

SIEMENS

SIMATIC HMI

WinCC V8.1 Update 2 SIMATIC HMI WinCC V8.1 スタート ガイド

入門書


ようこそ	1
アイコン	2
WinCC の操作	3
プロジェクトの作成	4
通信の設定	5
プロセス画面の設定	6
フェイスプレートタイプとタグ構造の設定	7
値のアーカイブと表示	8
メッセージの設定	9
検索、発見、フィルタリング	10


オンラインヘルプの印刷


法律上の注意

警告事項

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

 危険
回避しなければ、直接的な死または重傷に至る危険状態を示します。

 警告
回避しなければ、死または重傷に至るおそれのある危険な状況を示します。

 注意
回避しなければ、軽度または中度の人身傷害を引き起こすおそれのある危険な状況を示します。

通知
回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します。


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

有資格者

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

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

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

商標

®マークのついた称号はすべて Siemens 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	ようこそ	7
2	アイコン	11
3	WinCC の操作	13
4	プロジェクトの作成	17
4.1	"Quick_Start"プロジェクトの作成	18
5	通信の設定	21
5.1	WinCC のチャンネルと接続	23
5.2	WinCC のタグ	24
5.3	チャンネルの追加	26
5.4	接続の作成	31
5.5	タググループのプロセスタグの作成	35
5.6	WinCC でのプロセスタグのスケーリング	41
5.7	内部タグの作成	43
6	プロセス画面の設定	47
6.1	グラフィックシステム	49
6.2	プロセス画面の作成	51
6.3	プロセス画面の編集	55
6.3.1	ライブラリからのグラフィックオブジェクトの挿入	56
6.3.2	[スタティックテキスト]グラフィックオブジェクトの挿入	61
6.3.3	「SAMPLE.Pdl」プロセス画像の編集	64
6.4	プロセス画面のダイナミック化	66
6.4.1	フィルレベルインジケータのダイナミック化	67
6.4.2	I/O フィールド(グラフィックシステム - プロセス画像)の挿入とダイナミック化	71
6.5	ユーザー定義メニューおよびツールバーの作成	77
6.5.1	カスタマイズされたメニューおよびツールバーのプロシージャの作成	77
6.5.2	画面変更のためのカスタマイズされたメニューの作成	81
6.5.3	ランタイムを終了するカスタマイズされたツールバーの作成	84
6.6	ランタイム中のプロジェクトの実行	87
6.6.1	ランタイムタイムプロパティの定義(グラフィックシステム - プロセス画像)	87
6.6.2	プロジェクトの有効化(グラフィックシステム - プロセス画像)	90
6.6.3	プロジェクトのテスト(グラフィックシステム - プロセス画像)	93
6.6.4	ランタイムシステムダイアログの使用	97

7	フェイスプレートタイプとタグ構造の設定.....	103
7.1	フェイスプレートタイプの設定	103
7.2	構造タグの作成.....	106
7.2.1	構造タイプの作成	106
7.2.2	構造タグの作成.....	109
7.3	ポップアップ画面の設定	112
7.3.1	ポップアップ画面の作成とそのサイズの適合.....	112
7.3.2	ポップアップ画面にオブジェクトに配置.....	113
7.3.3	ポップアップ画面でオブジェクトをダイナミック化	116
7.4	フェイスプレートタイプの設定	120
7.4.1	フェイスプレートタイプの作成とオブジェクトの配置.....	120
7.4.2	構造タイプとフェイスプレートタイプのリンク	122
7.4.3	ポップアップ画面の設定.....	127
7.5	フェイスプレートインスタンスの作成	129
7.6	フェイスプレートタイプの変更	133
7.6.1	オブジェクトプロパティのアニメーション化.....	133
7.7	プロジェクトのテスト(グラフィックシステム - フェイスプレートタイプ).....	137
8	値のアーカイブと表示	143
8.1	値のアーカイブと表示	143
8.2	アーカイブシステム.....	144
8.3	タグロギングの開始.....	146
8.4	タイマーの設定.....	148
8.5	プロセス値アーカイブの作成.....	150
8.6	プロセス値アーカイブの編集.....	153
8.7	プロセス画面の設定(アーカイブシステム).....	156
8.7.1	[トレンド]ウィンドウの設定.....	156
8.7.2	[テーブル]ウィンドウの設定.....	164
8.8	ユーザー定義メニューのカスタマイズ(アラームシステム).....	170
8.9	ランタイムプロパティの定義(アラームシステム).....	173
8.10	プロジェクトの有効化とテスト(アーカイブシステム).....	176
9	メッセージの設定.....	179
9.1	メッセージシステム.....	180
9.2	アラームロギングの起動.....	182
9.3	メッセージブロックの設定	183

9.4	ビットメッセージの設定	186
9.4.1	ビットメッセージの作成	187
9.5	アナログメッセージの設定	191
9.5.1	制限値の設定	191
9.5.2	トレンドウィンドウでのメッセージの表示	194
9.6	メッセージステータスの色の定義	200
9.7	プロセス画面の設定(メッセージシステム)	202
9.7.1	アラームメッセージウィンドウの設定	202
9.7.2	スライダオブジェクトの挿入とダイナミック化	207
9.7.3	スケールの挿入	211
9.7.4	I/O フィールド(メッセージシステム)の挿入とダイナミック化	215
9.8	ユーザー定義メニューのカスタマイズ(メッセージシステム)	220
9.9	ランタイムプロパティの定義(メッセージシステム)	222
9.10	プロジェクトの有効化(アラームロギング)	225
9.11	プロジェクトのテスト(メッセージシステム)	229
10	検索、発見、フィルタリング	233
	用語集	237
	索引	247

WinCC「スタートガイド」へようこそ

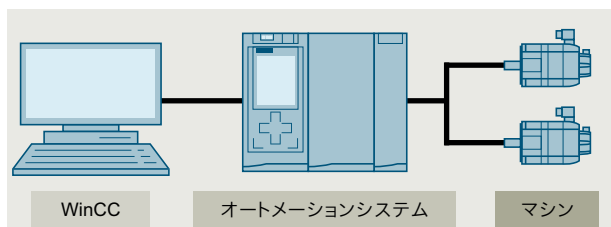
「スタートガイド」は、WinCC V8 に関するターゲットが絞られている概要ドキュメントです。

SIMATIC HMI WinCC

WinCC は、Microsoft Windows または Microsoft Windows Server で使用する強力な HMI システムです。

HMI は、"ヒューマンマシンインターフェース"、つまり人間と機械の間のインターフェースを表します。

WinCC は、マシンやプラントで実行されているプロセスのオペレータコントロールおよびモニタリングを有効にします。WinCC とマシンの間の通信は、AS を通して行なわれます。



スタートガイド

このドキュメントの全てのテーマを学習して、WinCC 設定の基本を学ぶには約 4 時間で十分です。

個別の設定ステップの画像表示により、エディタにすばやく移動することができます。

サンプルプロジェクト「Quick_Start」

サンプルプロジェクトの作成は、「スタートガイド」で説明します。このプロジェクトを使用して、水処理システムを制御できます。

そうするために、次のようなシステムを操作、監視するのに必要な様々なオブジェクトを「設定」します。

- 制御装置でプロセスを表示、操作する画像
- 操作対象装置と設備の間でデータを転送するタグ

- プロセスデータを保存するアーカイブ
- HMI デバイスのプラントの動作ステータスを示すメッセージ

「スタートガイド」は、以下のステップを通じてガイダンスを行います。

- プロジェクトの作成
ユーザーインターフェースを設定するためのベースとなる WinCC エクスプローラのプロジェクトを作成します。
- 通信の設定
[タグ管理]エディタで接続とタグを設定し、AS と HMI デバイスの間の通信を確立します。
- プロセス画像の設定
プロセスの表示、操作、モニタリングに使用する[グラフィックデザイナー]エディタでプロセス画像を設定します。
- フェイスプレートタイプと構造タイプの設定
プロセス画像と構造タググループのために、再利用可能なテンプレートを設定します。
- 値のアーカイブと表示
[タグロギング]エディタでアーカイブを設定し、アーカイブサイクルを定義し、プロセス値をアーカイブできるようにします。
- メッセージの構成
[アラームロギング]エディタでディスクリットアラームとアナログアラームを設定し、マシンやプラントのプロセスを監視できるようにします。

WinCC のインストール

WinCC セットアップには詳細なインストール説明書が含まれます。

WinCC セットアップには、設定ステップの実行に必要なすべてのプログラムが含まれています。

WinCC の技術情報およびトレーニング

WinCC の詳細については、次のインターネットを参照してください。

- オンラインサポートの SIMATIC WinCC V7 (<https://support.industry.siemens.com/cs/ww/en/view/93906404>):
WinCC に関連するすべての重要情報を一覧表示します。
- トレーニング:
 - SITRAIN コース:WinCC V7 (基本コース/システムコース) (<https://support.industry.siemens.com/cs/ww/en/view/109758633>)
 - SITRAIN 上級コース:WinCC V7 および WinCC オプション (<https://support.industry.siemens.com/cs/ww/en/view/109758660>)
- オートメーションおよびドライブテクノロジー用互換性ツール: (<https://support.industry.siemens.com/kompatool/index.html?lang=en>):
ツールは、Siemens 製品の互換性に関する情報を提供します。例えば、オペレーティングシステムのバージョンなど、他の製品もカバーされています。

下記も参照

プロジェクトの作成 (ページ 17)

通信の設定 (ページ 21)

プロセス画面の設定 (ページ 47)

フェイスプレートタイプとタグ構造の設定 (ページ 103)

値のアーカイブと表示 (ページ 143)

メッセージの設定 (ページ 179)









アイコン

概要

このセクションでは、「スタートガイド」で使用するアイコンに関する情報を説明します。

使用するアイコン

以下のアイコンを使用して、画像により手順の個別ステップを示します。

アイコン	意味
	マウスの左ボタンでクリック
	マウスの右ボタンでクリック
	マウスの左ボタンでダブルクリック
	キーボードでテキストを入力
	左マウスボタンを押し続けます(ドラッグアンドドロップ)
	左マウスボタンを押し続けてドラッグします(ドラッグアンドドロップ)
	ドラッグした後、マウスの左ボタンを放します(ドラッグアンドドロップ)
	個々の操作ステップのナンバリング

WinCC の操作

概要

WinCC はモジュラシステムです:

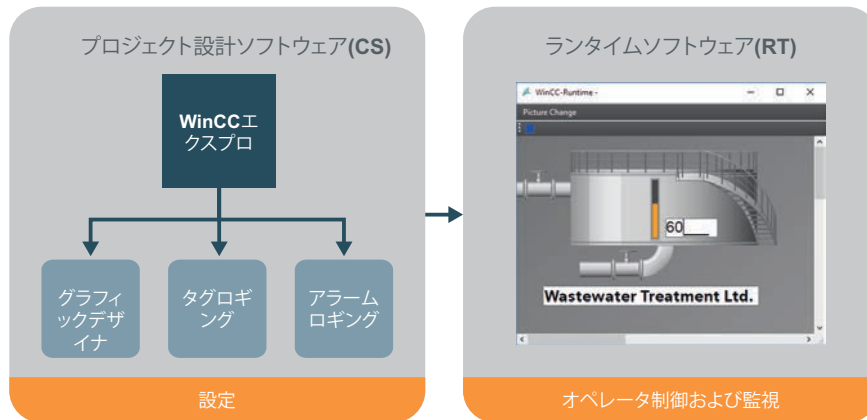
- WinCC を使用して、プロセスを視覚化し、グラフィカルユーザーインターフェースを設定します。
- プロセスのオペレータ制御および監視用ユーザーインターフェースを使用します。

WinCC の機能

- WinCC のプロセスをグラフィック表示します。
この理由で、プロセスは画面上にグラフィック表示されます。
プロセスの状態が変更されると、すぐに表示が更新されます。
- WinCC のプロセスを制御します。
例えば、ユーザーインターフェースからセットポイントを指定したり、バルブを開いたりすることができます。
- WinCC のプロセスを監視します。
アラームは、危機的なプロセス状態が発生すると、自動的にトリガされます。
例えば、事前定義された値を超過した場合は、画面にメッセージが表示されます。
- WinCC のプロセスをアーカイブします。
WinCC を使用すると、デジタルにメッセージをアーカイブできるだけでなく、値を処理したり、PDF として出力したりすることができます。
それにより、プロセス履歴を文書化し、過去の生産データにアクセスできます。

WinCC のコンポーネント

WinCC の基本コンポーネントは、設定ソフトウェア(CS)およびランタイムソフトウェア(RT)です。



プロジェクト設計ソフトウェア(CS)

WinCC エクスプローラはプロジェクト設計ソフトウェアの核となっています。

WinCC エクスプローラでは、プロジェクト構造全体が表示され、プロジェクトが管理されます。

WinCC エクスプローラからさまざまなエディタを開くことができます。各エディタは、WinCC のサブシステムに属しています。

WinCC の最も重要なサブシステムは、次のとおりです:

サブシステム	エディタ	ファンクション
グラフィックシステム	グラフィックデザイナー	設定画像
アラームロギング	アラームロギング	メッセージの設定
アーカイブシステム	タグロギング	アーカイブデータ
レポートシステム	レポートデザイナー	レイアウトの作成
ユーザー管理	ユーザー管理者	ユーザーおよびユーザー権限の管理
通信	タグ管理	通信の設定

WinCC Runtime ソフトウェア(RT)

WinCC Runtime は、プロセスモードでプロジェクトを実行します。すると、プロジェクトはランタイムになります。

WinCC Runtime は、プロセスのオペレータ制御および監視を有効にします。

ランタイムシステムには、特に以下のタスクが含まれます。

- CS データベースに保存されている設定データの読み取り
- 画面でのプロセス画像の表示
- AS との通信
- プロセス値やアラームイベントのような現在のランタイムデータのアーカイブ
- スルーセットポイント指定または電源オン/オフのようなプロセスの制御

プロジェクトの作成

概要

このセクションでは、WinCC エクスプローラのプロジェクトの作成方法について説明します。

プロジェクトは、WinCC のユーザーインターフェースの設定の基本です。プロジェクト内で、プロセスを操作、監視する必要がある全てのオブジェクトを作成、編集します。

原理

プロジェクトは WinCC エクスプローラで作成します。

WinCC エクスプローラは WinCC の設定システムです。WinCC エクスプローラを使用して、プロジェクトを管理します。

4.1 "Quick_Start"プロジェクトの作成

概要

次のステップは、WinCC の起動方法や"Quick_Start"プロジェクトの作成方法を示しています。

"Quick_Start"プロジェクトは、"シングルユーザープロジェクト"として作成します。

「シングルユーザープロジェクト」は、1 台の PC でのみ実行されます。他のコンピュータはこのプロジェクトにアクセスできません。

プロジェクトは、データを処理するサーバー、およびオペレータステーションとして動作する 1 台のコンピュータ上で実行されます。

WinCC エクスプローラの起動

初めて WinCC を起動すると、[WinCC エクスプローラ]が開きます。このダイアログで、プロジェクトのタイプを選択するか、既存のプロジェクトを開きます。

次に WinCC を起動するときは、前回作業を行なったプロジェクトが開きます。

WinCC を終了した時にプロジェクトが起動されていた場合、プロジェクトは有効な状態で再起動されます。

必要条件

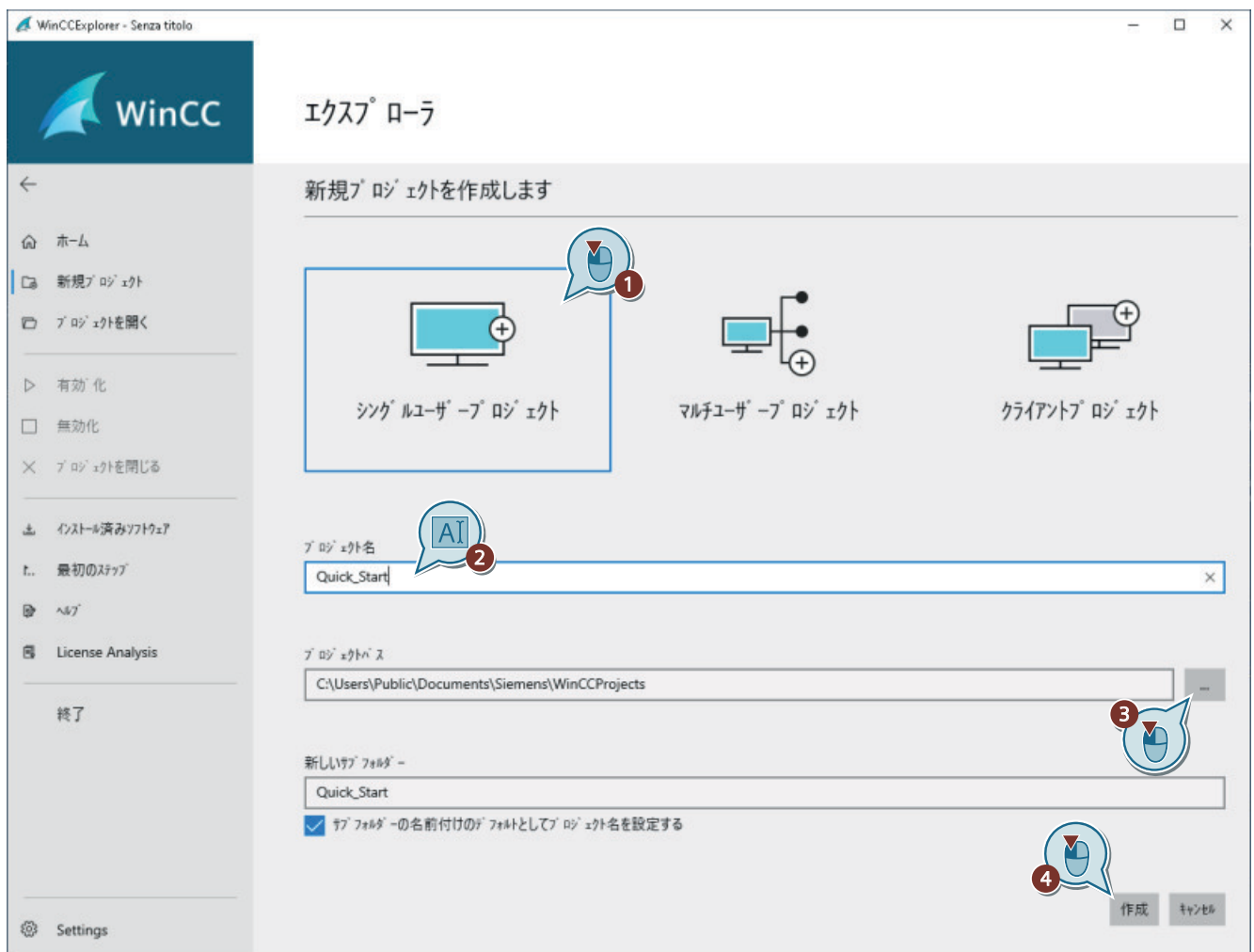
- WinCC の「標準」バリエーションが最低限インストールされていること。

手順

1. WinCC を起動します。
必要な場合、<Windows+R>を使用して入力画面を開きます。



2. [ようこそ]画面で、[シングルユーザープロジェクト]を選択し、プロジェクト情報を入力します。
[新規サブフォルダー]と[プロジェクトパス]のフィールドでデフォルト設定を受け入れるか、PCで別のパスを選択します。



4.1 "Quick_Start"プロジェクトの作成

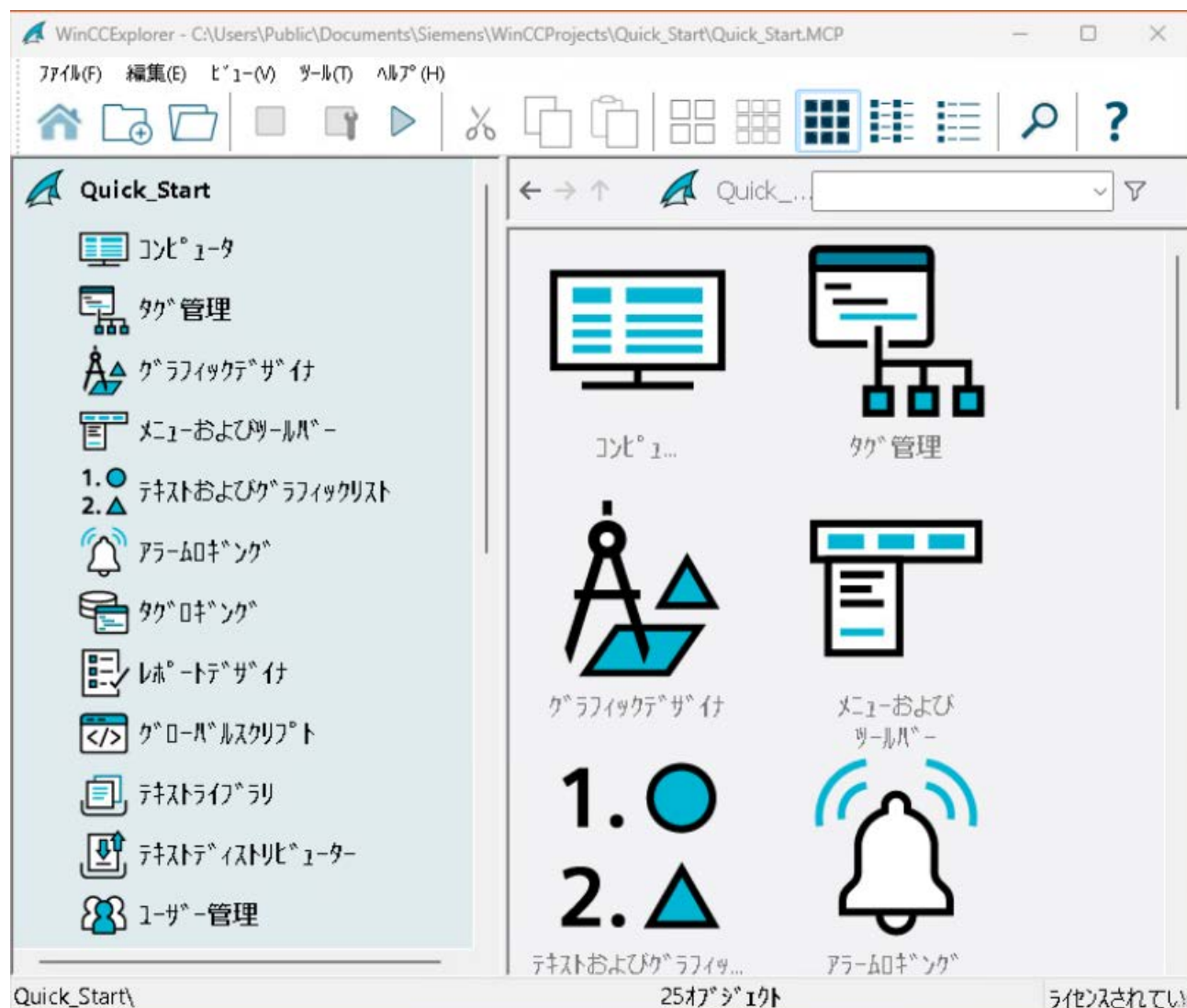
結果

"Quick_Start"プロジェクトが作成されています。

WinCC エクスプローラでプロジェクトが開きます。

必要なエディタのあるプロジェクト構造が WinCC エクスプローラの左ペインに表示されます。

左ペインにエディタが選択されると、右ペインにはエディタに属する要素が表示されます。



通信の設定

概要

このセクションでは、WinCC とオートメーションシステム間の通信の設定に関する情報を説明します。

このセクションでは、タグを作成し、スケーリングする方法について説明します。

原理

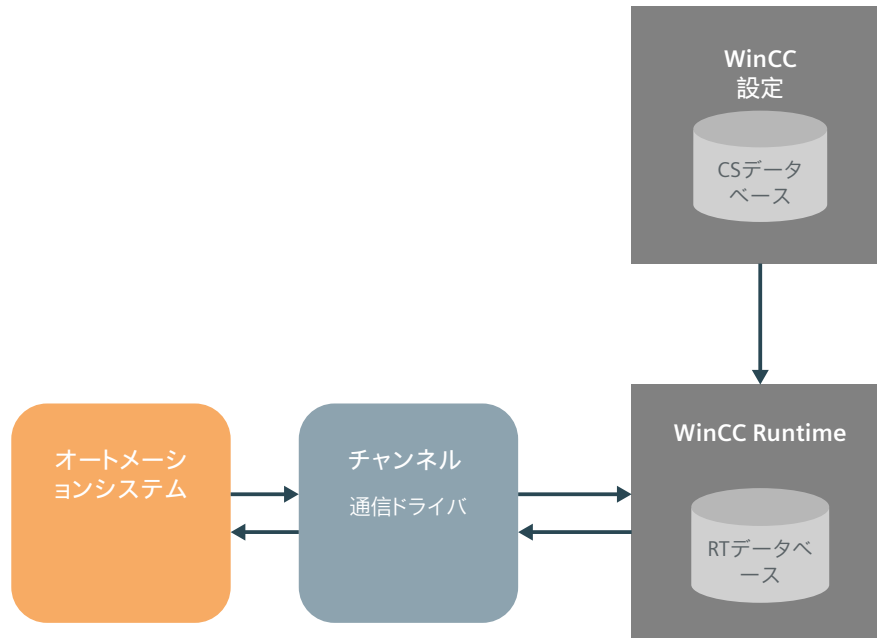
「WinCC Configuration Studio」を使用して通信を設定します。

通信を設定するには、少なくとも以下のコンポーネントが必要です。

- チャンネルユニット付きのチャンネルが1つ
- 1つの接続
- 1つのプロセスタグ

実際には、通信を設定することで AS の現在のプロセス値をアーカイブできるようになります。

WinCC システムでの通信



「Quick_Start」プロジェクトの設定

"Quick_Start"プロジェクトには、AS は必要ありません。

このプロジェクトでは、WinCC 内の内部タグで値が転送されます。

WinCC と AS の間の通信の設定は、プロセスタグのリニアスケールと同様に、実践的な経験を得るための練習です。

5.1 WinCC のチャンネルと接続

概要

チャンネルと接続によって、AS をどのように WinCC に接続するかを指定します。

チャンネル

チャンネルは、特殊な通信ドライバです。

チャンネルによって、AS からプロセスタグにプロセス値を供給することも、その逆を行うこともできます。

WinCC は、異なるオートメーションシステムの接続用の多様なチャンネルを提供します。

チャンネルユニット

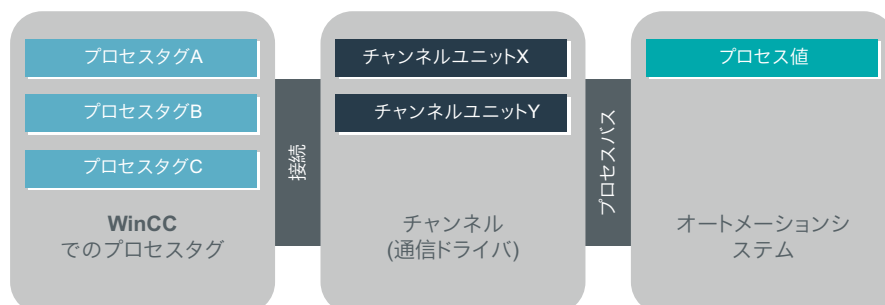
チャンネルには、異なる通信ネットワークのための異なるチャンネルユニットがあります。さらに、このチャンネルユニットは、特定のタイプの AS にアクセスするために使用されます。

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。

接続

チャンネルユニットで、多様なオートメーションシステムへの接続を設定します。

それぞれの接続は、定義された個々の AS へのインターフェースという意味になります。データ交換は接続を介してランタイムで行なわれます。



5.2 WinCC のタグ

概要

WinCC のタグは、外部値または内部値のいずれかを示します。内部値は、WinCC 内で計算またはシミュレーションに使用されます。外部値は、AS で読み込みなどに使用されます。

WinCC は、通信接続およびタグをタグ管理で管理します。

[WinCC Configuration Studio]の[タグ管理]エディタを使って、設定および管理を行います。

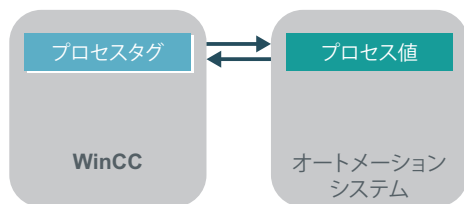
プロセスタグ

外部タグは、WinCC と AS 間のデータ交換用リンクとして機能します。

WinCC のそれぞれの外部タグは、接続された AS のいずれかのメモリ中の特定のプロセス値に対応します。そのため、外部タグはプロセスタグと呼ばれます。

ランタイムで、プロセスタグのプロセス値は WinCC によって決定、入力されます。

WinCC で、プロセスタグの値を決定することもできます。これらの値は、規定されたチャンネルを通じて AS へ転送されます。したがって、AS がプロセスを制御します。



内部タグ

WinCC で内部タグを使用して、プロセスリンクなしで WinCC 内で値を転送します。

タググループ

タググループは、[タグ管理]エディタのコンポーネントです。

タググループは、タグを明確な構造に整理するために使用します。

構造タグ

構造タグは、それぞれのケースで同じプロパティである複数のタググループを作成するために使用できるテンプレートと同じです。

作成される構造は、比較的構造化されているプラントコンポーネントを繰り返しビジュアル化するのに役立ちます。

このために、特定の要素とプロパティの構造タイプを定義します。構造タイプは好きなだけ数の構造タグとリンクされます。それぞれが同じ要素とプロパティとなります。

この方法で、同一のアプリケーションを複数回設定しなければならないことを回避します。

実際には、このような構造は PLC またはサードパーティのアプリケーションからよくインポートされます。そうすると、タグを別々に設定しないで、WinCC で構成を使用できます。

5.3 チャンネルの追加

概要

次のステップは、WinCC でのチャンネルの作成方法を示しています。

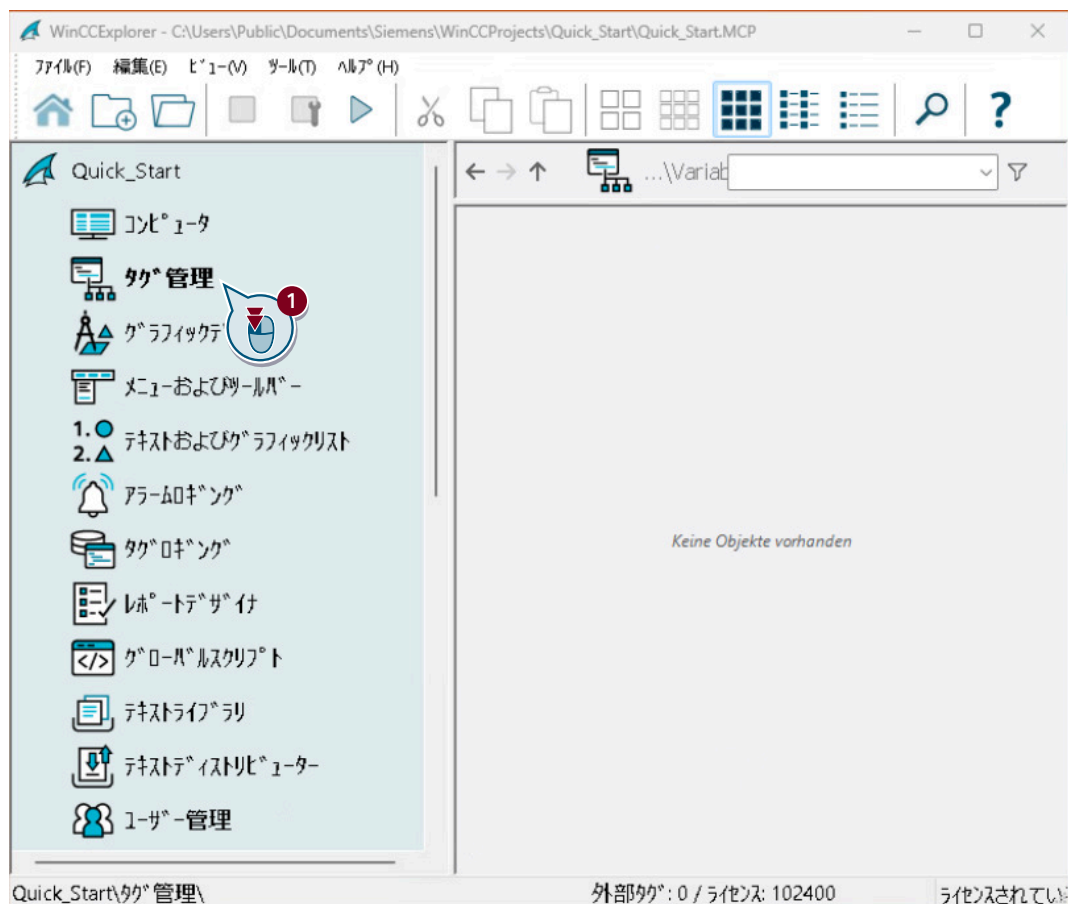
WinCC は、このチャンネルによってオートメーションシステムと通信します。これは、プロセス値を WinCC のプロセスタグにも提供します。

必要条件

- "Quick_Start"プロジェクトが開いている。

手順

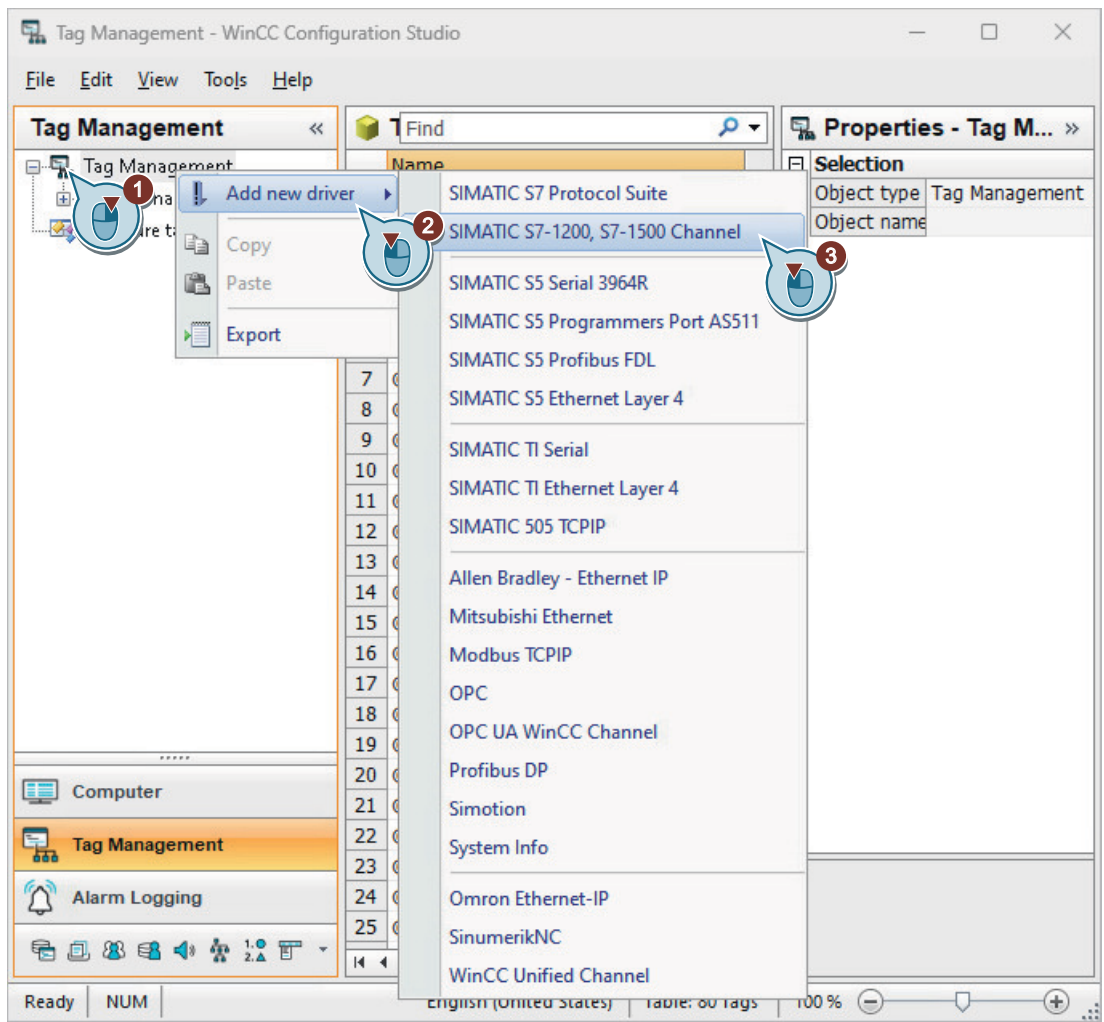
1. [タグ管理]エディタを開きます。



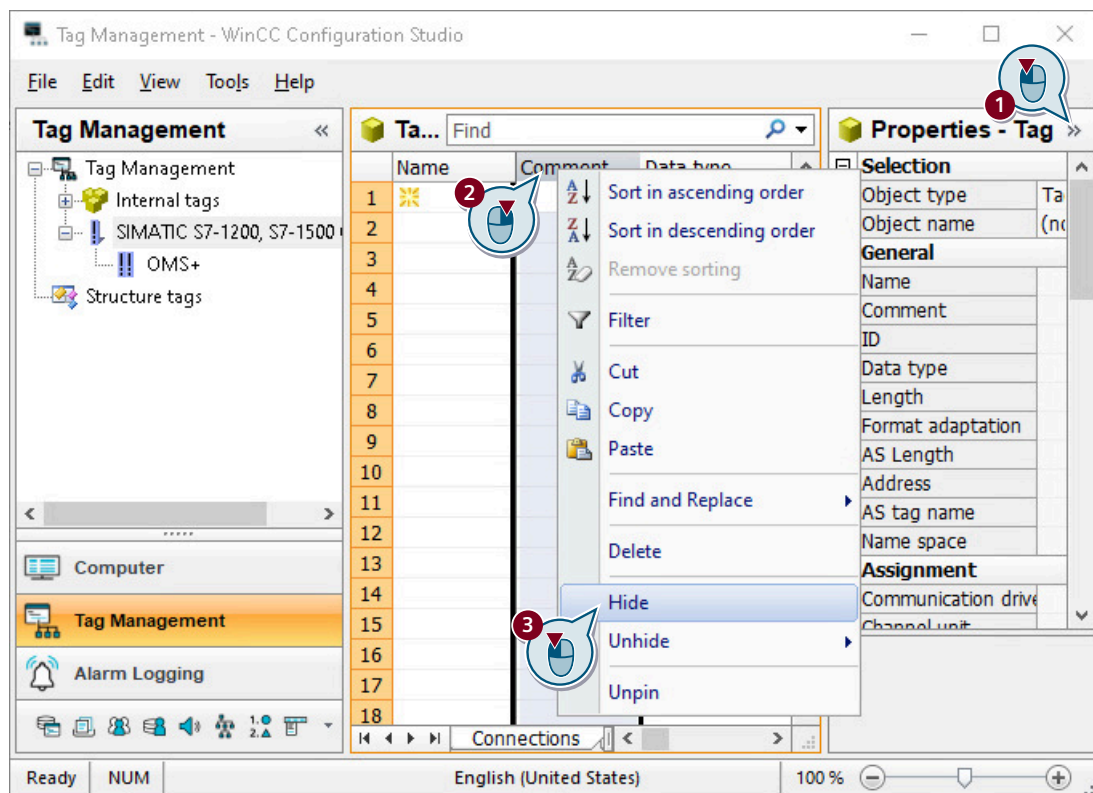
[タグ管理]エディタは、WinCC Configuration Studio で開きます。

2. "SIMATIC S7-1200、S7-1500 チャンネル"を挿入します。

5.3 チャンネルの追加



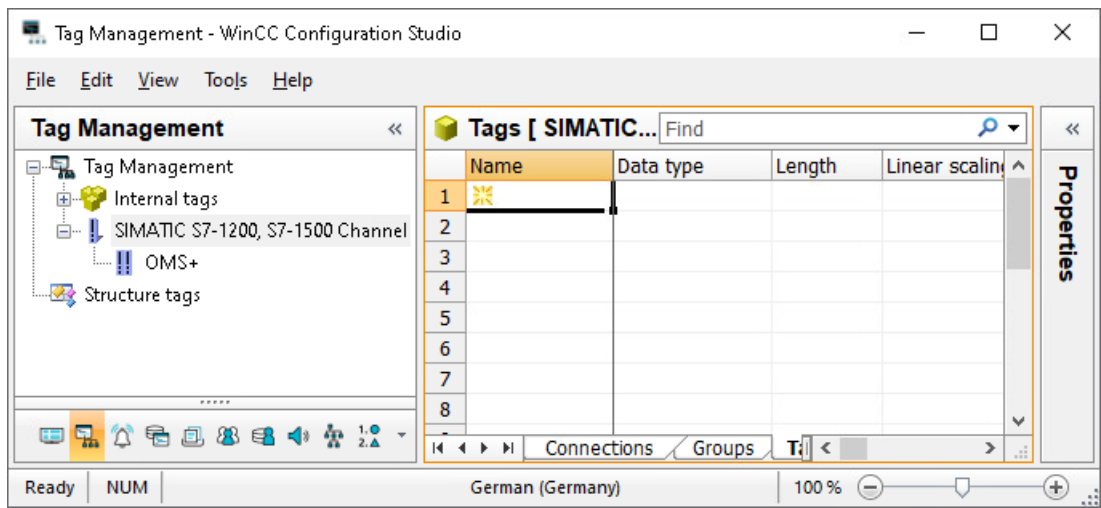
3. データ領域を拡大するには、**»** ボタンを使用して、[プロパティ]ウィンドウのサイズを減らします。
[プロパティ]ウィンドウを再度表示するには、**«** ボタンをクリックします。
4. WinCC Configuration Studio のエディタで、データ領域にある列を非表示および表示することができます。



結果

"SIMATIC S7-1200、S7-1500 チャンネル"がタグ管理に表示されます。
チャンネルユニット「OMS+」がチャンネルの下に表示されています。

5.3 チャンネルの追加



次のステップ

チャンネルユニット「OMS+」の下で AS への接続を作成します。

5.4 接続の作成

概要

次のステップは、チャンネルユニット"OMS+"の下でのオートメーションシステムへの接続の作成方法を示しています。

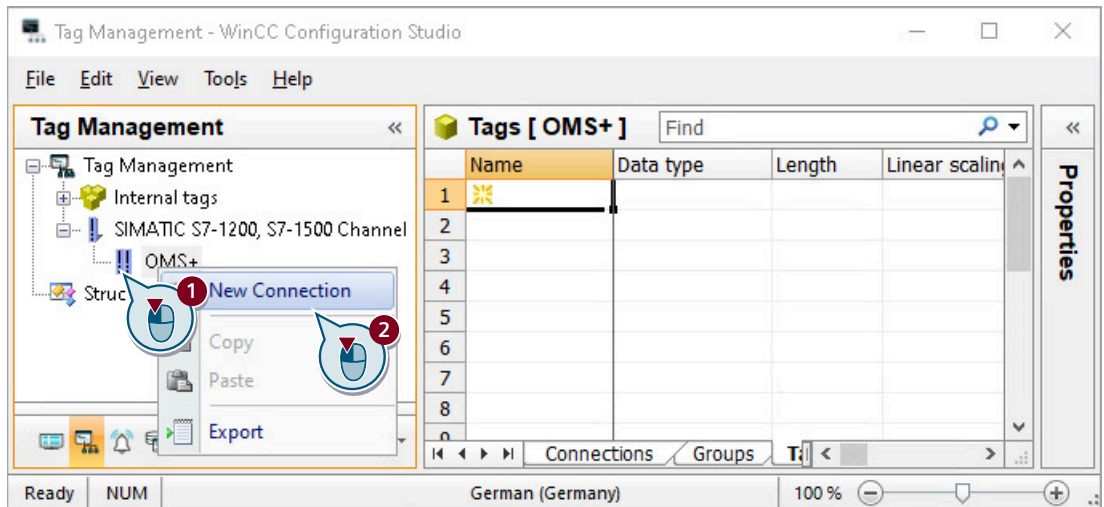
ランタイムには、この接続を通してデータ交換が行なわれます。

必要条件

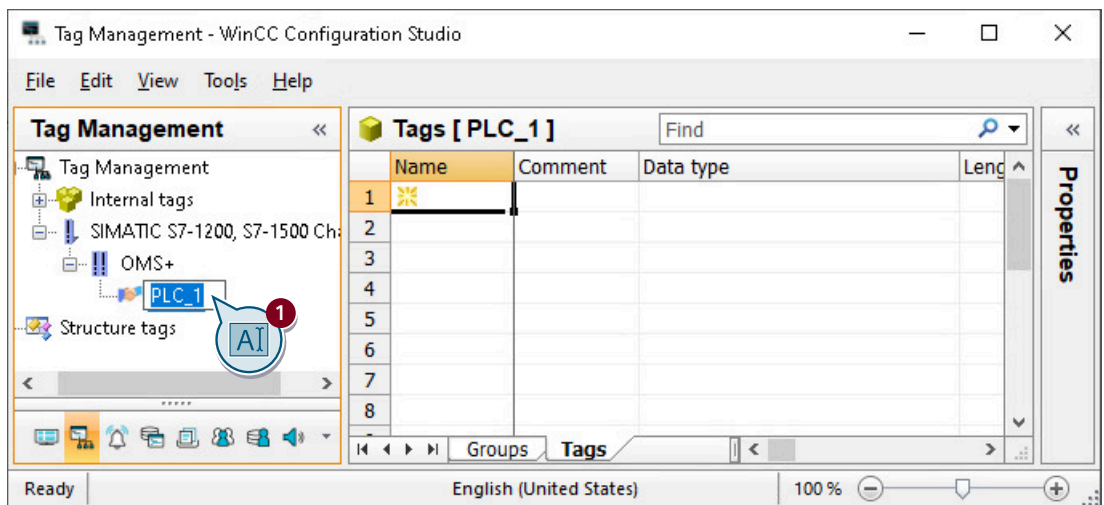
- "SIMATIC S7-1200、S7-1500 チャンネル"がタグ管理に追加されていること。

