
 - Tenant
 - Region
 - DeviceID
 「Info > Device Data」の「アセット設定」で「DeviceID」を確認できます。
アセットモデルはスクリプトとして表示されます。

Asset Model

```
{
  "id": "9bfa6958-097c-4d4a-a39d-e7d524f13435",
  "data": {
    "externalId": "68d7ea83-d08a-4bc5-8ce7-1d5d16fee4d0",
    "typeModel": {
      "aspectTypes": [],
      "assetTypes": []
    },
    "instanceModel": {
      "assets": []
    },
    "mappingModel": {
      "mappings": [
        {
          "dataPointId": "fb382439-4cc3-424c-9746-f0962c181979",
          "assetId": "eaa5da3878094abf8693eb1cd40f5f00",
          "aspectName": "c8y_Temperature",
          "variableName": "T",
          "measurement": "c8y_Temperature",
          "series": "T"
        }
      ]
    }
  }
}
```



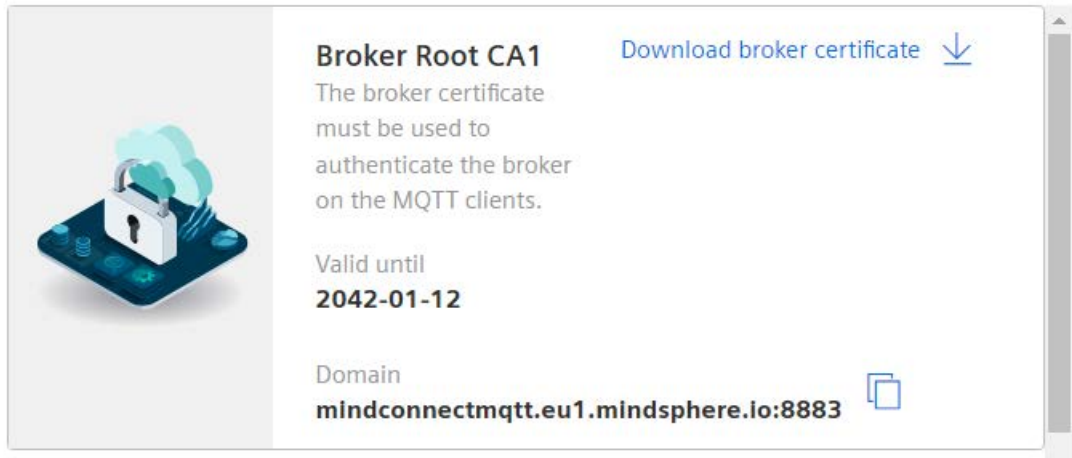
2. スクリプトの右上にあるボタンを使用してアセットモデルをコピーします。
3. TXT ファイルまたは JSON ファイルを作成し、スクリプトテキストを貼り付けます。
4. 「Manage MQTT Certificates」の「Asset Manager」で、Get a new certificate 経由で必要な証明書を作成します。
アセットが作成されます。

10.9 MindConnect IoT Extension (Mindsphere / Insights Hub)から MindConnect MQTT にアセットを移行する方法

5. 「Download broker certificate」 経由でルート証明書をダウンロードします。

Manage MQTT Certificates

Manage the certificates for securely connecting the MQTT agents.



6. WinCC Cloud Connector を開き、登録を解除します。
7. MindConnect MQTT の設定を指定します:
 - クラウドプロバイダ: Insights Hub (MindConnect EU1)
 - デバイス設定
8. ルート証明書(「CA 証明書」)、クライアント証明書、およびクライアントキーを選択します。
9. テナント ID を設定します。
10. [アセットモデル]フィールドで、エクスポートデータを保存した TXT ファイルまたは JSON ファイルを選択します。
11. 「設定をロード」 経由でアセットモデルの設定を Insights Hub にロードします。

結果

ランタイムを有効にすると、アセットのタグ値は MindConnect IOT Extension 経由ではなく MindConnect MQTT 経由で送信されます。