手順

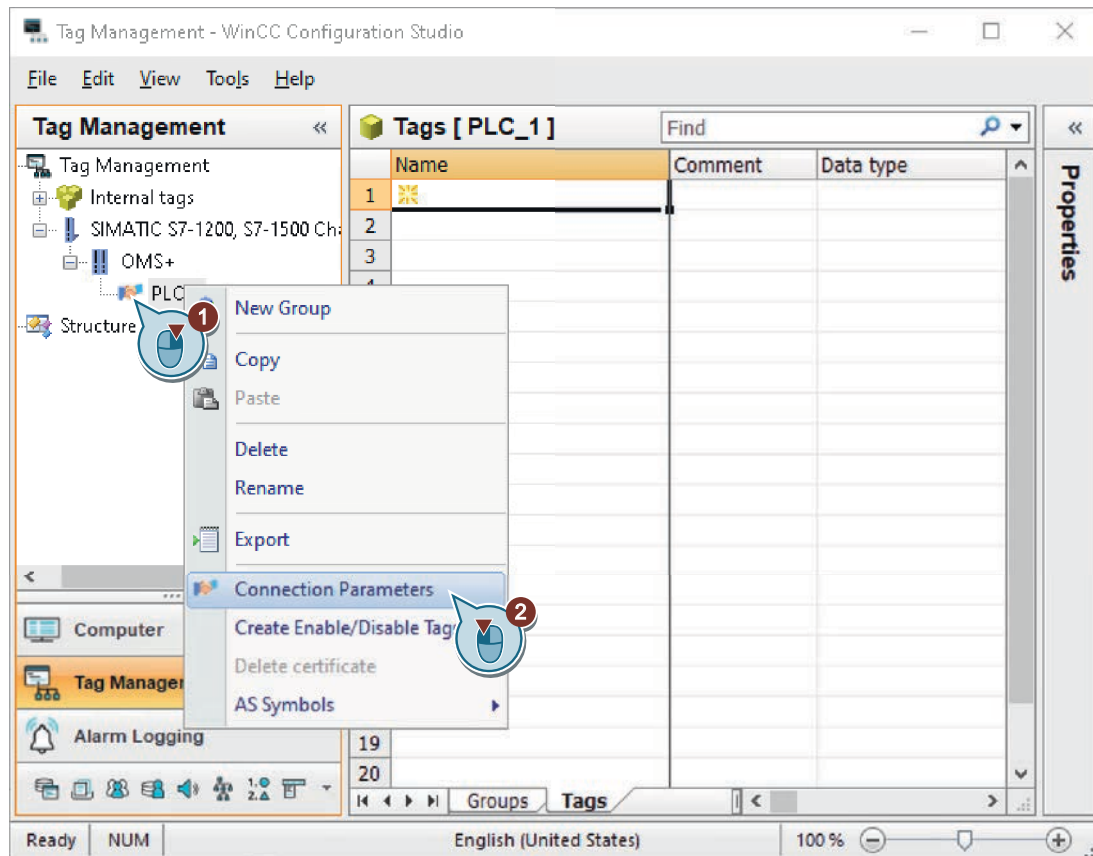
1. チャンネルユニット"OMS+"の下で、新しい接続を作成します。



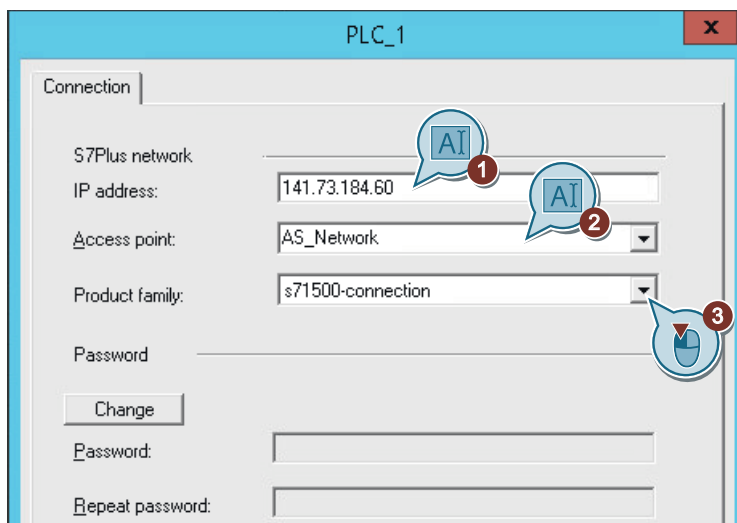
2. 接続名として"PLC_1"と入力します。



3. [接続パラメータ]ダイアログを開いて、接続を設定します。



4. 接続パラメータを入力します。

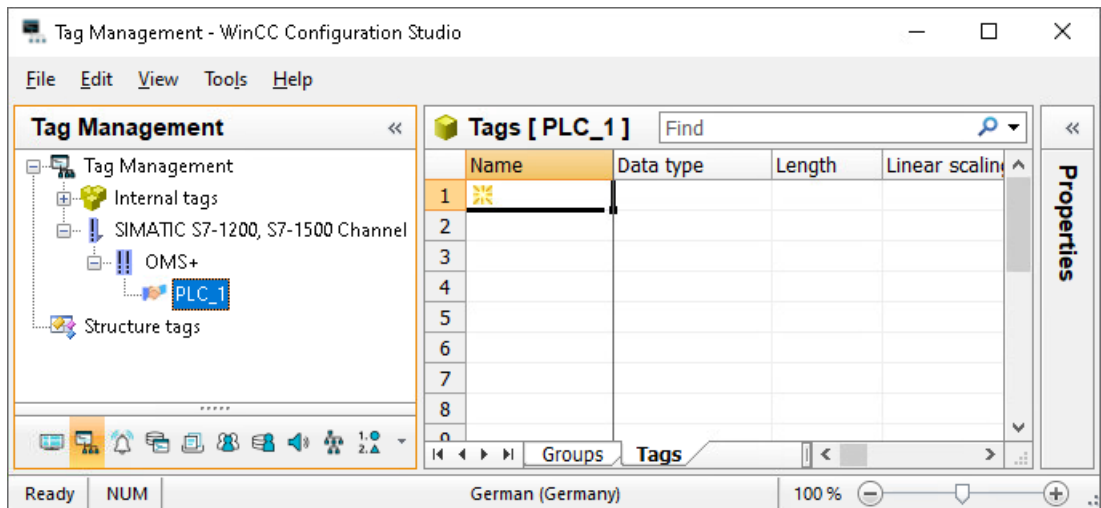


5. [OK]を押してダイアログを閉じます。

結果

オートメーションシステムへの"PLC_1"接続が確立されました。

接続は、タグ管理で表示されました。



次のステップ

次のステップで、[PLC_1]接続の下にタググループおよびタグを作成します。

5.5 タググループのプロセスタグの作成

概要

次のステップは、タググループでのプロセスタグの作成方法と、そのプロパティの決定方法について示しています。

タググループは、[タグ管理]エディタのコンポーネントです。これらのコンポーネントを使用して、構造化タグをソートし、明確に表示することができます。

プロセスタグについて、次のプロパティを指定します。

- 名前
- データタイプ
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。
WinCC のタグのデータタイプは、AS で使用されるデータタイプとは異なります。
- アドレス
アドレス指定によって、AS の特定のデータ範囲をプロセスタグに割り当てます。
アドレス指定のタイプは、通信パートナーのタイプに依存します。

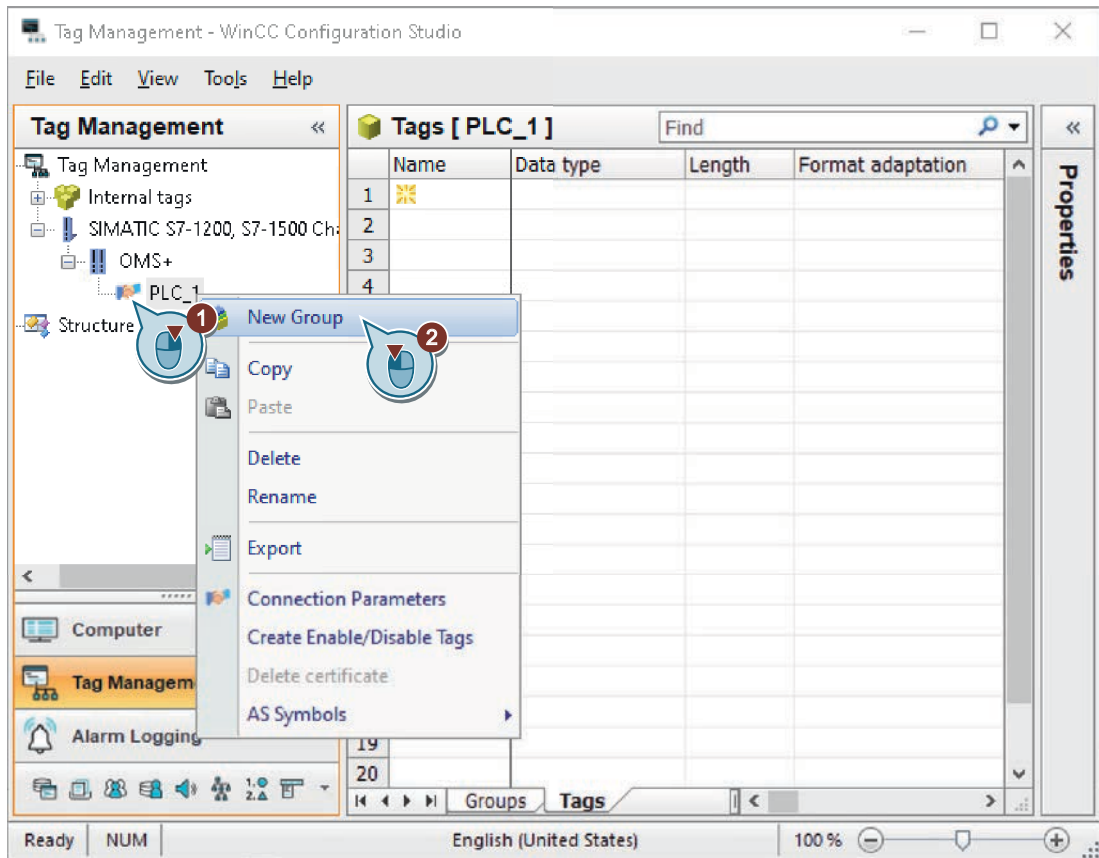
必要条件

- "PLC_1"接続がタグ管理で作成されていること。

5.5 タググループのプロセスタグの作成

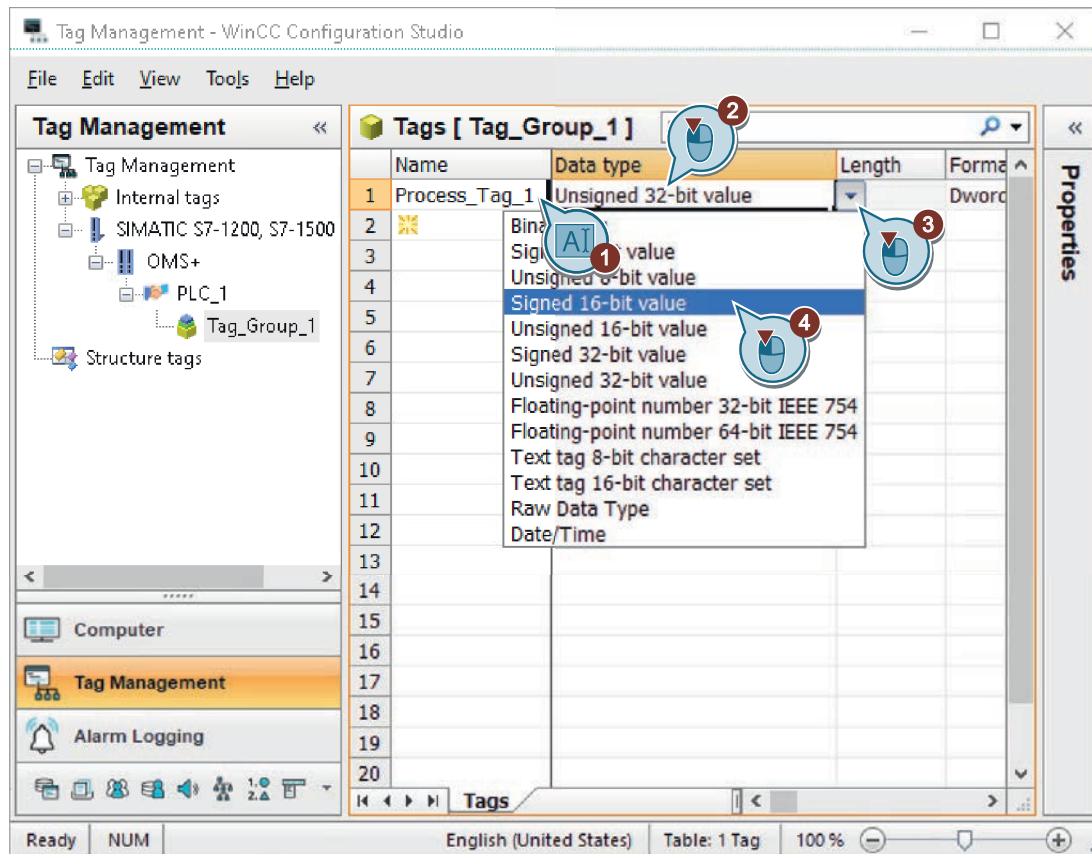
手順

1. タググループを作成します。



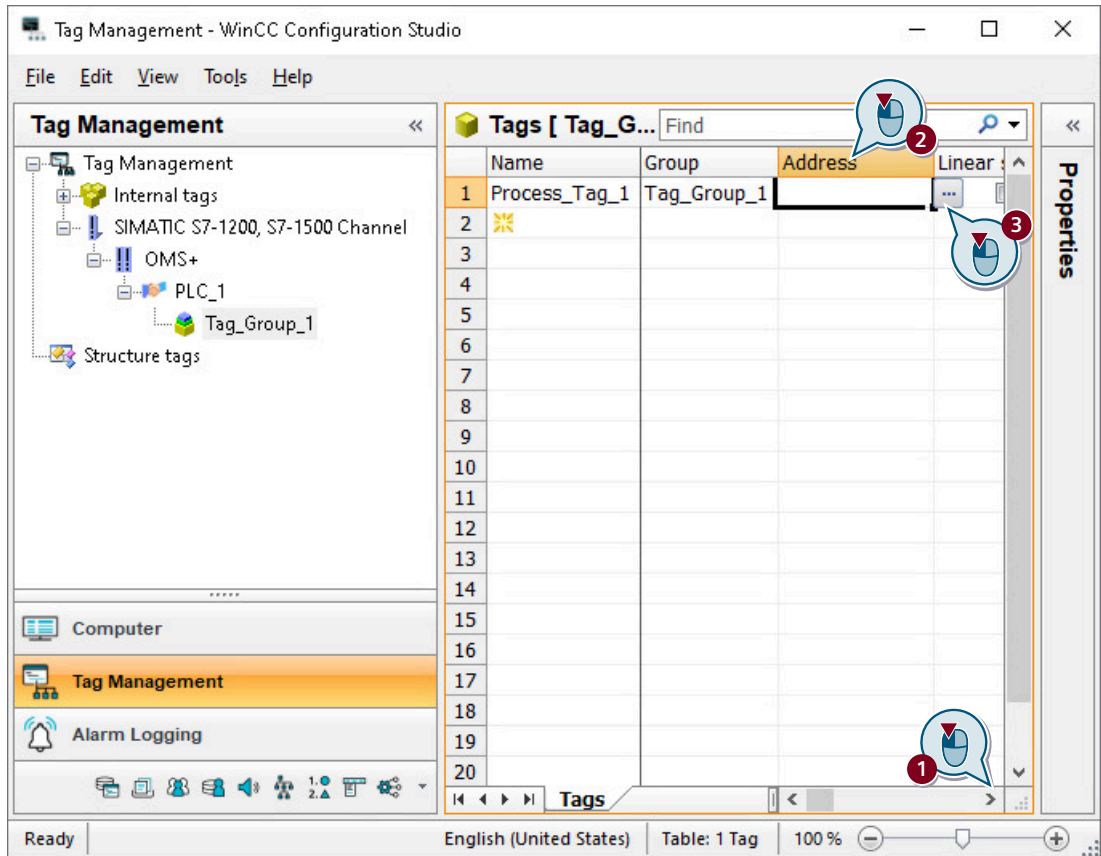
2. タググループ名として"Tag_Group_1"と入力します。
3. タググループの下にプロセスタグを作成するには、プロセスタグ名として、「Process_Tag_1」を[名前]列の一番上にあるフィールドに入力します。
「Process_Tag_1」プロセスタグが作成されました。

4. "符号付き 16 ビット値"データタイプを選択します。



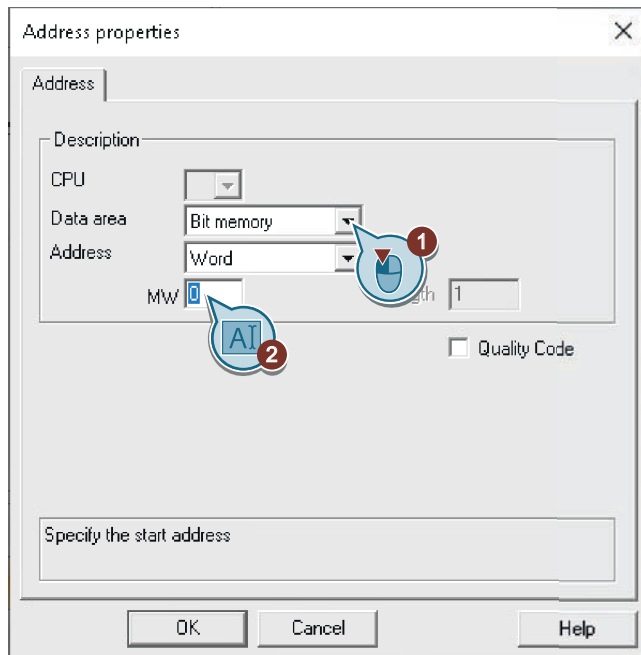
5.5 タググループのプロセスタグの作成

5. [アドレス]欄の[...]をクリックして、タグのアドレス指定のプロパティを指定します。



[アドレスプロパティ]ダイアログボックスが開きます。

6. アドレスの説明に情報を入力します。



7. [OK]を押してダイアログを閉じます。

結果

プロセスタグ"Process_Tag_1"が作成されます。

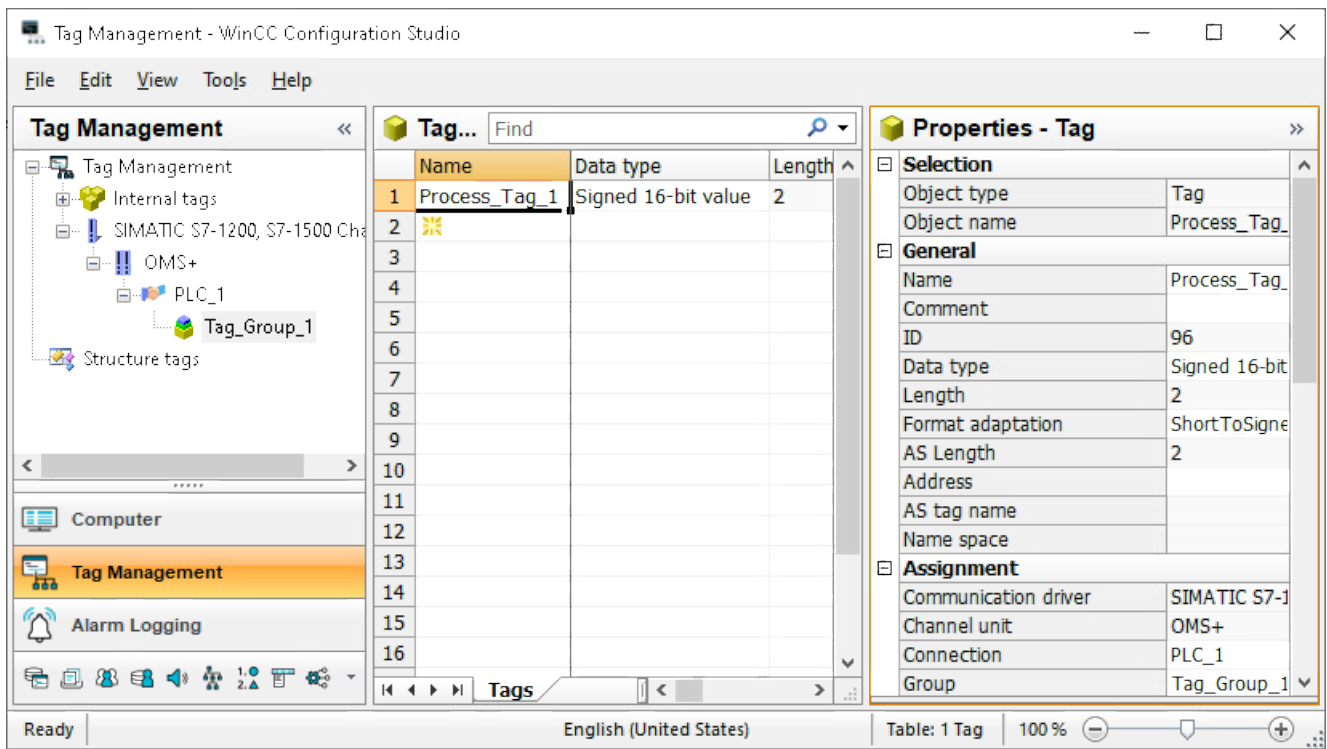
[タグ管理]エディタのテーブルエリアに、プロセスタグが表示されます。

[プロパティ-タグ]領域は、選択されたタグのすべてのプロパティを表示します。

タイプ変換によって、AS のデータフォーマットを WinCC フォーマットに変換できます。

プロセスタグは、WinCC と AS の間の通信を作成するのに必要な最後のコンポーネントです。

5.5 タググループのプロセスタグの作成



次のステップ

WinCC でプロセス値をスケールリングするためのリニアスケールリングのプロパティを定義します。

5.6 WinCC でのプロセスタグのスケールリング

概要

次のステップは、WinCC でのプロセスタグのスケール方法を示しています。

リニアスケールリングを使用して、プロセス値の値範囲をリニアに、WinCC のプロセスタグの定義されている値範囲にマップできます。プロセス値自体は変更されません。

リニアスケールリングについて、次のプロパティを指定します。

- [AS 値の範囲]列にプロセス値の範囲を設定します。
- [OS 値の範囲]列に WinCC のプロセスタグの値の範囲を設定します。

リニアスケールリングは、プロセスタグでのみ使用可能です。内部タグはスケールリングできません。

例:

温度センサはシステム内の抵抗を測定し、値を"オーム"単位に戻します。特定の抵抗値は特定の温度に対応します。

抵抗値を温度スケールに自動変換するためにリニアスケールリング機能を使用することができます。これにより、測定した抵抗値が瞬時にプロジェクトでは温度として表示されるようになります。

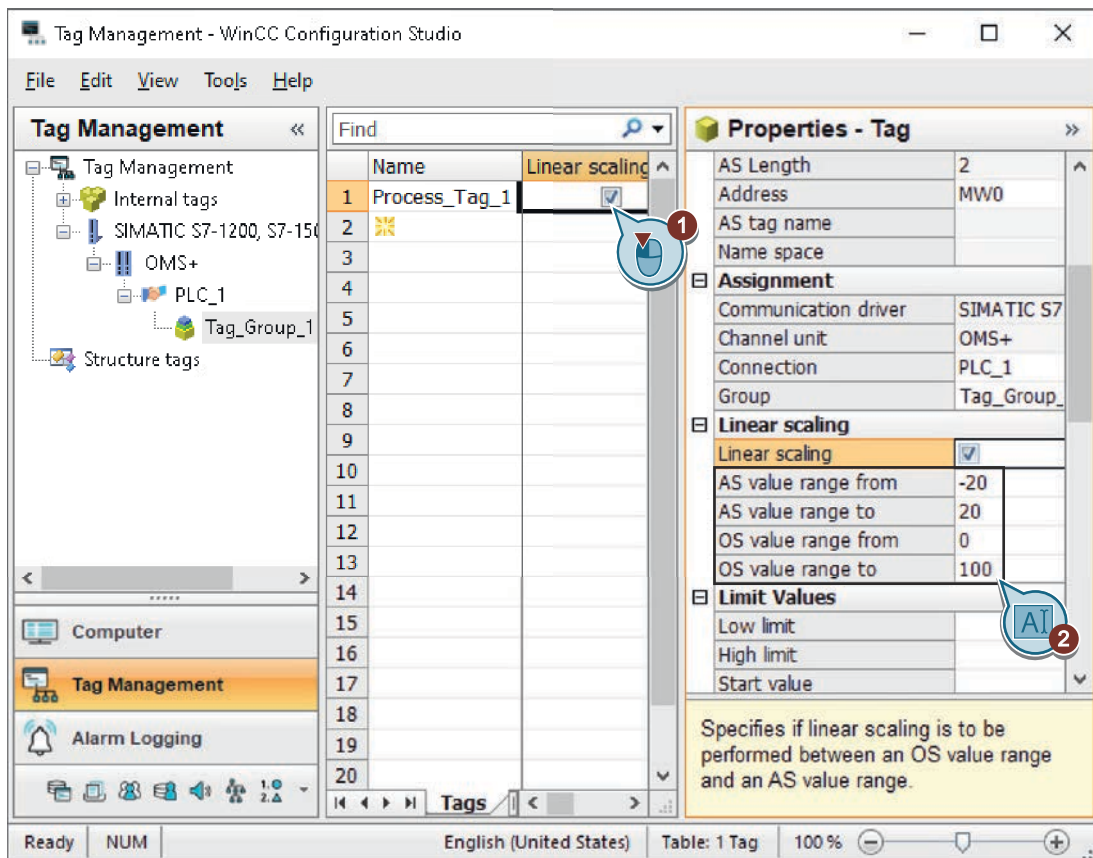
タグ管理では、抵抗値は[AS 値の範囲]列の設定に対応しています。温度は"OS 値の範囲"列の設定に対応しています。

必要条件

- [Process_Tag_1]プロセスタグがタグ管理で作成されていること。

手順

1. リニアスケーリングを有効にし、プロセスとタグの値範囲を定義します。



結果

"Process_Tag_1"プロセスタグのためのリニアスケーリングが有効になります。

プロセスとタグの値範囲が設定されます。プロセス値の値範囲[-20...20]が、プロセスタグの値範囲[0...100]として表示されます。

次のステップ

内部タグを作成します。

5.7 内部タグの作成

概要

次のステップは、タググループでの内部タグの作成方法と、そのプロパティの決定方法について示しています。

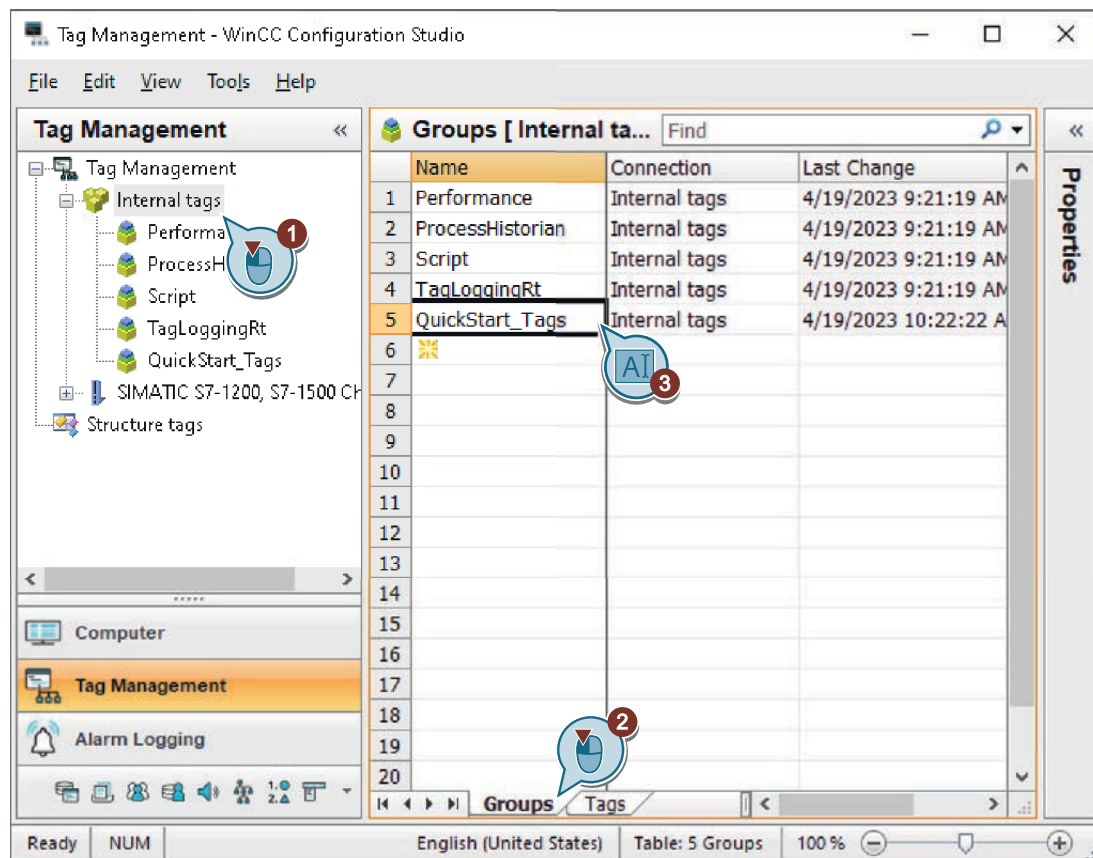
内部タグは、WinCC 内で値を転送するために使用します。

必要条件

- "Quick_Start"プロジェクトが開いている。

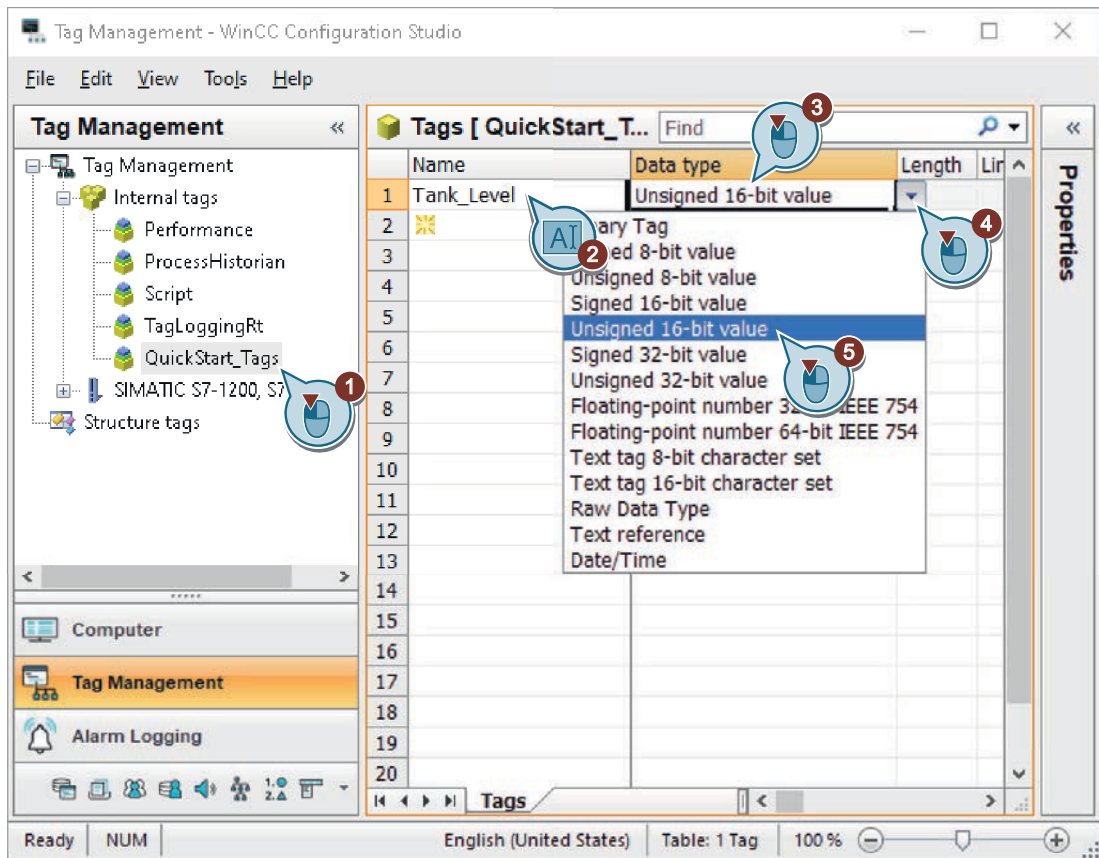
手順

1. タグ管理で、[内部タグ]エントリを選択します。
2. [グループ]タブで[QuickStart_Tags]タググループを作成します。



5.7 内部タグの作成

- 内部タグの名前として"Tank_Level"と入力して、内部タグをタググループで作成します。
- データタイプ[符号なし 16 ビット値]を選択します。



結果

内部タグ[Tank_Level]を作成しました。またそのデータタイプを[QuickStart_Tags]タググループで定義しました。

内部タグ[Tank_Level]がタグ管理で表示されています。

「@」で始まる名前のあるタグはシステムタグです。システムタグは、プロジェクトで使用できますが、変更はできません。

Tag Management - WinCC Configuration Studio

File Edit View Tools Help

Tag Management

- Tag Management
 - Internal tags
 - Performance
 - ProcessHistorian
 - Script
 - TagLoggingRt
 - QuickStart_Tags
 - SIMATIC S7-1200, S7-1500 CH
 - Structure tags

Computer

Tag Management

Alarm Logging

Tags [Internal tags] Find

	Name	Data type	Length	Lib
85	@RedundantServerStat	Unsigned 16-bit value	2	
86	@SCRIPT_COUNT_ACT	Unsigned 32-bit value	4	
87	@SCRIPT_COUNT_REQ	Unsigned 32-bit value	4	
88	@SCRIPT_COUNT_TAG	Unsigned 32-bit value	4	
89	@ServerName	Text tag 16-bit character set 256		
90	@ServerVersion	Text tag 16-bit character set 256		
91	@TLGRT_AVERAGE_TA	Floating-point number 64-bit 8		
92	@TLGRT_SIZEOF_NLL	Floating-point number 64-bit 8		
93	@TLGRT_SIZEOF_NOTI	Floating-point number 64-bit 8		
94	@TLGRT_TAGS_PER S	Floating-point number 64-bit 8		
95	Tank_Level	Unsigned 16-bit value	2	
96	✖			
97				
98				
99				
100				
101				
102				
103				
104				

Properties

Ready NUM English (United States) Table: 95 Tags 100%

プロセス画面の設定

概要

このセクションでは、グラフィックシステムに関する情報やプロセス画像の設定方法を説明します。

プロセス画像はプロジェクトの中心的要素です。プロセス画像はプロセスを表現し、このプロセスに関する操作や監視を可能にします。

原理

[グラフィックデザイナー]エディタを使用して、プロセス画像を設定できます。このエディタは、WinCC のグラフィックシステムの設定コンポーネントです。

各プロセス画像は、1 つ以上のオブジェクトで構成されます。

- スタティックオブジェクトは、ランタイム中に変化しません。
- ダイナミックオブジェクトは、個々のプロセス値に従って変更されます。
ダイナミックオブジェクトの例としては、バーがあります。バーの長さは、現在の温度値に応じて変化します。
- コントロール可能なオブジェクトを使用することで、プロセスに対して能動的な影響を与えることができます。
これらのオブジェクトには、ボタン、スライダ、あるいは特定のプロセスパラメータへの入力に使用する I/O フィールド(入力/出力フィールド)などがあります。

プロジェクトが複数のプロセス画像で構成されることがよくあります。それぞれのプロセス画像は、異なるプロセスステップを表示したり、特殊なプロセスデータを表示します。

「Quick_Start」プロジェクトの設定

このセクションでは、水処理システムを示すプロセス画像「START.Pdl」を作成します。2 番目のプロセス画像「SAMPLE.Pdl」の作成は練習として見なされます。

WinCC は、プロセス画像を設定するために必要なすべてのオブジェクトを提供します。

2つ目のステップで、前もって作成してある内部タブ「Tank_Level」を使用してプロセス画面「START.Pdl」をダイナミック化します。この結果、ウォータータンクの水位がタグ値の機能としてグラフィック表示されます。

6.1 グラフィックシステム

概要

グラフィックシステムは、WinCC のサブシステムです。このサブシステムは、プロセス画像を設定するために使用します。

グラフィックシステムは以下のタスクを処理します。

- テキスト、グラフィックまたはボタンなどのスタティックでオペレータ操作可能なオブジェクトを表示します。
- プロセス値に応じて棒グラフの長さを変更するなど、ダイナミックオブジェクトを更新します。
- ボタンをクリックしたり、テキストボックスにテキストを入力するなどのオペレータ入力に対応します。

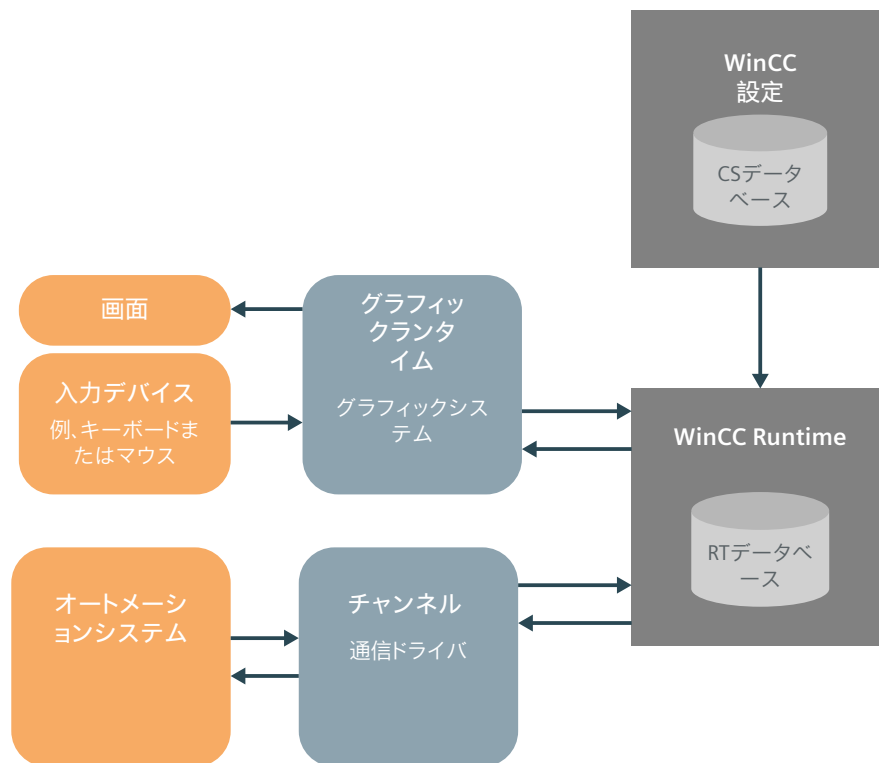
グラフィックシステムのコンポーネント

グラフィックシステムは、設定コンポーネントおよびランタイムコンポーネントで構成されています。

- [グラフィックデザイナー]エディタは、グラフィックシステムの設定コンポーネントです。このエディタで、プロジェクトのプロセス画像を作成します。
- グラフィックランタイムは、グラフィックシステムのランタイムコンポーネントです。グラフィックランタイムはランタイム中に画像を表示し、すべての入力および出力を管理します。

6.1 グラフィックシステム

WinCC システムのグラフィックシステム



6.2 プロセス画面の作成

概要

次のステップでは、WinCC エクスプローラでのプロセス画像の作成方法、名前の変更の方法を示しています。

新しい、空のプロセス画像は、[グラフィックデザイナー]エディタまたは WinCC エクスプローラを使用して作成することができます。

- すぐにプロセス画像を編集する場合は、[グラフィックデザイナー]エディタを使用して作成します。
- 処理する前に必要なプロセス画像をすべて作成しておく場合は、WinCC エクスプローラを使用することを推奨します。

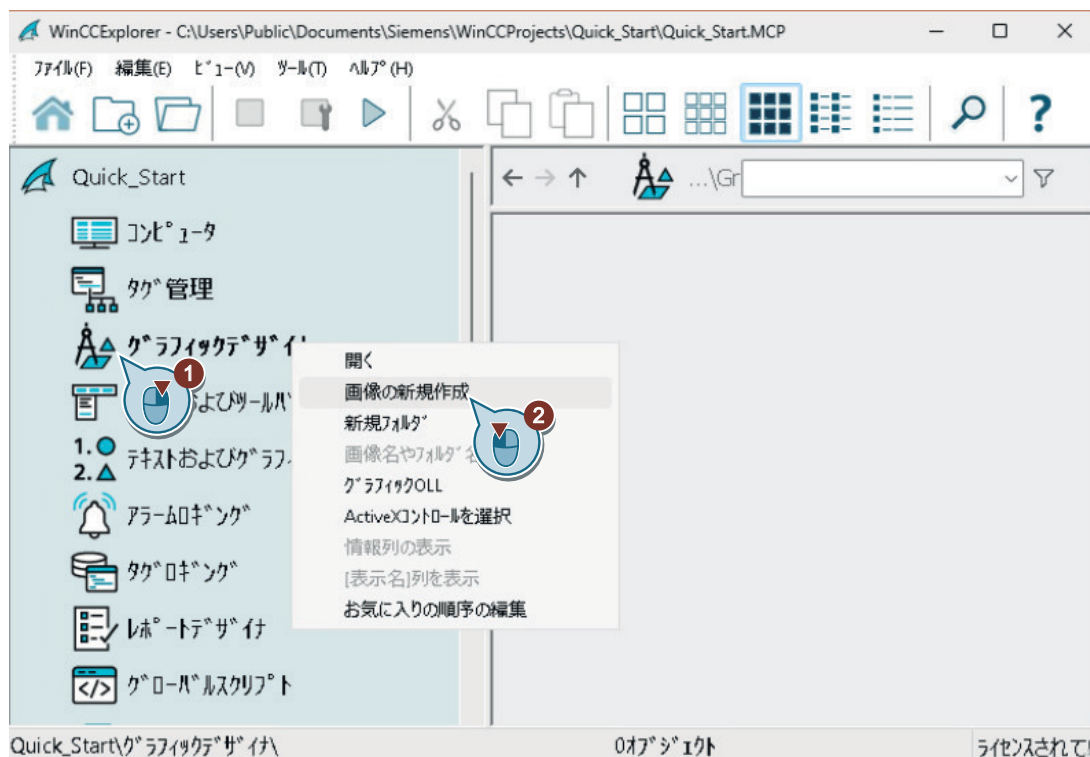
必要条件

- "Quick_Start"プロジェクトが開いている。

6.2 プロセス画面の作成

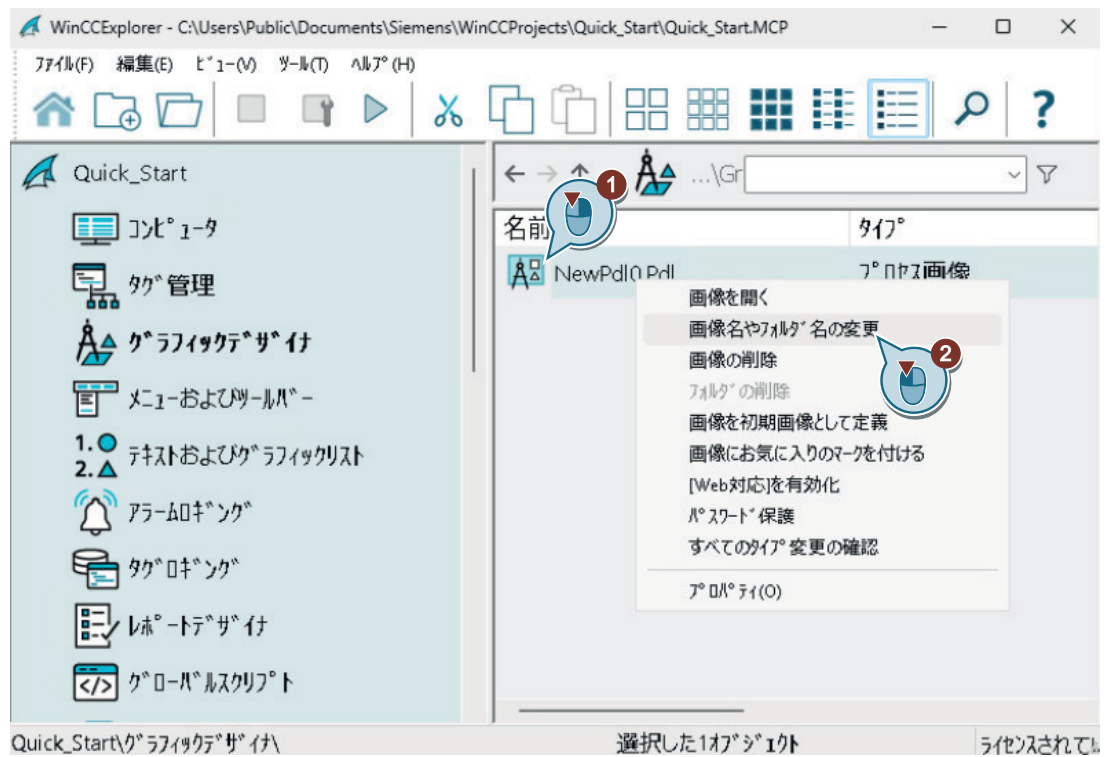
手順

1. 新しいプロセス画像を作成します。



プロセス画像は、WinCC エクスプローラの右ペインに表示されます。

2. 作成したプロセス画像の名前を「START.Pdl」に名前変更します。



「START.Pdl」プロセス画像は、WinCC エクスプローラの右ペインに表示されます。

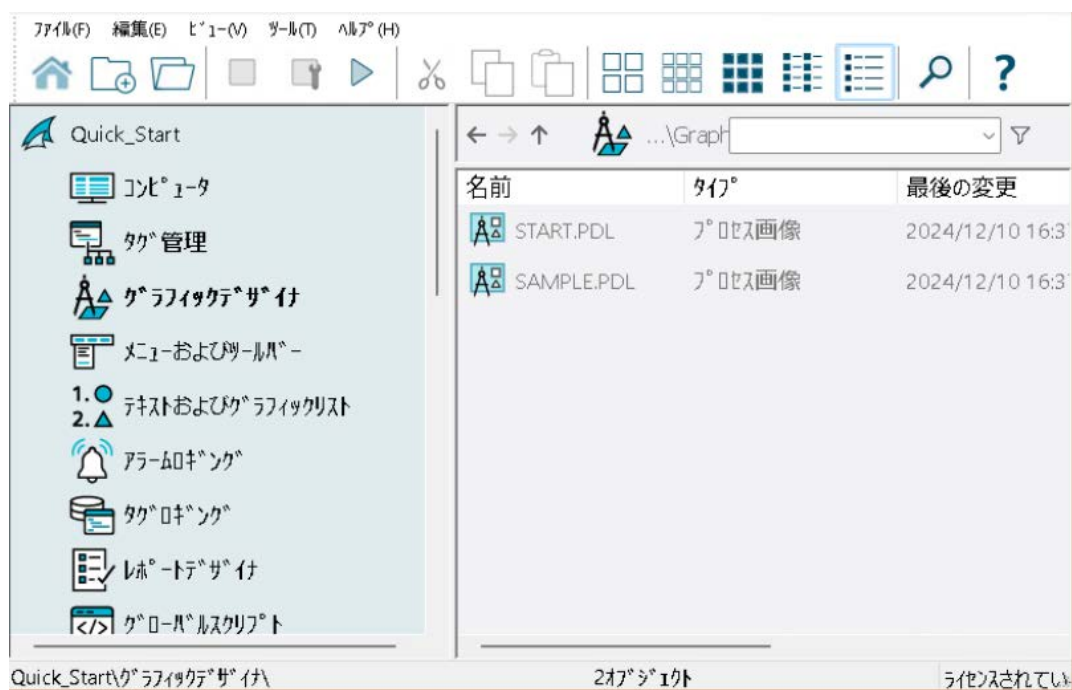
3. ステップ 1 およびステップ 2 と同様にして 2 番目のプロセス画像を作成し、それに「SAMPLE.Pdl」という名前を付けます。

結果

プロセス画像「START.Pdl」および「SAMPLE.Pdl」が作成されます。

これらは、WinCC エクスプローラの右ペインに表示されます。

6.2 プロセス画面の作成



次のステップ

「START.Pdl」プロセス画像に複数のオブジェクトを挿入することにより、水処理システムを画像で示します。

6.3 プロセス画面の編集

概要

このセクションでは、[グラフィックデザイナー]エディタを使用したプロセス画像の編集方法を説明します。

このセクションの最後には、自分の「SAMPLE.Pdl」画像を編集して、さらに深く学べる機会があります。

原理

[グラフィックデザイナー]エディタを使用して、プロセス画像を設定できます。このエディタの構造は描画プログラムと似ており、同じような方法で操作できます。

[グラフィックデザイナー]エディタは、プロセス画面を設定するためのオブジェクトとツールを提供します。「Quick_Start」プロジェクトに対しては、主としてオブジェクトパレットや[グラフィックデザイナー]エディタのライブラリを使用します。

オブジェクトパレットには、プロセス画像の処理で頻繁に必要となる様々なタイプのオブジェクトがあります。

オブジェクトパレットのオブジェクトを、ドラッグおよびドロップでプロセス画像に挿入します。または、オブジェクトを選択してから、プロセス画像の作業エリアでマウスの左ボタンを一度クリックします。

オブジェクトパレットには、画像を設定する以下のタイプのオブジェクトがあります。

- 標準オブジェクト:線、多角形、楕円、円、長方形、スタティックテキストなど
- スマートオブジェクト:アプリケーションウィンドウ、画像ウィンドウ、OLE オブジェクト、I/O フィールド、バー、ステータス表示など
- Windows オブジェクト:ボタン、チェックボックス、オプショングループ、スライダオブジェクトなど
- チューブオブジェクト:例、多角形チューブ、Tピース、ダブルTピース、チューブ曲がり
- コントロール:[コントロール]タブには、最も重要な ActiveX コントロールがあります。コントロールオブジェクトを使用して、追加のコントロールを統合することができます。

ライブラリを使用して画像を効率よく作成できます。ライブラリには、ドラッグとドロップを使用して画像に挿入できるグラフィックオブジェクトがあります。

6.3 プロセス画面の編集

[グラフィックデザイナー]エディタで、外部のグラフィックプログラムのグラフィックをインポートすることも可能です。

6.3.1 ライブラリからのグラフィックオブジェクトの挿入

概要

次のステップは、SVG ライブラリからのグラフィックオブジェクトをプロセス画像「START.Pdl」に挿入する方法を示しています。

SVG ライブラリは、[グラフィックデザイナー]エディタのコンポーネントです。

このコンポーネントは、グラフィックオブジェクトを保存、管理する多用途のツールです。ライブラリは、2つのエリアに分かれています。

- グローバルライブラリ
[グローバル SVG ライブラリ]エリアは、ディレクトリツリーの構造です。マシンやシステムのパーツ、測定デバイス、操作コントロールやビルなどの様々な事前に作成されたグラフィックオブジェクトを提供します。

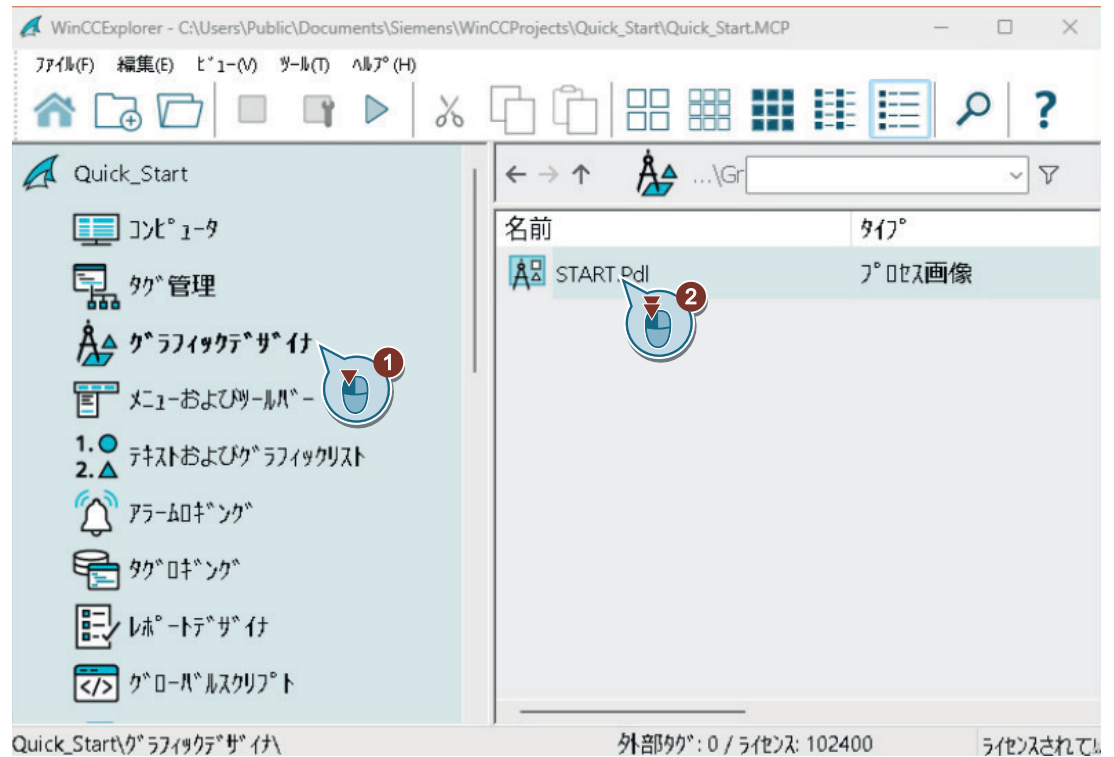
- プロジェクトライブラリ
[プロジェクト SVG ライブラリ]エリアには、カスタムプロジェクトを保存できます。

"Quick_Start"プロジェクトでは、[グローバル SVG ライブラリ]エリアのグラフィックオブジェクトだけが必要です。

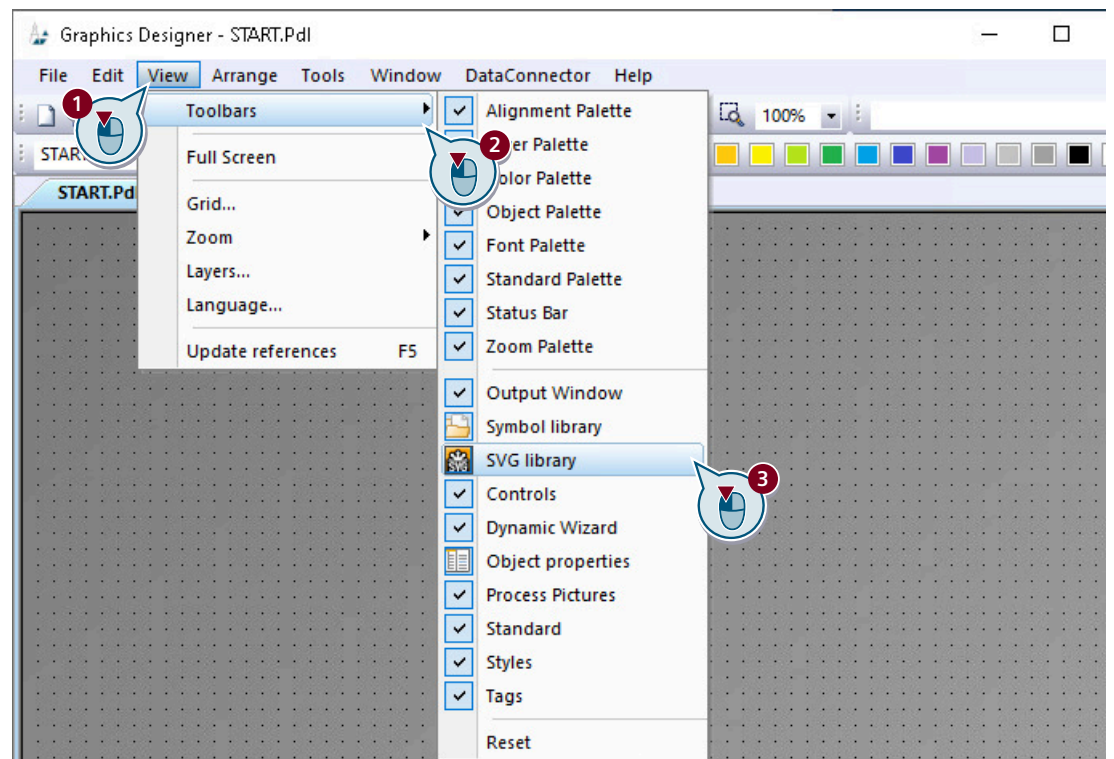
これらのオブジェクトを使用して、「START.Pdl」プロセス画像で水処理システムを示します。

手順

1. [グラフィックデザイナー]エディタで、「START.Pdl」プロセス画像を開きます。



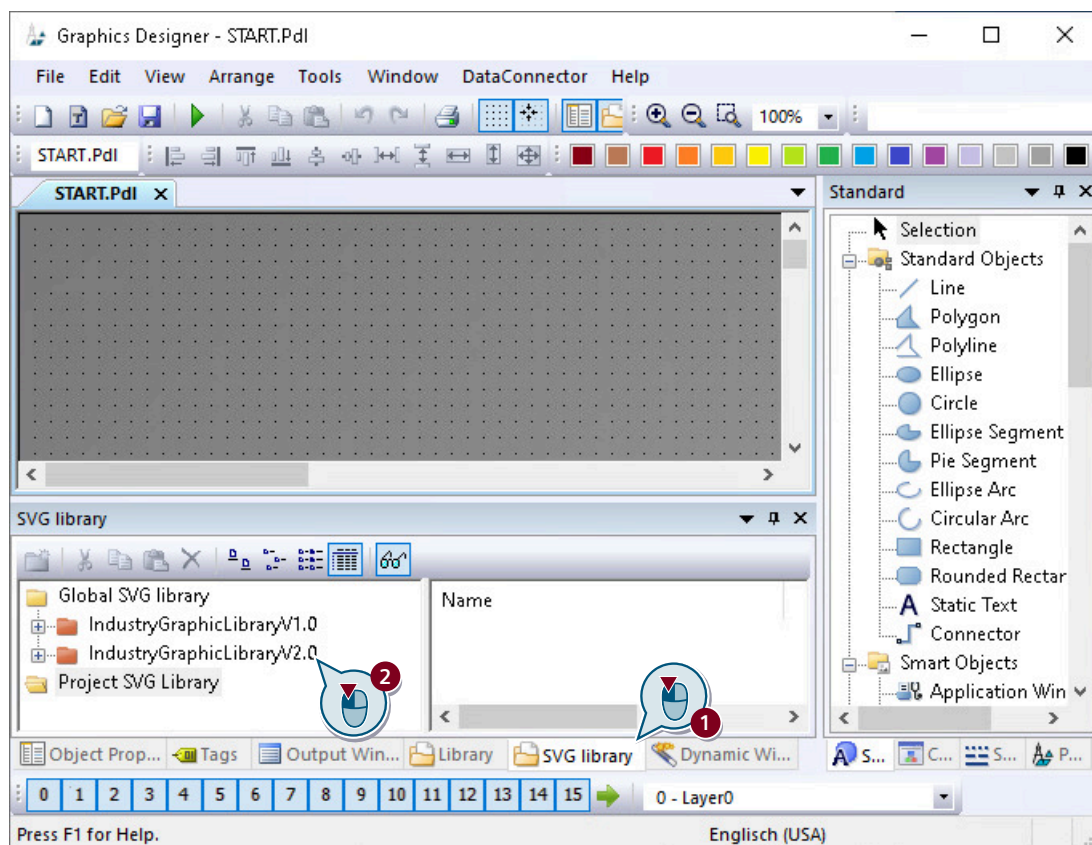
2. [SVG ライブラリ]ウィンドウが表示されない場合、ビューを有効にします。






6.3 プロセス画面の編集

[SVG ライブラリ]ウィンドウが開きます。

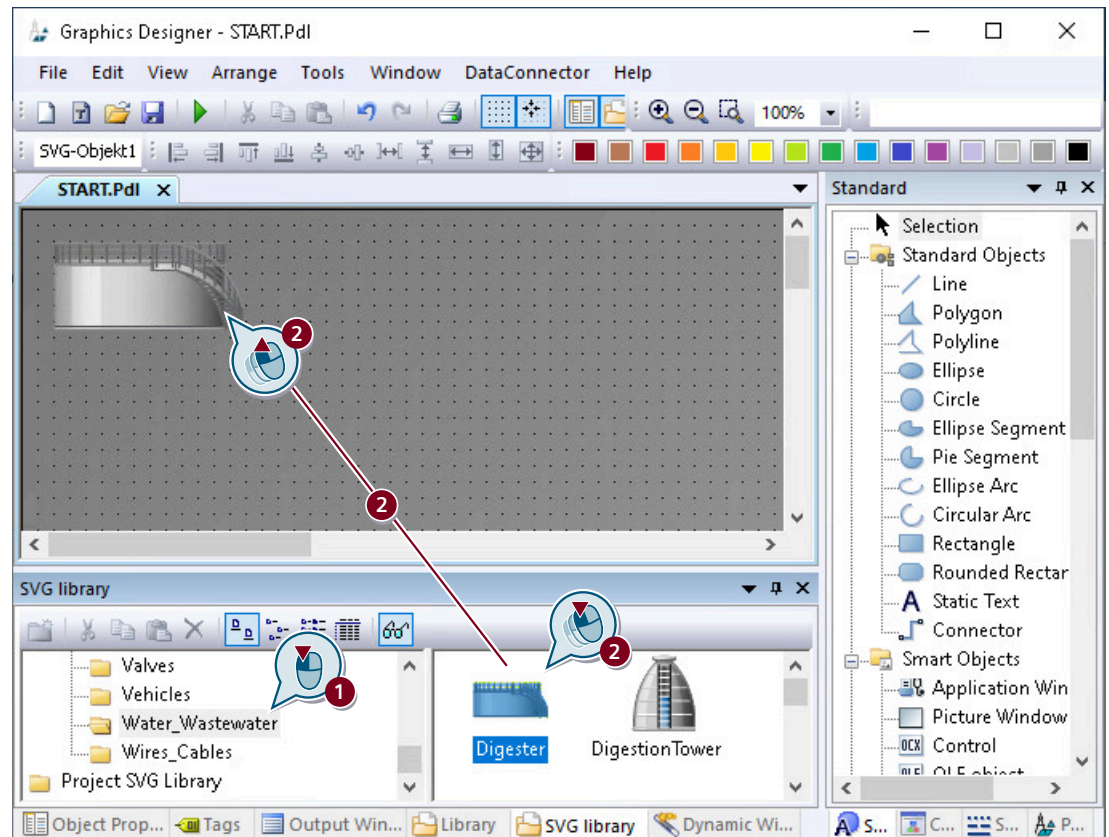
3. グローバル SVG ライブラリ "IndustryGraphicLibraryV2.0" を [SVG ライブラリ] ウィンドウで選択します。



ライブラリのツールバーの  ボタンを使用して、使用可能なオブジェクトのプレビューを表示します。

 と  ボタンを使用して表示されたシンボルのサイズを変更することができます。

4. [Digester]オブジェクトを[Water_Wastewater]フォルダから挿入します。



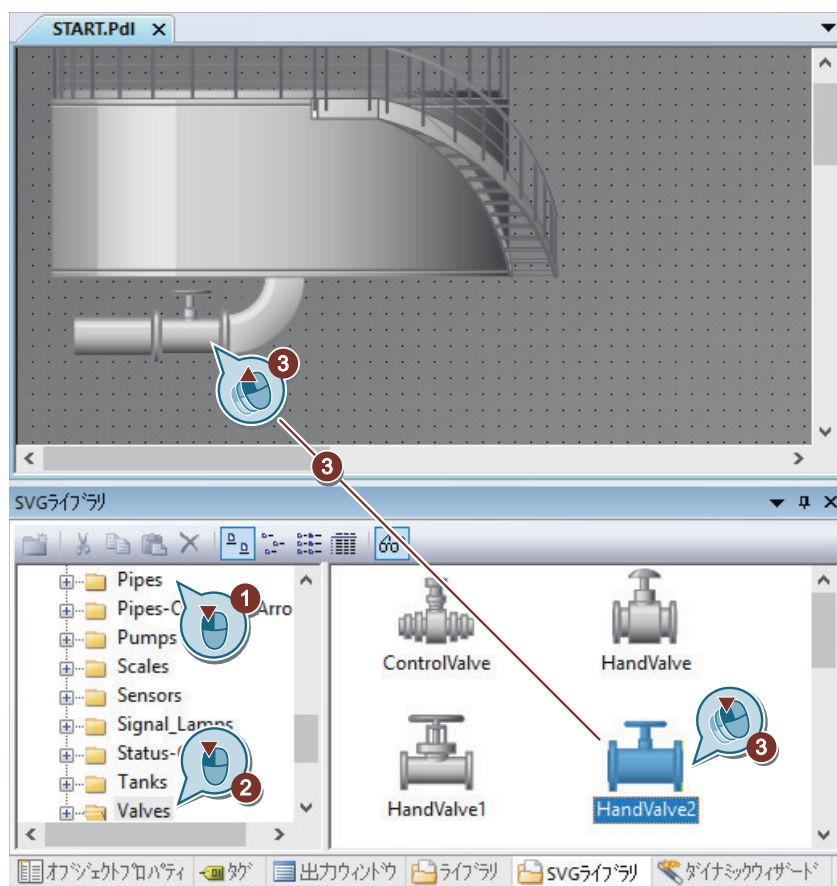
バイオコンバータの図が、作業エリアに表示されます。

マウスで表示されたポイントのいずれかをクリックアンドドラッグして画像を拡大します。

5. 他のオブジェクトの表示を追加します(例、パイプラインやバルブ)。

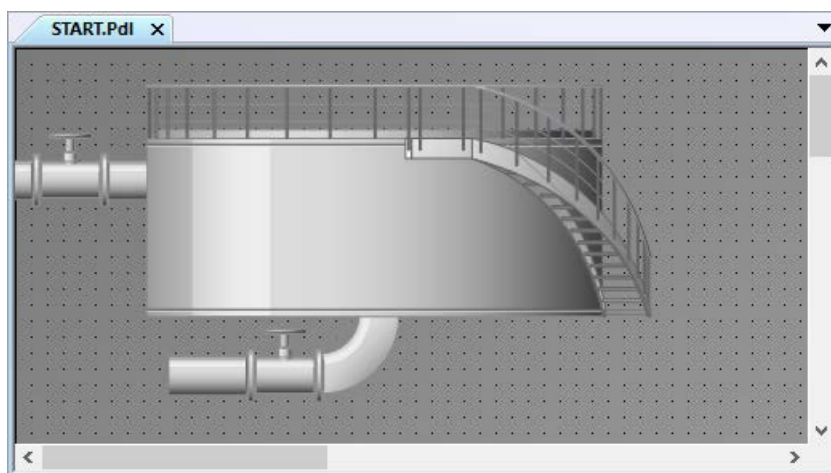
マウスまたはカーソルキーを使用して、プロセス画像上でオブジェクトの画像を移動します。

6.3 プロセス画面の編集



結果

水処理システムが描画されました。



次のステップ

プロセス画像で「静的テキスト」オブジェクトを挿入することにより、示されたプロセスにラベルを付けます。

6.3.2 [スタティックテキスト]グラフィックオブジェクトの挿入

概要

次のステップは、"スタティックテキスト"オブジェクトを挿入および編集する方法を示しています。

"スタティックテキスト"オブジェクトは、ランタイムで変化しないテキストフィールドです。

"Quick_Start"プロジェクトでは、表示されたプロセスにラベル付けするスタティックテキストが必要です。複数の画像を作成する場合は、示されたプロセスに明確なラベルを付けることが重要です。

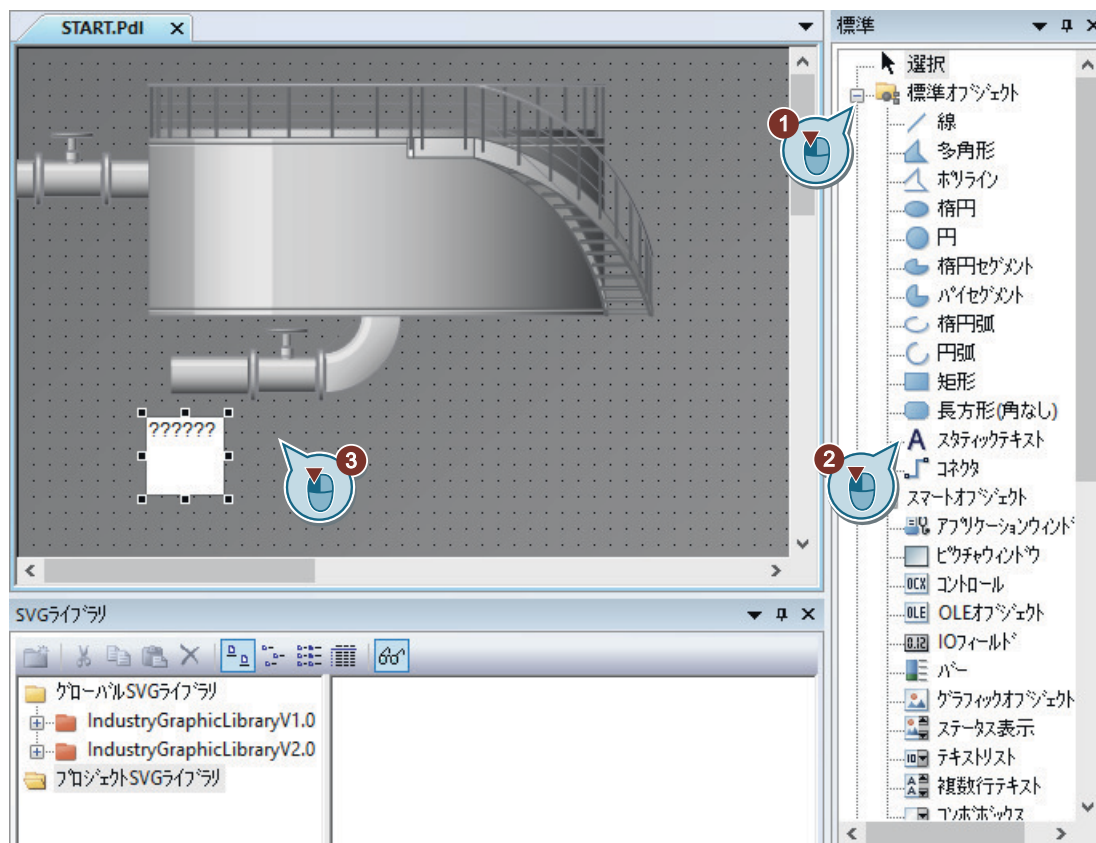
必要条件

- [グラフィックデザイナー]エディタで、プロセス画像「START.Pdl」が開いている。

6.3 プロセス画面の編集

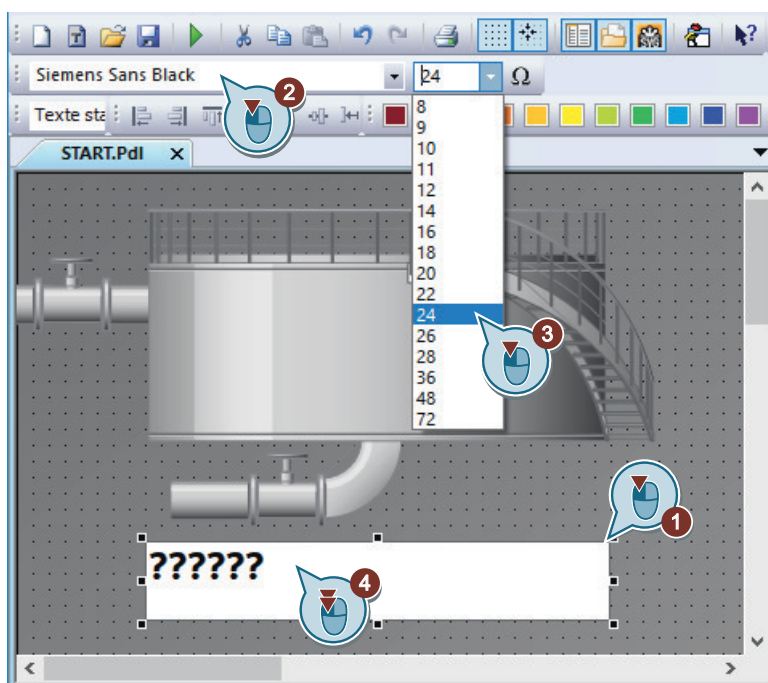
手順


1. [スタティックテキスト]オブジェクトを挿入するには、[標準]ウィンドウでオブジェクトをクリックしてから、プロセス画像をクリックします。



テキストフィールドが、プロセス画像に表示されます。

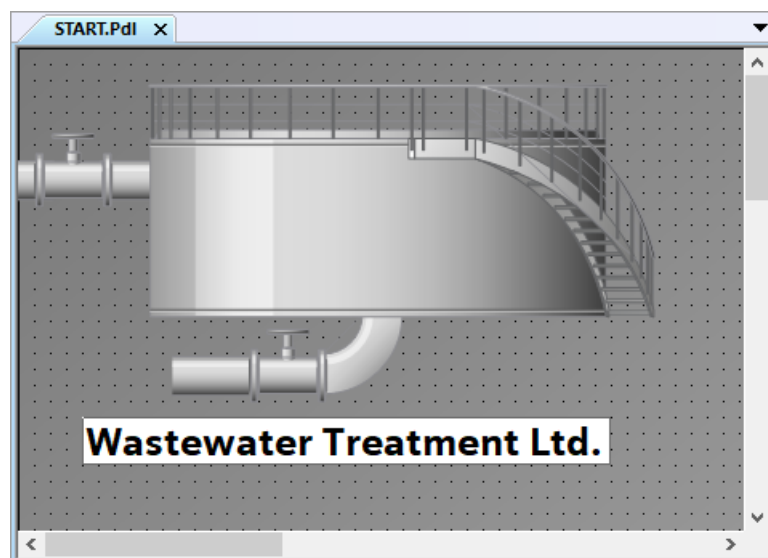
2. マウスを使用してテキストボックスをドラッグし、フォントタイプとサイズを設定します。



3. テキストフィールド内をダブルクリックし、キーボードを使用して「Wastewater Treatment Ltd」などのタイトルを入力します。
4. テキストに合わせて、テキストボックスのサイズを調整します。
5. ツールバーの  ボタンを使用して、プロセス画像「START.Pdl」を保存します。
6. 「START.Pdl」プロセス画像を閉じます。

結果

スタティックテキストフィールドが追加され、表示されたプロセスにラベルが付きます。



6.3 プロセス画面の編集

次のステップ

- プロセス画像の作成と編集をもっと練習する場合、追加のプロセス画像「SAMPLE.Pdl」を作成するオプションがあります。
- 入力フィールドを挿入し、内部タグを使用してそれをバイオコンバーターのグラフィックにリンクすることにより、「START.Pdl」プロセス画像でプラントをダイナミック化します。

6.3.3 「SAMPLE.Pdl」プロセス画像の編集

概要

次のステップで、「SAMPLE.Pdl」プロセス画像を編集します。



編集中に、ライブラリのグラフィックオブジェクトを使用してプロセスを表示します。

プロセスの表示は自由練習です。"Quick_Start"プロジェクトでは、このステップは必要ありません。

必要条件

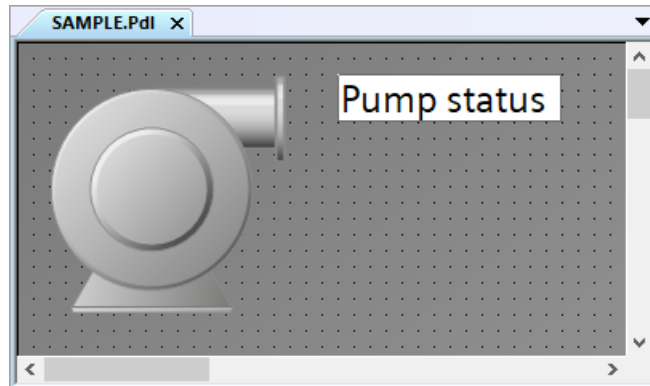
- 「SAMPLE.Pdl」プロセス画像が作成されている。
- [グラフィックデザイナー]エディタが開いている。

手順

1. [グラフィックデザイナー]エディタのツールバーの  ボタンを使用して、「SAMPLE.Pdl」プロセス画像を開きます。
2. ライブラリのグラフィックオブジェクトを使用して、例えば、ステータスディスプレイでポンプを示します。
3. ツールバーの  ボタンを使用して、プロセス画像「SAMPLE.Pdl」を保存します。
4. 「SAMPLE.Pdl」プロセス画像を閉じます。

結果

「SAMPLE.Pdl」プロセス画像を編集しました。



下記も参照

ライブラリからのグラフィックオブジェクトの挿入 (ページ 56)

6.4 プロセス画面のダイナミック化

概要

このセクションでは、プロセス画像をダイナミック化する方法を説明します。

原理

「Quick_Start」プロジェクトで、直接タグ接続によって「START.Pdl」プロセス画像をダイナミック化します。

タグ接続

直接タグ接続では、1つのタグをプロセス画像のダイナミックオブジェクトに接続します。ランタイムでタグが値を取ると、この値はダイナミックオブジェクトに直接転送されます。オブジェクトの動的表示は、タグ値に従ってランタイムで変化します。

実際には、プロセス画像のダイナミックオブジェクトはプロセスタグに接続されます。WinCC とオートメーションシステム(AS)の間に接続がある場合、AS はプロセスタグに値を提供します。ダイナミックオブジェクトは、ラインタイムでプロセス値の変化を示します。

[グラフィックデザイナー]エディタで、AS に値を転送するオブジェクトを設定することもできます。AS は、転送された値を使用して、プロセスを制御します。

内部タグを使用したダイナミック化

"Quick_Start"プロジェクトには、AS は必要ありません。このプロジェクトでは、内部タグ [Tank_Level] をバイオコンバータのグラフィック画像に接続します。

内部タグに値を指定するには、I/O フィールドを設定します。I/O フィールドは、タグ値を表示、変更するために使用する入力および出力フィールドです。

ランタイムに I/O フィールドの値を入力すると、この値は内部タグ「Tank_Level」によって受け取られます。内部タグは、入力された値をバイオコンバータを表すグラフィックオブジェクトに転送します。充填レベルインジケータが、タグ値に従って変化します。

WinCC Runtime の実行

プロジェクトが実行されたときに、WinCC Runtime が起動します。

WinCC Runtime は、プロジェクトをプロセスモードで実行します。すると、プロジェクトはランタイムになります。

ランタイムのプロセスのオペレータコントロールおよびモニタリングを実行します。WinCC エクスプローラでランタイムのプロパティを指定します。

6.4.1 フィルレベルインジケータのダイナミック化

概要

次のステップは、バイオコンバータの充填レベルインジケータをダイナミック化する方法を示しています。

フィルレベルインジケータのダイナミック化には、以下のステップが含まれます。

- バイオコンバータのグラフィック画像と内部タグ[Tank_Level]の接続
内部タグ[Tank_Level]への接続によって、バイオコンバータを表すグラフィックオブジェクトにタグ値を転送できます。充填レベルインジケータが、ランタイムでタグ値に従って変化します。
タグとオブジェクト間のリンクが、💡アイコンと太字で"オブジェクトプロパティ"ウィンドウに表示されます。
- 更新サイクルの指定
更新サイクルは、フィルレベルインジケータを更新する時間間隔を決定します。

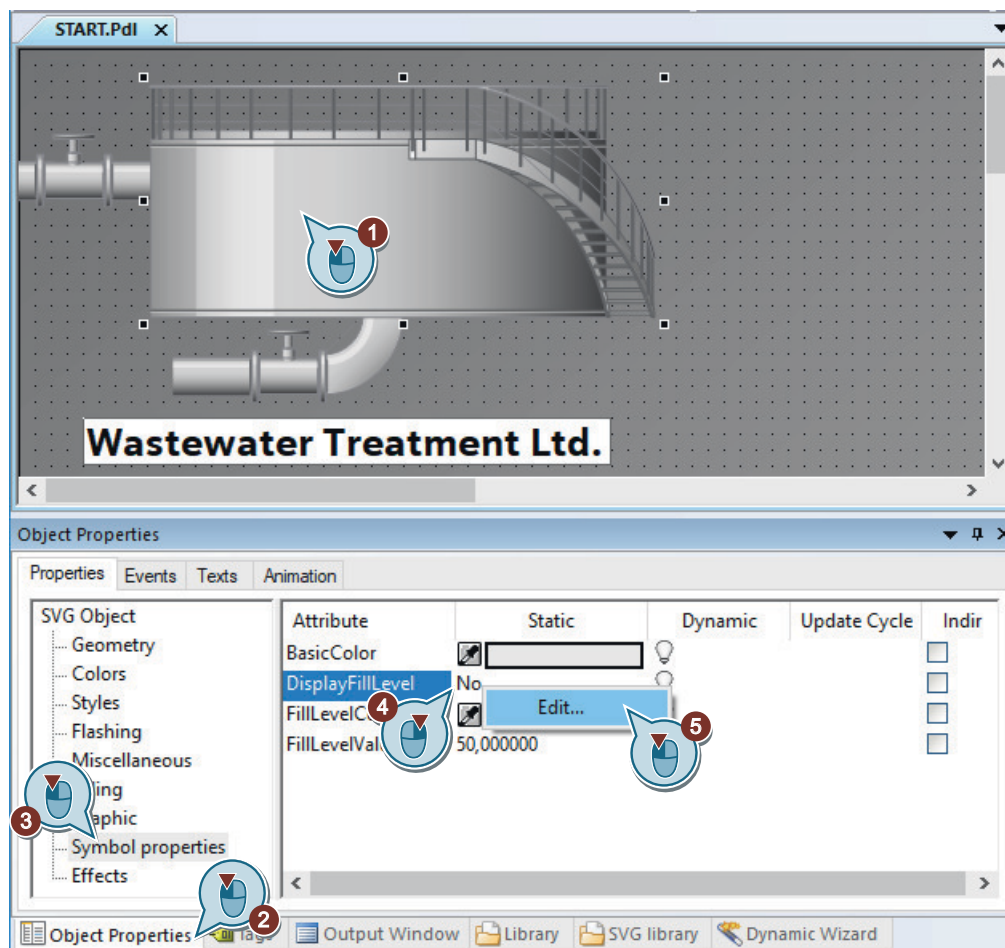
必要条件

- 「START.Pdl」プロセス画像が作成されている。
- 内部タグ"Tank_Level"が作成されている。
- SVG オブジェクト[Digester]が「START.Pdl」プロセス画像に、バイオコンバータのグラフィックの説明として挿入されていること。

6.4 プロセス画面のダイナミック化

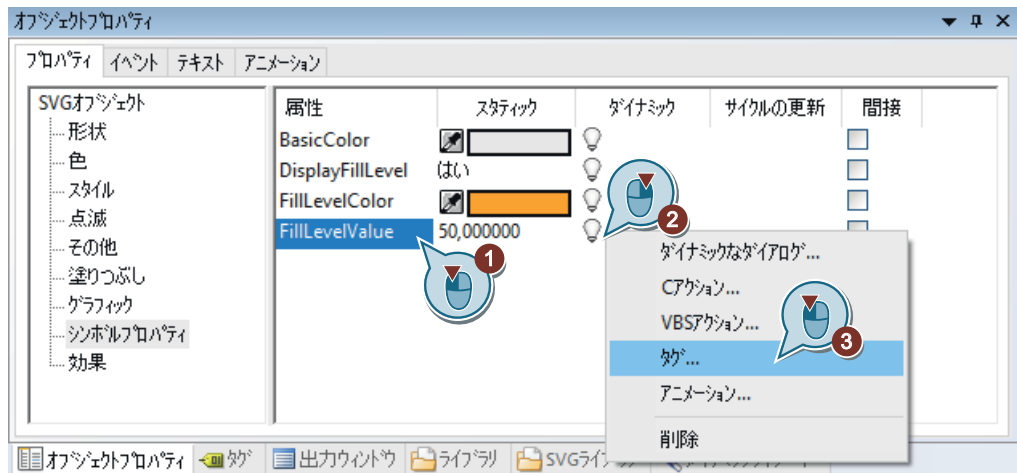
手順

1. 「START.Pdl」 プロセス画像を開き、[Digester]オブジェクトをクリックします。
2. [シンボルプロパティ]グループを[オブジェクトプロパティ]ウィンドウで開き、[DisplayFillLevel]属性を[なし]から[あり]に変更します。



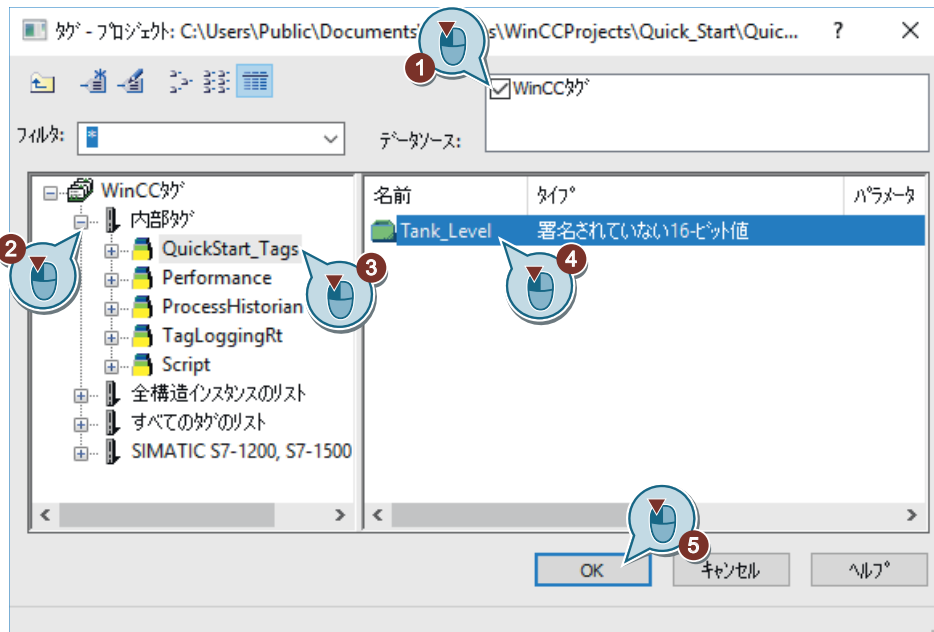
バーグラフが[Digester] SVG オブジェクトで、充填レベルインジケータとして表示されます。

3. [タグ]ダイアログを開き、[FillLevelValue]属性をタグとリンク付けします。



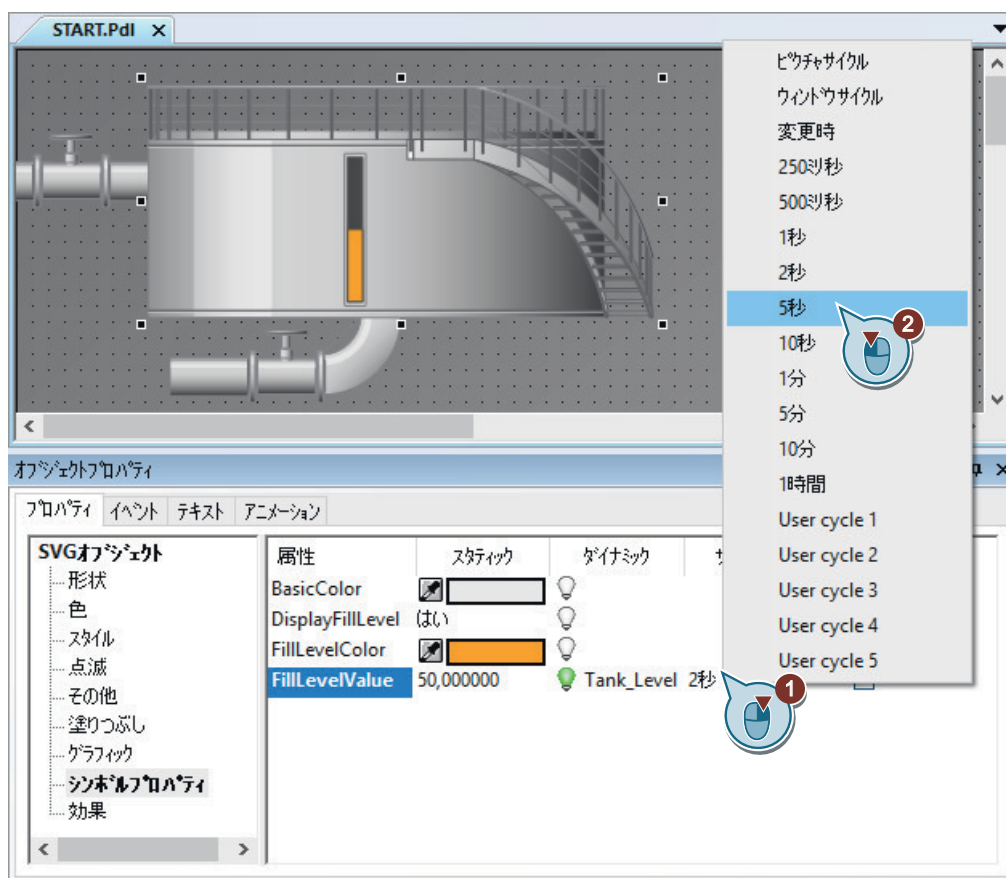
6.4 プロセス画面のダイナミック化

- 内部タグ"Tank_Level"を選択します。
必要な場合、"データソース"エリアで"WinCC タグ"オプションを選択します。



"FillLevelValue"の透明な電球が緑色に変わります。
"シンボルプロパティ"プロパティと"FillLevelValue"属性が太字で表示されます。

- 充填レベル更新サイクルを"5 秒"の値に設定します。



結果

内部タグ[Tank_Level]がバイオコンバータのグラフィック画像に接続されます。この接続によって、タグ値がグラフィックオブジェクトに転送されます。

次のステップ

I/O フィールドに、値の入力と出力に使用する「START.Pdl」プロセス画像を挿入します。

6.4.2 I/O フィールド(グラフィックシステム - プロセス画像)の挿入とダイナミック化

概要

次のステップは、I/O フィールドを挿入し、ダイナミック化する方法を示しています。

I/O フィールドは、タグ値を表示、変更するために使用する入力/出力フィールドです。

6.4 プロセス画面のダイナミック化

I/O フィールドのダイナミック化には、以下のステップが含まれます。

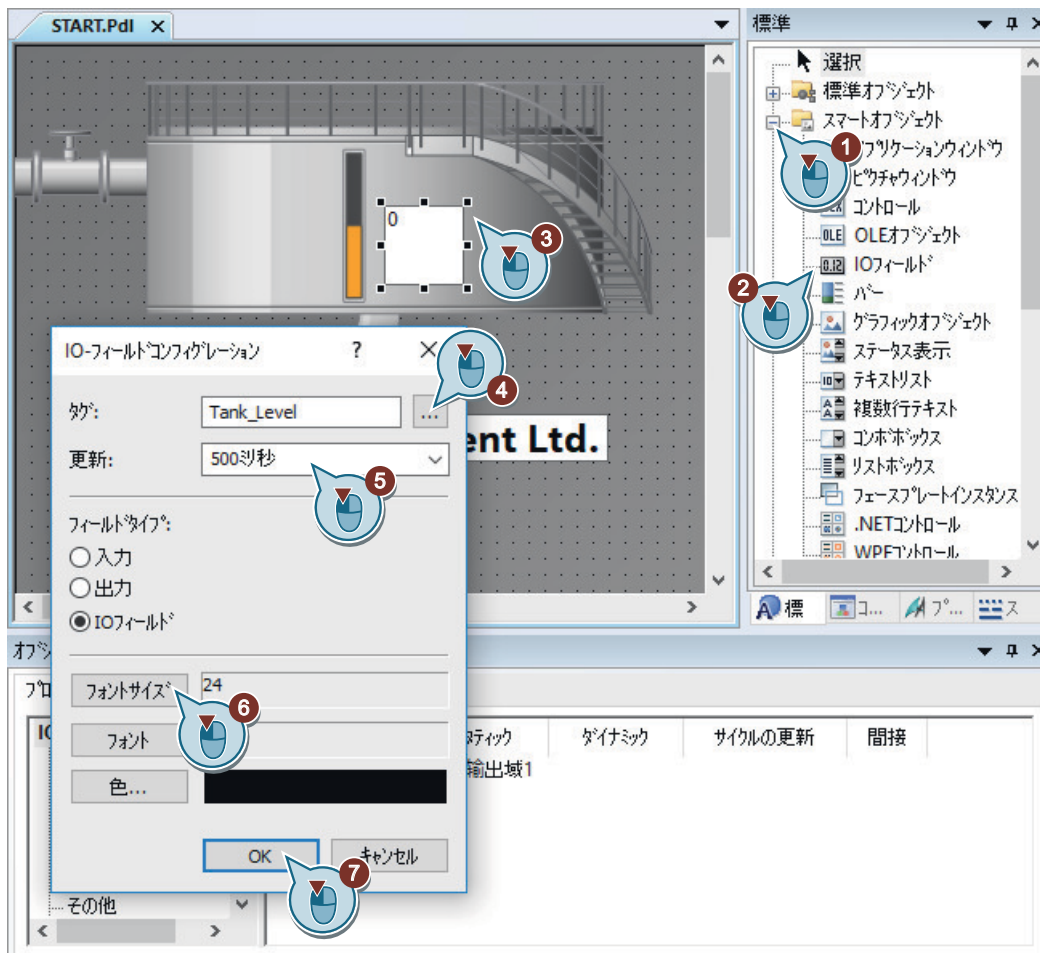
- I/O フィールドと内部タグ"Tank_Level"の接続
I/O フィールドを、"Quick_Start"プロジェクトの内部タグ"Tank_Level"と接続します。これにより、I/O フィールドとバイオコンバータのグラフィックの間に間接的接続が確立されます。
ランタイムに I/O フィールドの値を入力すると、この値は内部タグ"Tank_Level"によって取得されます。タグは、値をバイオコンバータを表すグラフィックオブジェクトに転送します。バイオコンバータの充填レベルインジケータが、タグ値に従ってランタイムに変化します。
- 更新の定義
更新によって、I/O フィールドの表示が更新される時間間隔を定義します。
- 属性[下限値]および[上限値]の定義
属性[下限値]および[上限値]によって、I/O フィールドへの入力を一定の値範囲に制限できます。設定した値範囲外の値はシステムによって拒否され、表示されません。

必要条件

- [グラフィックデザイナー]エディタで、プロセス画像「START.Pdl」が開いている。
- 内部タグ"Tank_Level"が作成されている。

手順

1. [I/O フィールド]スマートオブジェクトを挿入するには、[標準]ウィンドウでオブジェクトをクリックしてから、プロセス画像をクリックします。
I/O フィールドが、作業エリアに表示されます。[I/O フィールド設定]ダイアログが開きます。
2. 新しい I/O フィールドに"Tank_Level"タグをリンクします。
更新サイクルとして 500 ミリ秒、フォントサイズとして 24 を選択します。



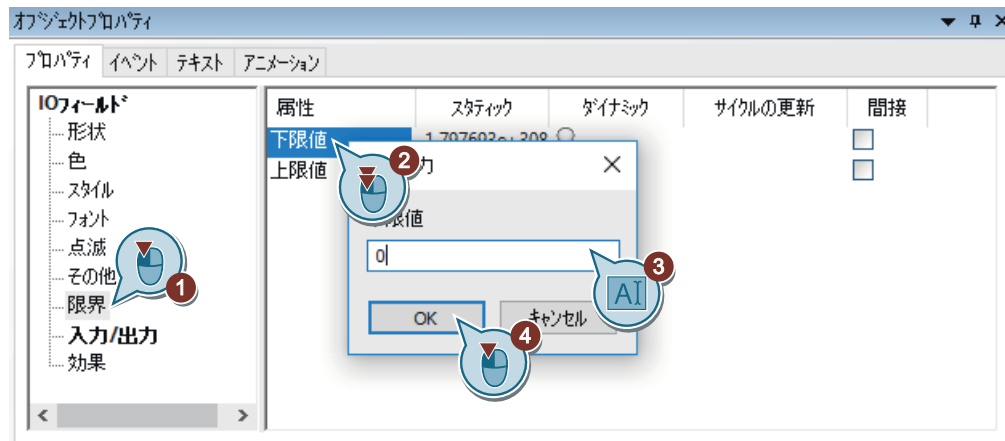
I/O フィールドのショートカットメニューから[設定ダイアログ]を選択することで、[I/O フィールド設定]ダイアログを再度開きます。

"出力/入力"プロパティが"オブジェクトプロパティ"ウィンドウで太字で表示されます。ここで、内部タグ[Tank_Level]が I/O フィールドに接続していることを確認できます。

"オブジェクトプロパティ"ウィンドウと同様に、"I/O フィールド設定"ダイアログでタグへの接続を作成できます。

6.4 プロセス画面のダイナミック化

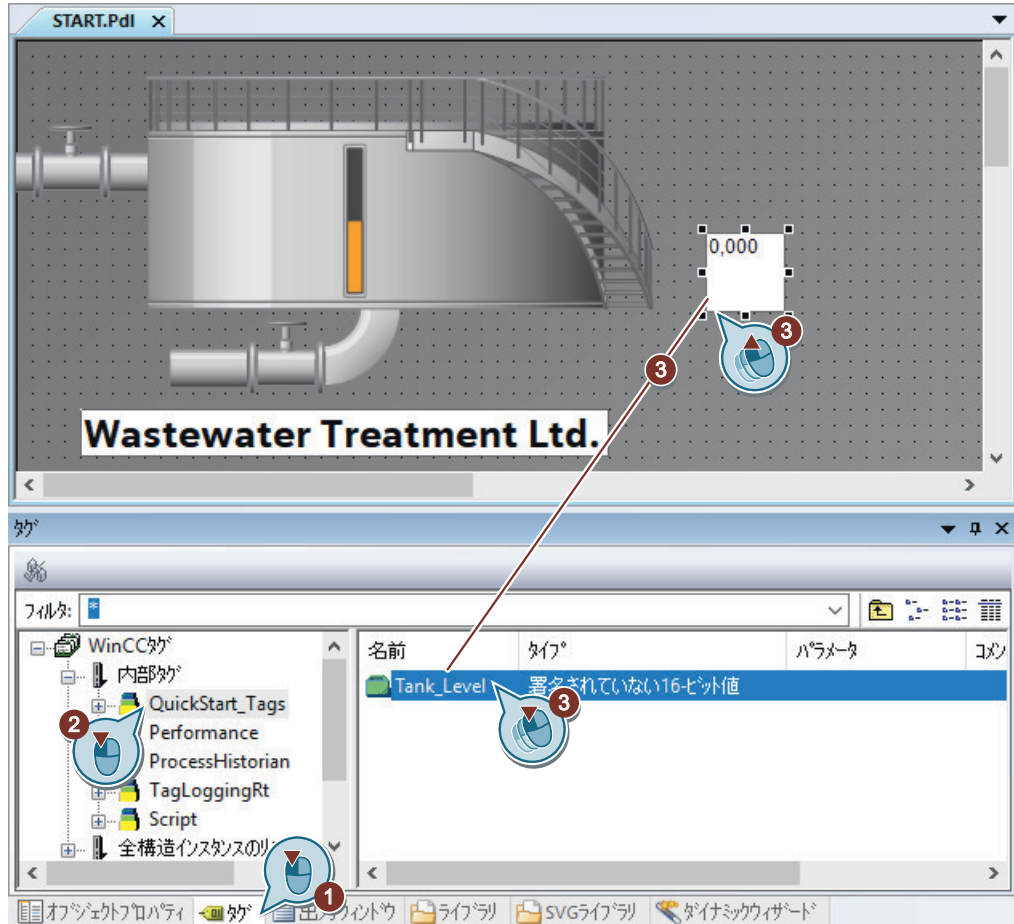
3. [制限]グループを[オブジェクトプロパティ]ウィンドウで開き、[下限値]を0に設定します。