ドキュメントに記載されているとおりに、MindConnect MQTT の必要な設定を指定します。

10.10 IOT エクステンションにより MindSphere 接続を設定する方法

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される、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。

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。

10.10 IOT エクステンションにより MindSphere 接続を設定する方法

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 でグレーアウトされます。
 - [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 619)

WinCC タグ管理での設定 (ページ 617)

MQTT 経由のクラウド接続の設定方法 (ページ 628)

REST 経由のクラウド接続の設定方法 (ページ 642)

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>)

10.11 EU1 ブローカにより MindSphere 接続を設定する方法

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される、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。

EU1 ブローカにより MindSphere に接続することは、IOT エクステンションにより接続するもう 1 つの方法で、少し遅延があります。

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。
- デバイス ID は MindSphere に登録されます。

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 screenshot shows the 'WinCC Cloud Connector構成' dialog box with the 'MQTT設定' tab selected. The 'REST設定' tab is also visible. The 'MQTT設定' section includes a checked checkbox for 'タグ値をMQTT経由でクラウドに送信'. Below this are sections for 'プロバイダ設定' and 'デバイス設定'. The 'プロバイダ設定' section has a dropdown menu for 'クラウドプロバイダ' set to 'MindSphere (MindConnect EU1)', and text boxes for 'ブローカーアドレス(R)' (mindconnectmqtt.eu1.mindsphere.io) and 'ブローカーポート(P)' (8883). The 'デバイス設定' section has a text box for 'デバイスID' set to 'WinCC'. There is also an unchecked checkbox for '変更された値のみを送信する'. The 'MindSphere' section contains text boxes for 'CA証明書(C)', 'クライアント証明書', 'クライアントキー(L)', 'テナントID', 'アセットモデル', and 'アセットインスタンス', each with a browse button (...). Below these are two buttons: '設定をロード' and '設定をエクスポート'. At the bottom right is a 'Test connection' button. At the very bottom are 'OK', 'Cancel', and 'Help' buttons.

4. プロバイダとデバイスの設定を指定します:
- クラウドプロバイダ: [MindSphere (MindConnect EU1)]
 - デバイス ID
- ブローカーアドレスが自動的に入力されます。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. 証明書およびクライアントキーを選択します。
6. テナント ID を入力します。
7. 使用可能なアセットモデルとアセットインスタンスがすでにある場合は、それらを選択して、ステップ 12 に進んでください。

10.11 EU1 ブローカにより MindSphere 接続を設定する方法

8. [設定のエクスポート]ボタンをクリックして、アセットモデルとアセットインスタンスを WinCC タグ設定から生成します。

構成のエクスポート

アスペクトタイプ名: WinCCAspectType_Clouds_2023_02_23T11_48_03

アスペクトタイプ説明: Aspect Type generated by export of Cloud Tags of WinCC Project C

アスペクト名: WinCCAspect_Clouds_2023_02_23T11_48_03

アセットタイプ名: WinCCAssetType_Clouds_2023_02_23T11_48_03

アセットタイプ説明: Asset Type generated by export of Cloud Tags of WinCC Project Cl

アセット名: WinCCAsset_Clouds_2023_02_23T11_48_03

アセット説明: Asset Instance generated by export of Cloud Tags of WinCC Projec

ファイルのエクスポート:

アセットモデルのファイル: ...

アセットインスタンスのファイル: ...

OK キャンセル

9. MindSphere 特有の値を入力します:
- アスペクトタイプ名
 - アスペクトタイプ説明
 - アスペクト名
 - アセットタイプ名
 - アセットタイプ説明
 - アセット名
 - アセット説明
10. アセットモデルとアセットインスタンスのファイル名とパスを選択します。
11. 入力を[OK]ボタンで確定します。
アセットモデルとアセットインスタンスは、MQTT 設定の入力フィールドに自動的に入力されます。
アセットモデルとアセットインスタンスが前にすでに生成されている場合、データの上書きを確定します。
12. 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。
13. [設定のロード]ボタンでアセットの設定を MindSphere にロードします。
14. [OK]をクリックして、ダイアログを閉じます。
15. [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

適用例:クラウドの WinCC データ接続 (<https://support.industry.siemens.com/cs/ww/en/view/109760955>)

10.12 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される、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 628)

[MindSphere (MindConnect IoT Extension)]または["MindSphere - MindConnect EU1]をクラウドプロバイダとして選択した場合は、次の関連の指示に従ってください:

- IOT エクステンションにより MindSphere 接続を設定する方法 (ページ 635)
- EU1 ブローカにより MindSphere 接続を設定する方法 (ページ 638)

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. [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' dialog box with the 'REST設定' tab selected. The 'MQTT設定' tab is also visible. The 'REST設定' section includes a checked checkbox for 'タグ値をREST経由でクラウドに送信'. Below this is a section for 'API設定' with fields for 'サービスアドレス(S)', 'サービスポート(P)' (set to 8080), and 'サービスパス(P)'. The '送信方式' dropdown is set to 'すべてのタグ値を"POST"コメントを使用して送信します.'. There is also an unchecked checkbox for '変更された値のみを送信する'. The '基本認証' section has fields for 'ユーザー名(U)' and 'パスワード(P)'.

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 628)

WinCC クラウドコネクタの MQTT 設定 (ページ 619)

WinCC タグ管理での設定 (ページ 617)

IOT エクステンションにより MindSphere 接続を設定する方法 (ページ 635)

10.12 REST 経由のクラウド接続の設定方法

WinCC クラウドコネクタの REST 設定 (ページ 626)

EU1 ブローカにより MindSphere 接続を設定する方法 (ページ 638)

10.13 クラウド接続の診断

概要

WinCC は、パフォーマンス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 WinCC インストールパスの[diagnose]フォルダで作成されます。

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 611)

WinCC クラウドコネクタの MQTT 設定 (ページ 619)

WinCC タグ管理での設定 (ページ 617)

Internet: <https://mosquitto.org/> (<https://mosquitto.org/>)

索引

"

"システム情報"チャンネル, 314
診断オプション, 314

A

A&E サーバー, 436, 438, 446, 448, 449
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, 449
WinCC メッセージシステムのマッピング, 438,
449
階層アクセス, 448
条件関連付けイベント, 436
条件付きイベント, 446
単純イベント, 446
追跡用イベント, 436, 446
ActiveX コントロールとしての, 298
Application Health Check, 130, 166
AS データタイプ, 255
タイプ変換, 269
Autostart, 121

B

BinWrite メカニズム, 283
一般的な手順, 283

C

CCCloudConnect サービス, 611
Channel DiagnosisChannelDiagnosis
KanalDiagnosismitChannelDiagnosis, 309

E

ERROR フラグ, 314, 330

I

I/O フィールド設定, 402
OPC DA の例, 402
INFO フラグ, 314, 330, 343, 354

M

MQTT プロトコル, 607

O

OPC

HDA サーバーブラウザ, 431
OPC A&E サーバーの機能, 446
OPC コンピュータでのユーザーアカウントの設
定, 396
ProgID, 400
WinCC OPC A&E サーバーの機能, 436
WinCC OPC DA サーバーの機能, 397
WinCC OPC HDA サーバーの機能, 418
WinCC OPC UA サーバーの動作原理, 464
WinCC の OPC, 394
チャンネル診断, 353
トレース, 506
互換性, 393
仕様, 393
新規ユーザー, 396
通信の概念, 394
複数の OPC DA サーバーの使用, 399
例, 405, 406, 407, 409, 412, 413, 414, 417
例:, 401, 402, 403
OPC A&E サーバー, 436, 439, 446, 448, 449
品質コード, 445, 457
OPC A&E サーバーのメッセージクラス, 449
OPC A&E サーバーのメッセージタイプ, 449
OPC A&E サーバーの機能
条件付きイベント, 446
単純イベント, 446
追跡用イベント, 446
OPC A&未処理データ, 445
OPC DA サーバー, 397
WinCC OPC DA サーバーの機能, 397
複数の OPC DA サーバーの使用, 399
OPC HDA サーバー, 418
WinCC OPC HDA サーバーの時間形式, 424
サポートされているファンクション, 423
データ構造, 420, 421
境界値, 420
項目 ID, 420
項目ハンドル, 420
集約, 422
書込みアクセス, 427