4. ステップ 3 と同様に、[上限値]を 100 に設定します。
5. 「START.Pdl」プロセス画像を保存します。
6. [グラフィックデザイナー]エディタを閉じます。

別の手順

1. [QuickStart_Tags]タググループの[タグ]ウィンドウで、[Tank_Level]タグを選択します。
2. タグをプロセス画像にドラッグアンドドロップします。



I/O フィールドが選択されたタグに対して作成されます。I/O フィールドは、[Tank_Level]タグと接続されています。

3. I/O フィールドのショートカットメニューから"設定ダイアログ"を選択することで、"I/O フィールド設定"ダイアログを開きます。
4. 更新サイクルとして 500 ミリ秒、フォントサイズとして 24 を選択します。
5. I/O フィールドを希望する位置に移動するには、マウスまたはキーボードの矢印キー(カーソルキー)を使用します。
上記手順のステップ 3 に進みます。

結果

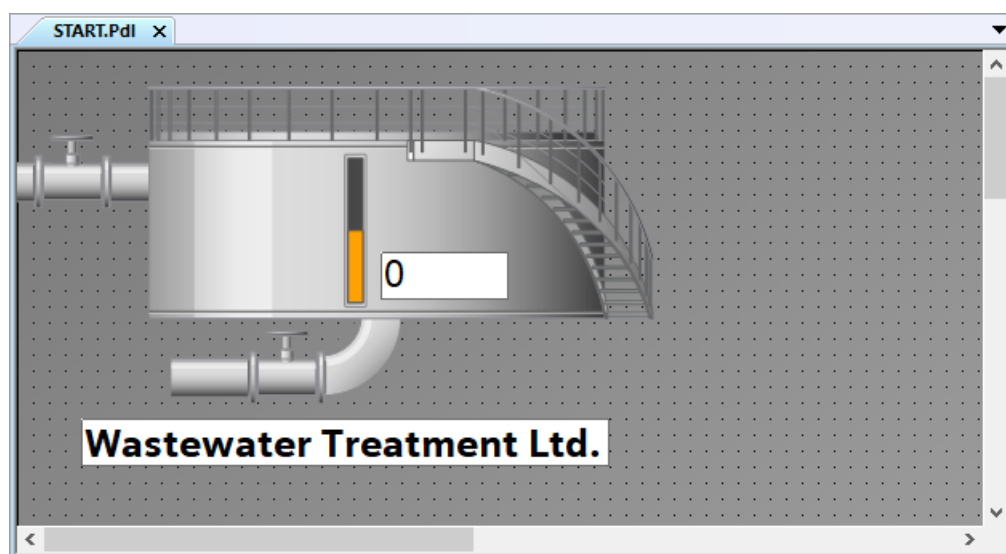
I/O フィールドを挿入し、これを内部タグ"Tank_Level"に接続します。

6.4 プロセス画面のダイナミック化

ランタイムに I/O フィールドの値を入力すると、この値は内部タグによってバイオコンバータのグラフィック画像に転送されます。充填レベルインジケータが、入力された値に従って変化します。

属性[下限値]および[上限値]によって、I/O フィールドの値範囲が定義されています。

ランタイムで I/O フィールドに値"0"を入力すると、空のバイオコンバータが表示されます。ランタイムで I/O フィールドに値"100"を入力すると、満水のバイオコンバータが表示されます。値範囲外の値は、システムで拒否されます。



次のステップ

- ランタイムの設定済み画像間の変更を行なえるようにするため、ユーザー定義メニューを作成します。
- 「START.Pdl」プロセス画像のダイナミック化を表示するために、WinCC Runtime のプロパティを定義し、「Quick_Start」プロジェクトを有効にします。

6.5 ユーザー定義メニューおよびツールバーの作成

概要

このセクションでは、カスタマイズされたメニューおよびツールバーを使用した画像の作成方法を説明します。

原理

[メニューとツールバー]エディタでは、カスタマイズされたメニューとツールバーを設定できます。

カスタマイズされたメニューおよびツールバーは、WinCC の"コンピュータ"エディタでプロジェクトに割り当てる設定ファイルに保存されます。

グローバルスクリプト手順を使って、メニュー項目とシンボルを接続します。

追加のコンフィグレーションの選択肢:

- 権限の割り当て
ログインしたオペレータは、対応する設定エレメントを表示するには、必要な権限を持っている必要があります。
- メニューコマンドとシンボルの非表示化または無効化
異なる機能を対象とする 2 つの設定ファイルを作成します。
適用例:ランタイムでユーザーが変更すると、現在の設定ファイルが 2 番目のファイルに置き換えられます。

6.5.1 カスタマイズされたメニューおよびツールバーのプロシージャの作成

概要

次のステップは、グローバルスクリプトのモジュールでのプロシージャの作成方法を示しています。

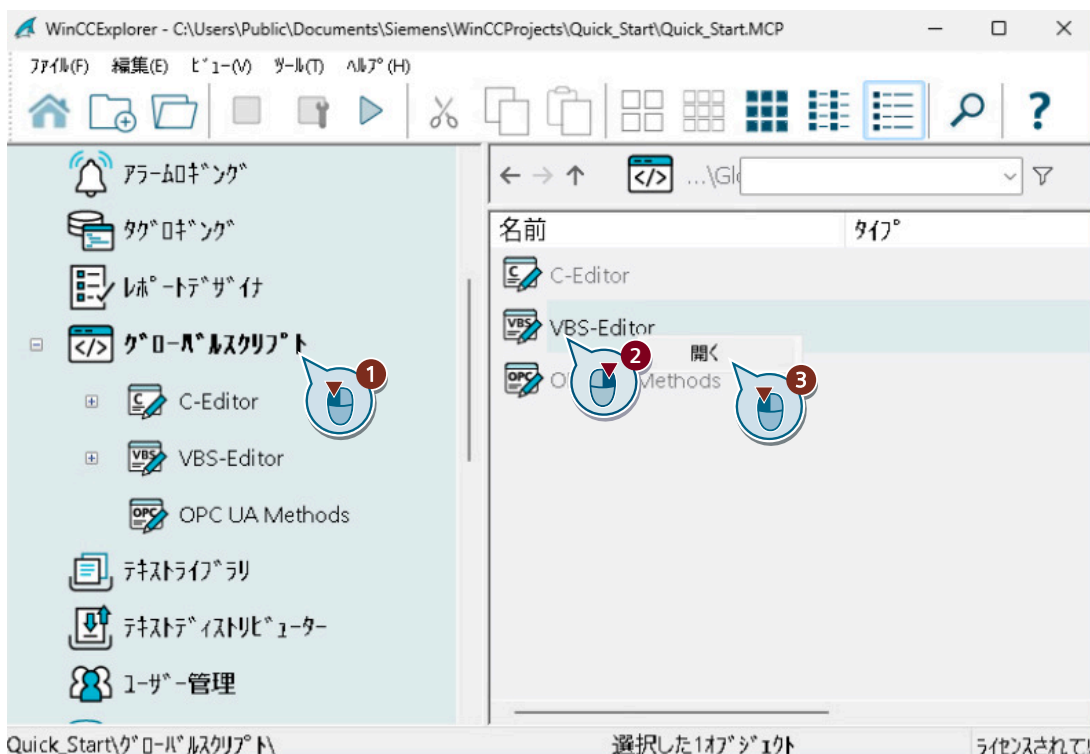
カスタマイズされたメニューおよびツールバーを機能させるには、2 つのプロシージャが必要になります。

- `ActivatePicture(ByVal PictureName):`
"PictureName"パラメータと一緒に転送される画像に対して、画像の変更を実行します。
- `StopRuntime(ByVal Item):`
ランタイムを終了します。

6.5 ユーザー定義メニューおよびツールバーの作成

手順

1. VBS-Editor を開きます。



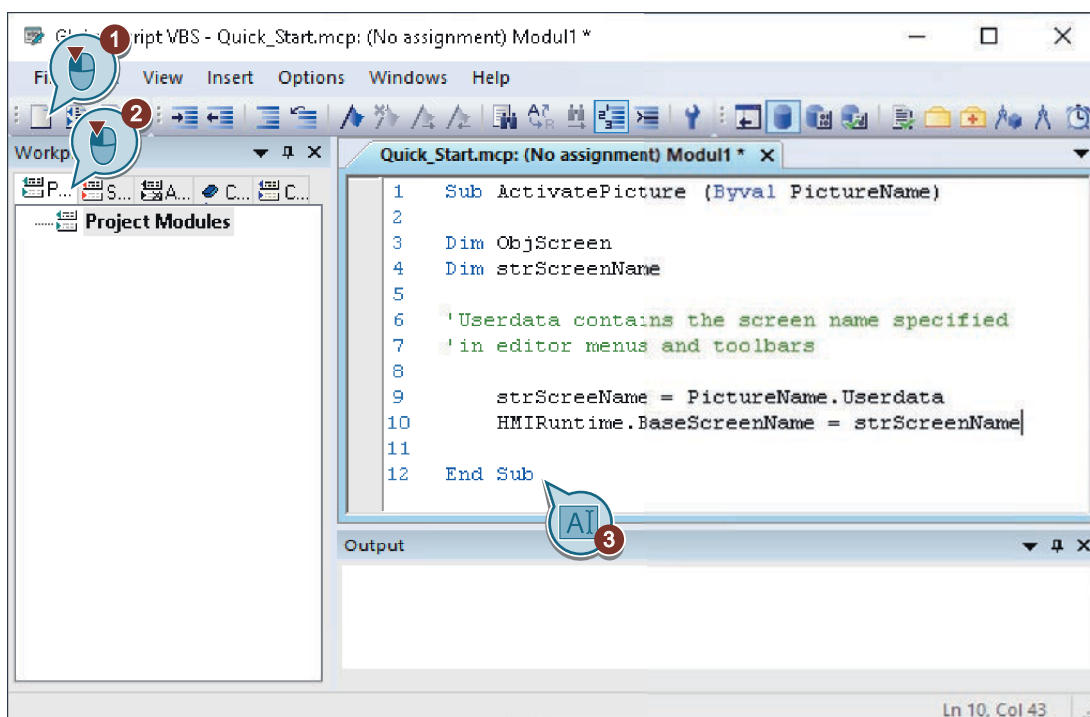
2. [プロジェクトモジュール]タブを選択し、以下のプロシージャコードを書き込みます。
Sub ActivatePicture (Byval PictureName)

```
Dim objScreen
Dim strScreenName
```

'Userdata contains the screen name specified in editor menus and toolbars

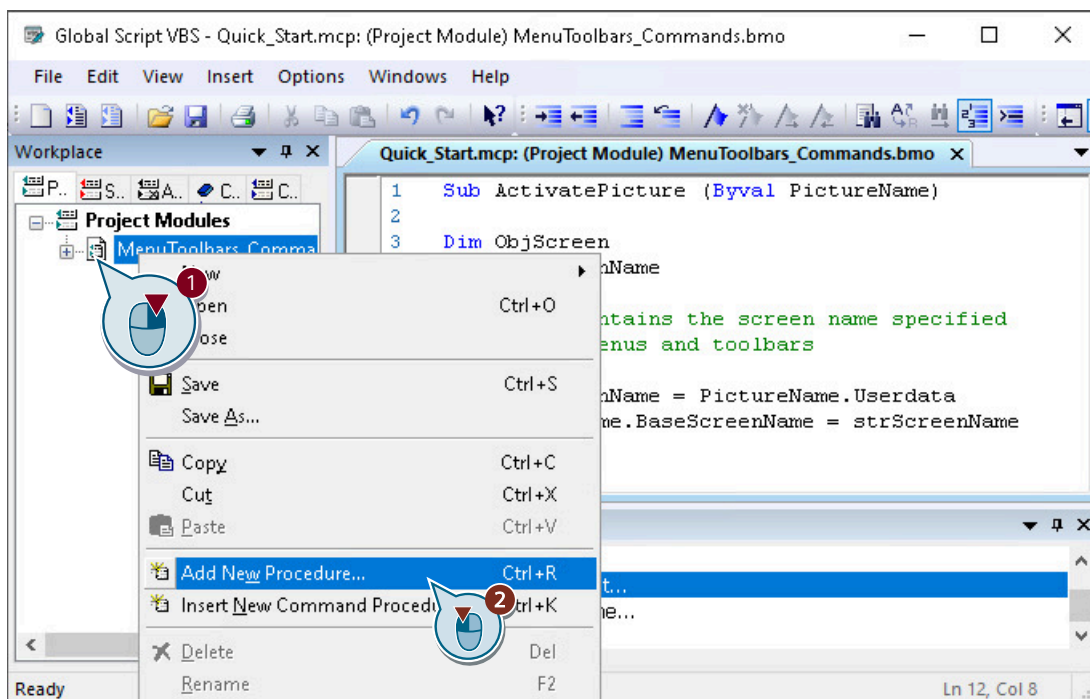
```
strScreenName = PictureName.Userdata
HMIRuntime.BaseScreenName = strScreenName
```

```
End Sub
```



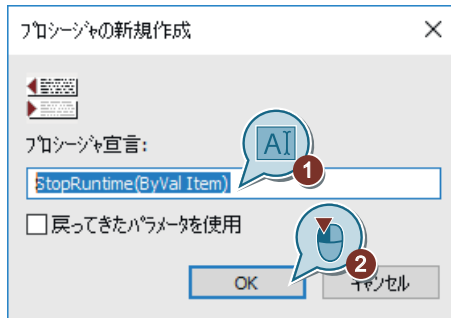
3. [ファイル]>[保存]を使用して、モジュールを「MenuToolbars_Commands.bmo」ファイルとして保存します。

4. 新しいプロシーダを挿入します。



6.5 ユーザー定義メニューおよびツールバーの作成

- 名前"StopRuntime(ByVal 項目)"を入力します。



- 以下のプロシージャコードを書き込みます。

```
Sub ActivatePicture (Byval PictureName)

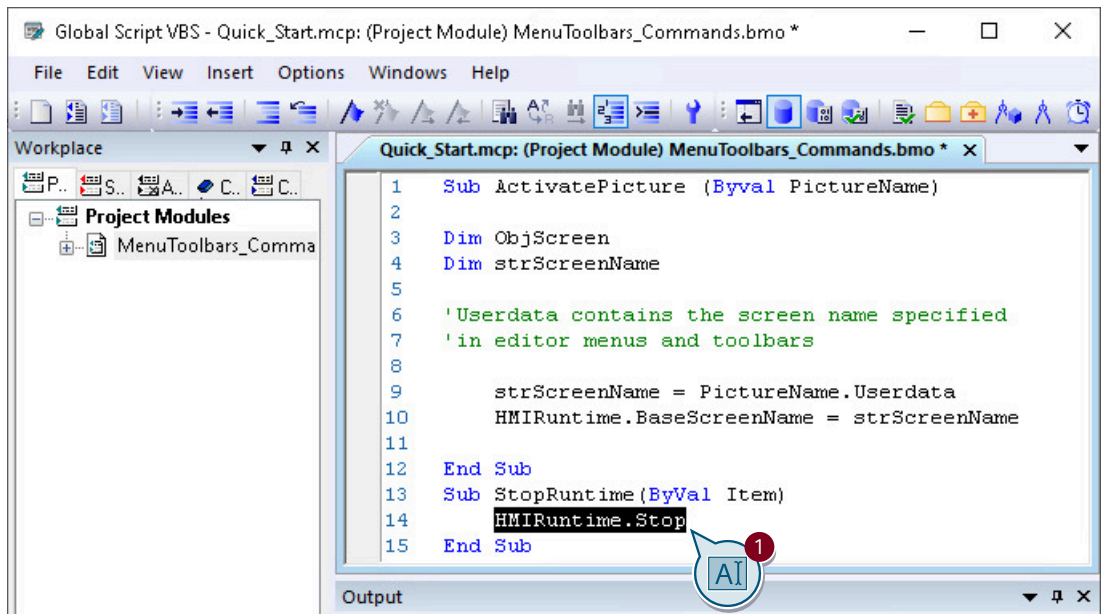
Dim objScreen
Dim strScreenName

'Userdata contains the screen name specified in editor menus and
toolbars

strScreenName = PictureName.Userdata
HMIRuntime.BaseScreenName = strScreenName

End Sub

Sub StopRuntime (ByVal Item)
    HMIRuntime.Stop
End Sub
```



7. モジュールを保存します。
8. VBS-Editor を閉じます。

結果

プロシージャ"ActivatePicture(ByVal PictureName)"および"StopRuntime(ByVal Item)"が作成されます。

次のステップ

- 「START.Pdl」 および 「SAMPLE.Pdl」 の画像に画像変更を実行するカスタマイズされたメニューを作成します。
- カスタマイズされたツールバーを使用して、ランタイムを終了します。

6.5.2 画面変更のためのカスタマイズされたメニューの作成

概要

次のステップは、2つのメニュー項目[開始]と[サンプル]を持つ、カスタマイズされたメニュー[画像変更]を作成する方法を示しています。

メニューエントリ"開始"と"サンプル"をプロシージャ"ActivatePicture(ByVal PictureName)"にリンクさせます。

変更するプロセス画像の名前を、"ユーザーデータ"ボックスに入力します。

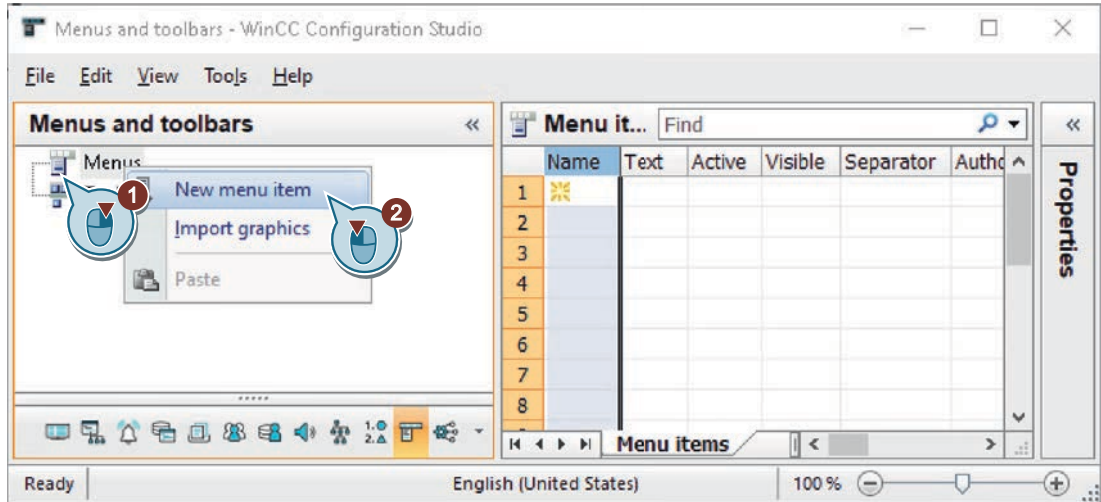
必要条件

- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。

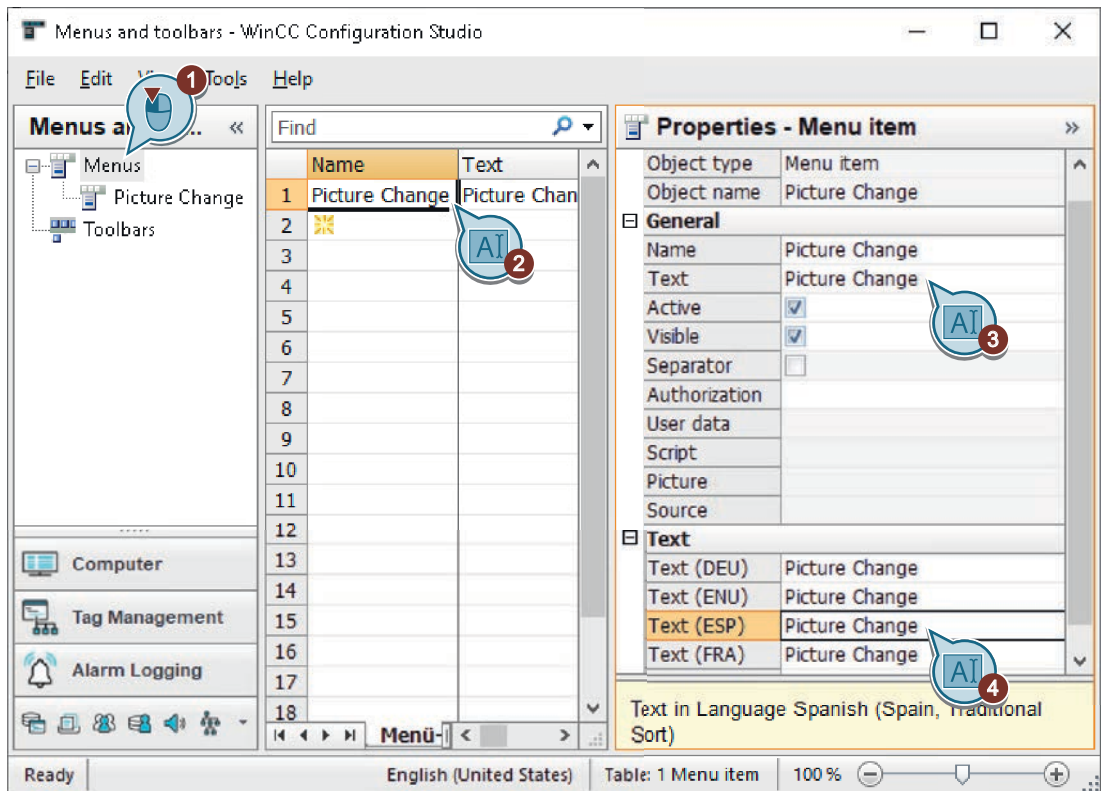
6.5 ユーザー定義メニューおよびツールバーの作成

手順

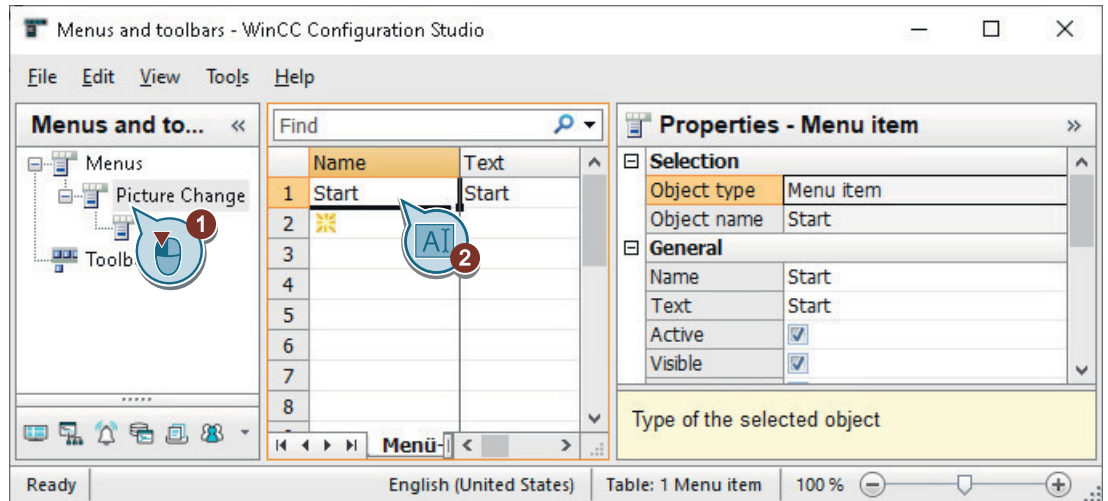
1. [メニューとツールバー]エディタを開きます。
2. 新しいメニュー項目を作成します。



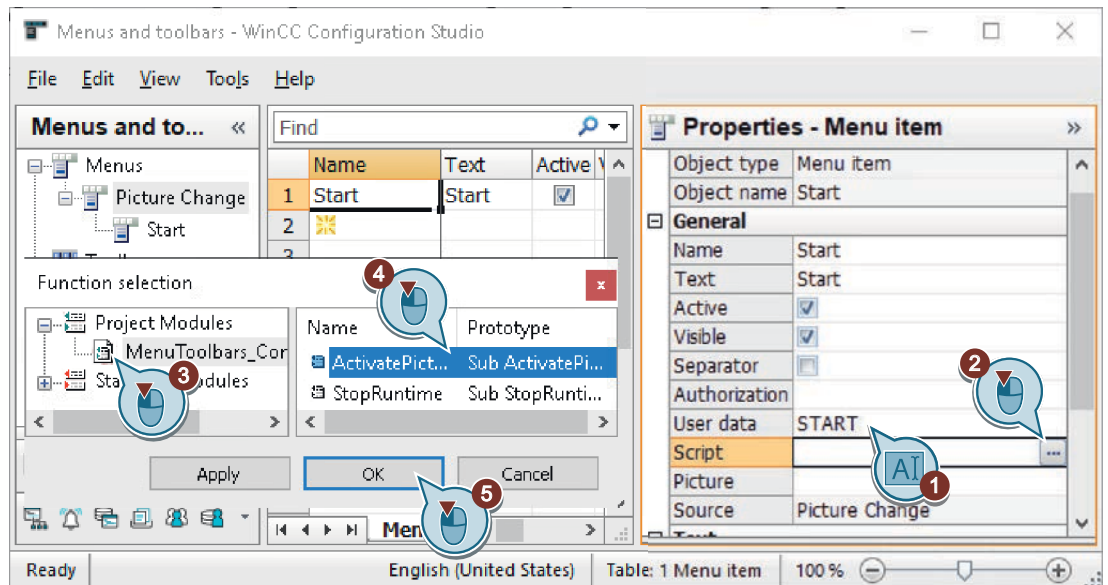
3. メニュー項目に"Picture Change"という名前を付けます。
"プロパティ"ウィンドウでは、"テキスト"エリアにその他のランタイム言語の名前を入力します。



4. [スタート]メニューコマンドを作成します。



5. メニュー項目[開始]を設定して、「START.Pdl」画像に画像変更を実行できるようにします。



6. 同じ手順を使用して、「SAMPLE.Pdl」画像に画像変更を実行するメニュー項目[サンプル]を作成します。

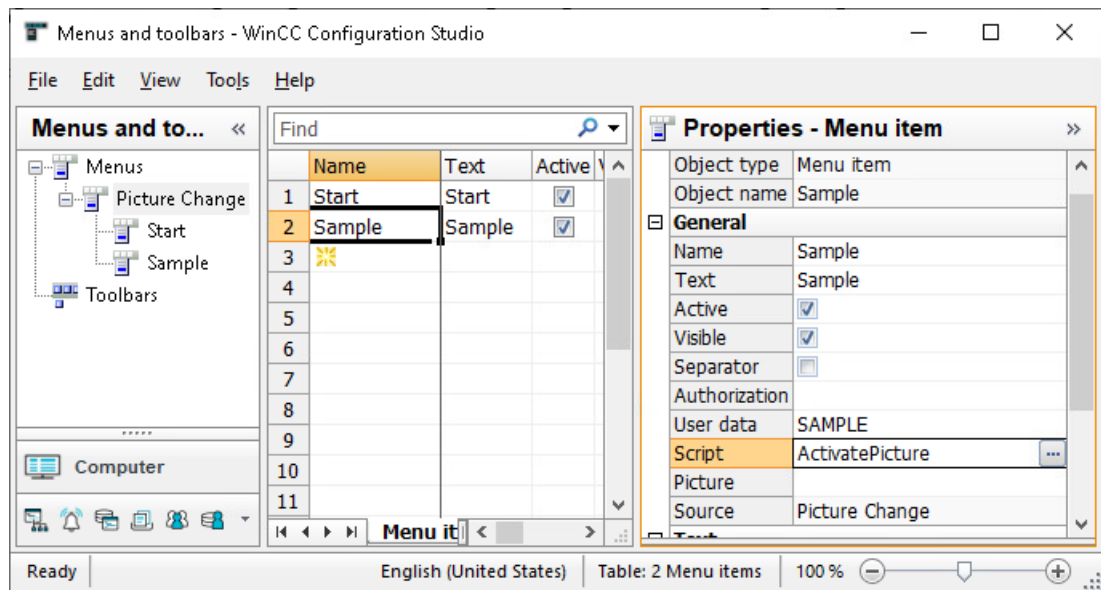
7. [ファイル]>[保存]を使用して、設定に「ActivatePicture」という名前を付けて保存します。

結果

2つのメニューコマンドを持つ[画像変更]メニューが作成されます。

ランタイムでこれらメニューコマンドを使用して、「START.Pdl」と「SAMPLE.Pdl」のプロセス画像を変更することができます。

6.5 ユーザー定義メニューおよびツールバーの作成



6.5.3 ランタイムを終了するカスタマイズされたツールバーの作成

概要

次のステップは、ランタイムを終了するアイコンがある、カスタマイズされたツールバーの作成方法を示しています。

シンボルのグラフィックファイル

シンボルとして表示するグラフィックファイルを次のフォルダにコピーします。

- <プロジェクトフォルダ>\GraMT\mtimages

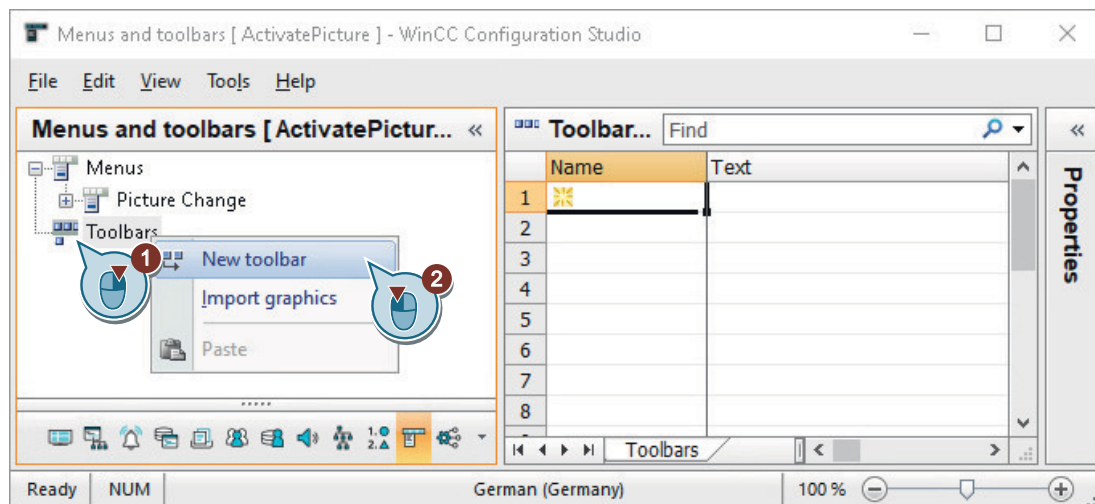
シンボルとして、BMP、PNG、JPG または GIF フォーマットのグラフィックを使用します。

必要条件

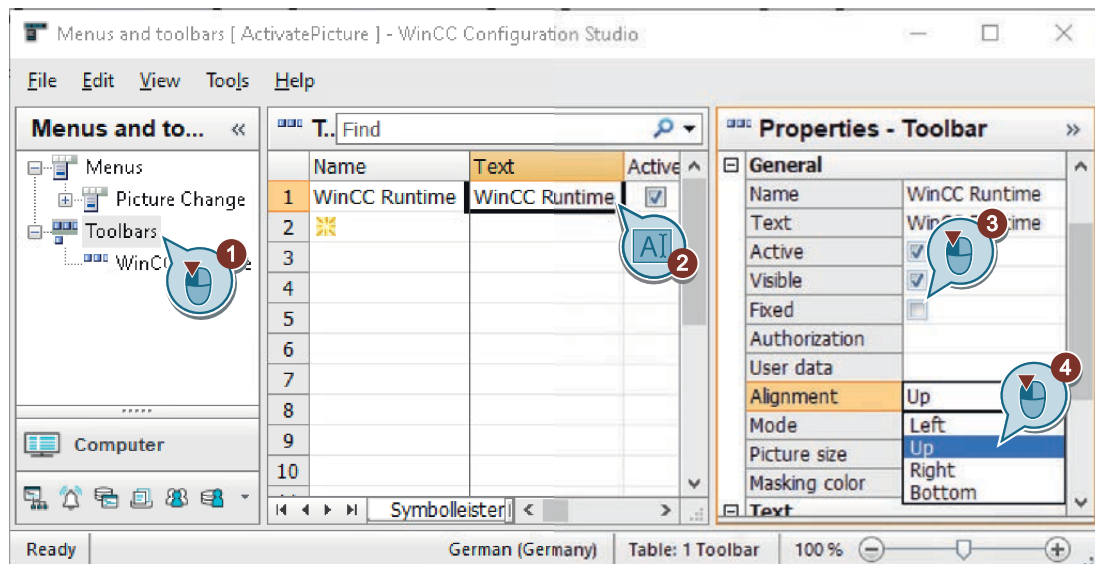
- [ActivatePicture]設定が、[メニューとツールバー]エディタで開いていること。
- "StopRuntime(ByVal 項目)"プロシージャが作成されていること。
- グラフィックファイルが"ランタイムの終了"のシンボルとして WinCC プロジェクトフォルダにコピーされていること。例:
 - C:\Users\Public\Documents\Siemens\WinCCProjects\Quick_Start\GraMT\mtimages

手順

1. 新規ツールバーを作成し、「WinCC Runtime」を名前として入力します。

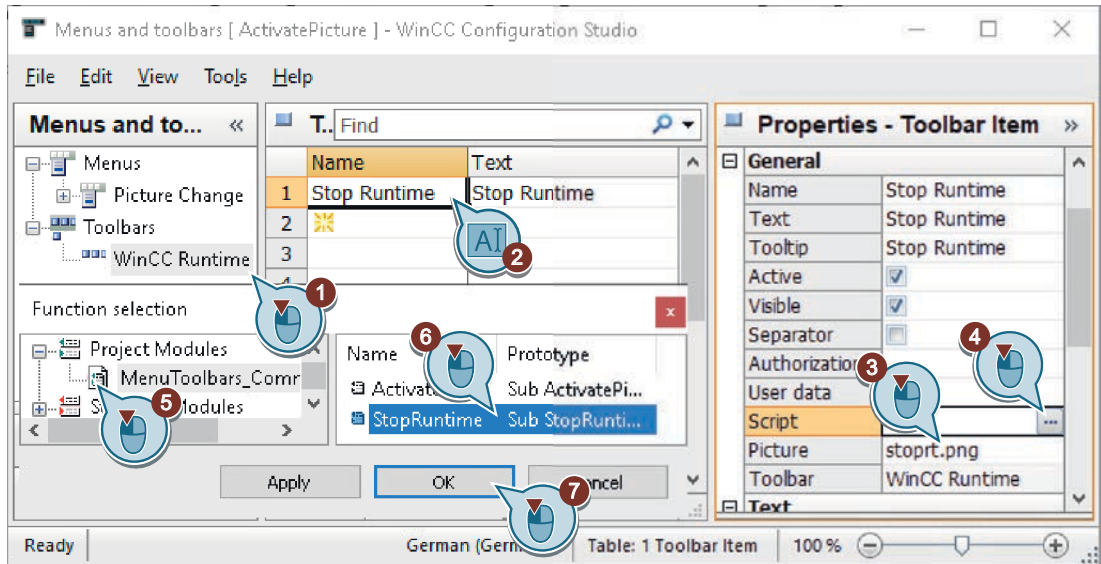


2. デフォルトで画像の上部端に表示されるようにツールバーを設定しますが、ユーザーはこの位置を変えることができます。



6.5 ユーザー定義メニューおよびツールバーの作成

3. [ランタイムの停止]エレメントを作成し、ランタイムを停止するアイコンを設定します。この例では、グラフィック「stoprt.png」がアイコンとして使用されています。



4. [ActivatePicture]設定を保存します。
5. "メニューとツールバー"エディタを閉じます。

結果

ランタイムを終了するアイコンのあるツールバーが作成され、設定が保存されます。

次のステップ

ランタイムでメニューバーおよびツールバーを表示するには、ランタイムのプロパティを指定しているときに設定ファイルを選択します。

そうすると、ランタイム時に、それぞれのプロセス画像にはユーザー定義のメニューとツールバーが表示されます。

「スタートガイド」のコースで追加のプロセス画像を生成する場合、ここに示すステップを使用して追加の図を含むようにメニューの設定を拡張できます。

6.6 ランタイム中のプロジェクトの実行

6.6.1 ランタイムタイムプロパティの定義(グラフィックシステム - プロセス画像)

概要

次のステップは、WinCC Runtime のプロパティの定義方法を示しています。

WinCC Configuration Studio の[コンピュータ]エディタで WinCC Runtime のプロパティを設定します。

プロジェクトを有効にするときにグラフィックランタイムが実行されるように WinCC Runtime をセットアップします。

「START.Pdl」プロセス画像を、ランタイムウィンドウの初期画像として選択します。

作成した設定ファイルをプロセス画像でユーザー定義メニューおよびツールバーを表示するために読み込みます。

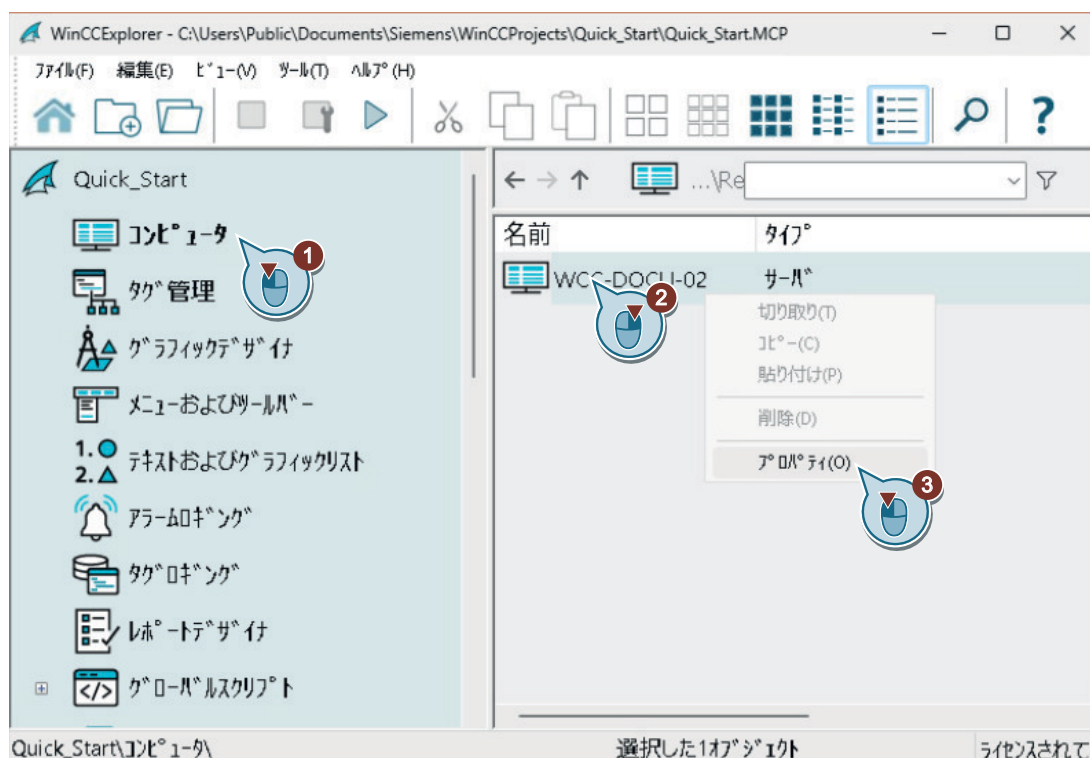
必要条件

- "Quick_Start"プロジェクトが開いている。

6.6 ランタイム中のプロジェクトの実行

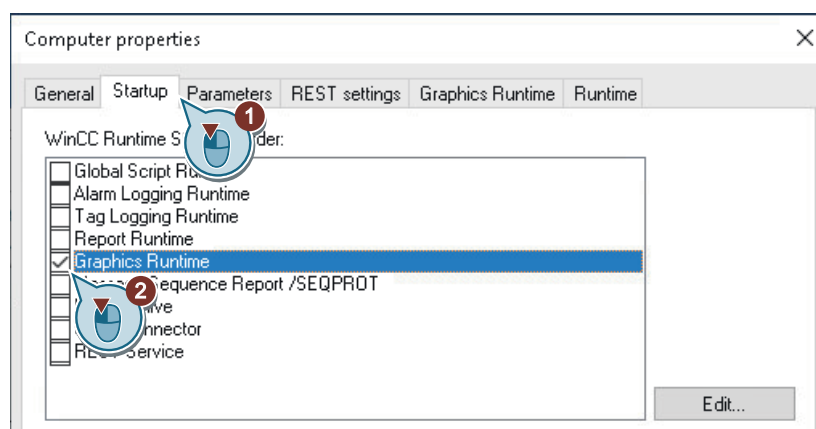
手順

1. ショートカットメニューで[コンピュータ]と[プロパティ]のエントリのコンピュータ名を選択します。

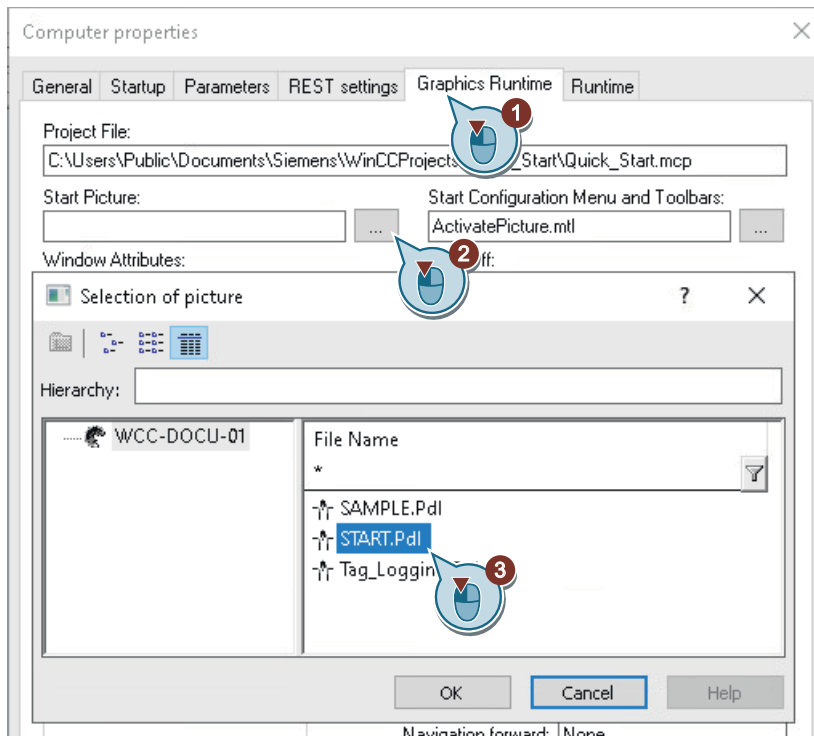


[コンピュータ]エディタが開かれます。

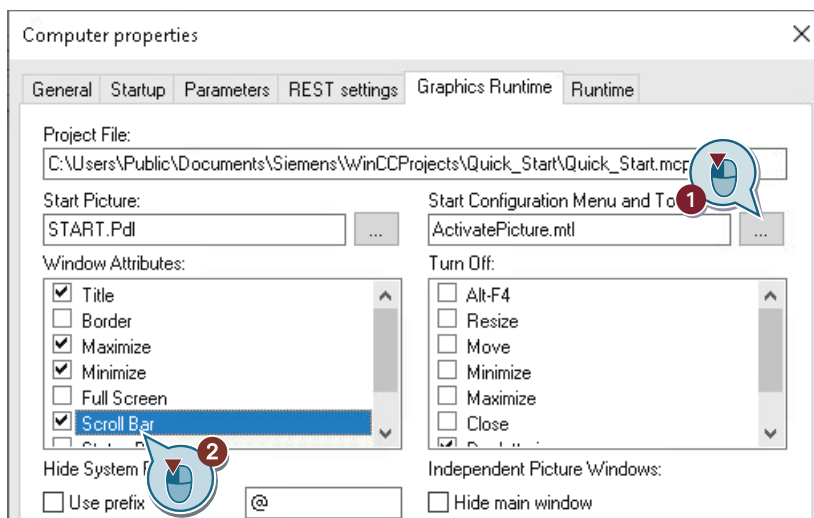
2. [スタートアップ]タブの[グラフィックスランタイム]アプリケーションを選択します。



3. [グラフィックランタイム]タブで、開始画像としてプロセス画像「START.Pdl」を定義します。



4. メニューおよびツールバーの設定ファイル"ActivatePicture.mtl"を選択します。 .
5. "タイトル"、"最大化"、"最小化"および"スクロールバー"ウィンドウ属性を有効にします。



結果

WinCC Runtime のプロパティが定義されています。

6.6 ランタイム中のプロジェクトの実行

"Quick_Start"プロジェクトが有効にされている場合、[グラフィックランタイム]コンポーネントが実行されます。開始画像として、「START.Pdl」プロセス画像が表示されます。

ランタイムをナビゲート、終了するために、すべてのプロセス画像で、カスタマイズされたメニューとツールバーが表示されます。

このウィンドウ属性を使用して、ランタイムウィンドウの追加機能を決定します。

次のステップ

「Quick_Start」プロジェクトを有効にします。

6.6.2 プロジェクトの有効化(グラフィックシステム - プロセス画像)

概要

次のステップは、「Quick_Start」プロジェクトを有効にする方法や、ランタイムでの動的なプロセス画像「START.Pdl」の操作方法を示しています。

プロジェクトを有効にすると、WinCC Runtime が開始します。WinCC Runtime は、プロセスモードでプロジェクトを実行します。

ランタイムで、設定済みの I/O フィールドを操作し、フィルレベルインジケータの変化を監視します。

必要条件

- ランタイムプロパティが定義されている。

手順


1. WinCC エクスプローラの▶ ツールバーボタンを使って"Quick_Start"プロジェクトを有効にします。
短いロード時間の後に、ランタイムウィンドウが開きます。「START.Pdl」プロセス画像が表示されます。



2. I/O フィールドに 0 から 100 の間の値を入力し、入力内容を<Return>で確定します。
フィルレベルインジケータの変化を監視します。
3. 「SAMPLE.Pdl」プロセス画像に切り替えるには、[画像変更]メニューで[サンプル]エントリをクリックします。



6.6 ランタイム中のプロジェクトの実行

4. 「START.Pdl」プロセス画像に戻るには、[画像変更]メニューで[開始]エントリをクリックします。
5. 設定したツールバーボタンをクリックして、ランタイムを終了します。
あるいは、WinCC エクスプローラツールバーの  ボタンを使用して、ランタイムを停止します。



結果

"Quick_Start"プロジェクトが有効になり、WinCC Runtime が開始します。動的なプロセス画像「START.Pdl」がランタイムウィンドウに表示されます。

I/O フィールドに値を入力すると、この値は内部タグ"Tank_Level"によってバイオコンバータのグラフィック画像に転送されます。これによって、バイオコンバータの充填レベルインジケータを監視することができます。

次のステップ

内部タグ「Tank_Level」をシミュレートするために、「Quick_Start」プロジェクトをテストし、[タグシミュレーション]エディタを使用します。

下記も参照

ランタイムを終了するカスタマイズされたツールバーの作成 (ページ 84)

6.6.3 プロジェクトのテスト(グラフィックシステム - プロセス画像)

概要

次のステップは、WinCC タグシミュレータによって「Quick_Start」プロジェクトをテストする方法を示しています。

WinCC タグシミュレータによって、まだ開発段階であるプロジェクトをテストできます。テスト中に、オートメーションシステム(AS)に接続した時に、プロジェクトがどのように動作するかを確認します。