- 属性, 421
 - 動作の原則, 418
 - 品質コード, 426, 445
 - 未処理データ, 418
 - OPC UA
 - WinCC タグの表示, 477
 - アーカイブタグの表示, 478
 - プロファイル, 473
 - 適合ユニット, 473
 - OPC UA Historical Access, 479
 - OPC UA サーバー
 - URL, 464
 - アプリケーション証明書, 466
 - インスタンス証明書, 466
 - サポートされる仕様, 464
 - セキュリティポリシー, 465
 - 証明書, 466
 - 信頼されるクライアント証明書, 467
 - 設定ファイル, 493
 - 通信プロファイル, 464
 - 動作, 464
 - OPC UA データアクセス, 479
 - OPC UA メソッド, 490
 - OPCScout 新規プロジェクト 1
 - OPC DA の例, 409
 - OPC インターフェース, 102
 - クライアント/サーバーシステムで, 102
 - OPC チャンネル, 353
 - 診断オプション, 353
 - OPC 項目マネージャ, 400
- P**
- PROFIBUS システムパラメータ, 336
 - ProgID
 - 問い合わせ, 400
 - PSK キー, 18, 86, 88
- R**
- REST サービス, 615
 - REST プロトコル, 607
- S**
- S5 Profibus FDL
 - チャンネル診断, 342
 - S7 Protocol Suite
 - チャンネル診断, 329
 - S7DOS, 247
 - SelfDiagnosis, 130
 - ServiceMode, 107
 - SIMATIC S5 PROFIBIS FDL チャンネル, 342
 - 診断オプション, 329
 - SIMATIC S7 Protocol Suite チャンネル, 329
 - 診断オプション, 329
 - SIMATIC S7-1200, S7-1500 Channel
 - システム診断の設定, 323
 - チャンネル診断, 319
 - SIMATIC Shell, 83, 88
- U**
- URL
 - OPC UA サーバー, 464
- W**
- WinCC, 14
 - ServiceMode, 107
 - WinCC の OPC, 394
 - クライアント/サーバーシステム, 14
 - サービスモード, 107
 - WinCC Explorer-OPC_Client.MPC, 400
 - WinCC OPC A&E サーバー
 - 階層アクセス, 448
 - WinCC OPC A&E サーバーの機能
 - 単純イベント, 436
 - WinCC OPC UA サーバー, 464
 - 設定, 500
 - 発見サーバー, 465
 - WinCC ServiceMode, 107, 116
 - 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, 107
 - WinCC タイプ変換, 255
 - WinCC タグ@RM_MASTER, 158
 - WinCC タグ@RM_MASTER_NAME, 158
 - WinCC チャンネル診断, 316, 338, 349, 360
 - WinCC データタイプ, 255, 260
 - WinCC データタイプ, 260
 - タイプ変換, 260, 269
 - WinCC と Microsoft Excel 間の接続, 413
 - OPC DA の例, 413
 - WinCC と SIMATIC NET FMS OPC サーバー間の接続, 405
 - OPC DA の例, 405
 - WinCC と SIMATIC NET S7 OPC サーバー間の接続, 407
 - OPC DA の例, 407
 - WinCC プロジェクト, 83
 - WinCC プロジェクトダイアログ, 77

リモートの実行, 98
 リモート終了, 100
 呼び出し, 77
 WinCC プロセス通信, 247
 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, 439
 OPC A&OPC-A&WinCC メッセージシステム上
 の, 438
 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, 439, 440
 属性, 440, 452
 WinCC 間の接続, 401
 OPC DA の例, 401
 WinCC 通信, 248
 原理, 248
 Windows
 タスクバー, 113
 トレイエリア, 113