「Quick_Start」プロジェクトで、WinCC タグシミュレータによって内部タグ「Tank_Level」の値をシミュレーションします。WinCC Configuration Studio の[タグシミュレーション]エディタでシミュレーションを設定します。

WinCC タグシミュレータは、内部タグ「Tank_Level」に異なる値を割り付けます。バイオコンバータのグラフィック描写に内部タグを接続すると、バイオコンバータの充填レベルインジケータはタグ値に従って変化します。

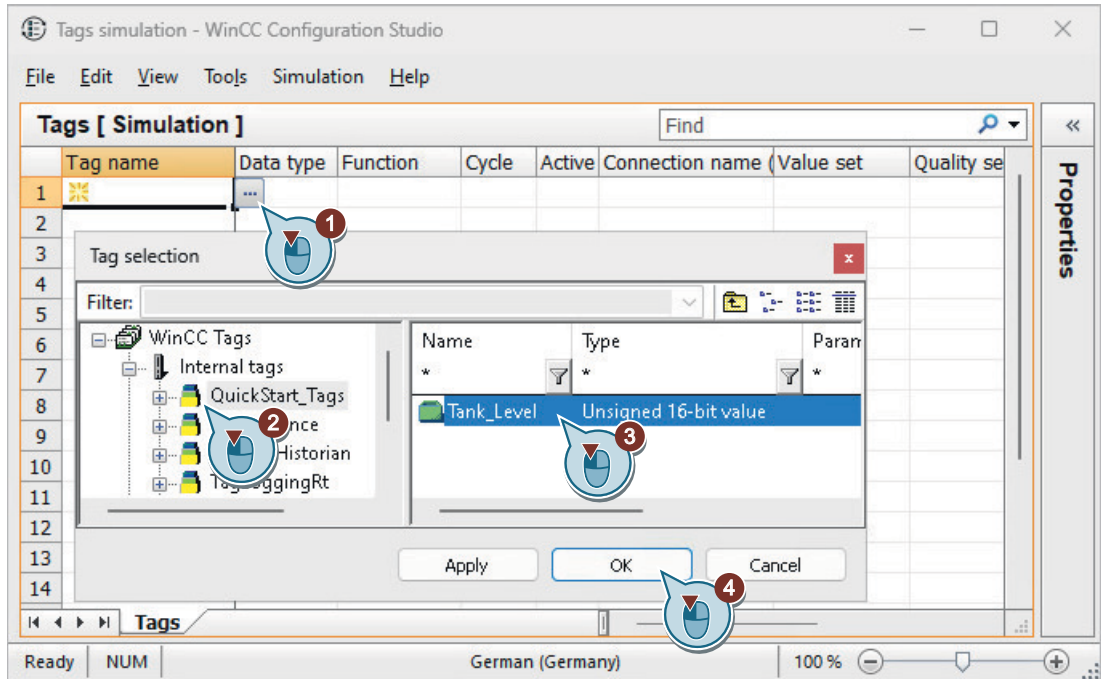
必要条件

- ランタイムプロパティが定義されている。

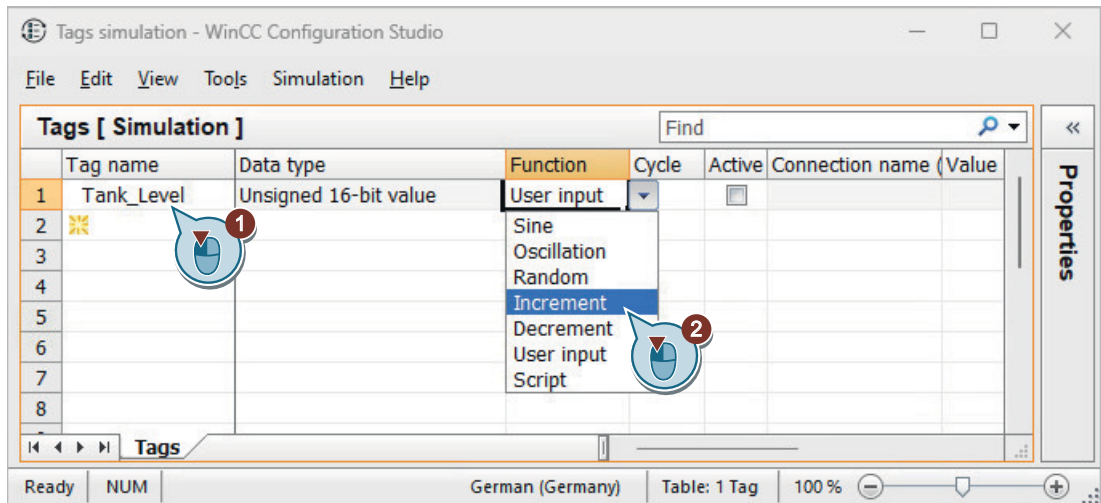
6.6 ランタイム中のプロジェクトの実行

手順

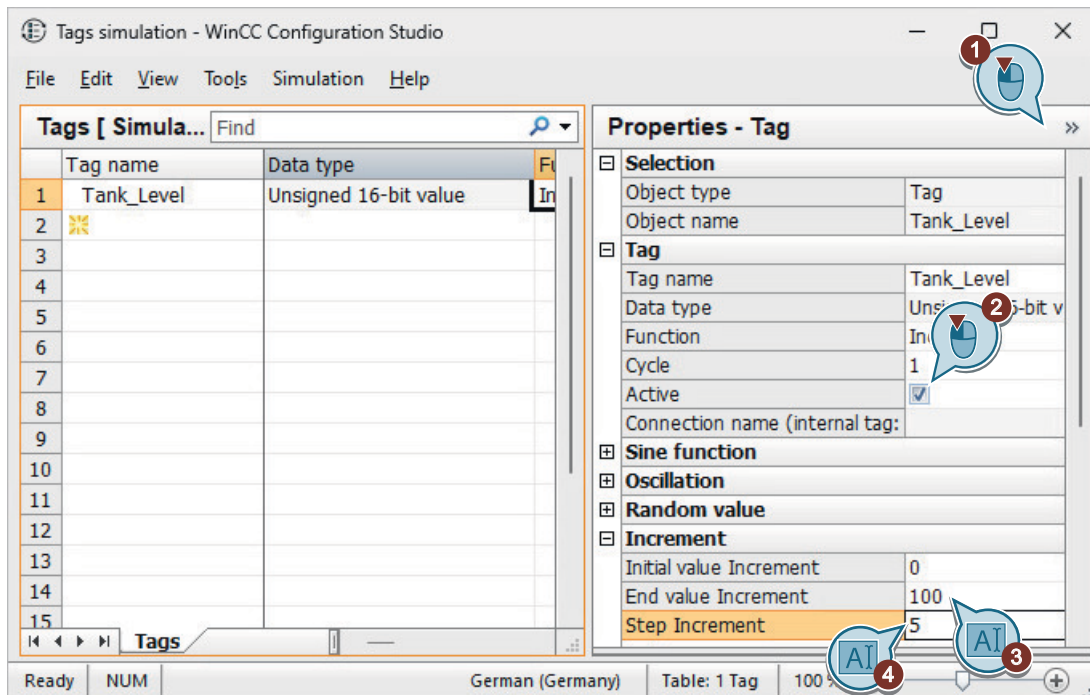
1. [タグシミュレーション]エディタを起動します。
[タグシミュレーション]エディタが開きます。
2. タグ選択ダイアログを開き、内部タグ[Tank_Level]をタググループ[QuickStart_Tags]から選択します。



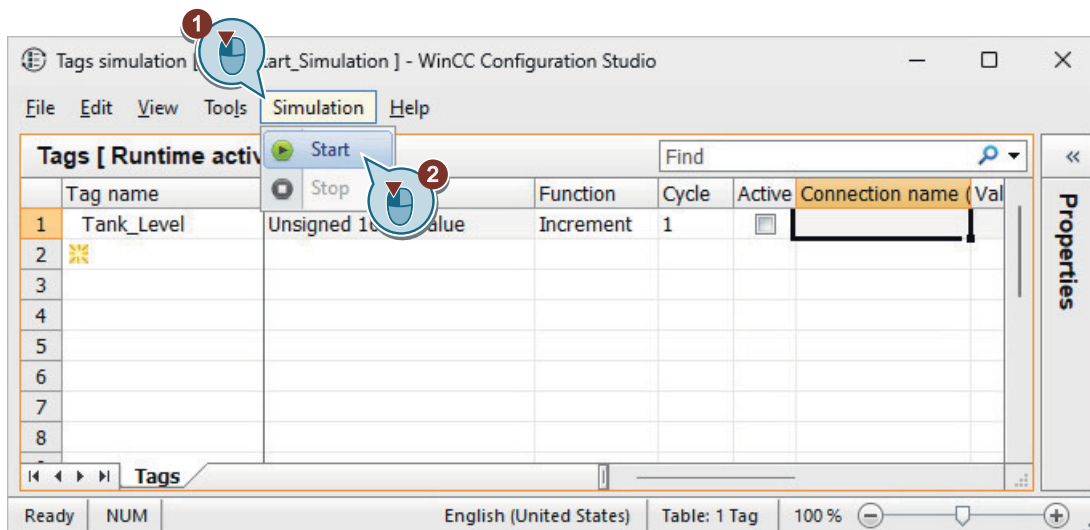
3. シミュレーションタイプ[インクリメント]を選択します。



4. [プロパティ]ウィンドウを開き、シミュレーションタイプのプロパティを定義します。

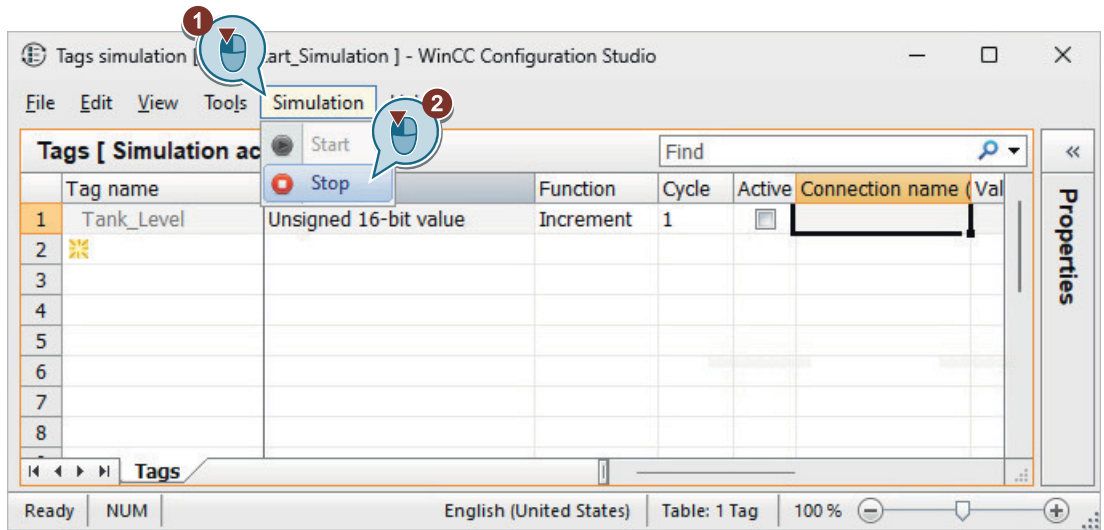


5. シミュレーションを"QuickStart_Simulation"として保存します。
6. ▶ ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。
7. [タグシミュレーション]エディタでシミュレーションを起動します。



6.6 ランタイム中のプロジェクトの実行

8. [タグシミュレーション]エディタとランタイムウィンドウを、両方を同時に監視できるように位置調整します。
シミュレーション値の変化が、フィルレベルインジケータにどのように影響するかを監視します。
設定更新サイクルは異なる表示の遅延を引き起こします。
9. シミュレーションおよび WinCC Runtime を終了します。



結果

WinCC タグシミュレータによって、"Quick_Start"プロジェクトがテストされます。
テストは、プロセス値を提供された時のプロジェクトの動作を示します。

6.6.4 ランタイムシステムダイアログの使用

概要

ランタイムシステムダイアログは、ランタイム中に、頻繁に必要なアクションを実行するのに使用できます。画像内のアクションを設定する必要はありません。

可能なアクションには、以下が含まれています。

- ランタイムシステムダイアログボックスを閉じる
- 画像の開始を表示
- 直前の画像を表示
- 次の画像を表示
- お気に入りの画像を表示
- 言語の選択

ランタイムシステムダイアログを有効にする手順は以下の通りです。

例では、2つの画像を取り替える手順を示しています。

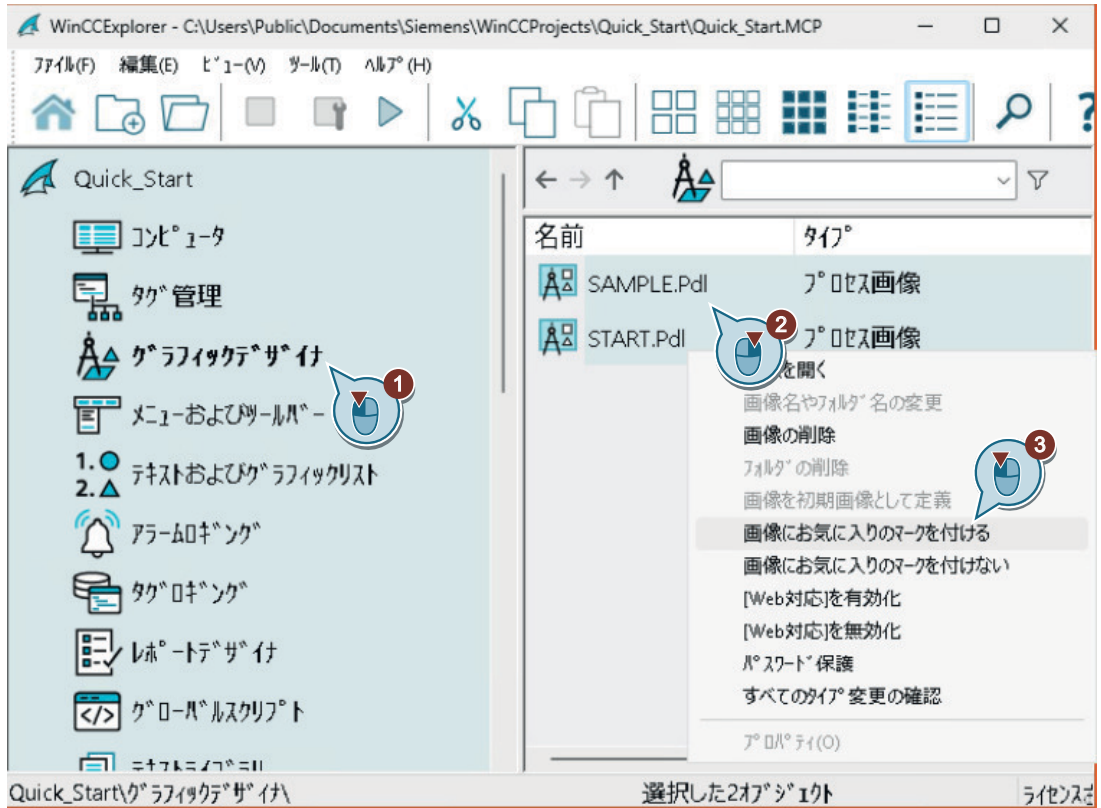
必要条件

- 「START.Pdl」と「SAMPLE.Pdl」のプロセス画像が作成されている。
- 開始画像として、「START.Pdl」プロセス画像が定義されている。

6.6 ランタイム中のプロジェクトの実行

手順

1. WinCC エクスプローラで、「START.Pdl」と「SAMPLE.Pdl」のプロセス画像を選択します。両方の画像を選択するには、クリックするときに<Ctrl>ボタンを押します。
2. プロセス画像をお気に入りとしてマーキングします。



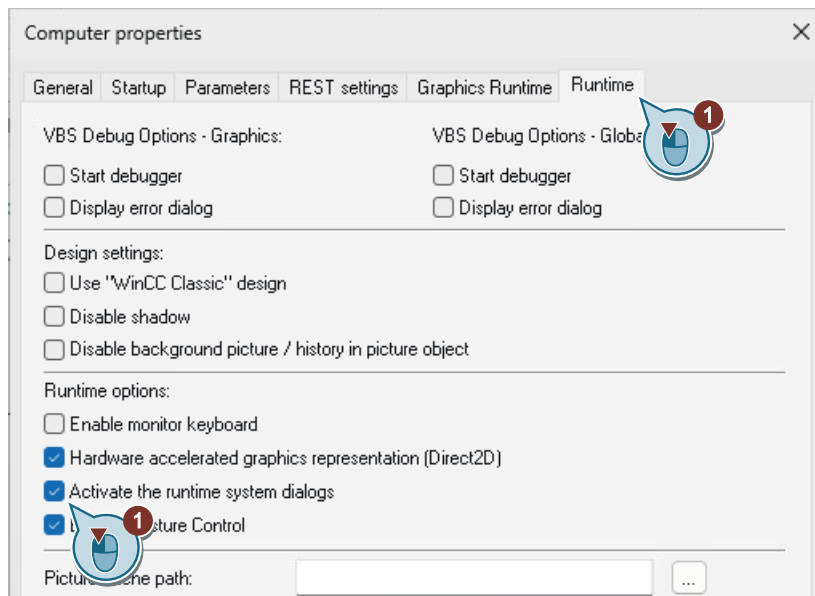
「START.Pdl」と「SAMPLE.Pdl」のプロセス画像が WinCC エクスプローラの右ペインにお気に入りとして表示されます。

3. [コンピュータ]エディタのナビゲーションエリアでショートカットメニューによりコンピュータのプロパティを開きます。



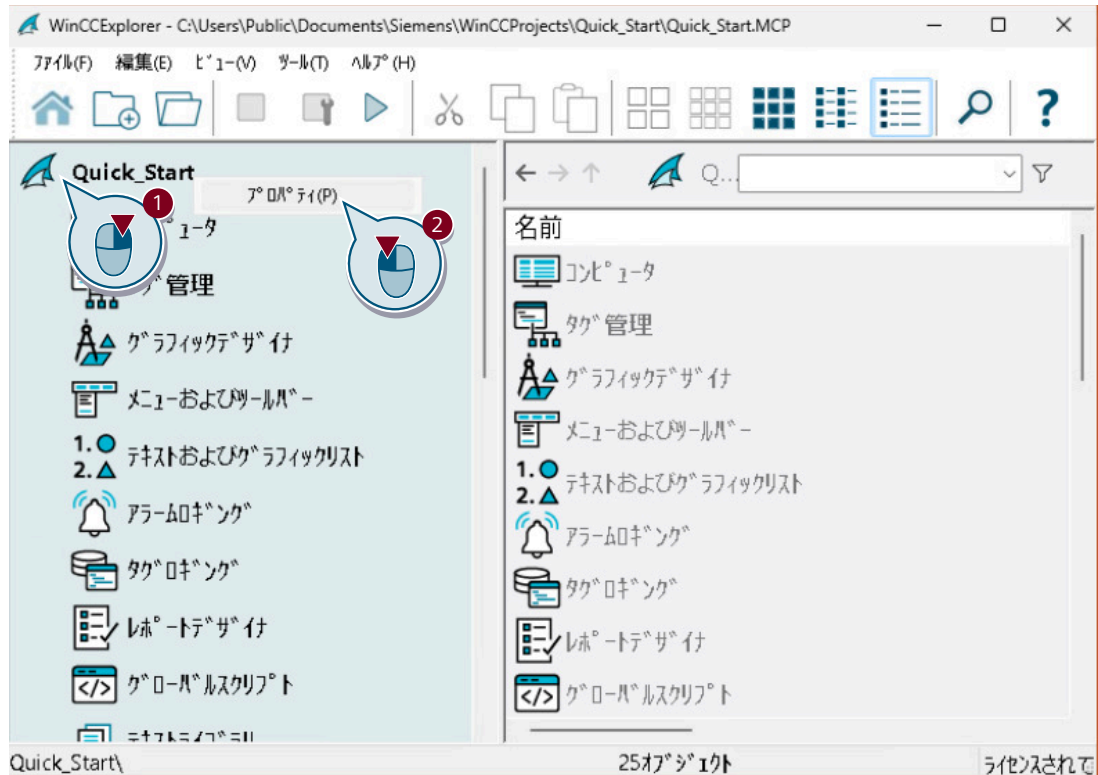
[コンピュータ]エディタが開かれます。

4. [ランタイム]タブの[ランタイムオプション]エリアで、[ランタイムシステムダイアログの有効化]オプションを有効にします。



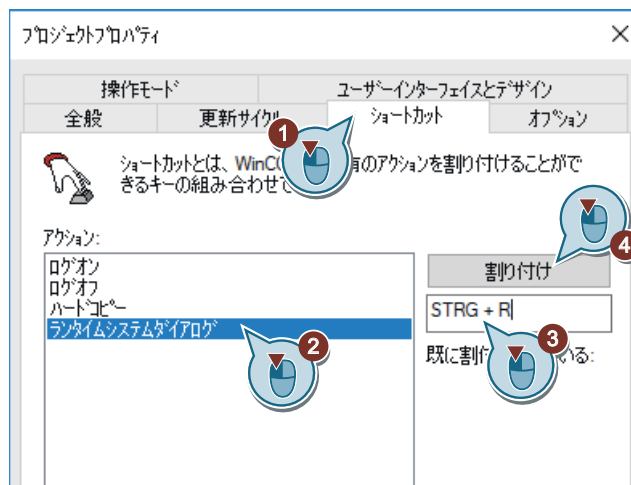
6.6 ランタイム中のプロジェクトの実行

5. 「Quick_Start」のショートカットメニューによりプロジェクトプロパティを開きます。



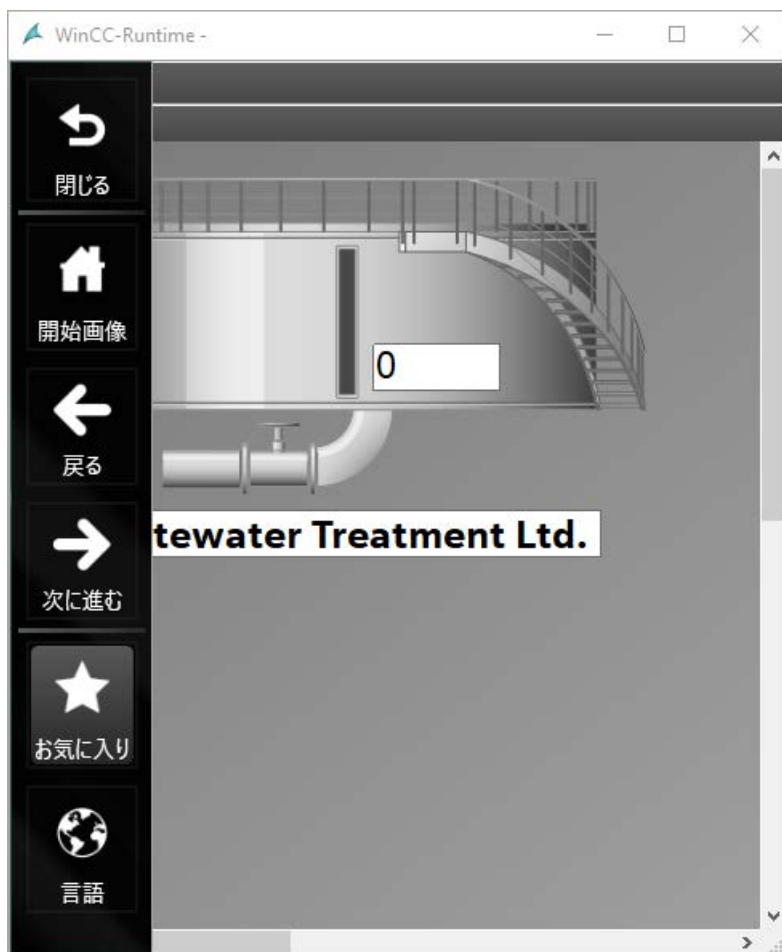
プロジェクトプロパティが開きます。

6. [ショートカット]タブで、[ランタイムシステムダイアログ]アクションに<Ctrl+R>などのキーボードショートカットを割り付けます。



7. "Quick_Start"プロジェクトを有効にします。
短いロード時間の後に、ランタイムウィンドウが開きます。「START.Pdl」プロセス画像が表示されます。

8. <Ctrl+R>などの割り付けられたキーボードショートカットを使ってランタイムシステムダイアログを起動します。
ランタイムウィンドウが開き、システムメニューが表示されます。



9. [お気に入り]および[SAMPLE]プロセス画像をクリックします。

6.6 ランタイム中のプロジェクトの実行



ランタイムシステムダイアログが非表示になり、「SAMPLE.Pdl」プロセス画像が表示されま
す。

結果

ランタイムシステムダイアログを有効にし、テストしました。

システムダイアログを使用して作成したプロセス画像間をナビゲートしたり、WinCC
Runtime の言語を変更したりすることができます。

フェイスプレートタイプとタグ構造の設定

7.1 フェイスプレートタイプの設定

概要

このセクションは、フェイスプレートタイプと構造タグの使用方法に関する情報を提供し、プロセス画像のために再利用可能なカスタマイズされているテンプレートを設定できるようにします。

フェイスプレートタイプと構造タグを使用すると、複数の使用に対してオブジェクトとタグを設定できます。

フェイスプレートのタイプは個別ファイルとして作成されます。WinCC でこれらのファイルの管理は、プロセス画像の処理に対応しています。

原理

フェイスプレートタイプ

プロセス画像とまったく同じ方法で、[グラフィックデザイナー]エディタのフェイスプレートタイプを設定します。スタティックで動的かつ操作可能なオブジェクトを配置し、それらをタグにリンクします。これにより、プロセス画像のフェイスプレートインスタンスとして好きな頻度で使用できるテンプレートが生成されます。このタイプインスタンスモデルにより、設定の労力が大幅に軽減されます。

複数の使用の場合、フェイスプレートタイプを構造タイプとリンクすることにより匹敵する個別のタググループをリンクします。

タグ管理:構造タイプ

構造タイプは、タグの特定のコンパイルーションのスキームのためのテンプレートです。構造タイプは別の構造タグに割り付けられます。この結果、これらの構造タグは常に、構造タイプと同じ要素で構成されますが、スタンドアロンの構造インスタンスとして構成されます。

構造タイプがフェイスプレートタイプとリンクされている場合、ドラッグアンドドロップで構造インスタンスをプロセス画像へドラッグすることにより、フェイスプレートインスタンスを配置します。

実際には、構造タイプと関連するインスタンスは、PLC またはサードパーティのアプリケーションからインポートされ、手動で個別に設定する必要がありません。

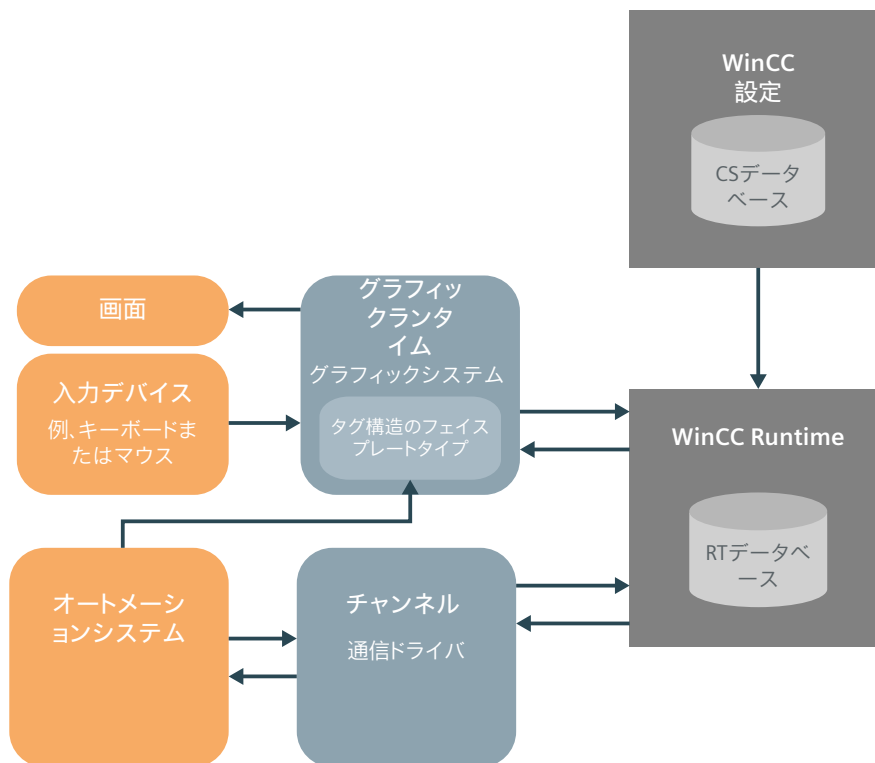
7.1 フェイスプレートタイプの設定

データレコードを TIA Portal プロジェクトからエクスポートするには、「SIEMENS SIMATIC SCADA Export」 ツールを使用します。

Industry Online Support ページで、ツールをダウンロードしたり、追加情報を取得したりできます。

- Industry Online Support: 「TIA Portal の SIMATIC SCADA Export」 をダウンロード(ID 109748955) (<https://support.industry.siemens.com/cs/ww/en/view/109748955>)
- Industry Online Support: 「SIMATIC SCADA Export」 ドキュメント(ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

WinCC システムのフェイスプレートタイプとタグ構造



「Quick_Start」プロジェクトの設定

「START.Pdl」プロセス画像は、2つのバルブがあります。これらはグラフィックな表示以外は、これまで何の役割も担っていません。

プロジェクトが拡張され、各バルブをクリックすると、ポップアップ画面が表示されるようになりました。ポップアップ画面は、温度や流量などの追加情報を表示します。フェイ

スプレートタイプや構造タイプを使用して、関連のタグやポップアップ画面を複数回作成しなければならなくなることを回避します。

このセクションでは、次のものを作成します。

- 1つの構造ラベルタイプと2つの構造タグ
- 構造タイプとリンクされている1つのフェイスプレートタイプ
- フェイスプレートタイプとリンクされている1つのポップアップ画像

作成されているフェイスプレートタイプをプロセス画像のフェイスプレートインスタンスとして使用します。

フェイスプレートタイプの付加価値を示すために、バルブ状態のカラーによるビジュアル化をプロジェクトに追加します。

下記も参照

Industry Online Support: 「SIMATIC SCADA Export」ドキュメント(ID 101908495) (<https://support.industry.siemens.com/cs/ww/en/view/101908495>)

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

7.2 構造タグの作成

7.2.1 構造タイプの作成

概要

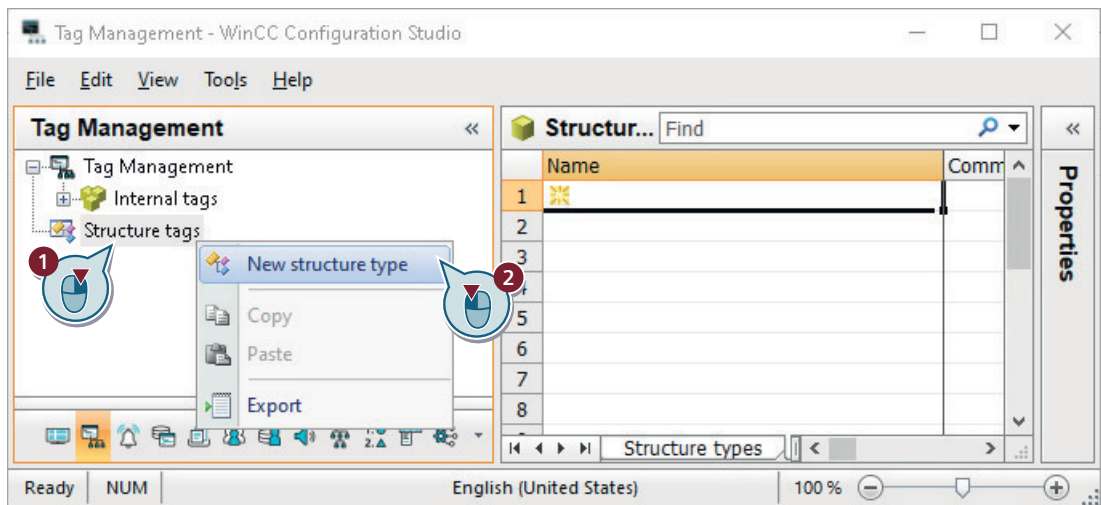
次のステップは、構造タイプと関連の構造タイプ要素を作成する方法を示しています。

必要条件

- "Quick_Start"プロジェクトが開いている。

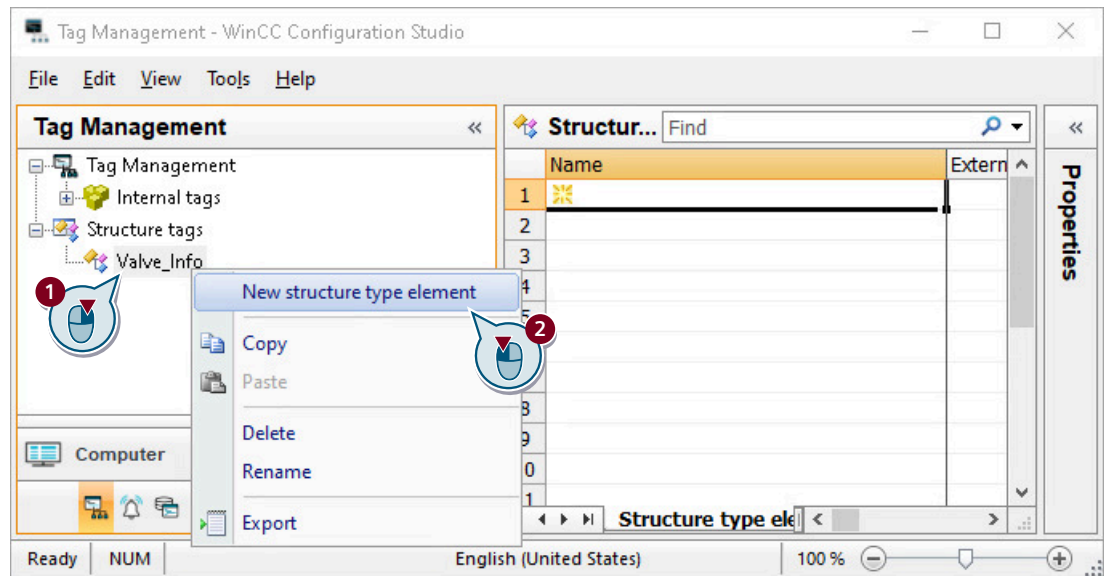
手順

1. 新しい構造タイプをタグ管理の[構造タグ]エントリで作成します。



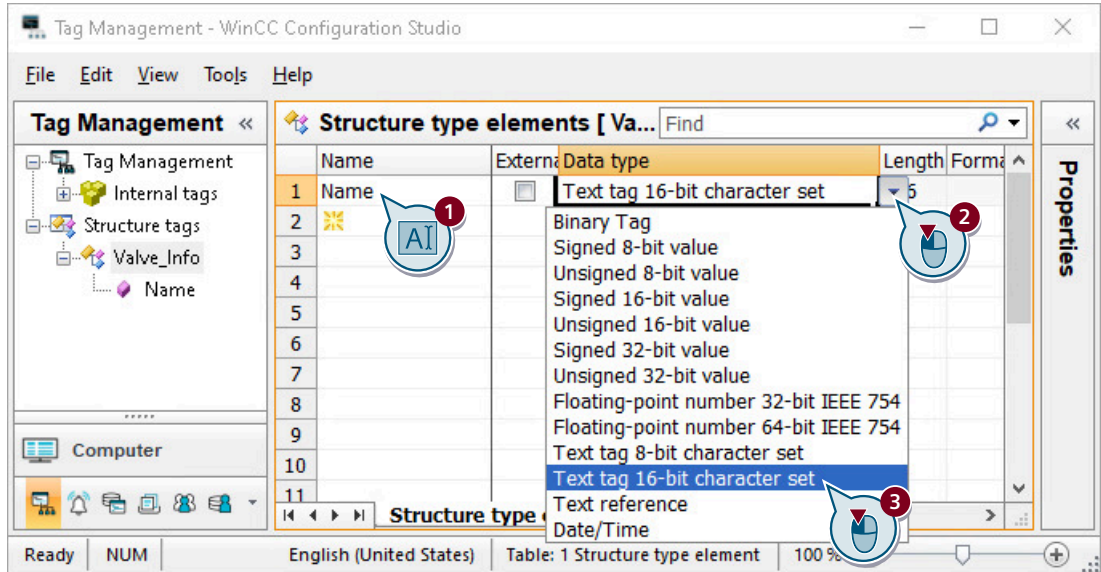
2. 構造タイプの名前として「Valve_Info」を入力します。

3. 「Valve_Info」構造タイプの構造タイプ要素を作成します。



7.2 構造タグの作成

- 「Name」を名前として入力して、データタイプ[テキストタグ 16 ビット文字セット]を選択します。



- ショートカットメニューにより次のプロパティの追加の構造タイプ要素を作成します。

名前	データタイプ
ステータス	2 進タグ
温度	浮動小数点数 64 ビット IEEE 754
圧力	浮動小数点数 64 ビット IEEE 754
流量	浮動小数点数 64 ビット IEEE 754

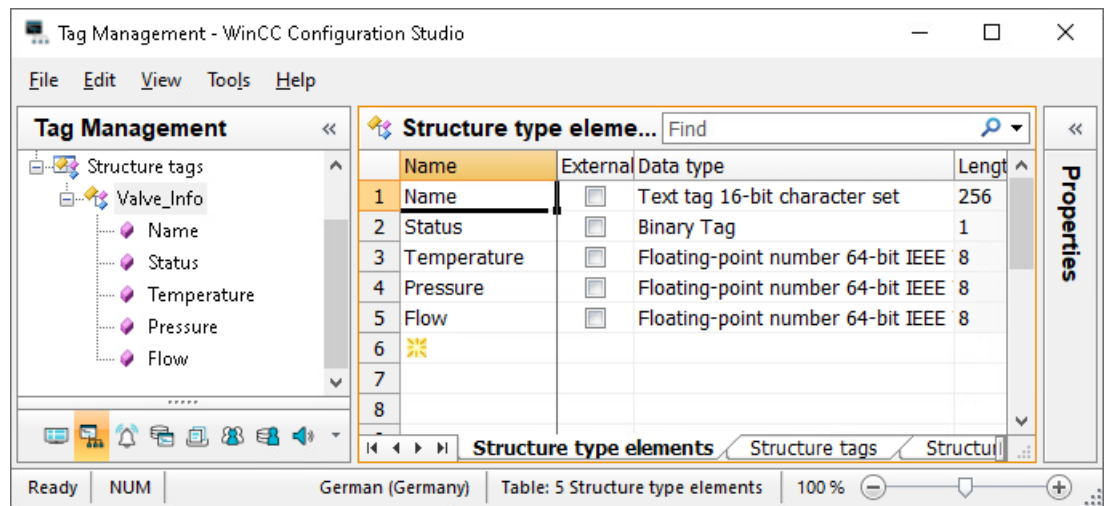
あるいは、次に利用可能なフィールドをクリックして、構造タイプ要素の名前を直接入力します。

シングルアクションで複数の構造タイプを作成するには、「Name」という名前のフィールド内をクリックして、フィールドの右下の隅をマウスで下にドラッグします。Microsoft Excel のオートフィル機能と同様に、行の列エントリは下の行に適用されます。

結果

「Valve_Info」構造タイプを作成し、5つの構造タイプ要素を定義しています。

「Valve_Info」構造タイプはタグ管理の[構造タグ]の下に表示されます。



次のステップ

定義されている構造タイプに基づいて構造タグを作成します。

7.2.2 構造タグの作成

概要

次のステップは、前に作成した構造タイプとデータタイプを使用する構造タグを作成する方法を示しています。

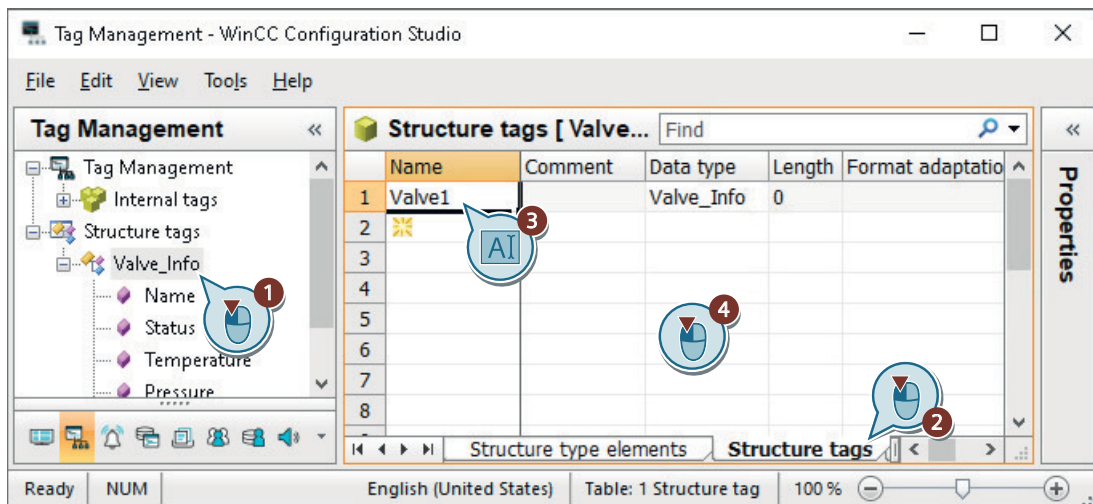
必要条件

- 「Valve_Info」構造タイプが作成され、定義されている。

7.2 構造タグの作成

手順

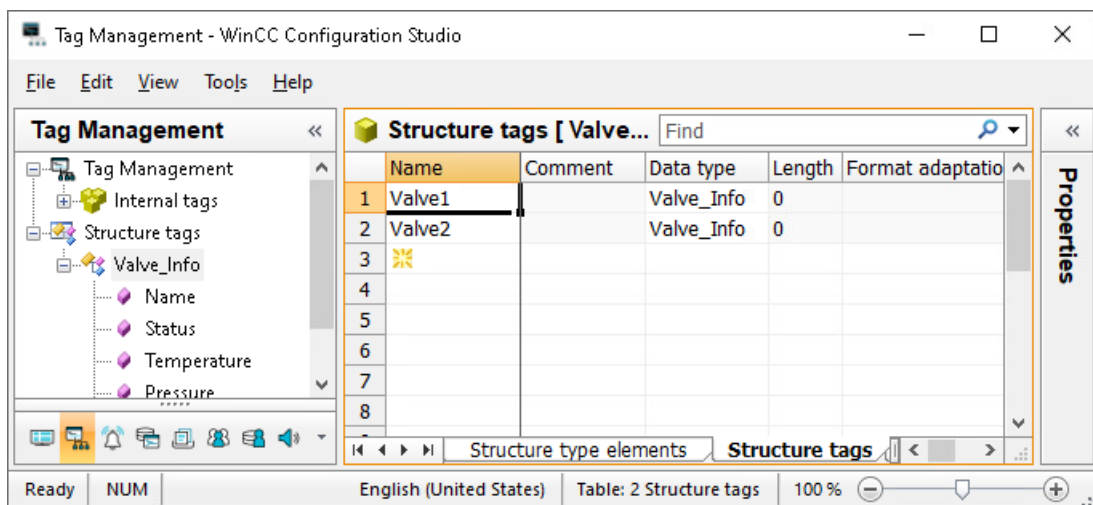
1. [構造タグ]タブで、「Valve_Info」構造タイプのために構造タグ「Valve1」を作成します。



2. 利用可能なフィールドをクリックします。
これまでに 1 つの構造データタイプしか作成されていないため、「Valve_Info」が[データタイプ]列に自動的に入力されます。
3. 「Valve2」という名前の別の構造タグを作成します。

結果

「Valve_Info」構造タイプに基づいて 2 つの構造タグを作成しています。



タグのインスタンスとして、関連の構造タグ要素が自動的に作成されます。

Tag Management - WinCC Configuration Studio

File Edit View Tools Help

Tag Management <<

- Tag Management
 - Internal tags
 - Structure tags
 - Valve_Info
 - Name
 - Status
 - Temperature
 - Pressure
 - Flow

Structure tag elements [Valve_Info] Find

	Name	Data type	Length	Format adaptation	Connection
1	Valve1.Flow	Floating-point number 64-bit IEEE	8		Internal tags
2	Valve1.Name	Text tag 16-bit character set	256		Internal tags
3	Valve1.Pressure	Floating-point number 64-bit IEEE	8		Internal tags
4	Valve1.Status	Binary Tag	1		Internal tags
5	Valve1.Temperature	Floating-point number 64-bit IEEE	8		Internal tags
6	Valve2.Flow	Floating-point number 64-bit IEEE	8		Internal tags
7	Valve2.Name	Text tag 16-bit character set	256		Internal tags
8	Valve2.Pressure	Floating-point number 64-bit IEEE	8		Internal tags
9	Valve2.Status	Binary Tag	1		Internal tags
10	Valve2.Temperature	Floating-point number 64-bit IEEE	8		Internal tags
11					

Structure type elements Structure tags **Structure tag elements**

Ready NUM English (United States) Table: 10 Tags 100%

7.3 ポップアップ画面の設定

7.3.1 ポップアップ画面の作成とそのサイズの適合

概要

次のステップは、フェイスプレートインスタンスをクリックすると開くポップアップ画面の作成方法を示しています。

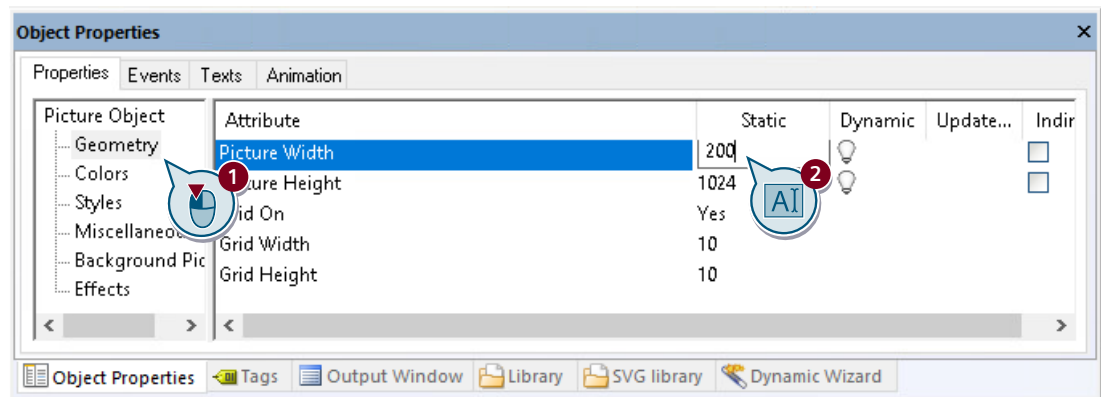
ポップアップ画面で下のコンテンツが非表示になるのを防止するために、プロセス画像のサイズを適合します。

必要条件

- プロセス画像「POPUP.Pdl」が作成されていること。
プロセス画面の作成 (ページ 51) で説明されている手順に従ってください。

手順

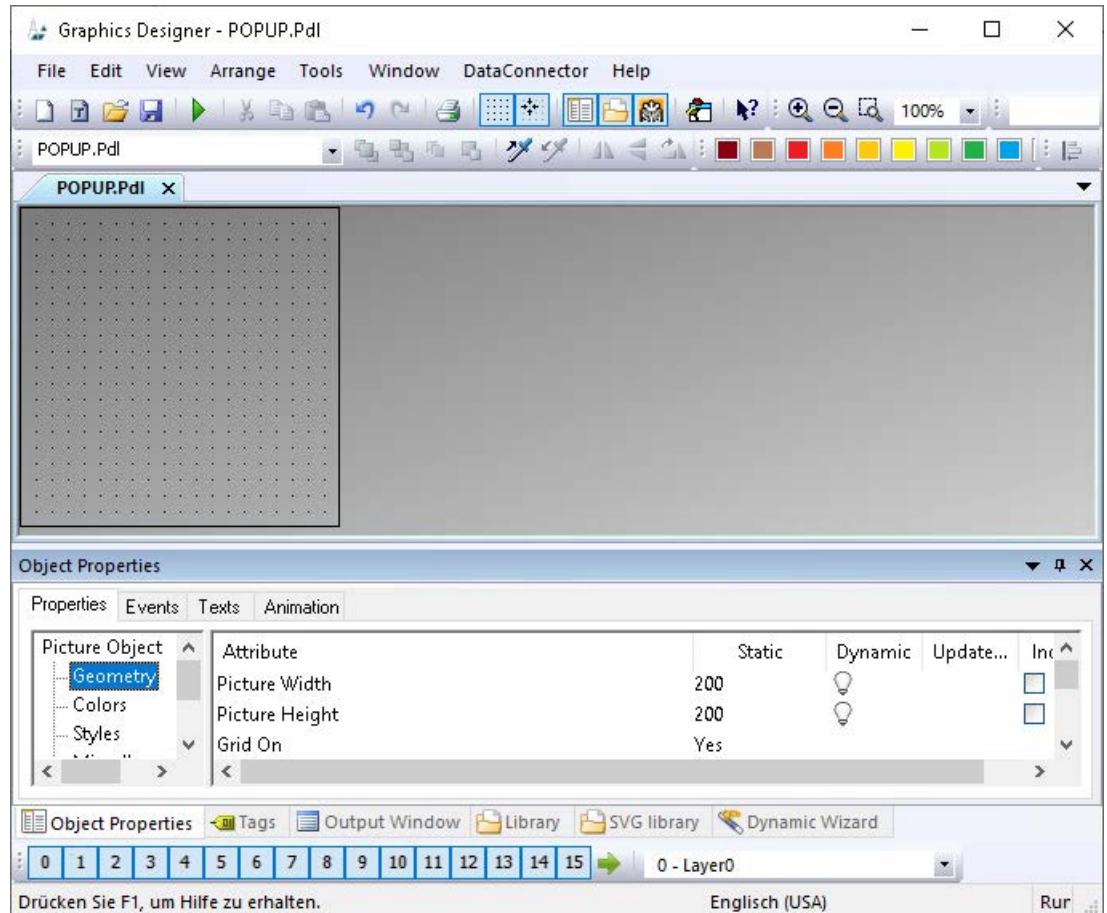
- [グラフィックデザイナー]エディタで、「POPUP.Pdl」プロセス画像を開きます。
- [ジオメトリ]グループを[オブジェクトプロパティ]ウィンドウで開き、[画像幅]を値「200」に設定します。



- ステップ 2 と同様に、[画像高さ]を値「200」に設定します。

結果

プロセス画像の寸法を 200 x 200 ピクセルに変更しています。



次のステップ

「POPUP.Pdl」プロセス画像にすべての必要なオブジェクトを配置します。

7.3.2 ポップアップ画面にオブジェクトに配置

概要

次のステップは、必要なオブジェクトをポップアップ画面に挿入する方法を示しています。

7.3 ポップアップ画面の設定

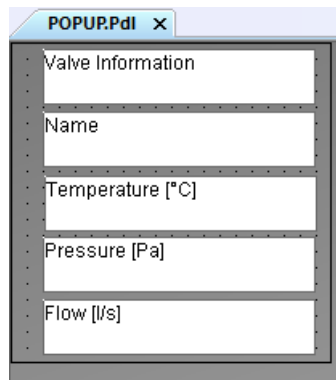
スタティックテキストをラベルとして使用し、プロセス値を出力するために I/O フィールドを使用します。

必要条件

- 「POPUP.Pdl」 プロセス画像が開いていること。

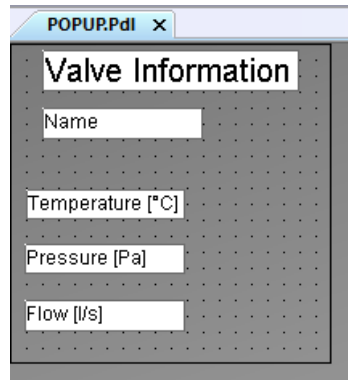
手順

1. [スタティックテキスト]タイプの 5 つのオブジェクトを挿入します。
プロセス画面の編集 (ページ 55) に説明されている手順に従ってください。
2. 次のテキストを入力します。
 - バルブの情報
 - 名前
 - 温度[°C]
 - 圧力[Pa]
 - 流量[l/s]

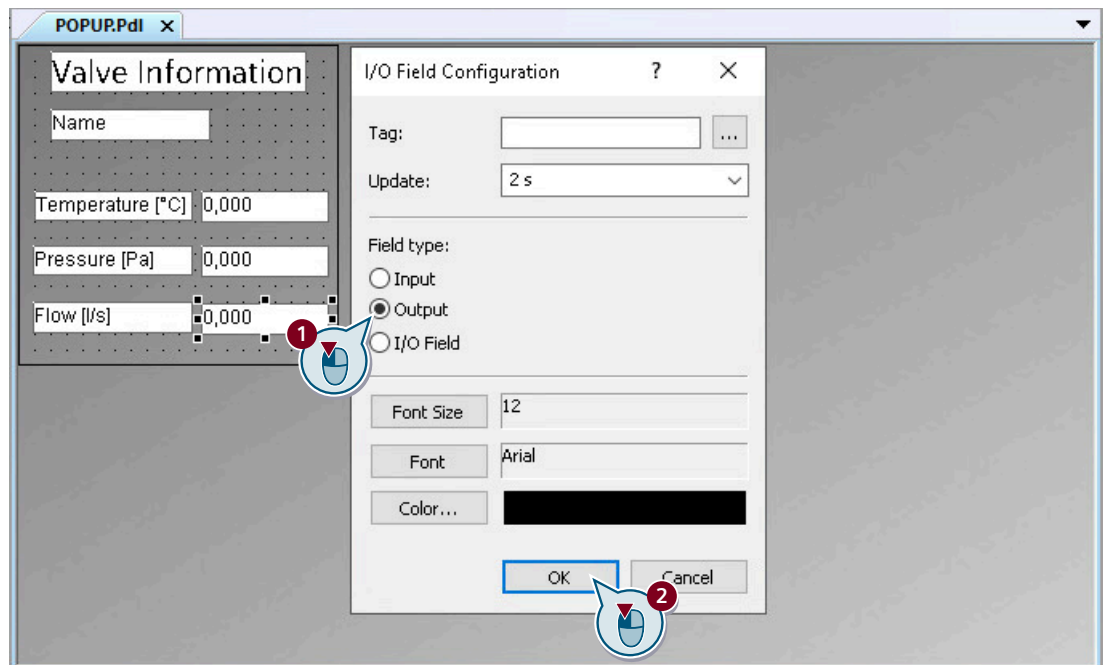


3. テキストボックスでフォントサイズを定義します。

4. テキストボックスのサイズと位置を適合します。



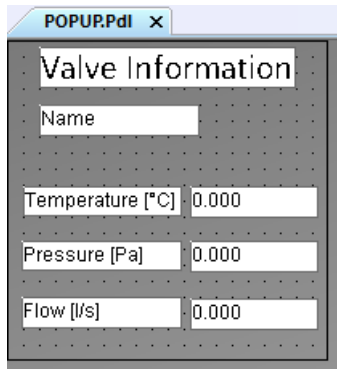
5. [I/O フィールド]タイプの3つのオブジェクトを挿入し、各ケースのフィールドタイプとして[出力]を選択します。[タグ]フィールドを空にしておきます。



7.3 ポップアップ画面の設定

結果

ポップアップ画面にすべての必要なオブジェクトを挿入しています。



次のステップ

挿入されているオブジェクトをダイナミック化します。

7.3.3 ポップアップ画面でオブジェクトをダイナミック化

概要

次のステップは、ポップアップ画面でオブジェクトをダイナミック化する方法を示しています。

レベルインジケータをダイナミック化するのに使用したのと同じ手順に従っていると、同じタグがすべてのインスタンスで出力されます。この場合、バルブのすべてのポップアップ画面は同じコンテンツになります。

別のバルブにたいして異なる値を表示するには、特定のタグではなく構造タイプ要素をリンクします。この場合、表記「.Temperature」をタグ名として使用します。

オブジェクトは2つのステップでダイナミック化します。

- スタティックテキスト「Name」のオブジェクトプロパティ[テキスト]のダイナミック化:
それぞれのバルブの名前のダイナミック表示
- I/O フィールドのオブジェクトプロパティ[出力値]のダイナミック化:
そのバルブのそれぞれのプロセス値のダイナミック表示

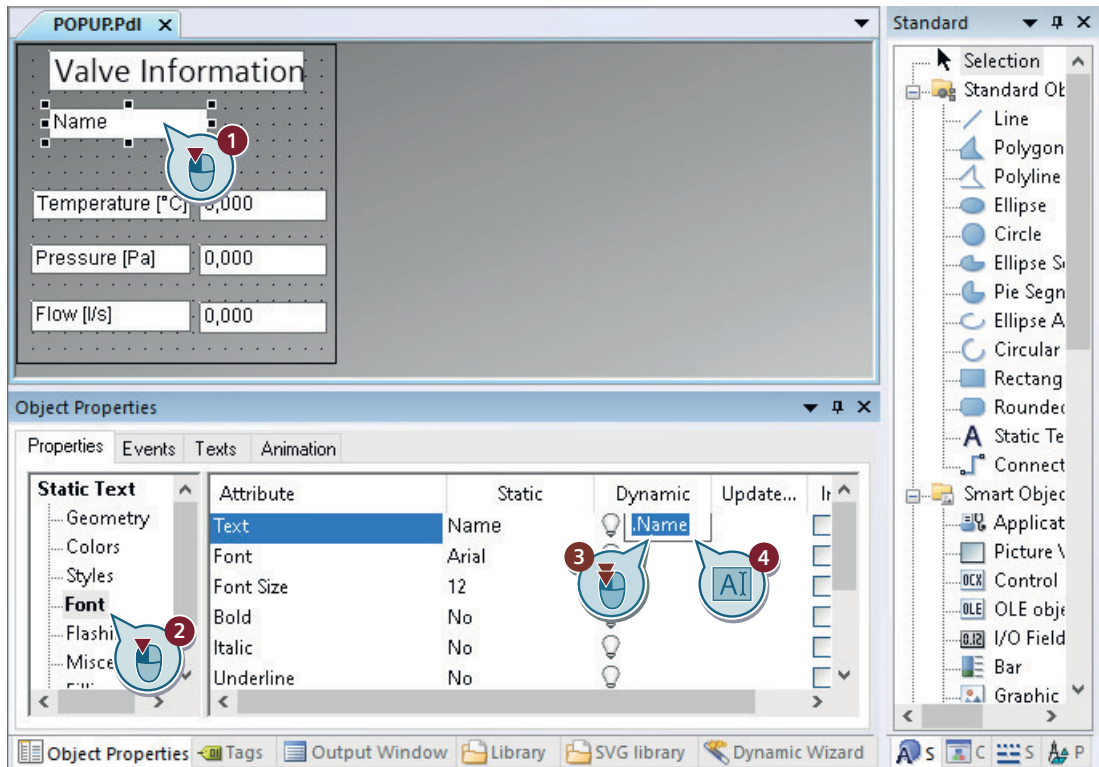
必要条件

- 「POPUP.Pdl」プロセス画像が開いていること。

7.3 ポップアップ画面の設定

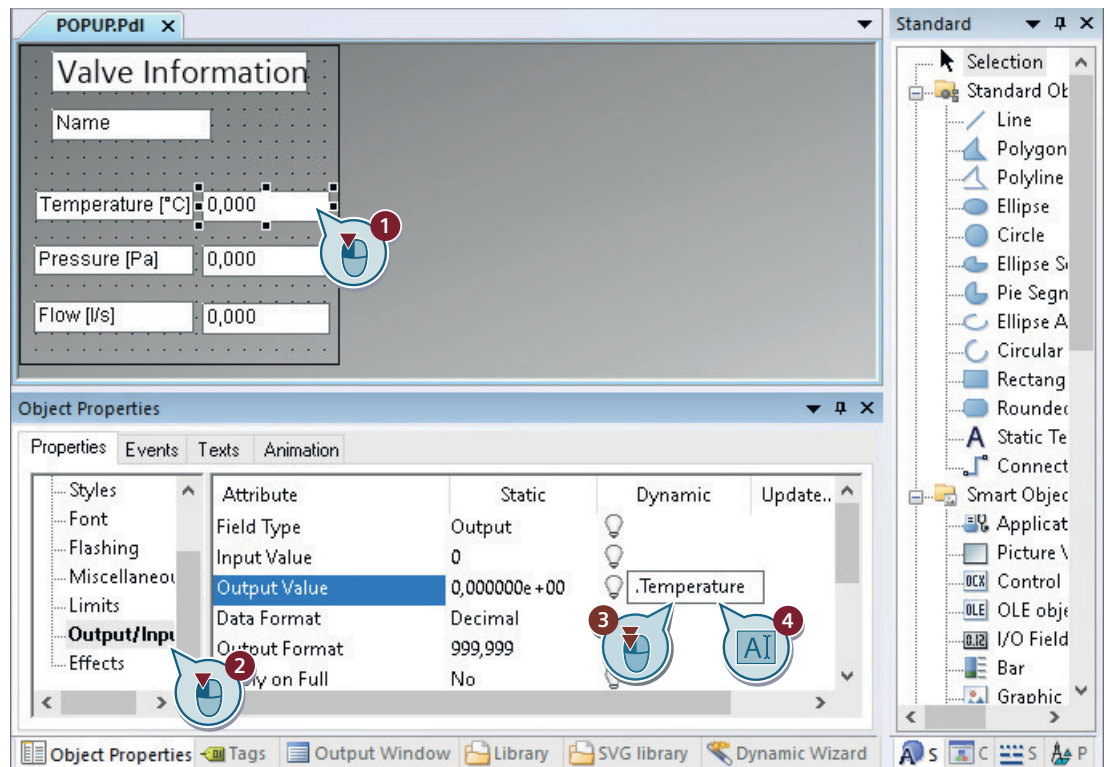
手順

1. 「Name」ラベルのある[スタティックテキスト]オブジェクトの場合、オブジェクトプロパティ[テキスト]を構造タグ要素「.Name」でダイナミック化します。



ライトの電球が緑色になります。テキストがダイナミック化されています。

2. 「Temperature [°C]」ラベルの横にある[I/O フィールド]オブジェクトの場合、オブジェクトプロパティ[出力]値を構造タグ要素「.Temperature」でダイナミック化します。



ライトの電球が緑色になります。出力値がダイナミック化されています。

3. 他の2つのI/Oフィールドに対してステップ2を繰り返します。
ダイナミック化のために次の構造タグ要素を使用します。
 - ".Pressure"
 - ".Flow"

結果

ポップアップ画面のすべてのオブジェクトをダイナミック化しています。

7.4 フェイスプレートタイプの設定

7.4.1 フェイスプレートタイプの作成とオブジェクトの配置

概要

次のステップは、フェイスプレートを作成し、必要なオブジェクトを配置する方法を示しています。

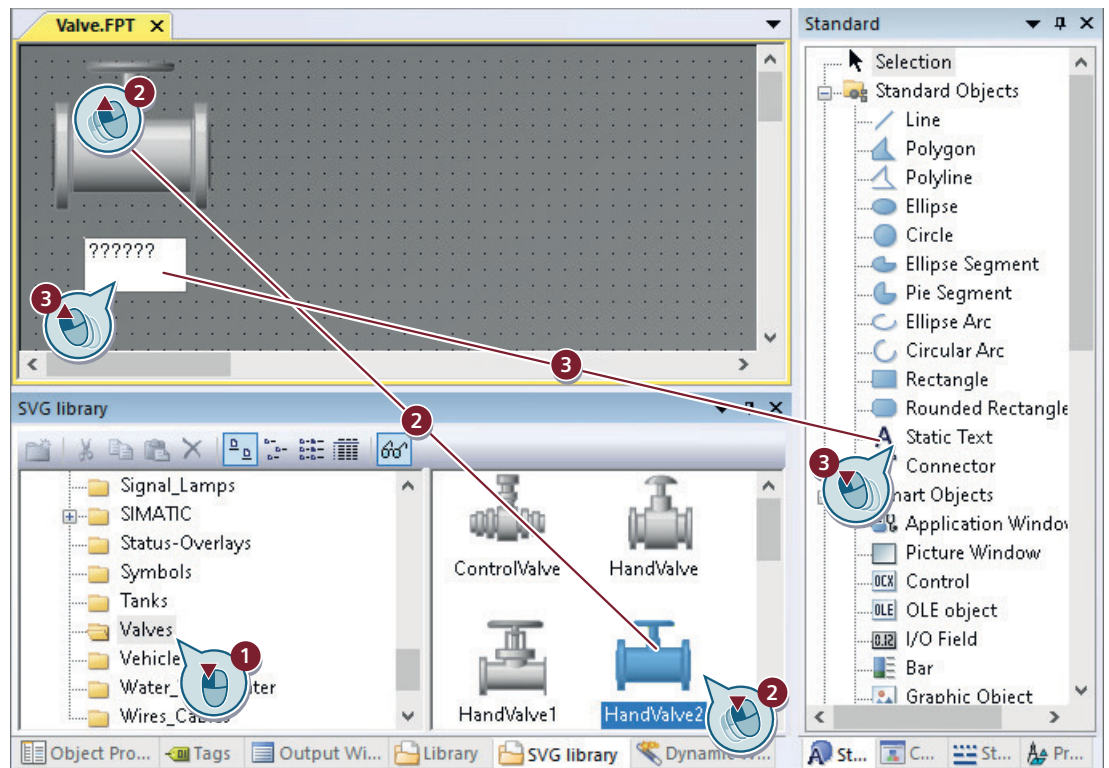
必要条件

- [グラフィックデザイナー]エディタが開いている。

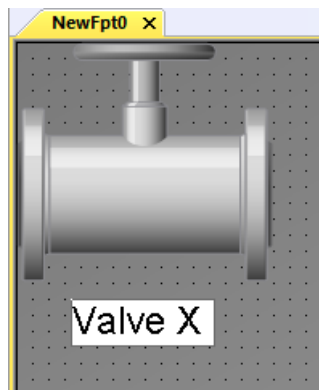
手順

1. [ファイル]メニューの[新規フェイスプレートタイプ]エントリを選択します。
*.FPT 形式で新しい画像が開きます。
2. [ファイル]メニューの[保存]エントリを選択します。
3. フェイスプレートタイプに「Valve.Fpt」という名前を付けます。
[保存]を押して確定します。
「Valve.Fpt」フェイスプレートタイプが「GraCS」プロジェクトフォルダに作成されています。
4. グローバル SVG ライブラリ"IndustryGraphicLibraryV2.0"を[SVG ライブラリ]ウィンドウで選択します。
5. [HandValve2]オブジェクトを[Valves]フォルダから挿入します。

6. [スタティックテキスト]タイプのオブジェクトを挿入し、テキスト「Valve X」をラベルとして入力します。

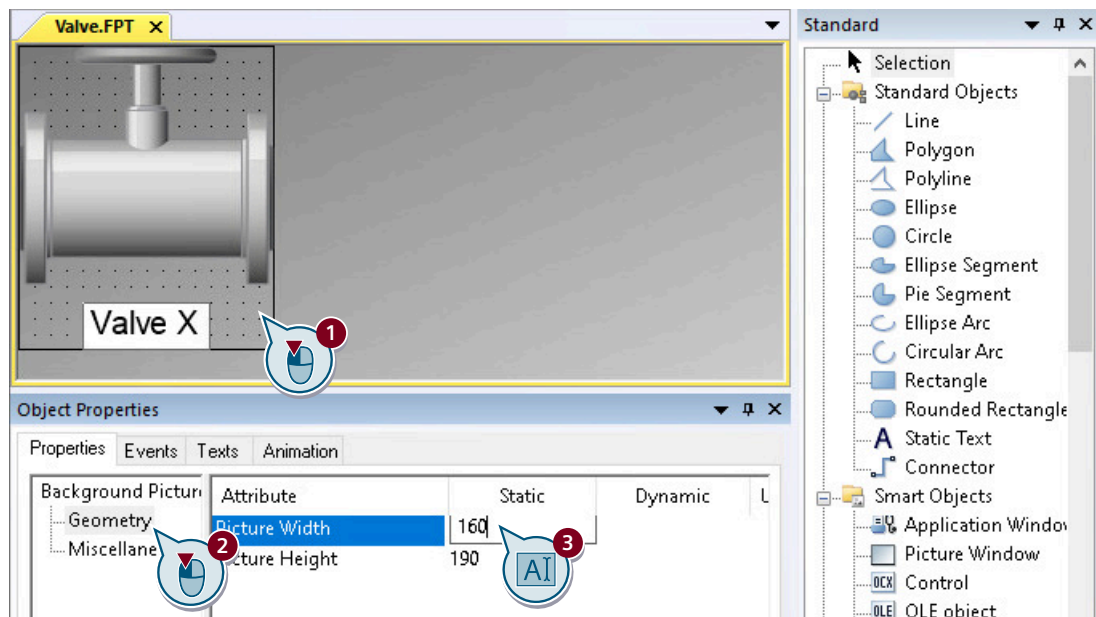


7. オブジェクトの位置を適合します。



7.4 フェイスプレートタイプの設定

8. 選択されているオブジェクトがない状態になるように、背景をクリックします。プロセス画像のオブジェクトプロパティが表示されます。
9. [画像幅]を値「160」に、[画像高さ]を値「190」に設定します。



結果

[バルブ]フェイスプレートタイプを作成し、すべての必要なオブジェクトを配置しています。

次のステップ

「Valve_Info」構造タイプを[バルブ]フェイスプレートタイプとリンクします。

7.4.2 構造タイプとフェイスプレートタイプのリンク

概要

次のステップは、フェイスプレートタイプの設定方法を示しています。

フェイスプレートタイプを作成してある構造タグとリンクし、ポップアップ画面を構成します。

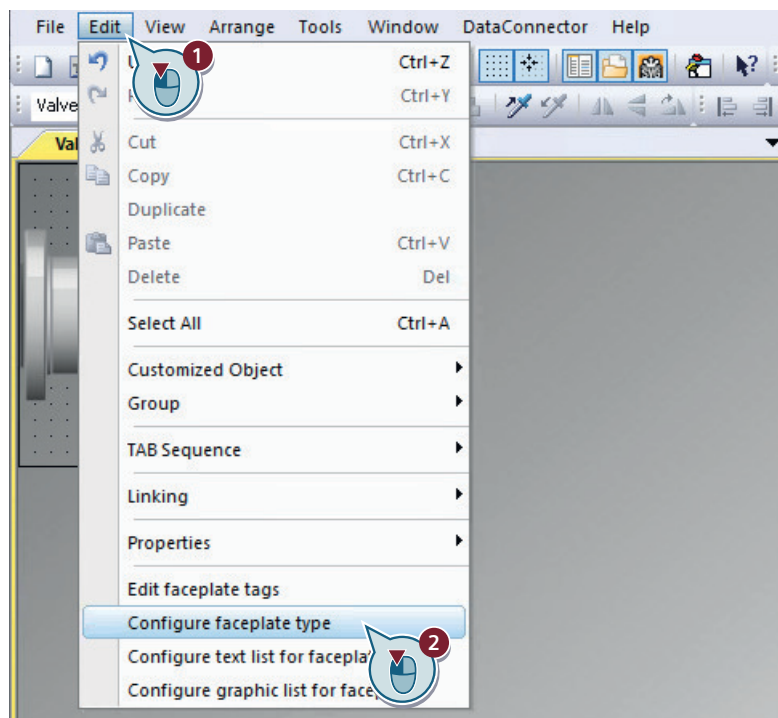
必要条件

- 「Valve.Fpt」フェイスプレートタイプが開かれていること。

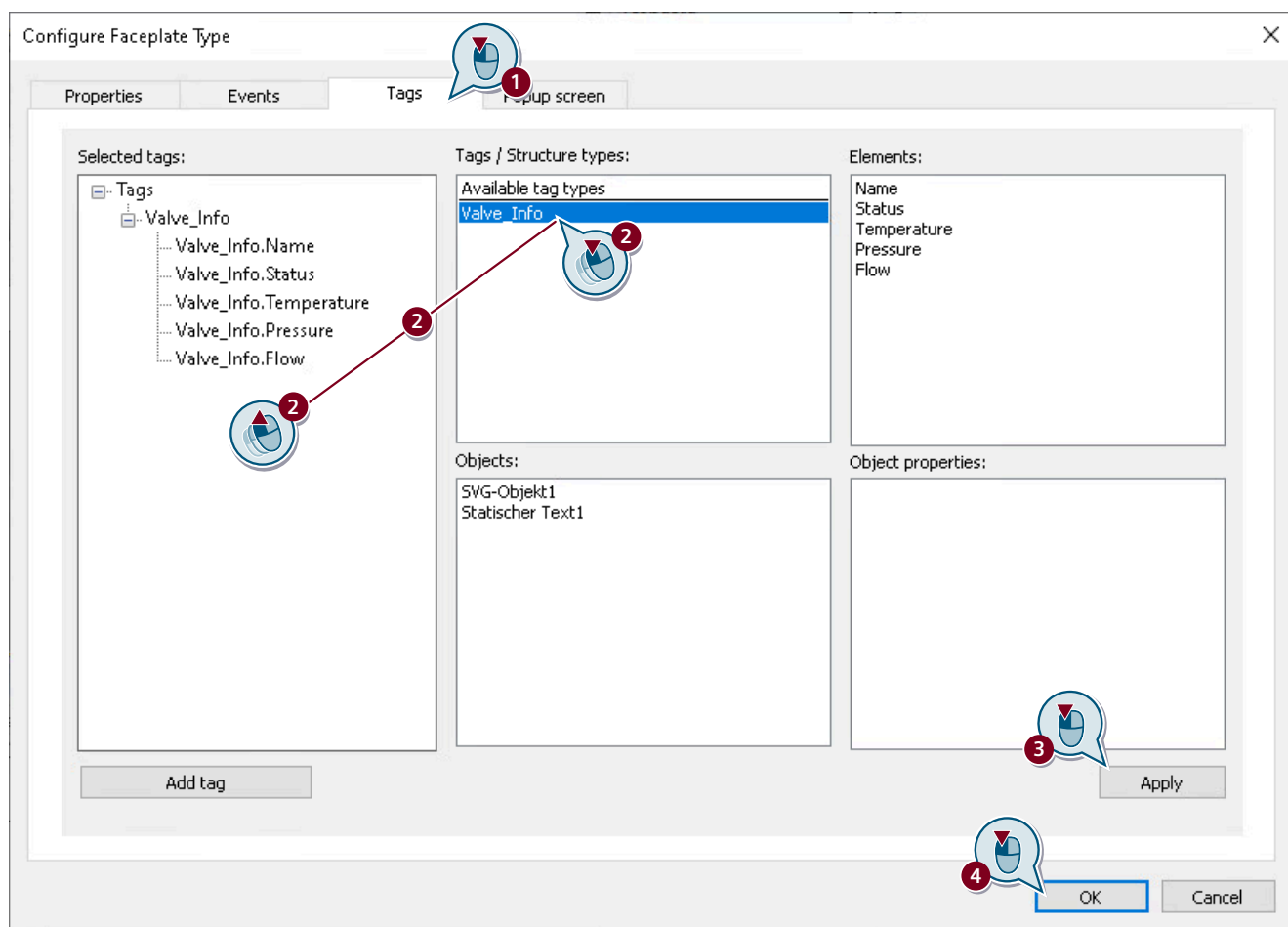
7.4 フェイスプレートタイプの設定

手順

1. フェイスプレートタイプのプロパティを定義するには、[設定]ダイアログを開きます。



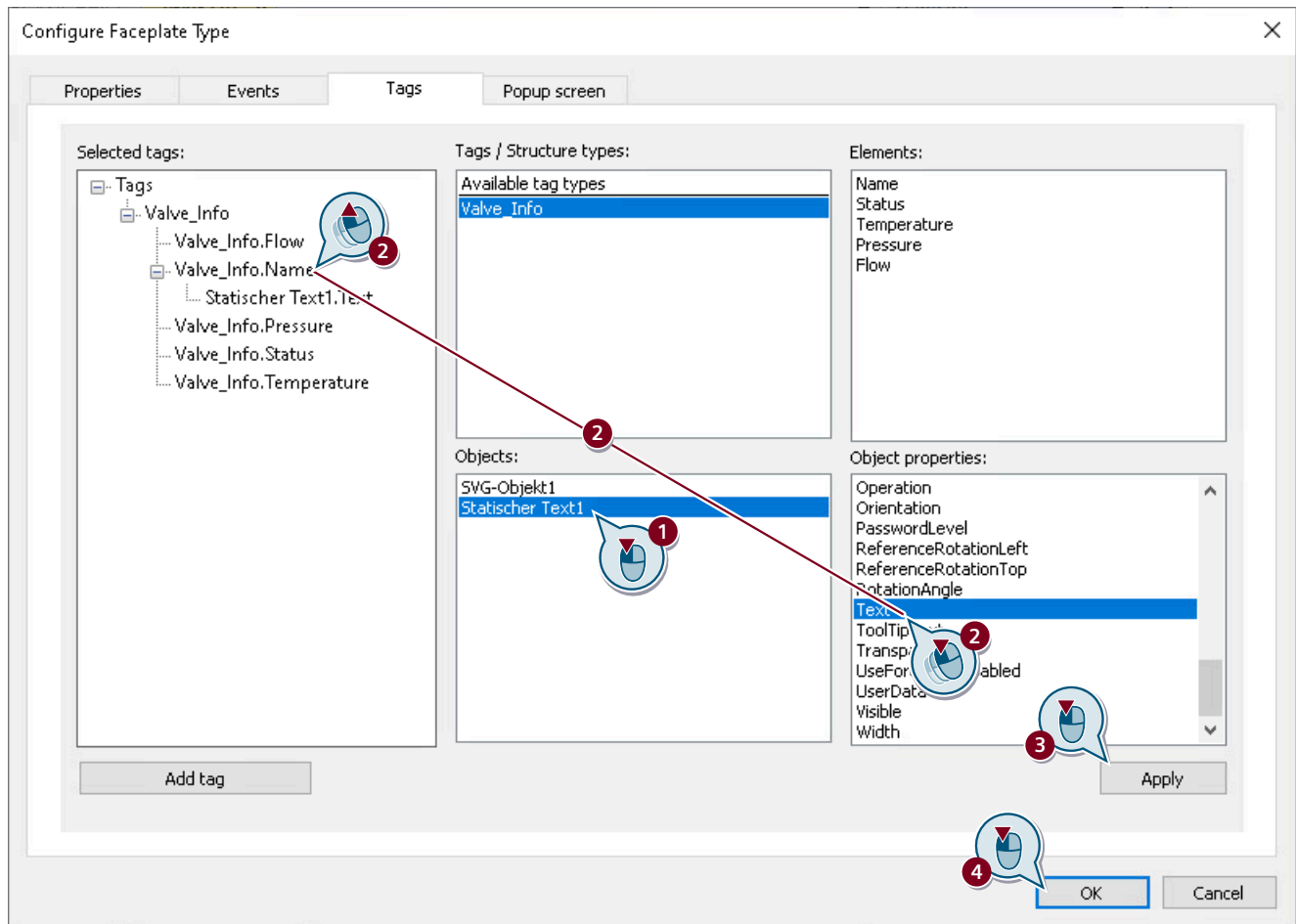
2. [タグ]タブで、「Valve_Info」構造タイプを選択されているタグとして設定します。



構造タイプがインターフェースタグとして定義されています。
フェイスプレートタイプのダイナミック化を行うために、インターフェースタグを使用します。

7.4 フェイスプレートタイプの設定

3. [テキスト]オブジェクトプロパティを「Valve_Info.Name」タグとリンクします。



4. 設定を保存するには、[適用]をクリックして、[はい]で確定します。

結果

作成した構造タイプをインターフェースタグとして定義し、タグを割り付けることによりテキストをダイナミック化しています。

次のステップ

「POPUP.Pdl」プロセス画像をポップアップ画面として設定します。

7.4.3 ポップアップ画面の設定

概要

次のステップは、フェイスプレートタイプでポップアップ画面を設定する方法を示しています。

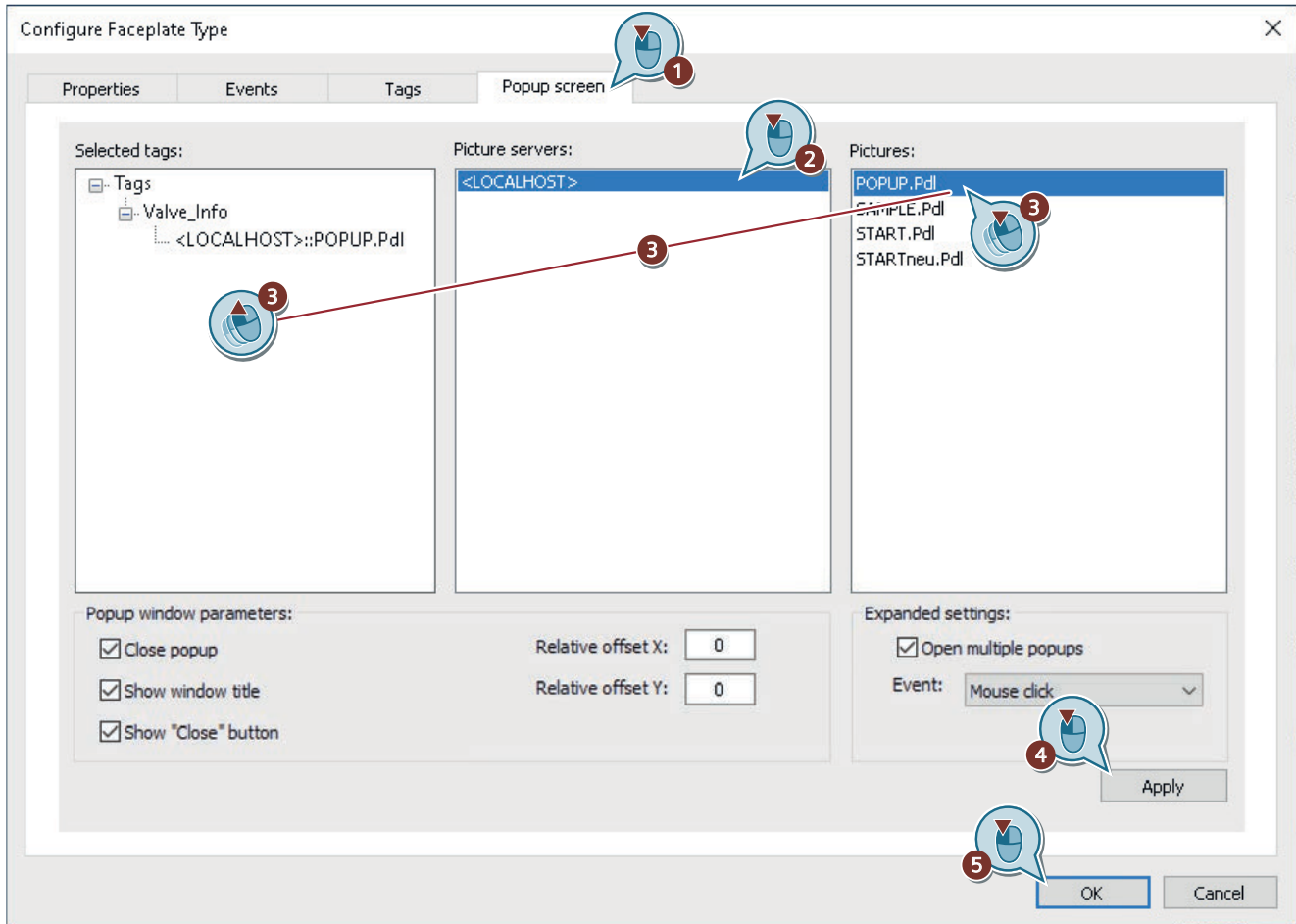
必要条件

- 「Valve.Fpt」フェイスプレートタイプが開かれていること。

7.4 フェイスプレートタイプの設定

手順

1. [編集]メニューの[フェイスプレートタイプの設定]エントリを選択することにより、設定ダイアログを開きます。
2. [ポップアップ画面]タブで「POPUP.Pdl」プロセス画像をポップアップ画面として設定します。



3. 設定を保存するには、[適用]をクリックして、[はい]で確定します。
画像がポップアップ画面として設定されています。

結果

「POPUP.Pdl」画像をポップアップ画面として設定しています。フェイスプレートインスタンスをクリックすると、ランタイムで画像が開きます。

関連するスクリプトが自動的に作成されます。

7.5 フェイスプレートインスタンスの作成

概要

次のステップは、フェイスプレートインスタンスをプロセス画像に挿入する方法を示しています。

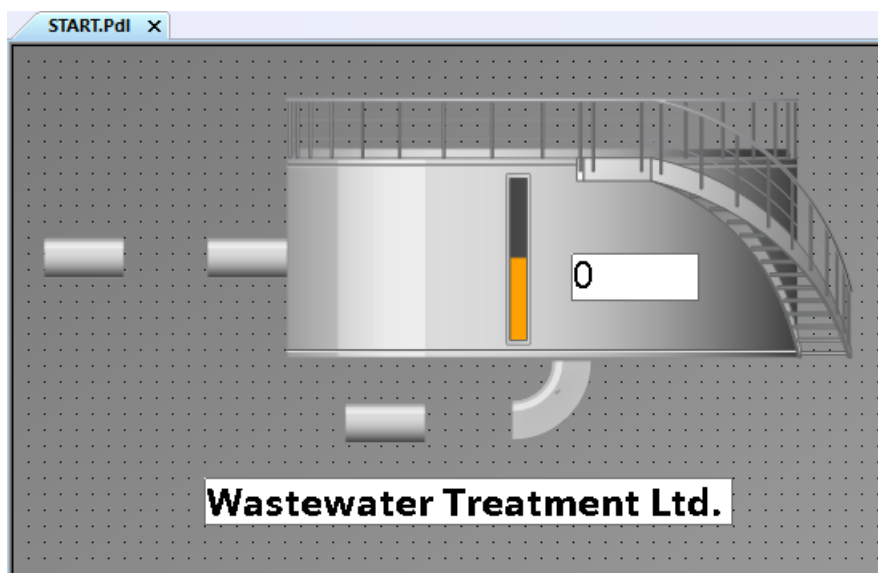
このために、ドラッグアンドドロップで構造タグをプロセス画像にドラッグします。

必要条件

- [グラフィックデザイナー]エディタが開いている。
- 「START.Pdl」プロセス画像が開いている。

手順

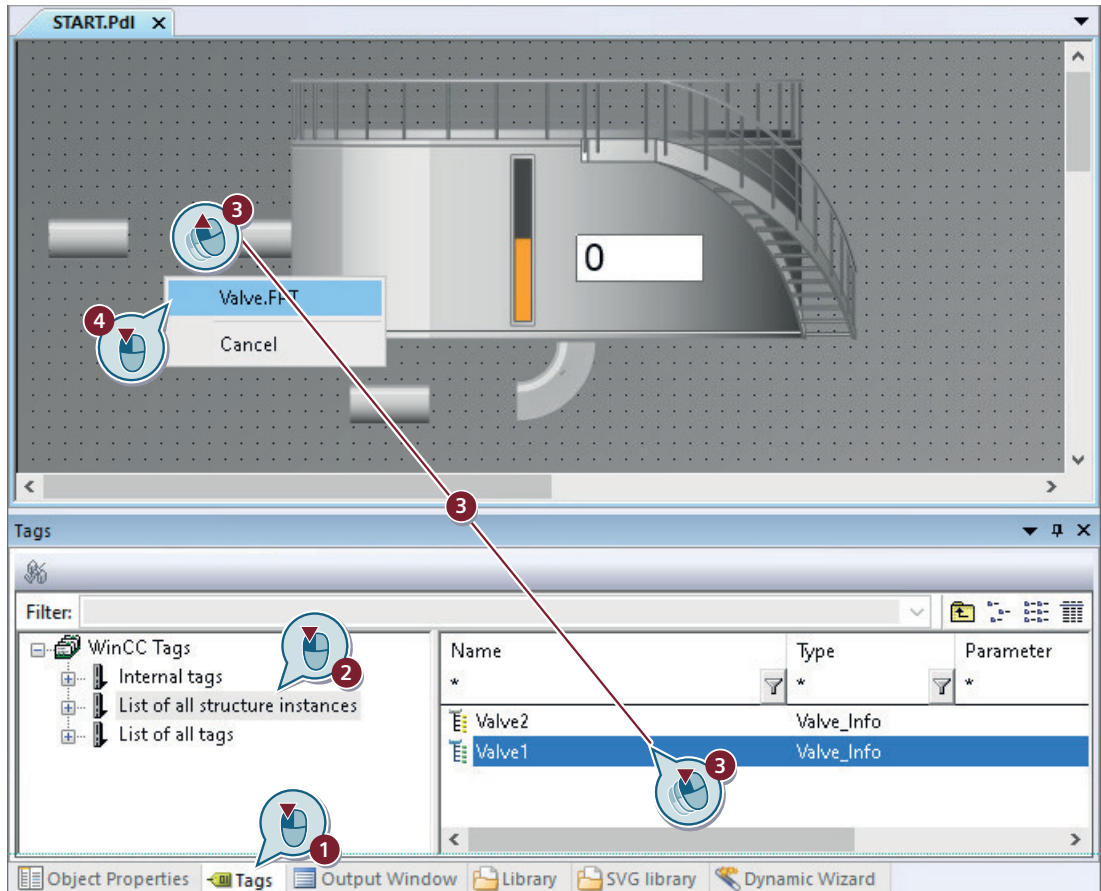
1. プロセス画像の2つのSVGオブジェクト「HandValve2」のそれぞれを右クリックして、[削除]を選択します。
2つのSVGオブジェクトが削除されます。



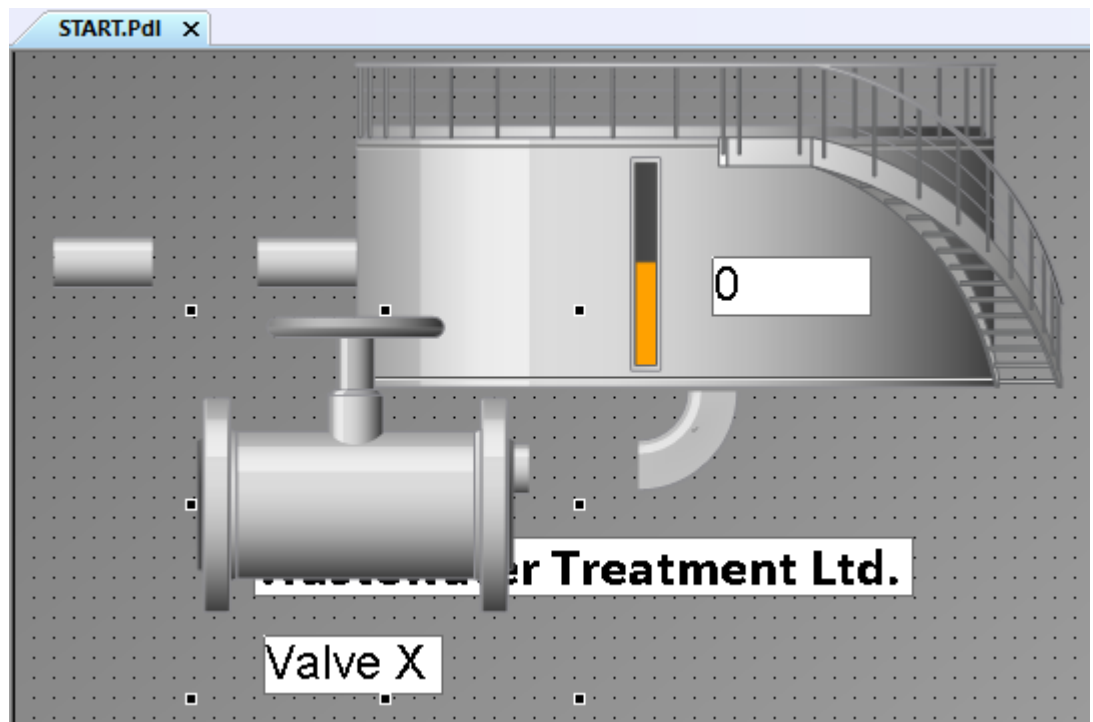
2. すべての構造インスタンスのリストを[タグ]選択ウィンドウに表示します。
[タグ]ウィンドウが表示されない場合、[ツールバー]の[ビュー]メニューの[タグ]エントリを有効にします。