ア

アーカイブ ` メモリ, 159, 162

イ

インストール, 105
 ファイルサーバー, 105
 インポート, 50
 パッケージ, 50

ウ

ウェブクライアント, 18

エ

エクスポート, 37
 パッケージ, 37

オ

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, 133

ク

クライアント, 18, 45
 インポートパッケージ, 50
 ウェブクライアント, 18
 オペレータ権限, 33
 コンピュータのプロパティの設定, 41
 サーバーでプロジェクトを開く, 95
 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, 41
 サーバープロジェクトの終了, 100
 サーバープロジェクトへのアクセス, 92
 サーバーへの表示付き, 41
 サーバー画像の編集, 97
 サーバー上での同時起動, 21
 シャットダウン, 81
 シンクライアント, 18
 ピクチャの変更のコンフィグレーション, 64
 ブート, 78
 プロジェクト新規作成, 47
 マルチユーザーシステム, 41
 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, 41
 メッセージシーケンスレポートの設定, 70
 リモートサーバープロジェクトの実行, 98
 開始画像の設定, 60
 設定, 45
 複数サーバーのピクチャの表示, 62
 複数サーバーのメッセージの表示, 69
 複数のサーバーからのデータの表示, 66
 分散システムにおける設定, 45
 クライアント/サーバーシステム, 14
 OPC インターフェースのアプリケーション, 102
 クライアントおよびサーバーの可能数, 14
 クライアントタイプ, 18
 クライアントのシャットダウン, 81
 クライアントのスタートアップ, 78
 サーバーのシャットダウン, 81
 サーバーのスタートアップ, 77
 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, 73
 使用方法, 14
 冗長性, 14, 24
 数量構造, 21
 設定, 18
 必要条件, 14
 標準サーバー, 24
 複合設定, 18

複数ステーションシステム, 14
分散システム, 14, 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, 47
クラウドコネクタ, 607

コ

コンピュータ
ステータス, 113
コンピュータリスト, 32
クライアントの受入れ, 32

サ

サーバー, 24, 28
SOFTNET ドライバの特殊機能, 79
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, 98, 100
クライアント側からの画像編集, 97
サーバープロジェクトにおけるクライアントの設定, 41
サーバー間通信, 24
シャットダウン, 81
ファイルサーバー, 24
ブート, 77, 79
プロジェクト新規作成, 30
リモートコンフィグレーション(必要条件), 32
リモート設定, 33
設定, 28
標準サーバー, 24
標準サーバーの設定, 53
複数のクライアントからのプロジェクトアクセス, 92
複数のネットワークカードを使った特殊機能, 79
優先サーバーの構成, 57
サーバーデータ, 37
パッケージインポート, 50
パッケージエクスポート, 37
標準サーバーの設定, 53
優先サーバーの構成, 57
サーバーでの SOFTNET ドライバ常時使用, 79
サーバーパッケージ:, (以下の項目を参照してください:
パッケージ)
サーバー間通信, 24
サーバー接続語, 60, 62, 64, 66
サービスプロジェクト, 107, 116
Autostart, 121
インタラクティブな操作の起動, 127
インタラクティブな操作の終了, 127
スクリプト, 111

スタートアップリスト, 111
ログオフ, 126
ログオン, 126
診断情報, 111
制約, 110
接続ステーション, 111
設定, 109
定義, 120
非リリース構成要素, 111
必要条件, 118
標準プロジェクトへの変更, 121
用途, 110
サービスモード, 107
サブネット, 88, (サブネットを参照)

シ

システムエラー, 73
システムタグ, 168
システムパラメータ, 248, 347
システムメッセージ, 173
システム情報
チャンネル診断, 314
システム診断, 323
デバイス表示, 319
詳細表示, 320
診断バッファ表示, 321
システム診断表示, 322
システム動作, 73
シナリオ, 24, 158
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, 26
シャットダウン, 81
クライアントの, 81
サーバーの, 81
シリアルケーブル, 133
シンクライアント, 18