7.5 フェイスプレートインスタンスの作成

3. ドラッグアンドドロップで「Valve1」構造インスタンスをプロセス画像にドラッグします。「Valve.Fpt」フェイスプレートタイプを選択します。

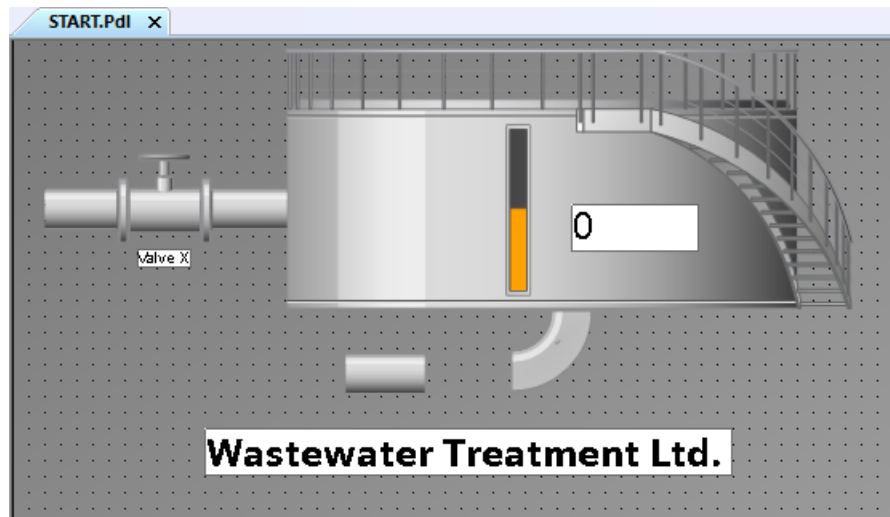


フェイスプレートインスタンスがプロセス画像に配置されていて、「Valve1」構造インスタンスの構造タグとダイナミック化されています。

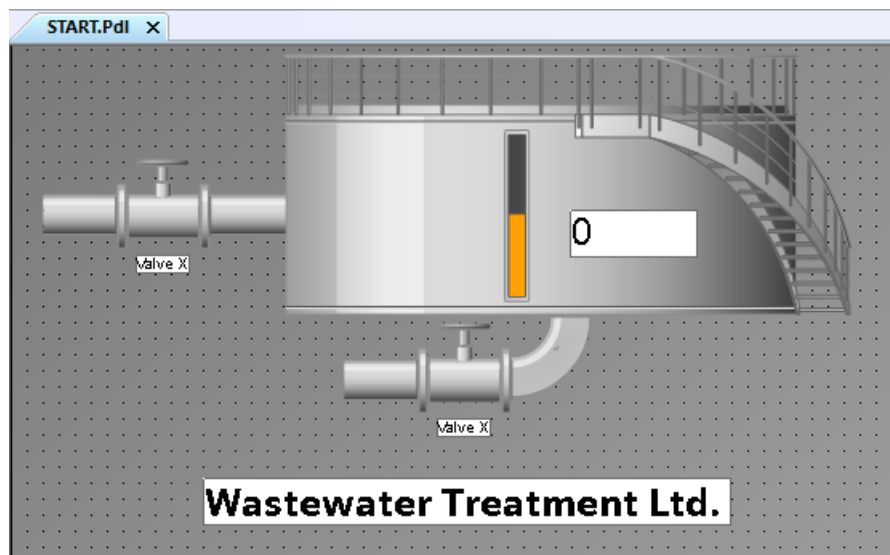


7.5 フェイスプレートインスタンスの作成

- フェイスプレートインスタンスの位置とサイズを適合します。



- 「Valve2」構造インスタンスに対してステップ3～4を繰り返します。



結果

構造インスタンスを挿入し、その結果としてプロセス画像で2つのフェイスプレートインスタンスを作成しています。

7.6 フェイスプレートタイプの変更

概要

このセクションは、フェイスプレートタイプを使用することにより、設定の労力を軽減した状態で変更を実装する方法について、情報を提供します。

原理

フェイスプレートタイプを使用する主なメリットは、容易で時間を節減できる形で変更を実装できることです。

フェイスプレートタイプを変更すると、変更はすべてのインスタンスに適用されます。これにより、個別オブジェクトの変更を時間をかけないで行うことができます。

「Quick_Start」プロジェクトの設定

さらに、「Valve.Fpt」フェイスプレートタイプは、カラーを使用することにより、バルブのステータスを示します。

タグを使用して、バルブの基本カラーをアニメーション化します。

フェイスプレートタイプでアニメーションを設定するため、変更はプロセス画像の両方のバルブに適用されます。

7.6.1 オブジェクトプロパティのアニメーション化

概要

次のステップは、オブジェクトプロパティのアニメーション化の方法を示しています。

希望のプロパティをタグとリンクします。オブジェクトプロパティが、タグ値の機能としてランタイムで変更されます。

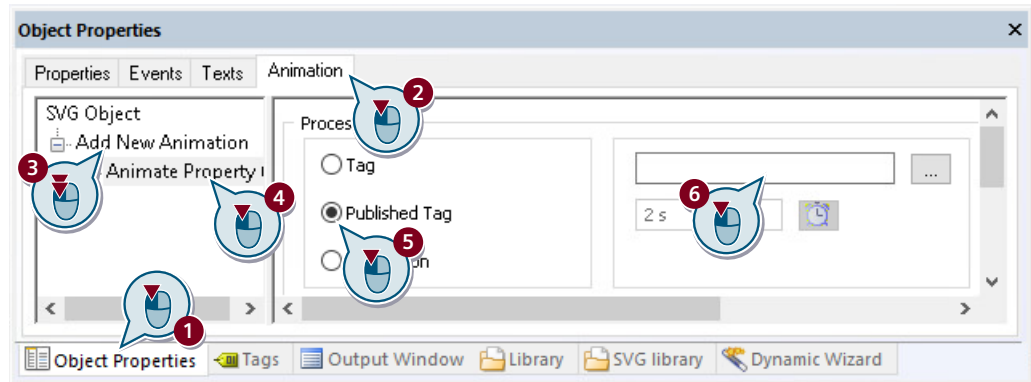
必要条件

- 「Valve.Fpt」フェイスプレートタイプが開かれていること。

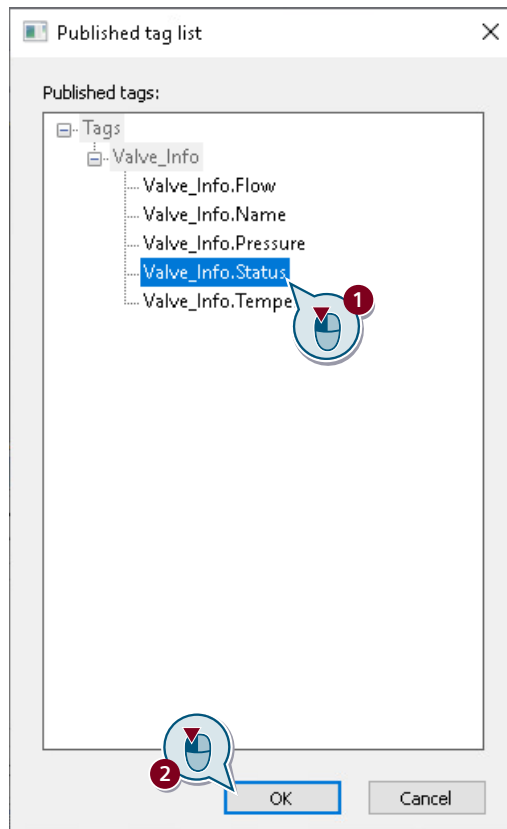
7.6 フェイスプレートタイプの変更

手順

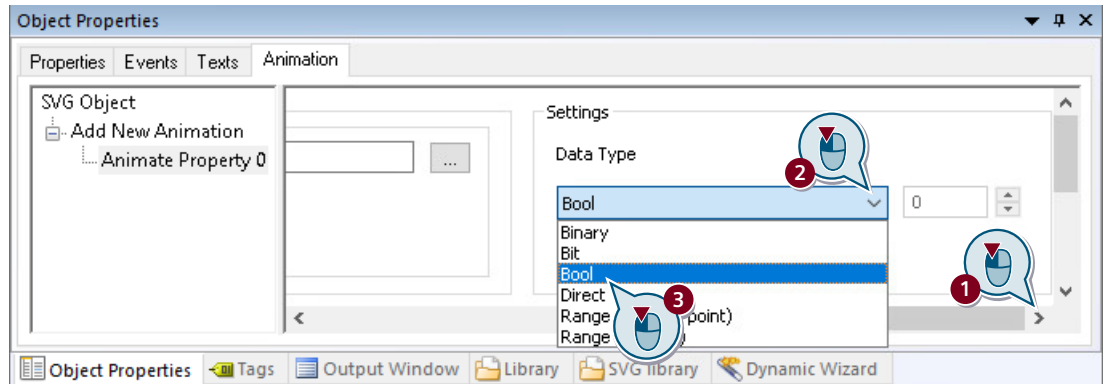
1. フェイスプレートタイプの SVG ライブラリのバルブオブジェクトをクリックします。
SVG オブジェクトが選択され、[オブジェクトプロパティ]選択ウィンドウで表示されます。
2. パブリッシュされているインターフェースタグを使用する新しいアニメーションを作成します。



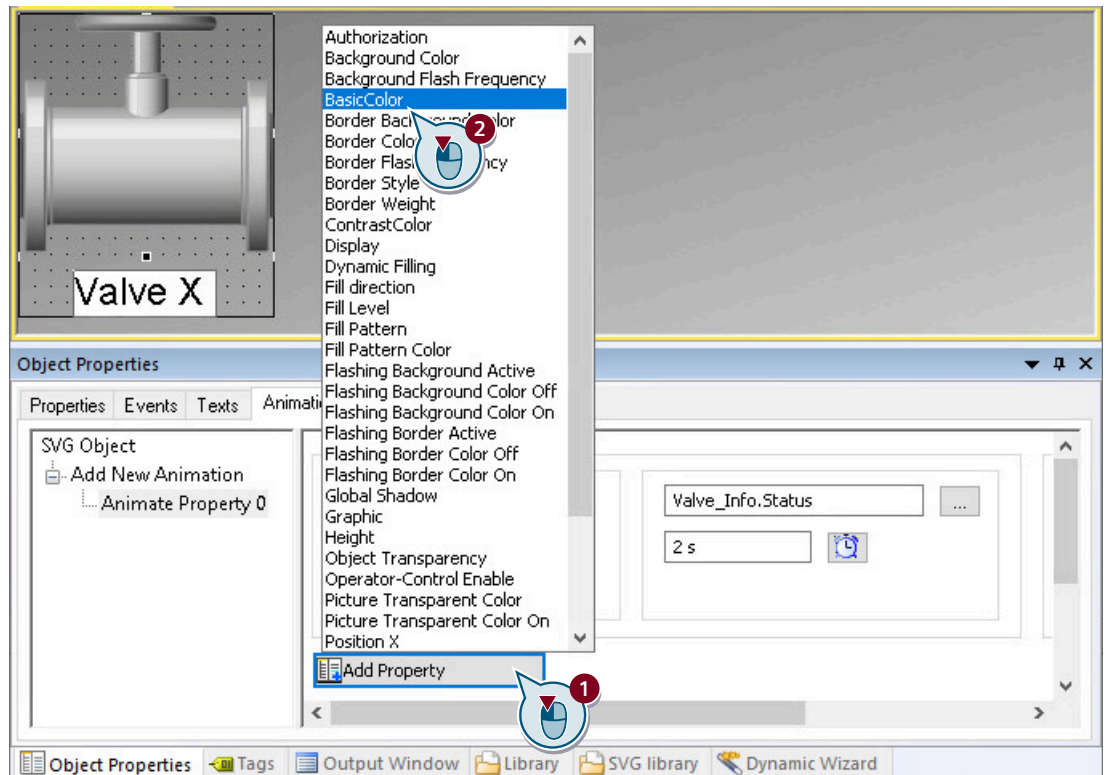
3. 「...」 を使用して、タグの選択を開きます。
4. インターフェースタグ「Valve_Info.Status」 を選択して、[OK]で確定します。
アニメーションは「Valve_Info」構造タイプの構造タイプ要素「.Status」 でコントロールされます。



5. [ブーリアン]データタイプを選択します。

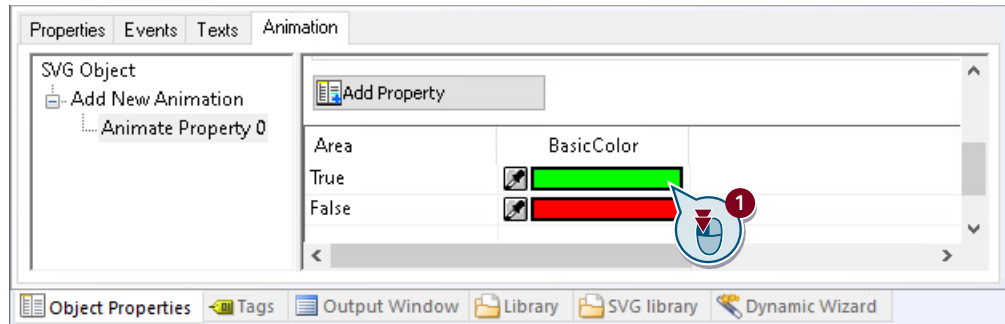


6. [BasicColor]プロパティを追加します。



7.6 フェイスプレートタイプの変更

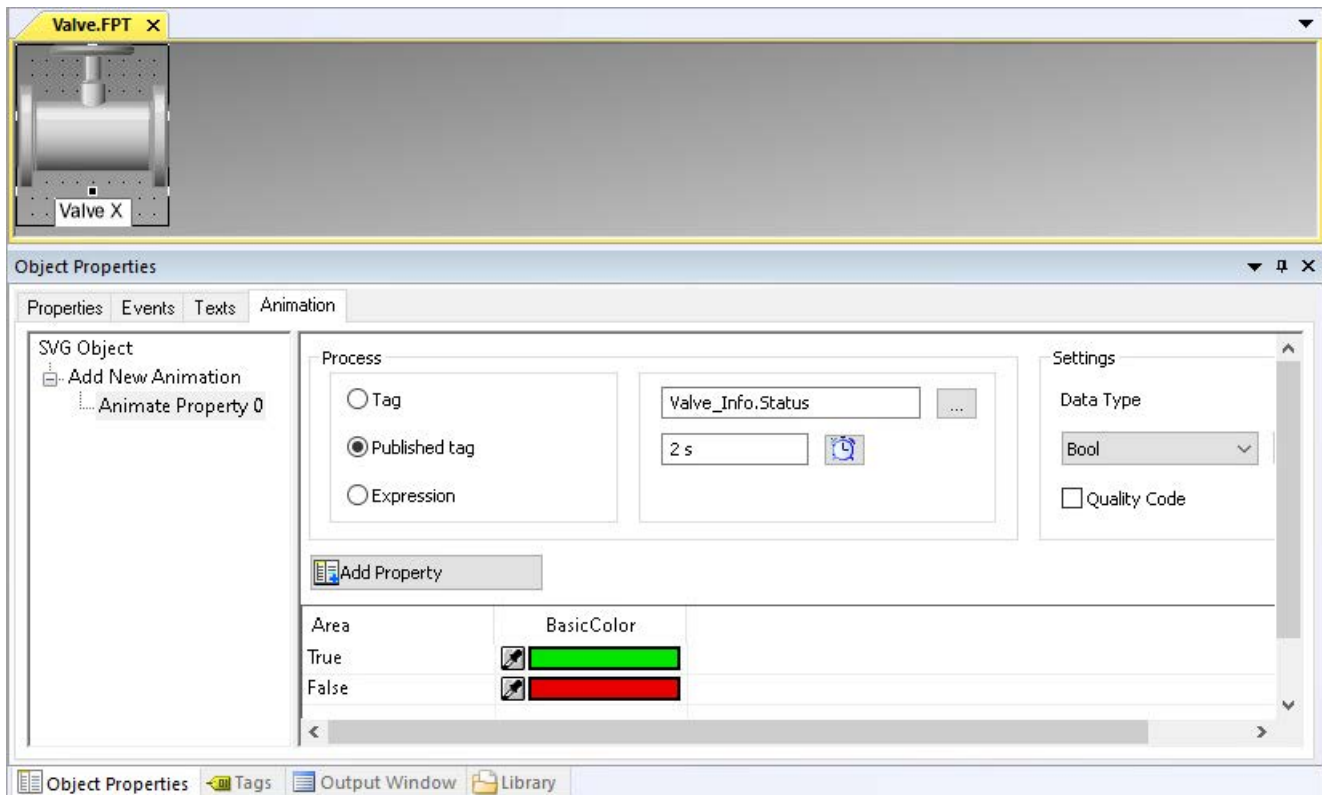
7. 緑色を[真]、赤色を[偽]に対して選択します。



8. プロセス画像を保存します。

結果

「Valve_Info.Status」 インターフェースタグで SVG オブジェクトをアニメーション化しています。バルブのステータスに応じて、バルブは緑色あるいは赤色でランタイムで示されます。



7.7 プロジェクトのテスト(グラフィックシステム-フェイスプレートタイプ)

概要

次のステップは、WinCC タグシミュレータによって拡張している「Quick_Start」プロジェクトをテストする方法を示しています。

拡張している「Quick_Start」プロジェクトで、WinCC タグシミュレータによって構造タグの値をシミュレーションします。

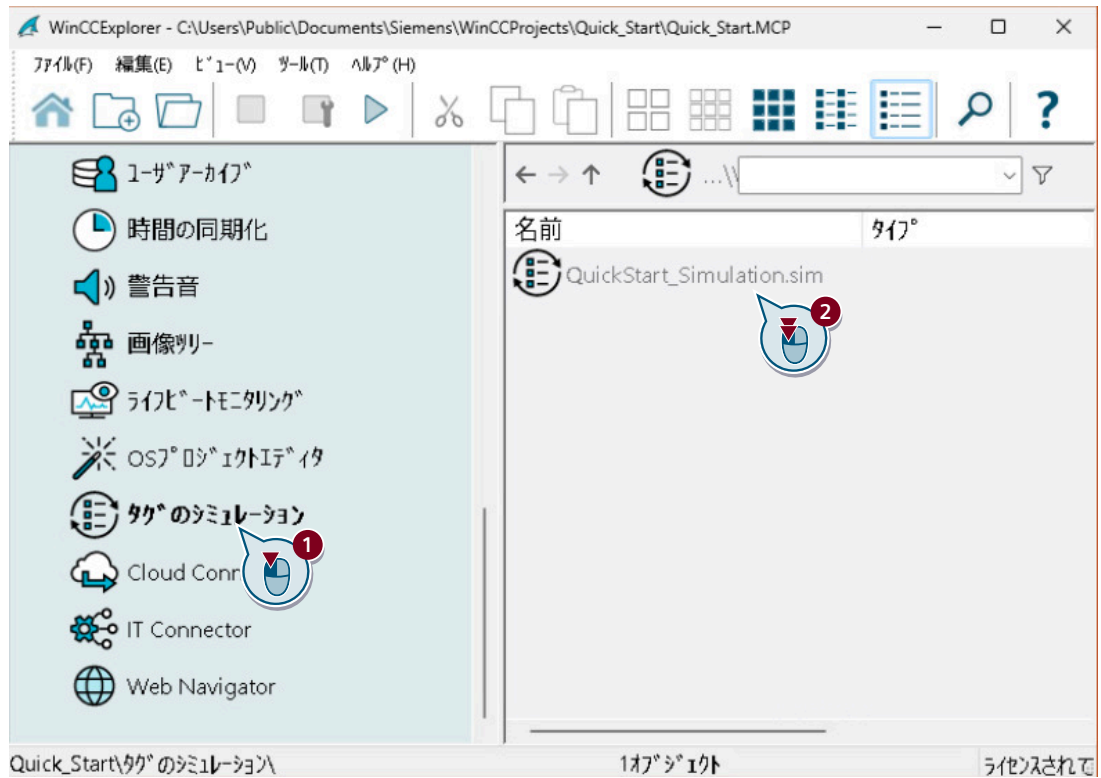
WinCC タグシミュレータは、構造タグエレメントに対して様々な値を割り付けます。構造タグ要素がポップアップ画面の I/O フィールドおよびバルブの表示にリンクされています。タグ値に従ってランタイムの I/O フィールドおよびバルブ表示の値は異なります。

必要条件

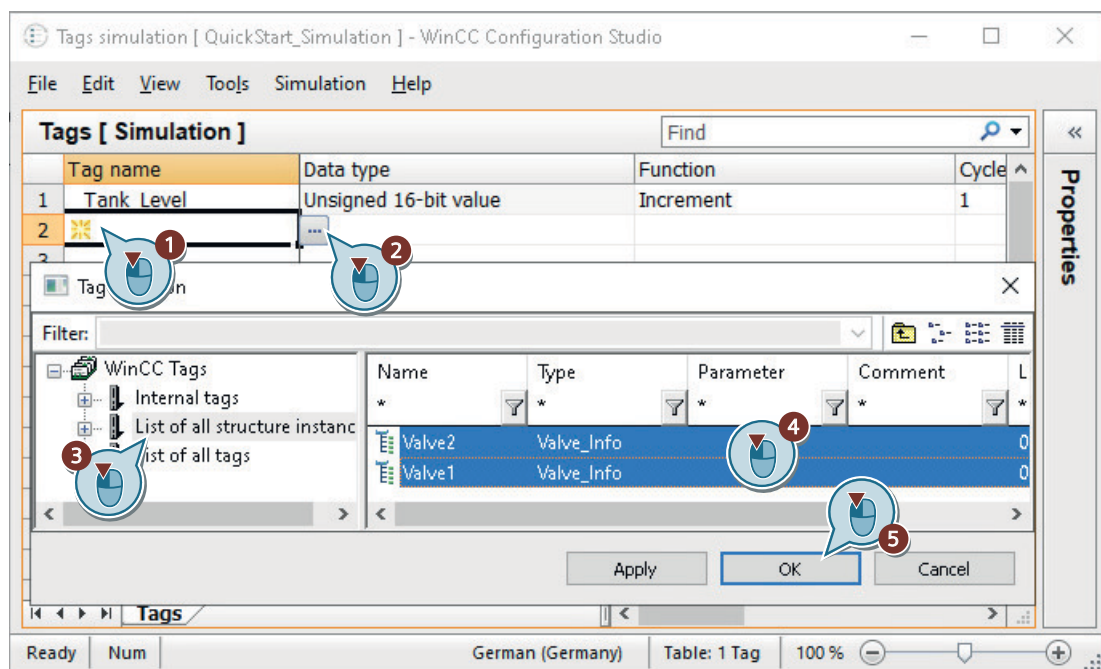
- ランタイムプロパティが定義されていること。
ランタイムタイムプロパティの定義(グラフィックシステム-プロセス画像)(ページ87)に説明されている手順に従ってください。

手順

1. 「QuickStart_Simulation.sim」設定を[タグシミュレーション]エディタで開きます。

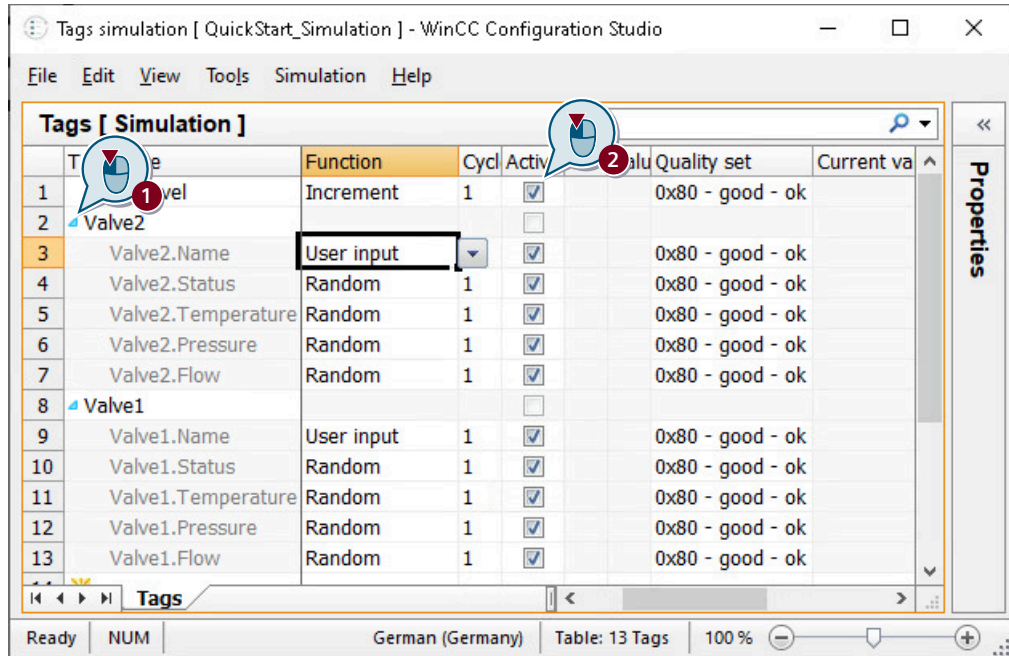


2. タグ選択ダイアログを開き、「Valve1」と「Valve2」のタグを構造インスタンスのリストから選択します。



7.7 プロジェクトのテスト(グラフィックシステム - フェイスプレートタイプ)

3. ▶ ボタンを使用して、構造タグ要素を拡張します。
4. [アクティブ]列ですべてのタグを有効にします。



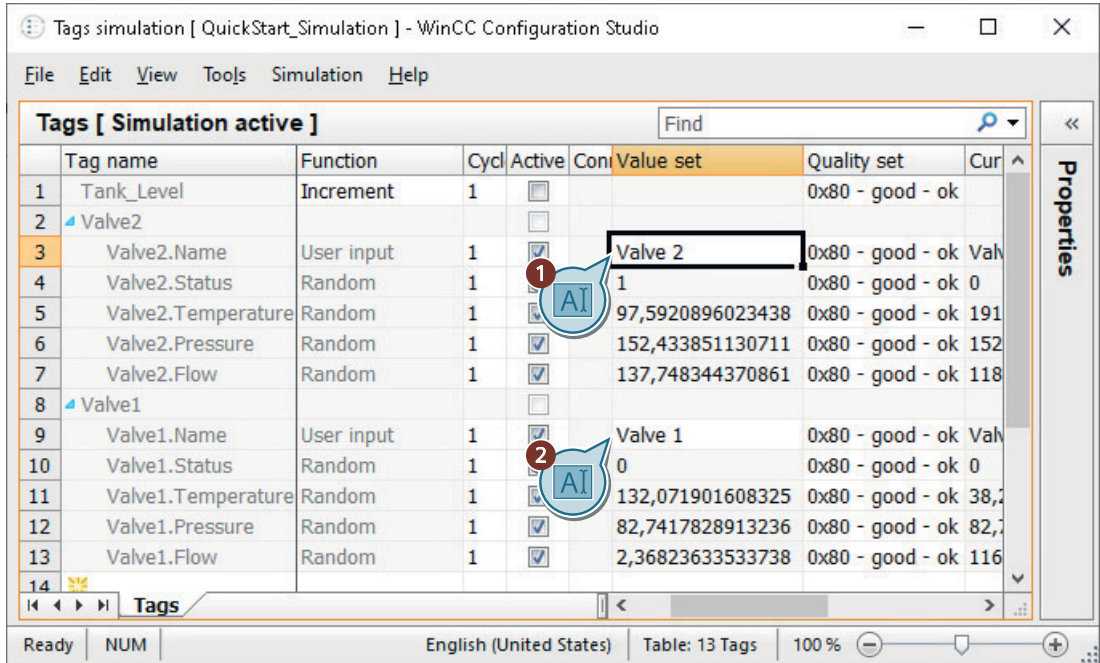
5. タグの次のプロパティを定義します。

タグ名	ファンクション	サイクル	品質セット
Tank_Level	インクリメント	1	0x80 - 良好 - OK
Valve2.Name	ユーザー入力	1	0x80 - 良好 - OK
Valve2.Status	ランダム	1	0x80 - 良好 - OK
Valve2.Temperature	ランダム	1	0x80 - 良好 - OK
Valve2.Pressure	ランダム	1	0x80 - 良好 - OK
Valve2.Flow	ランダム	1	0x80 - 良好 - OK
Valve1.Name	ユーザー入力	1	0x80 - 良好 - OK
Valve1.Status	ランダム	1	0x80 - 良好 - OK
Valve1.Temperature	ランダム	1	0x80 - 良好 - OK
Valve1.Pressure	ランダム	1	0x80 - 良好 - OK
Valve1.Flow	ランダム	1	0x80 - 良好 - OK

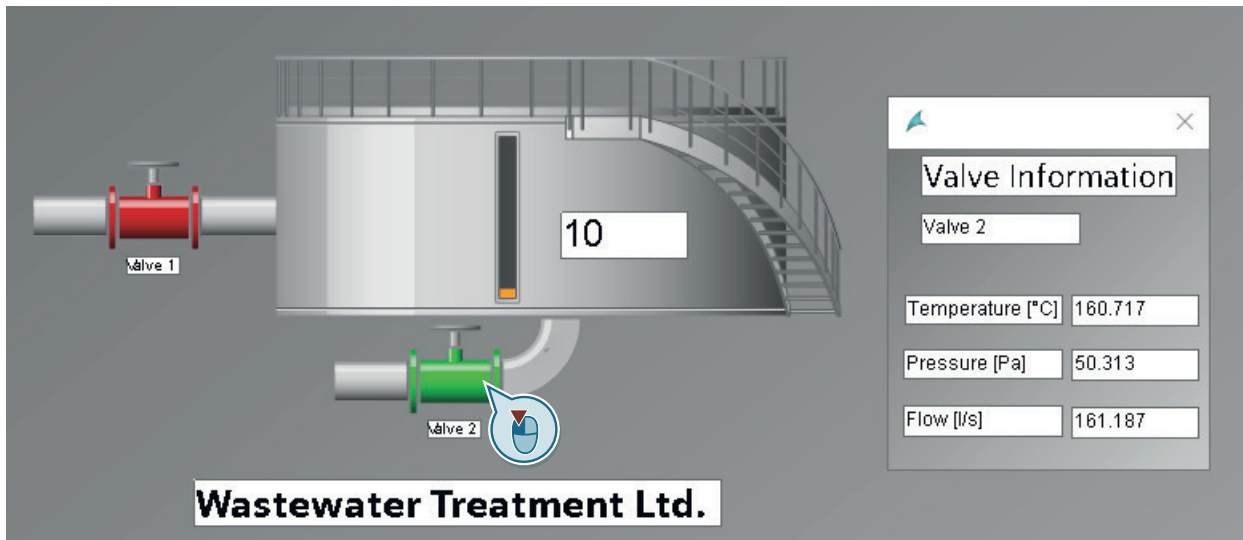
6. 設定を保存します。
7. ▶ ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。
8. [タグシミュレーション]エディタでシミュレーションを起動します。

7.7 プロジェクトのテスト(グラフィックシステム-フェイスプレートタイプ)

- バルブの名前を表示するには、名前を構造タグ要素「Valve1.Name」と「Valve2.Name」のための現在のタグ値として入力します。



- [タグシミュレーション]エディタとランタイムウィンドウを、両方を同時に監視できるように位置調整します。
2つのバルブのうち1つでクリックすることによりポップアップ画面を開き、様々なシミュレーション値がディスプレイに影響を与える方法を監視します。
「Valve(x).Status」タグの値の機能としてバルブの色が変更する方法を監視します。



- シミュレーションおよび WinCC Runtime を終了します。

結果

WinCC タグシミュレータを使用して「Quick_Start」プロジェクトをテストし、拡張しています。

テストは、プロセス値を提供された時のプロジェクトの動作を示します。

7.7 プロジェクトのテスト(グラフィックシステム- フェイスプレートタイプ)

値のアーカイブと表示

8.1 値のアーカイブと表示

概要

このセクションは、アーカイブシステムに関する情報を提供し、プロセス値アーカイブでの値の保存方法を説明します。

原理

プロセス値アーカイブを使用して、プロセス値の経時的な展開を、例えばダイアグラムやテーブルなどのように表示します。実際には、早期に問題を認識できる可能性があるため、こうした暫定的な表示は非常に重要です。

プロセス値アーカイブの別の活用方法として、個別の履歴的なプロセス値へのアクセスがあります。たとえば、生産上の問題が発生した場合に、特定の値がどの程度高かったのかを特定するために役立ちます。

「Quick_Start」プロジェクトの設定

"Quick_Start"プロジェクトでは、プロセス値は必要ありません。

このプロジェクトでは、内部タグ「Tank_Level」を使用します。WinCC タグシミュレータによって、これらのタグの値をシミュレートします。シミュレートしたタグの値は、プロセス値アーカイブに保存されます。

保存した値のシーケンスは、トレンドダイアグラムやテーブルとして、プロセス画像に入力されます。このために、[グラフィックデザイナー]エディタの[コントロール]選択画面にあるコントロールを使用します。

ランタイムで設定済みコントロールのシミュレーション値の変更を監視します。

8.2 アーカイブシステム

概要

アーカイブシステムは、WinCC のサブシステムです。このサブシステムは、プロセス値とメッセージをアーカイブするために使用します。

アーカイブシステムのコンポーネント

プロセス値のアーカイブシステムは、設定コンポーネントとランタイムコンポーネントで構成されています。

- アーカイブシステムの設定コンポーネントは"タグロギング"エディタです。このエディタで、(その中の)以下のタスクを実行できます。
 - 取得サイクルとアーカイブサイクルの定義
 - プロセス値アーカイブと圧縮アーカイブの設定
 - アーカイブするプロセス値の定義
- タグロギングランタイムは、アーカイブシステムのランタイムコンポーネントです。タグロギングランタイムには、次のようなタスクがあります。
 - プロセス値のプロセス値アーカイブへの書き込み
 - プロセス値アーカイブからの、アーカイブ済みプロセス値の読み取り

アーカイブ

プロセス値は、コンピュータのハードディスクのアーカイブデータベース、またはタグロギングランタイムのメインメモリに、保存されます。

アーカイブされたプロセス値を圧縮し、データ量を減らすことができます。

アーカイブ期間

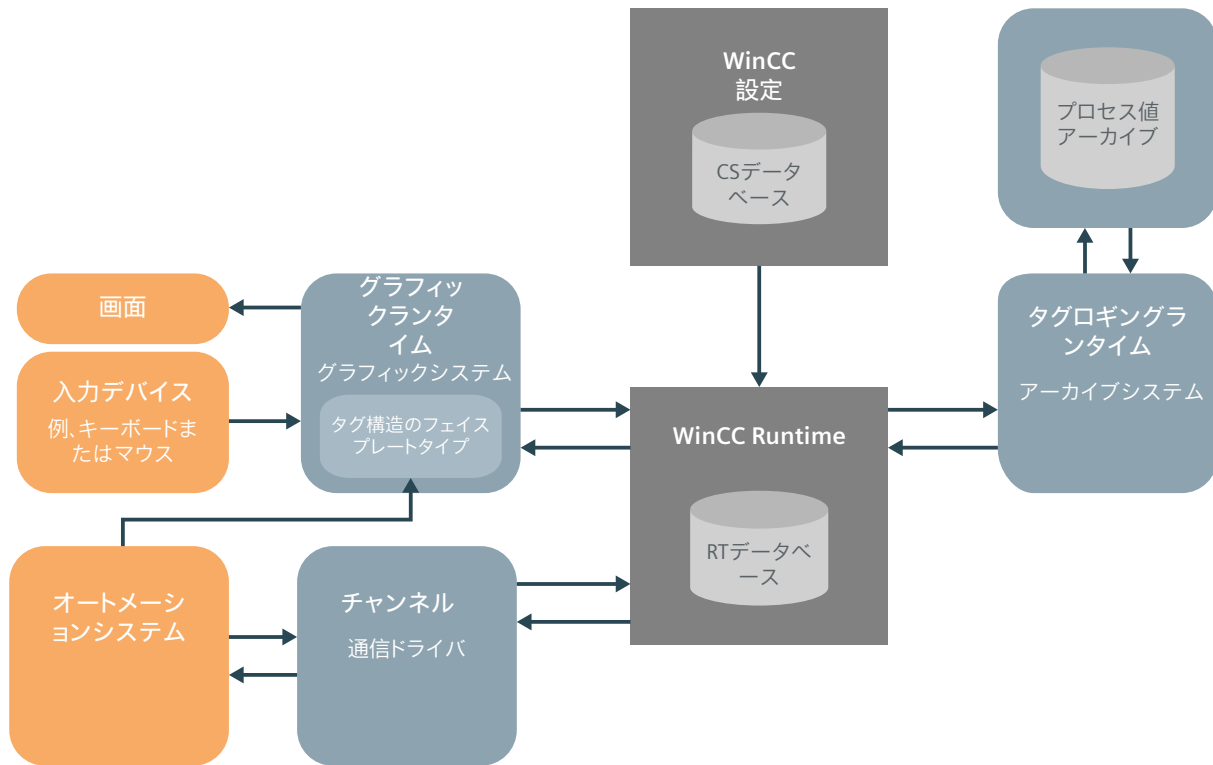
アーカイビングのサイクルとイベントによりアーカイビング時間を制御します。

一定間隔、または定義されている量やパーセンテージでプロセス値が変更されたときのみなどに、プロセス値をアーカイブできます。

ライセンスニング

WinCC 基本システムで、追加のライセンスなしで 512 のロギングタグを設定することができます。

WinCC システムのアーカイブ



8.3 タグロギングの開始

概要

次のステップは、"タグロギング"エディタの開始方法を示しています。

"タグロギング"エディタで、取得サイクルやアーカイブサイクルの時間と共に、プロセス値アーカイブを設定します。

必要条件

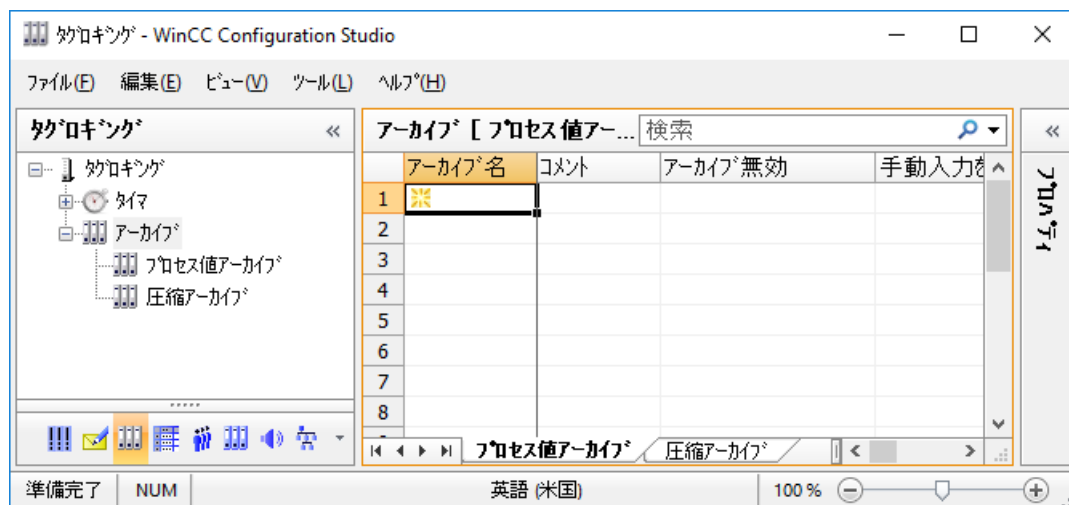
- "Quick_Start"プロジェクトが開いている。

手順

1. "タグロギング"エディタを開始します。
"タグロギング"エディタが開きます。

結果

"タグロギング"エディタが開いています。



次のステップ

- [タグロギング]エディタで、取得サイクルとアーカイブサイクルの時間を設定します。
- 設定された時間で、タグ値を取得し、アーカイブする時間間隔を定義します。
- プロセス値アーカイブを作成します。

8.4 タイマーの設定

概要

"Quick_Start"プロジェクトで、取得サイクルおよびアーカイブサイクルの新しい時間を設定します。

この時までには、タグ値を取得、アーカイブする時間間隔を決めておきます。

取得サイクル

取得サイクルは、プロセスタグのプロセス値を読み出す間隔を決定します。

取得サイクルは、WinCC Runtime が起動するとすぐに開始します。

アーカイブサイクル

アーカイブサイクルは、プロセス値がアーカイブデータベースに保存される時間間隔です。

- アーカイブサイクルは、常に設定された取得サイクルの整数倍です。
- アーカイブサイクルは、WinCC Runtime の起動時またはユーザーが定義した時点に開始します。

開始ポイントを指示すると、値の時間差アーカイブやアーカイブ負荷の分散が可能になります。

システムは、取得とアーカイブの間に最大で取得サイクルの長さまでの遅延を引き起こすことがあります。

"タグロギング"エディタは、異なる標準時間を提供します。

これらの時間を変更することはできません。標準時間が十分でない場合は、新しい時間を設定してください。

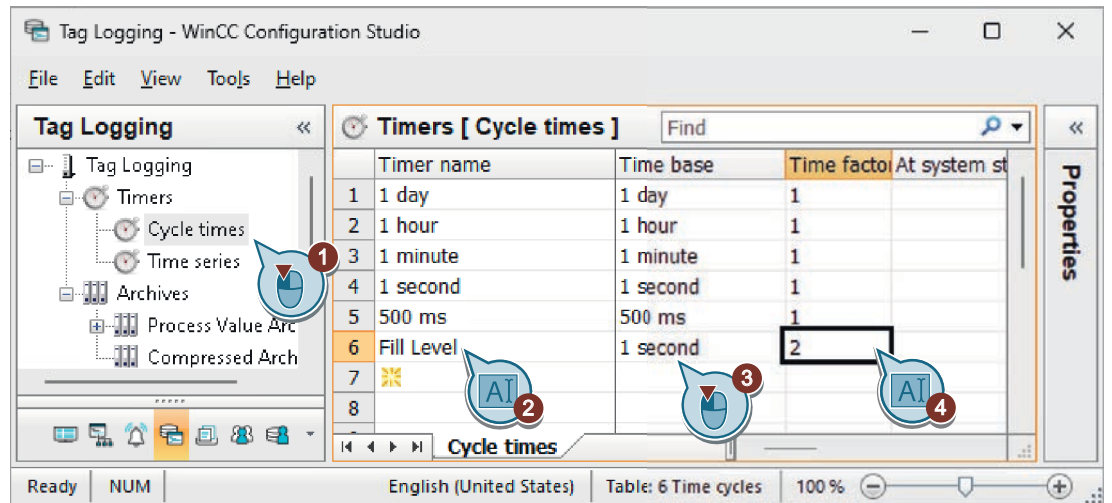
時間を設定する場合、基本時間と時間係数を定義します。基本時間と時間係数の積が、2つのアーカイブの間の時間間隔を決定します。例えば、時間ベースを「1秒」に、時間係数を「5」に設定すると、プロセス値は5秒毎にアーカイブされます。

必要条件

- [タグロギング]エディタが開いている。

手順

1. 新しいサイクルタイム[Fill_Level]を作成します。
2. 新規タイマを作成するには、先頭の空白セルをクリックして、テーブルペインの[タイマ名]列に名前を入力します。



結果

取得サイクルおよびアーカイブサイクルの新しい時間[Fill_Level]が設定されます。
設定した時間によって、タグ値は2秒毎に取得、アーカイブされます。

次のステップ

プロセス値アーカイブを作成し、内部タグの値をアーカイブできるようにします。

8.5 プロセス値アーカイブの作成

概要

次のステップは、"Quick_Start"プロジェクトのプロセス値アーカイブの作成方法を示しています。内部タグ"Tank_Level"の値は、このアーカイブに保存されます。

[タグロギング]エディタでアーカイブを作成します。

プロセス値アーカイブを作成する場合、値をアーカイブするタグも定義します。

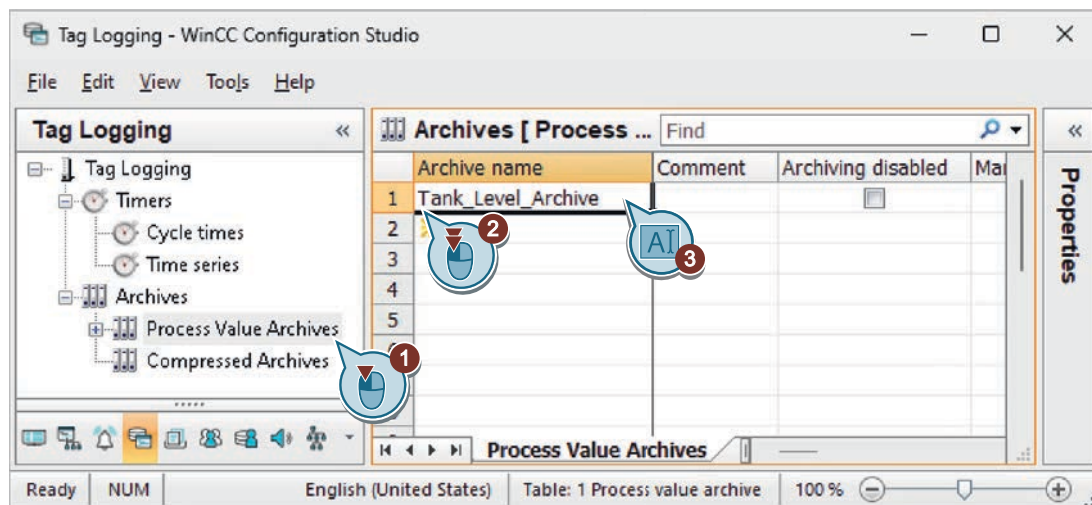
タグを定義すると、[タグロギング]はプロセス値アーカイブにアーカイブタグを作成します。アーカイブする値は、アーカイブタグに保存されます。

必要条件

- [Tank_Level]タグがタグ管理で設定されている。

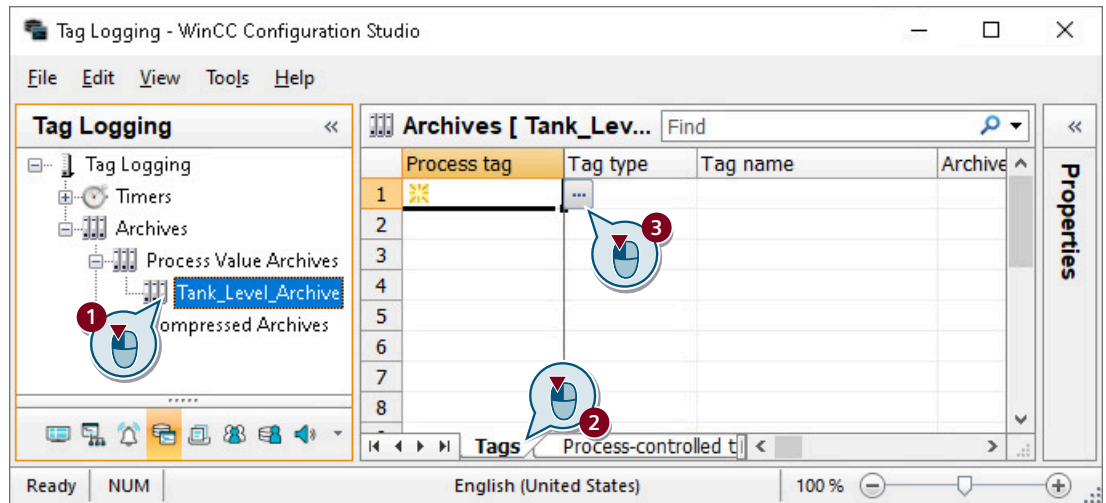
手順

1. [タグロギング]エディタのナビゲーションエリアで、[プロセス値アーカイブ]フォルダを選択します。
2. テーブルエリアの[アーカイブ名]列で、上部の空白行をクリックして、名前「Tank_Level_Archive」を入力します。

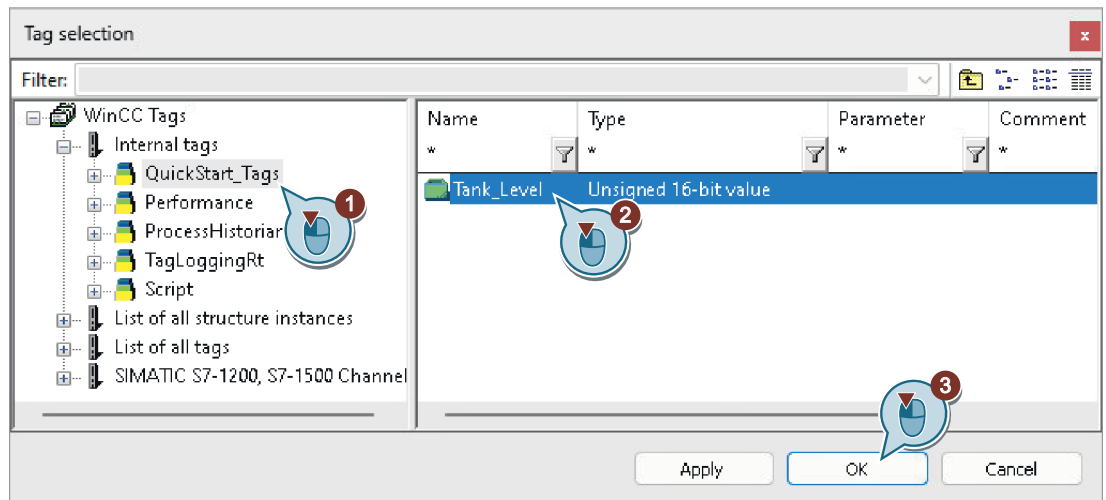


3. ナビゲーションエリアで、アーカイブのフォルダを選択します。
テーブルエリアで[タグ]タブを選択します。

4. テーブルで[プロセスタグ]列の先頭の空き行をクリックしてから、[...]ボタンをクリックします。
タグの選択用ダイアログが開きます。



5. 内部タグ[Tank_Level]を[QuickStart_Tags]タググループで選択します。



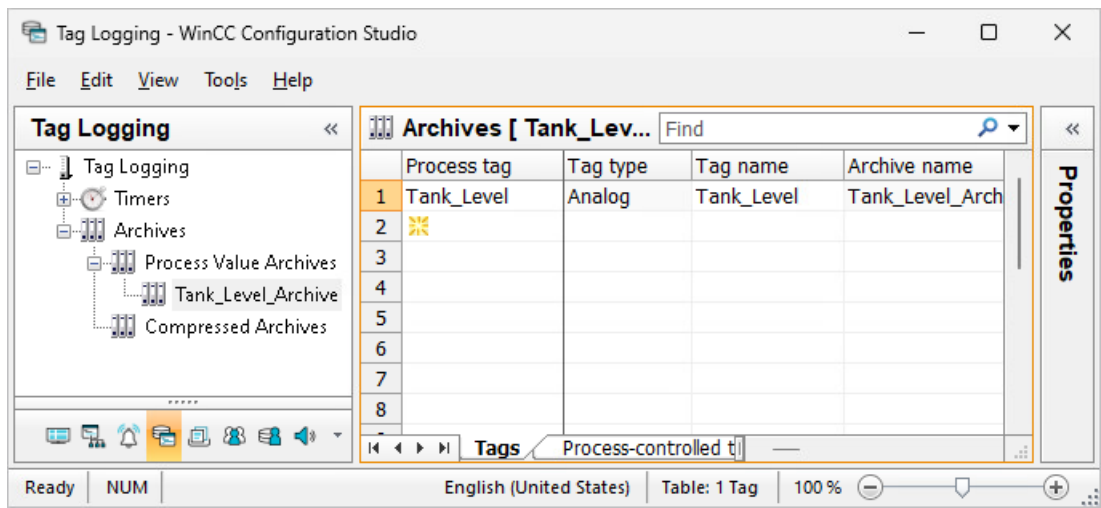
6. [OK]を押してダイアログを閉じます。
アーカイブタグが作成されました。内部タグの名前が使用されます。

結果

[Tank_Level_Archive]プロセス値アーカイブが作成されます。

内部タグ"Tank_Level"の値は、このプロセス値アーカイブに記入されます。

8.5 プロセス値アーカイブの作成



次のステップ

作成したプロセス値アーカイブを編集します。

8.6 プロセス値アーカイブの編集

概要

次のステップは、プロセス値アーカイブ"Tank_Level_Archive"の編集方法を示しています。プロセス値アーカイブの編集は、以下のステップで行なわれます。

- アーカイブタグの名前の変更
- 取得サイクルおよびアーカイブサイクルへの設定時間"Fill_Level"の割り付け
- プロセス値アーカイブの保存場所の定義
- プロセス値アーカイブのサイズの定義


「Quick_Start」プロジェクトのメインメモリを、プロセス値アーカイブの保存場所として選択します。メインメモリは、ランタイム中だけプロセス値アーカイブを提供します。