ス

スクリプト
サービスプロジェクト, 111
スタートアップリスト
サービスプロジェクト, 111
スタートアップ動作, 77, 78
クライアントの, 78
サーバーの, 77
ステータス表示, 113, 253, 316, 338, 349, 360

タ

- タイプ変換, 255, 269
- タグ, 255, 362
 - BinWrite 付きタグの設定, 285
 - Check, 318, 362
 - オートメーションシステムの外部タグのアドレス指定, 255
 - タグの設定のテスト, 336, 347
 - テキストタグの構成, 259
 - テキストタグの長さ情報, 255
 - パワータグ, 255
 - パワータグの構成, 259
 - 外部, 255
 - 外部タグの構成, 259
 - 確認, 340, 351
 - 線形スケーリングの構成, 259
 - 内部タグのチェック, 388
 - 品質コード, 374
- タグ^々, 372
 - HDA サーバーブラウザ, 431
 - OPC DA の例, 409, 412, 417
 - ステータス, 372, 383
 - 品質, 372
 - 品質コード, 372
- タグステータス, 372, 383, 386, 387
 - グローバルアクションによるタグステータスのモニタ, 387
 - ダイナミックダイアログでのタグステータスのモニタ, 386
- タグの追加
 - OPC DA の例, 403
- タグの同期化, 138
- タグの品質コード, 372, 374
- タグプロパティ, 259
- タグ名
 - 接頭語, 258
 - 接尾語, 258
- タスクバー, 113

チ

- チャンネル, 248, 360
 - テスト, 316, 338, 360
 - 確認, 349
 - 接続の確立/終了, 253
 - 論理接続ステータス, 253, 295
- チャンネルユニット, 248
- チャンネル診断, 297
 - "システム情報"チャンネル, 314

- OPC チャンネル, 353
- PROFIBUS システムパラメータ, 336
- SIMATIC S5 PROFIBUS FDL チャンネル, 342
- SIMATIC S7 Protocol Suite チャンネル, 329
- SIMATIC S7-1200, S7-1500 Channel, 319
- WinCC チャンネル診断, 316, 338, 349, 360
- Windows アプリケーションとしてのチャンネル診断, 309
 - グローバルアクションによるタグステータスのモニタ, 387
 - システムパラメータ, 347
 - ステータスメッセージの概要, 316, 338, 349, 360
 - ダイナミックダイアログでのタグステータスのモニタ, 386
 - タグのチェック, 318, 362
 - タグの確認, 340, 351
 - チャンネルのトレースファンクションの設定, 311
 - チャンネルのテスト, 316, 338, 349, 360
 - チャンネル診断, 297, 298, 309
 - トレースファンクションの標準フラグ, 311
 - によるチャンネル診断, 298
 - プロパティ - CP5613, 333, 335, 344, 346
 - ログファイル, 314, 330, 343, 354
 - 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, 306, 310
 - 次によるチャンネル診断, 297
 - 診断オプション, 314, 329, 342, 353
 - 接続ステータス, 295
 - 接続の確認, 316, 338, 349, 360
 - 接続パラメータ, 347
 - 設定データのチェック, 358
 - 通信プロセッサの設定確認, 333, 335, 344, 346
 - 内部タグのチェック, 388
 - 例, 387
 - 論理接続ステータス, 295
- チャンネル診断チャンネル
 - テスト, 306, 310
- 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, 298
- チャンネル診断トレースファイル, 297

ツ

- ツールヒント, 490

- テ
- データ, 62
 - 複数サーバーの表示, 62
 - 複数のサーバーからの表示, 66
 - データタイプ, 255
 - タイプ変換, 269
 - テーブルエリア, 490
 - テキストタグ, 255
 - 構成, 259
- ト
- トランスポートパラメータ, 248
 - トレイエリア, 113
 - トレースファンクション, 311
 - 設定, 311
- ナ
- ナビゲーションエリア, 490
 - ナビゲーションバー, 490
- ネ
- ネットワーク, 88, 245
 - サブネット, 88
 - ルータ, 88
- ハ
- バイト単位のアクセス, 283
 - パッケージ, 37, 50
 - WinCC エクスプローラの表示, 37, 50
 - インポートの設定, 50
 - エクスポートの設定, 37
 - ファイルシステムにおける保存, 37
 - 自動更新, 37
 - パフォーマンス, 21
- ヒ
- ビットアクセス/バイトアクセス, 283
 - ビットタグ/バイトタグ, 285
 - ビット単位のアクセス, 283
- フ
- ファイルサーバー, 24, 105
 - インストール, 105
 - 設定, 105
 - ブート, 77, 78
 - クライアントの, 78
 - サーバーの, 77, 79
 - プロジェクト, 30, 47
 - OPC DA の例, 403
 - WinCC のステータス, 113
 - クライアントプロジェクトの新規作成, 47
 - 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, 92
 - リモートオープン, 95
 - 実行, 98
 - 終了, 100
 - 複製, 149, 152
 - プロジェクトデュプリケータ, 149, 152
 - プロパティ
 - ポップアップヒント, 490
 - プロパティ - CP5613, 333, 335, 344, 346
 - プロパティエリア, 490
- マ
- マルチユーザーシステム, 14, 24
 - クライアントの設定, 41
 - 設定, 28
 - マルチユーザープロジェクト, 30
 - 作成, 30
- メ
- メッセージ, 69
 - クライアント側でのメッセージシーケンスレポートの設定, 70
 - 複数のサーバーからの表示, 69
 - メッセージクラスのロック, 133
 - パッシブ, 133
 - メッセージシーケンスレポート, 70
 - メッセージのロック, 133
 - アクティブ, 133
 - パッシブ, 133
 - メッセージのロック(アクティブ), 133
 - メッセージのロック(パッシブ), 133
- 652

ユ

ユーザーアカウント, 396
OPC コンピュータへの通知, 396

ラ

ランタイム, 73
アーカイブ, 73
グラフィック, 73
システムエラー時の動作, 73
システム動作, 73
スクリプト, 73
テキストライブラリ, 73
メッセージ, 73
ユーザー管理者, 73
レポート, 73

リ

リダンダントサーバーの起動, 140
リダンダントサーバー間のシリアル接続, 143
リモートコンフィグレーション
プロジェクトの実行, 98
リモート設定, 83
サーバープロジェクトへのアクセス, 92
プロジェクトの終了, 100
プロジェクトを開く, 95
画像の編集, 97

ル

ルータ, 88

ロ

ログファイル, 314, 330, 343, 354
ログブックエントリ, 314, 330, 343, 354
ログブックファイル, 297

ワ

ワイルドカード, 601

安

安全な通信, 88

暗

暗号化通信, 18

画

画像, 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, 64
クライアント側での複数サーバーの表示形式, 62
サーバープロジェクトの編集, 97
サーバー接頭語, 60
開始画像の設定, 60

開

開く
プロジェクト, 95
開始画像, 60
クライアントの設定, 60

確

確認セオリー, 443
確認ポリシー, 454

境

境界値, 420

互

互換性, 393

項

項目 ID, 420
項目ハンドル, 420

試

試運転, 506

時

時間同期化, 133
時系列順メッセージ, 133

- 自**
自動更新, 37, 50
- 実**
実行, 98
プロジェクト, 98
- 冗**
冗長システム
プロジェクトの複製, 149, 152
設定, 142
冗長システムの必要条件, 133
冗長化の設定, 140
冗長性, 24, 130
Application Health Check, 130, 166
アーカイブの切り替え, 130
オペレーティングシステム, 133
サーバーの起動, 130
サーバー間のシリアル接続, 143, 158
システムメッセージ, 159
時間同期化, 133
時系列順メッセージ, 133
設定, 140
必要条件, 133
優先サーバー, 24
優先サーバーの構成, 57
- 新**
新規タグの定義, 409
OPC DA の例, 409
新規ユーザー, 396
- 診**
診断
接続ステータス, 253, 295
診断オプション, 314, 329, 342, 353
"システム情報"チャンネル, 314
OPC チャンネル, 353
SIMATIC S5 PROFIBIS FDL チャンネル, 342
SIMATIC S7 Protocol Suite チャンネル, 329
診断情報
サービスプロジェクト, 111
診断能力
SIMATIC S7-1200, S7-1500 Channel, 319
- 数**
数値タグタイプ, 255
リニアスケールリング, 255
数量構造, 21
- 切**
切り捨て, 601
- 接**
接続, 248, 360
テスト, 316, 338, 360
ランタイムでのステータス, 253, 295
確認, 349
確立/終了, 253
新規接続の作成, 258
接続の設定のテスト, 336, 347
接続ステーション
サービスプロジェクト, 111
接続ステータス, 113
テスト, 295
接続パラメータ, 248, 347
接続プロパティ, 258
接頭語, 258
接尾語, 258
- 設**
設定, 18, 28, 45, 105, 140, 142
WinCC OPC UA サーバー, 500
クライアント, 45
クライアント/サーバーシステム, 18
サーバー, 28
セキュリティ設定, 500
ファイルサーバー, 105
ポート番号, 500
マルチユーザーシステム, 28
ユーザー ID, 501
リモート, 83
最適 WinCC アーカイブ書き込みアクセス, 501
分散システム, 28
設定データ, 358
設定ファイル
WinCC OPC UA サーバーの URL, 493
WinCC ログイン書き込みアクセス, 493
セキュリティ設定, 493
ポート番号, 493

ユーザー ID, 493
 レイアウト, 493
 証明書, 493
 保存先, 494

値

値の範囲, 269

中

中央アーカイブサーバー, 26

通

通信, 245

AS データタイプ, 259, 269
 AS データタイプによるフォーマット調整のソート, 269
 BinWrite メカニズム, 283
 BinWrite メカニズムの原則, 283
 BinWrite 付きタグの設定, 285
 IPv4 プロトコル, 247
 WinCC タイプ変換, 255
 WinCC データタイプ, 255
 WinCC データタイプによるフォーマット調整のソート, 260
 WinCC とオートメーションシステム間の通信, 248
 WinCC プロセス通信, 247
 WinCC 通信の原則, 248
 オートメーションシステムのアドレス指定, 255
 タイプ変換, 255, 269
 タグプロパティ, 259
 チャンネルユニット, 248
 データタイプ, 255, 269
 テキストタグの長さ情報, 255
 パワータグ, 255
 パワータグの構成, 259
 ビットアクセス/バイトアクセス, 283
 ビットタグ/バイトタグ, 285
 基本, 245
 新規接続の作成, 258
 数値タグタイプのリニアスケールリング, 255
 接続, 248
 接続プロパティ, 258
 値の範囲, 269
 通信ドライバ, 248
 通信ドライバ, 248
 接続の確立/終了, 253

通信プロセッサ, 333, 335, 344, 346
 通信プロセッサの設定確認, 333, 335, 344, 346

標

標準サーバー, 24
 設定, 53
 標準フラグ, 311
 標準プロジェクト, 107, 116
 サービスプロジェクトへの変更, 121

表

表示, 62
 複数サーバーから, 62
 複数のサーバーからのデータ, 66

品

品質コード, 426, 445, 457

複

複合設定, 21

分

分散システム, 14, 24
 クライアントコンフィグレーション, 45
 クライアントでのデータ出力の設定, 66
 クライアントプロジェクトの作成, 47
 クライアント側でのサーバーピクチャの表示, 62
 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, 64
 クライアント側でのメッセージシーケンスレポートの設定, 70
 クライアント側でのメッセージの表示, 69
 パッケージインポート, 50
 リモート設定, 83
 設定, 28

変

変更のオンラインロード
 冗長システムのユーザーアーカイブへの変更, 148

未

未処理データ, 418

無

無停電電源装置(UPS), 133

優

優先サーバー, 24
構成, 57

例

例:, 387