プロセス値アーカイブに保存されたデータセットの数によって、プロセス値アーカイブのサイズを定義できます。

必要条件

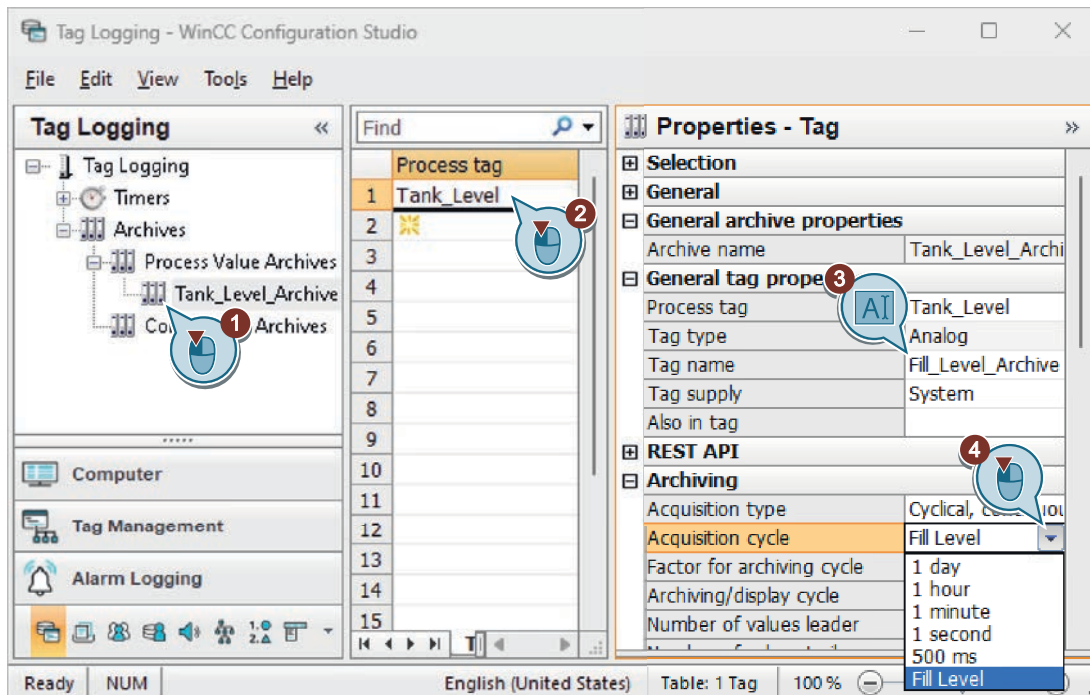
- プロセス値アーカイブ"Tank_Level_Archive"が作成されている。

手順

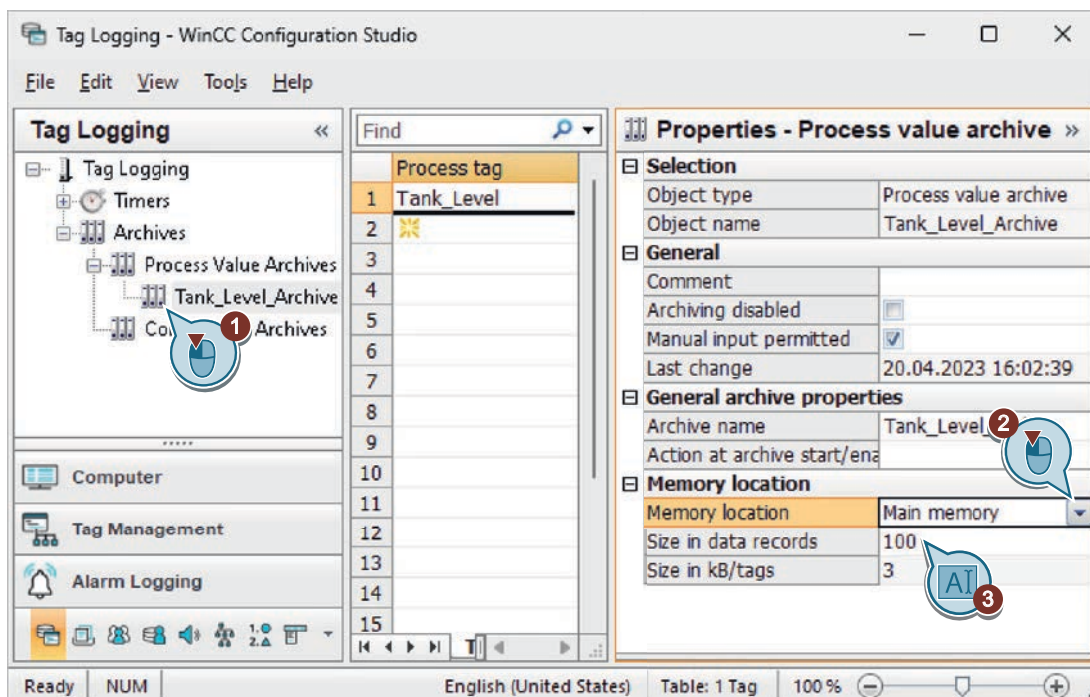
1. 設定済みのアーカイブタグでラインにあるテーブルエリアをクリックし、 ボタンを使用して[プロパティ]ウィンドウを開きます。
必要な場合は、プロパティエリアを拡大します。
2. アーカイブタグ名として"Fill_Level_Archive"を入力します。

8.6 プロセス値アーカイブの編集

3. 取得サイクルおよびアーカイブサイクルに設定時間"Fill_Level"を割り付けます。



4. 作成されたプロセス値アーカイブ"Tank_Level_Archive"にあるナビゲーションエリアでクリックします。
5. プロセス値アーカイブ"Tank_Level_Archive"のメモリの場所とサイズを定義します。



6. "タグロギング"エディタを閉じます。

結果

プロセス値アーカイブが編集されます。

内部タグ[Tank_Level]の値が2秒毎に取得され、アーカイブタグ[Fill_Level_Archive]に保存されます。

タグ値がメインメモリにアーカイブされ、ランタイム中だけ利用できます。

次のステップ

保存された値をランタイムのトレンドとして出力するには、[グラフィックデザイナー]エディタの[トレンド]ウィンドウを設定します。

8.7 プロセス画面の設定(アーカイブシステム)

概要

このセクションでは、プロセス値アーカイブからの出力値を表示するプロセス画像の設定方法を説明します。

原理

プロセス画像を[グラフィックデザイナー]エディタで設定します。

このため、以下のオブジェクトを使用します。

- WinCC OnlineTrendControl
"WinCC OnlineTrendControl"オブジェクトは、[トレンド]ウィンドウを作成するために使用します。
[トレンド]ウィンドウでは、プロセス値アーカイブからの値はトレンドとして出力されます。
- WinCC OnlineTableControl
"WinCC OnlineTableControl"オブジェクトは、[テーブル]ウィンドウを作成するために使用します。
[テーブル]ウィンドウでは、プロセス値アーカイブからの値はテーブルとして出力されます。

各出力はランタイムで行なわれます。

8.7.1 [トレンド]ウィンドウの設定

概要

次のステップは、[トレンド]ウィンドウの設定方法を示しています。

[トレンド]ウィンドウは、[グラフィックデザイナー]エディタで設定します。

このため、新しいプロセス画像を作成します。プロセス画像に、オブジェクト"WinCC OnlineTrendControl"を挿入します。

"Quick_Start"プロジェクトで、このオブジェクトをアーカイブタグ"Fill_Level_Archive"と接続させます。

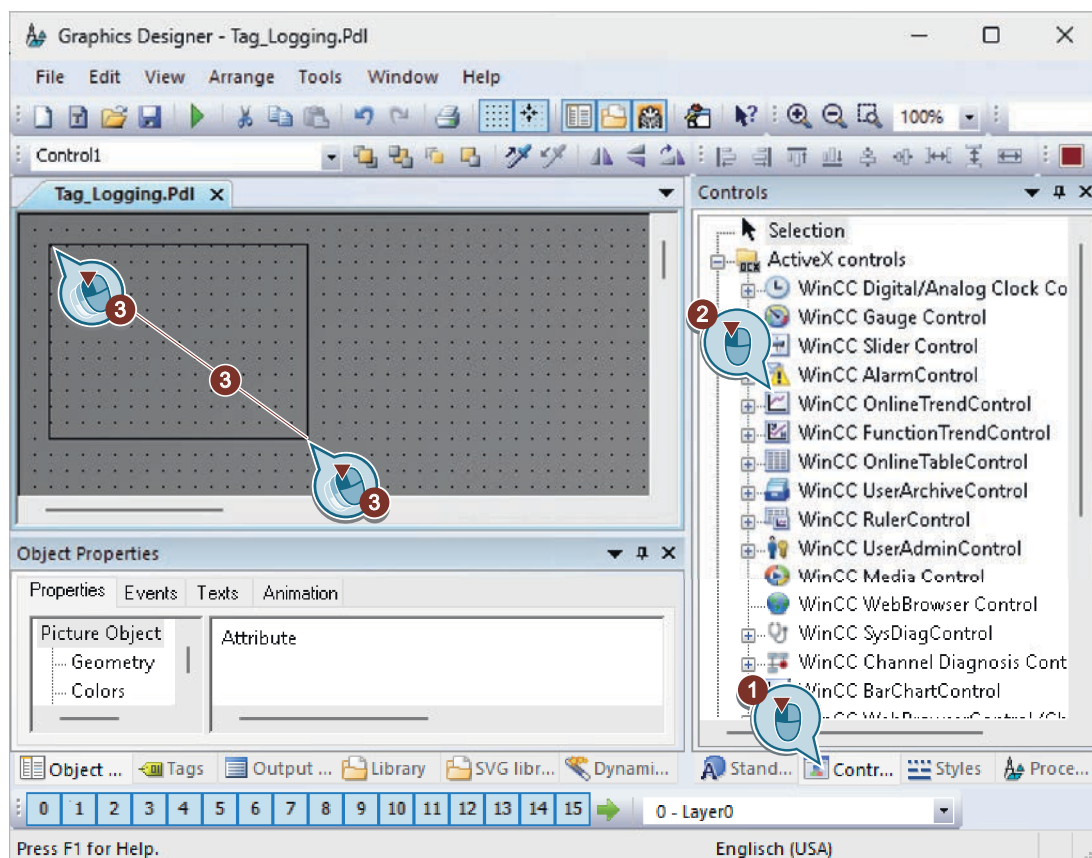
これで、ランタイムでアーカイブタグに保存された値がトレンドとして出力されます。

必要条件

- プロセス値アーカイブ"Tank_Level_Archive"が作成されている。

手順

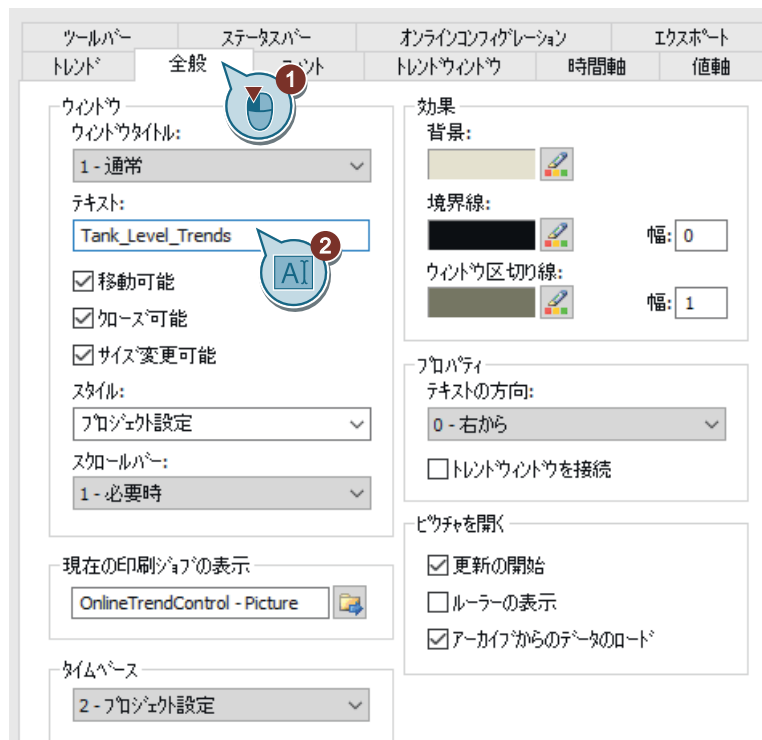
1. 「Tag_Logging.Pdl」という名前の新しいプロセス画像を作成し、[グラフィックデザイナー]エディタで開きます。
2. [コントロール]ウィンドウから、「Tag_Logging.Pdl」プロセス画像に、「WinCC OnlineTrendControl」オブジェクトを挿入します。



[グラフィックデザイナー]エディタの作業エリアに、[トレンド]ウィンドウが表示されます。
[WinCC OnlineTrendControl のプロパティ]ダイアログが開きます。

8.7 プロセス画面の設定(アーカイブシステム)

- [トレンド]ウィンドウのタイトルとして"Tank_Level_Trends" と入力し、デフォルト設定を適用します。




4. 時間軸の設定を定義します。

ツールバー	ステータスバー	オンラインコンフィグレーション	エキスポート
トレンド	全般	トレンド	値軸
時間軸: <input checked="" type="checkbox"/> 時間軸 1		オフセット: 時間軸 1	
<input type="button" value="新規作成"/> <input type="button" value="削除"/> <input type="button" value="上へ"/> <input type="button" value="下へ"/>		トレンド単位: トレンド単位ウ 1	
		ラベル: _____	
		配置: 0 - 下揃え	
		<input checked="" type="checkbox"/> 最新の情報の更新	
時間範囲			
設定: 2 - 測定点の数		開始時間: 19.10.2018 14:37:42	
測定点の数: 120		終了時間: 19.10.2018 14:38:42	
		時間範囲: 1 x 1分	
効果			
時間の形式: 自動		色: [黒色]	
日付の形式: 自動		<input type="checkbox"/> トレンド色を使用	
<input checked="" type="checkbox"/> 日付の表示			

8.7 プロセス画面の設定(アーカイブシステム)

5. 値軸の設定を定義します。

ツールバー	ステータスバー	オンラインコンフィグレーション		エキスポート
トレンド	全般	フォント	トレンドウィンドウ	時間軸 値軸
値軸: <input checked="" type="checkbox"/> 値軸 1		オブジェクト名: 値軸 1		
<input type="button" value="新規作成"/> <input type="button" value="削除"/> <input type="button" value="上へ"/> <input type="button" value="下へ"/>		トレンドウィンドウ: トレンドウィンドウ 1		
		ラベル: [空欄]		
		配置: 0 - 左揃え		
値範囲 開始: 0 終了: 10 <input checked="" type="checkbox"/> 自動		スクーリング: 0 - 線形		
効果 小数点以下の桁数: 2 <input checked="" type="checkbox"/> 自動 <input type="checkbox"/> 指数表記		色: [黒色] <input type="checkbox"/> トレンド色を使用		
ユーザースクーリング <input type="checkbox"/> 使用		エリア名 <input type="checkbox"/> 表示		

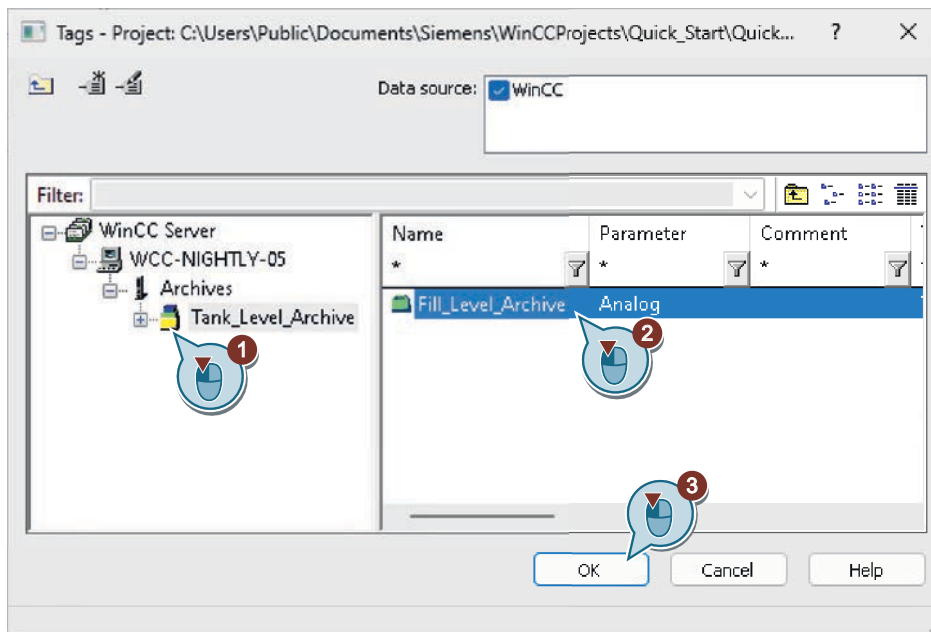
6. トレンド名前"Tank_Level"を入力し、をクリックします。



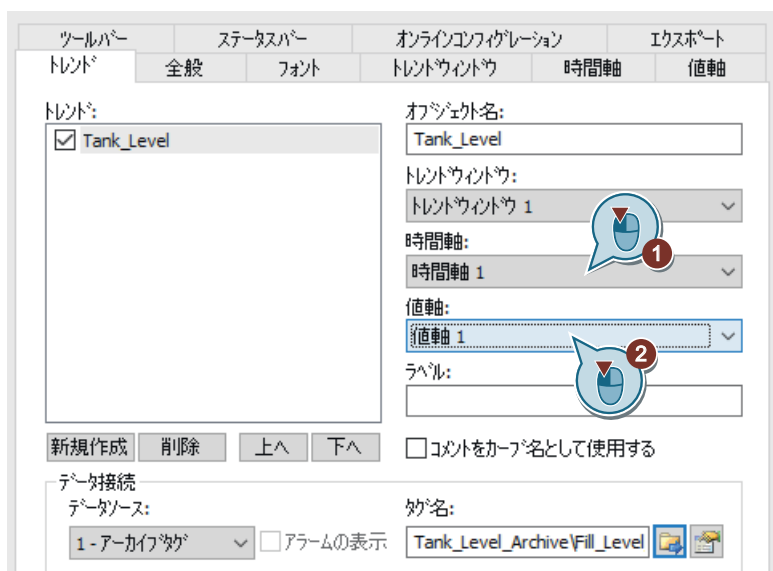
タグの選択用ダイアログが開きます。

8.7 プロセス画面の設定(アーカイブシステム)

- [Fill_Level_Archive]アーカイブタグを[タグ]ダイアログで選択します。
タグを表示するには、[データソース]エリアで[WinCC タグ]オプションの選択が必要な場合があります。



- 時間軸と値軸を選択します。



- [OK]をクリックして、ダイアログボックスを閉じ、設定を適用します。
- 「Tag_Logging.Pdl」プロセス画像を保存します。

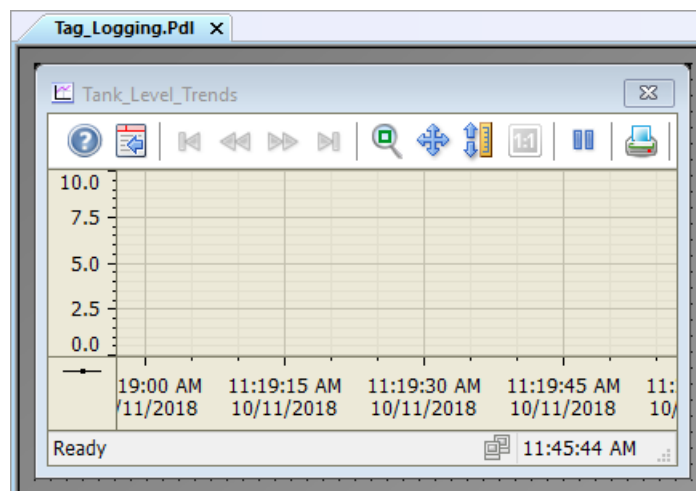
別の手順

1. [タグロギング]エディタのテーブルペインにあるアーカイブタグ"Fill_Level_Archive"の付いた行を選択します。
マウスを選択長方形の端に動かします。
カーソルが十字型ポインタ[+]に変わります。
2. タグの選択内容をグラフィックデザイナーの画像の空白エリアにドラッグしている間、マウスの左ボタンを押したままにします。
3. 画像でマウスボタンから指を放します。
WinCC OnlineTrendControl が作成されます。
作成されたコントロールには、選択した[Fill_Level_Archive]アーカイブタグのデータ接続のあるトレンドが含まれています。
4. 上記手順で説明したように、ステップ 3 から 5 および 8 に進みます。

結果

[トレンド]ウィンドウ"Tank_Level_Trends"が設定されます。

ランタイムで、このウィンドウは、トレンドとしてアーカイブタグ[Fill_Level_Archive]に保存される値を出力します。



次のステップ

ロギングタグに保存されている値をランタイムのテーブルとして出力するには、[グラフィックデザイナー]エディタの[テーブル]ウィンドウを設定します。

8.7 プロセス画面の設定(アーカイブシステム)

8.7.2 [テーブル]ウィンドウの設定

概要

次のステップは、[テーブル]ウィンドウの設定方法を示しています。

[テーブル]ウィンドウは、"グラフィックデザイナー"エディタで設定します。このために「Tag_Logging.Pdl」プロセス画像を使用します。

プロセス画像に、オブジェクト「WinCC OnlineTableControl」を挿入します。

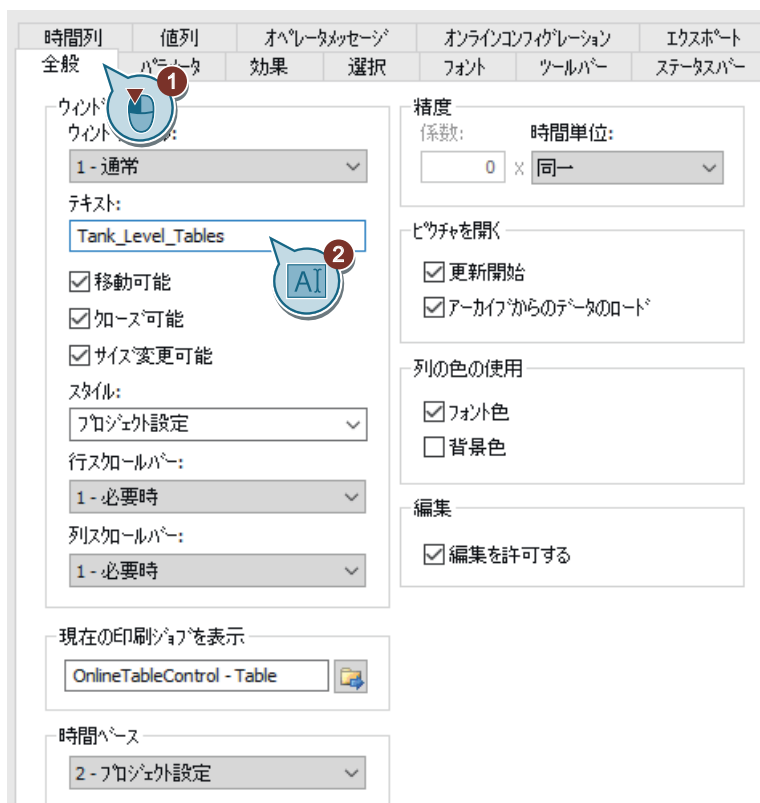
"Quick_Start"プロジェクトで、このオブジェクトをアーカイブタグ"Fill_Level_Archive"と接続させます。


これで、ランタイムでアーカイブタグに保存された値がテーブルとして出力されます。

必要条件

- プロセス値アーカイブ"Tank_Level_Archive"が作成されている。
- 「Tag_Logging.Pdl」プロセス画像が開いている。

8.7 プロセス画面の設定(アーカイブシステム)



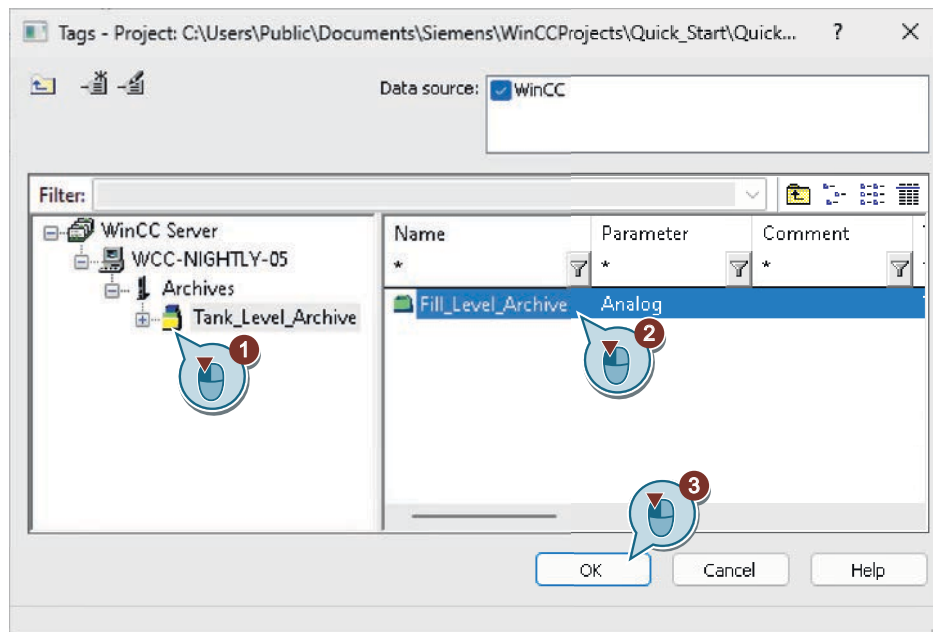
3. [値]列に"Tank_Level"という名前を入力し、をクリックします。



アーカイブタグ選択のダイアログが開きます。

8.7 プロセス画面の設定(アーカイブシステム)

4. [Fill_Level_Archive]アーカイブタグを[タグ]ダイアログで選択します。
タグを表示するには、[データソース]エリアで[WinCC タグ]オプションの選択が必要な場合があります。

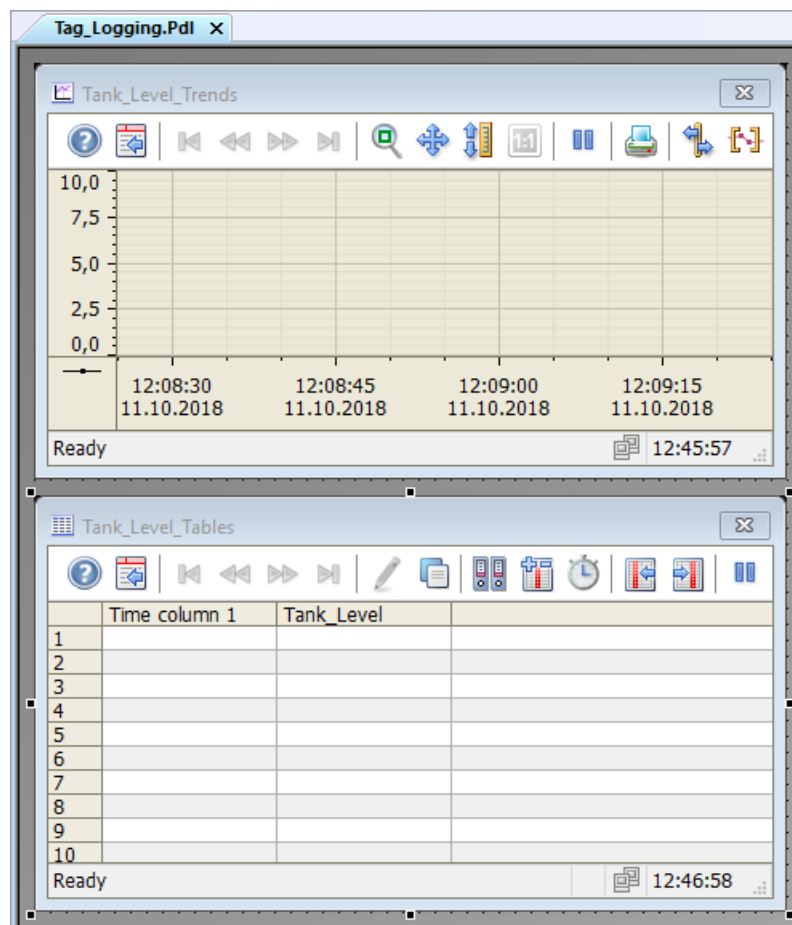


5. [OK]をクリックして、[WinCC OnlineTableControl のプロパティ]ダイアログを閉じます。
6. 「Tag_Logging.Pdl」プロセス画像を保存します。
7. グラフィックデザイナーを閉じます。

結果

[テーブル]ウィンドウ"Tank_Level_Tables"が設定されます。

ランタイムで、このウィンドウは、表としてアーカイブタグ[Fill_Level_Archive]に保存される値を出力します。



次のステップ

- 「Tag_Logging.Pdl」 ページをユーザー定義メニューに追加します。
- WinCC Runtime のプロパティを定義します。

8.8 ユーザー定義メニューのカスタマイズ(アラームシステム)

概要

次のステップは、[タグロギング]メニュー項目をカスタマイズされた[画像変更]メニューに追加する方法を示しています。

メニューエントリ"タグロギング"をプロシージャ"ActivatePicture(ByVal PictureName)"にリンクさせます。

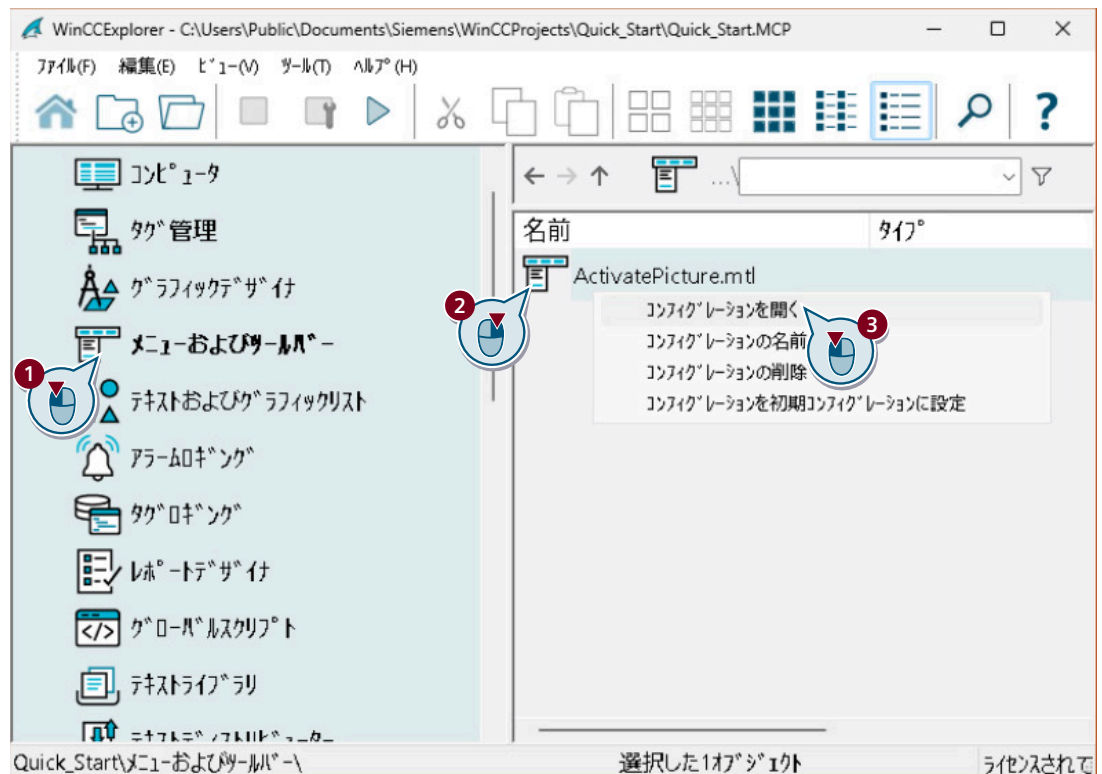
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

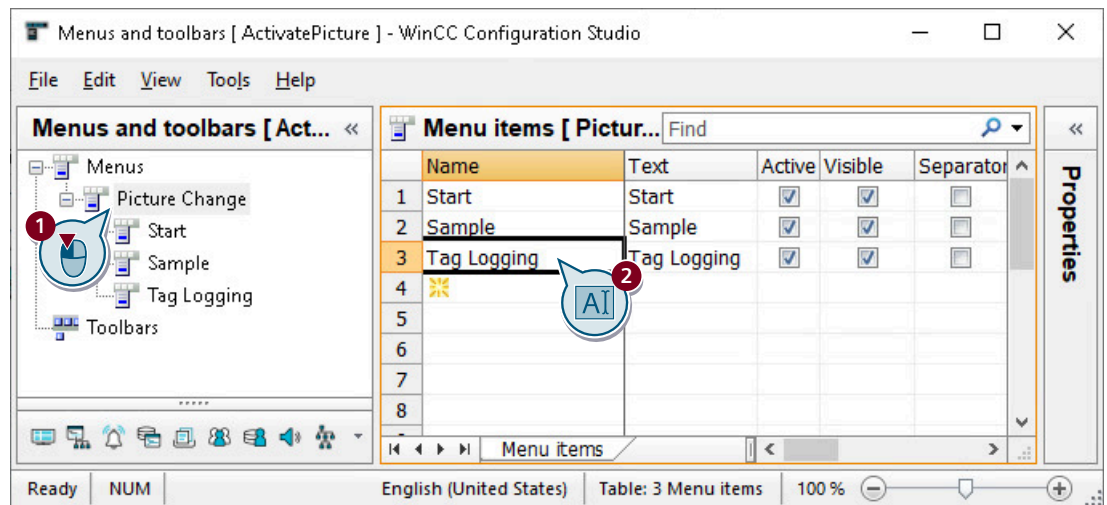
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- ユーザー定義の[画像変更]メニューが「START.Pdl」および「SAMPLE.Pdl」のプロセス画像用に作成されていること。

手順

1. 保存された設定[ActivatePicture.mtl]を[メニューとツールバー]エディタで開きます。

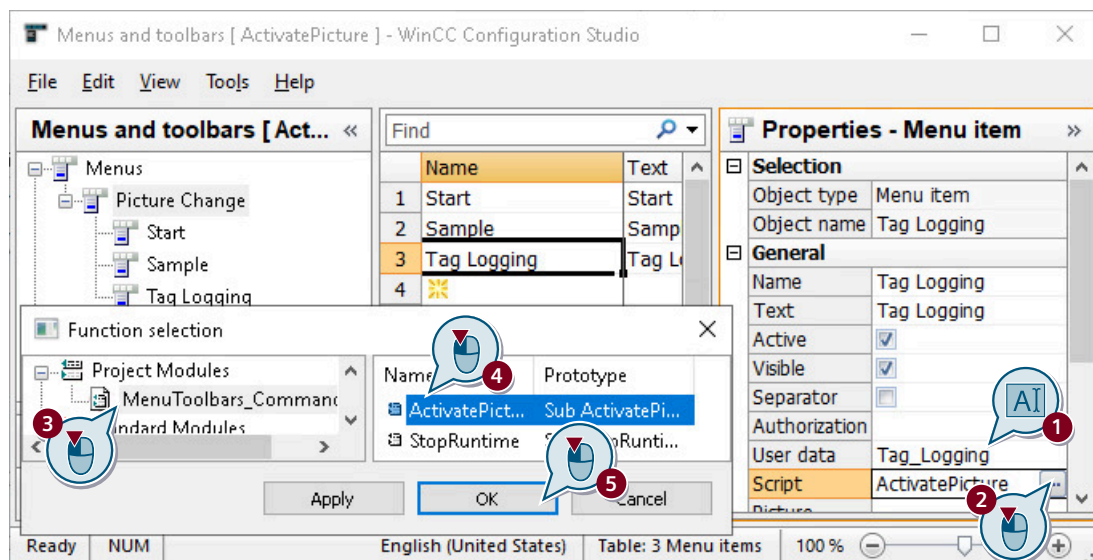


2. [タグロギング]メニューコマンドを作成します。



8.8 ユーザー定義メニューのカスタマイズ(アラームシステム)

3. [タグロギング]メニュー項目を設定して、「Tag_Logging.Pdl」画像に画像変更を実行できるようにします。



4. [メニューとツールバー]エディタで変更内容を保存します。
5. "メニューとツールバー"エディタを閉じます。

結果

[タグロギング]メニュー項目を[画像変更]メニューに追加しました。

メニュー項目を使用して、ランタイムで「START.Pdl」、「SAMPLE.Pdl」および「Tag_Logging.Pdl」のプロセス画像に切り替えます。

8.9 ランタイムプロパティの定義(アラームシステム)

概要

次のステップは、WinCC Runtime のプロパティの定義方法を示しています。

このセクションでは、プロジェクトの起動時にタグロギングランタイムが実行されるように WinCC Runtime をセットアップします。「Tag_Logging.Pdl」プロセス画像を、ランタイムウィンドウの開始画像として選択します。

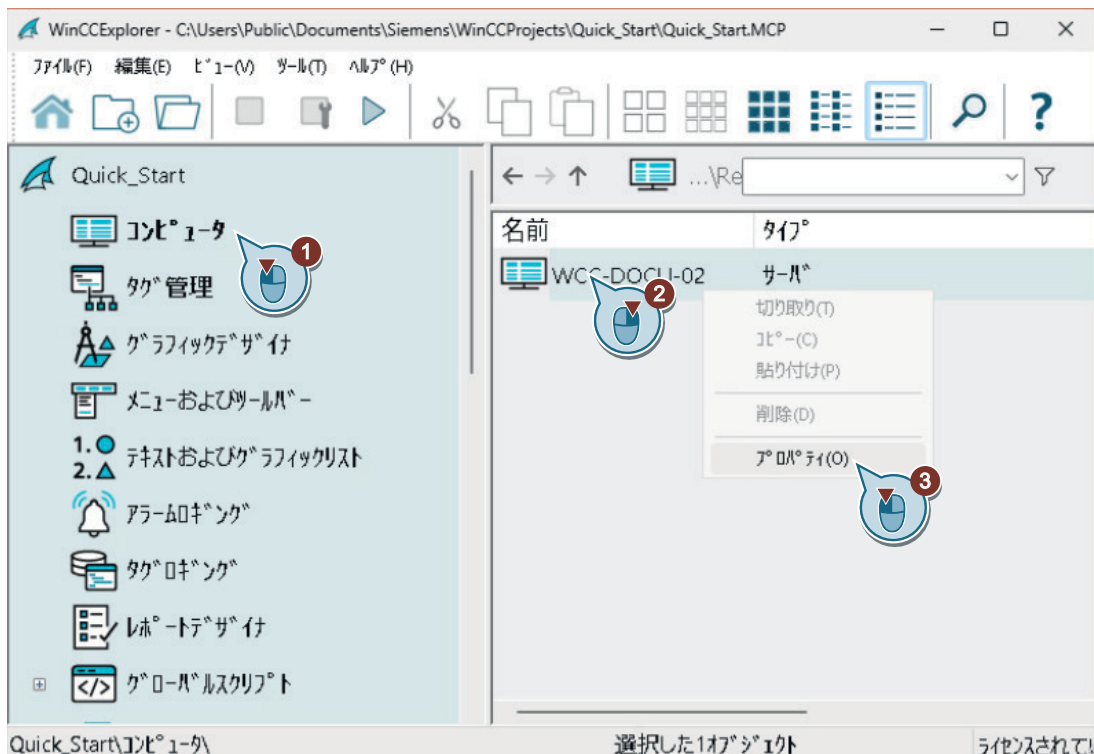
必要条件

- 「Tag_Logging.Pdl」プロセス画像が作成されている。

8.9 ランタイムプロパティの定義(アラームシステム)

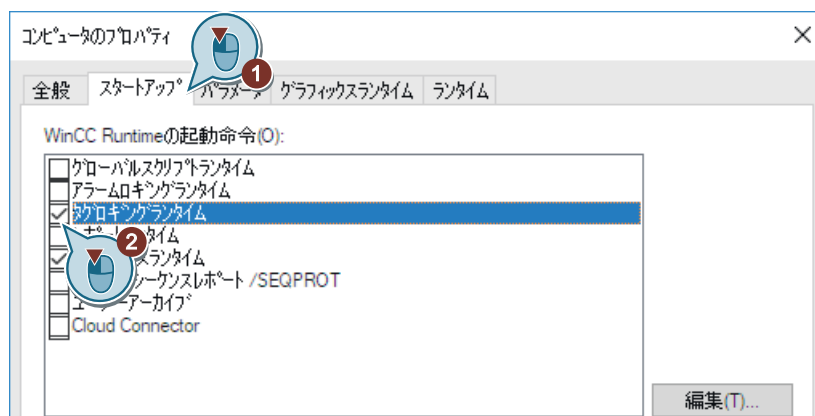
手順

1. [コンピュータ]エディタのナビゲーションエリアでショートカットメニューによりコンピュータのプロパティを開きます。



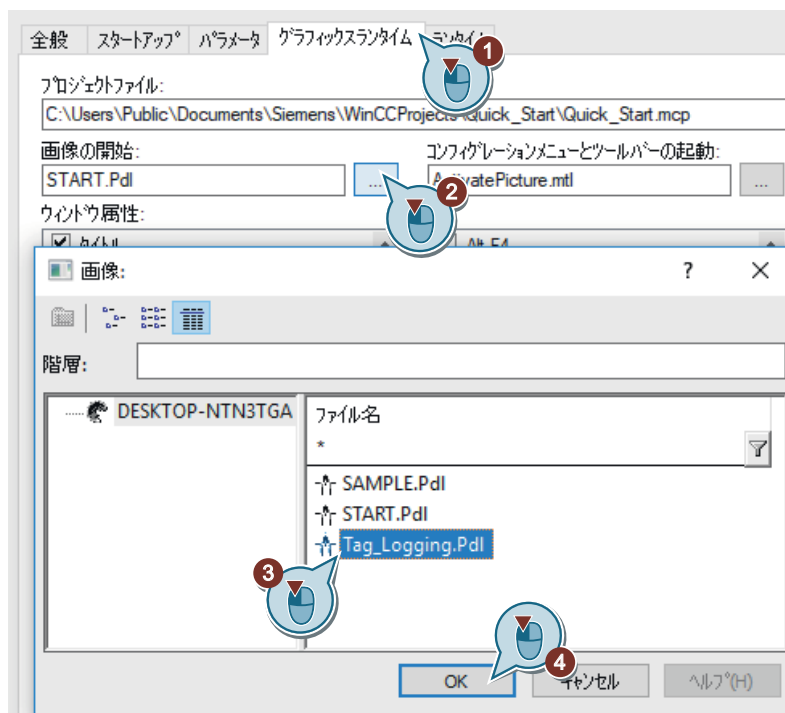
コンピュータのプロパティが開きます。

2. [スタートアップ]タブの[タグロギングランタイム]アプリケーションを選択します。



3. [グラフィックランタイム]タブで、開始画像として「Tag_Logging.Pdl」プロセス画像を定義します。

8.9 ランタイムプロパティの定義(アラームシステム)



結果

WinCC Runtime のプロパティが定義されています。「Quick_Start」プロジェクトを有効にすると、タグロギングランタイムが実行され、「Tag_Logging.Pdl」プロセス画像が表示されます。

次のステップ

「Quick_Start」プロジェクトを有効にし、テストします。

8.10 プロジェクトの有効化とテスト(アーカイブシステム)

概要

次のステップは、"Quick_Start"プロジェクトで設定の有効化およびテスト方法を示しています。

WinCC タグシミュレータは、ランタイムの内部タグ「Tank_Level」に値を割り付けます。


これらの値は"Quick_Start"プロジェクトで2秒毎に取得され、アーカイブタグ [Fill_Level_Archive]に保存されます。

タグロギングランタイムがアーカイブ値を読み取り、[トレンド]ウィンドウおよび[テーブル]ウィンドウに転送します。値は、トレンドおよびテーブルとして出力されます。

必要条件

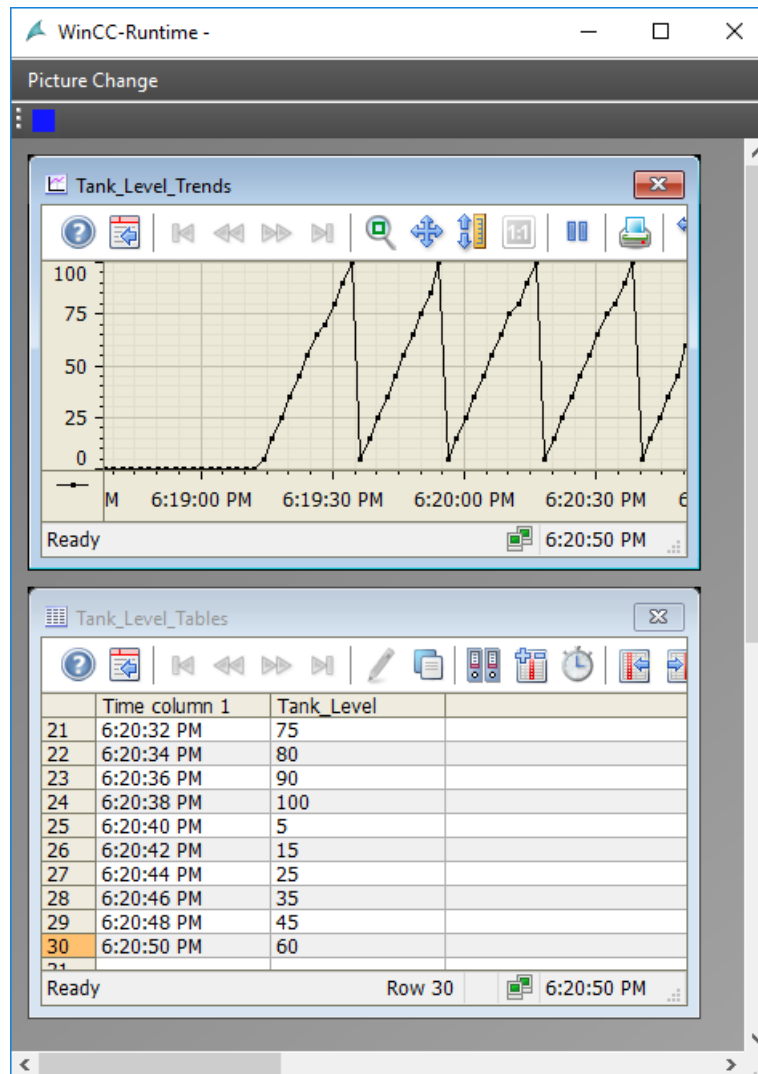
- ランタイムプロパティが定義されている。

手順

1. 「QuickStart_Simulation.sim」設定を[タグシミュレーション]エディタで開きます。
2.  ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。
3. [シミュレーション]>[起動]を使用して、[タグシミュレーション]エディタでシミュレーションを起動します。

8.10 プロジェクトの有効化とテスト(アーカイブシステム)

4. 「Tag_Logging.Pdl」プロセス画像で、シミュレーション値の出力を監視します。



5. [シミュレーション]>[停止]を使用して、[タグシミュレーション]エディタでシミュレーションを停止します。
6. ツールバーの ボタンを使用して、WinCC Runtime を終了します。

結果

"Quick_Start"プロジェクトが有効化され、内部タグ"Tank_Level"の値がシミュレーションされます。

設定済みの[トレンド]ウィンドウおよび[テーブル]ウィンドウに、値シーケンスが表示されます。

8.10 プロジェクトの有効化とテスト(アーカイブシステム)

メッセージの設定

概要

このセクションでは、アラームロギングに関する情報や[アラームロギング]エディタでのメッセージの設定方法を説明します。

原理

アラームロギングはプロセスを監視します。

アラームシステムの[アラームロギング]エディタで次のメッセージを設定できます。

- ディスクリートアラーム
プロセスのステータスの変化を表示します。
ディスクリートアラームは、PLCによってトリガされます。
- アナログアラーム:
制限値違反を表示します。
アナログアラームは、設定した制限値が違反されているときにトリガされます。

「Quick_Start」プロジェクトの設定

"Quick_Start"プロジェクトで、メッセージを設定して、給水バルブやバイオコンバータの充填レベルを監視します。

- 給水バルブをシミュレートするには、新しい内部タグを作成します。
[アラームロギング]エディタで、給水バルブの異なるステータスに対して、ディスクリートアラームを設定します。
タグ値で特定のビットが設定されると、ディスクリートアラームがトリガされます。
- 内部タグ[Tank_Level]の値を使用して、"Quick_Start"プロジェクトのバイオコンバータの充填レベルをシミュレーションします。
フィルレベルの下限値および上限値を設定します。
内部タグ"Tank_Level"の値が限度に違反すると、対応するアナログメッセージがトリガされ、ランタイムで表示されます。

9.1 メッセージシステム

はじめに

メッセージシステムは、プロセスの監視に使用する WinCC のサブシステムです。

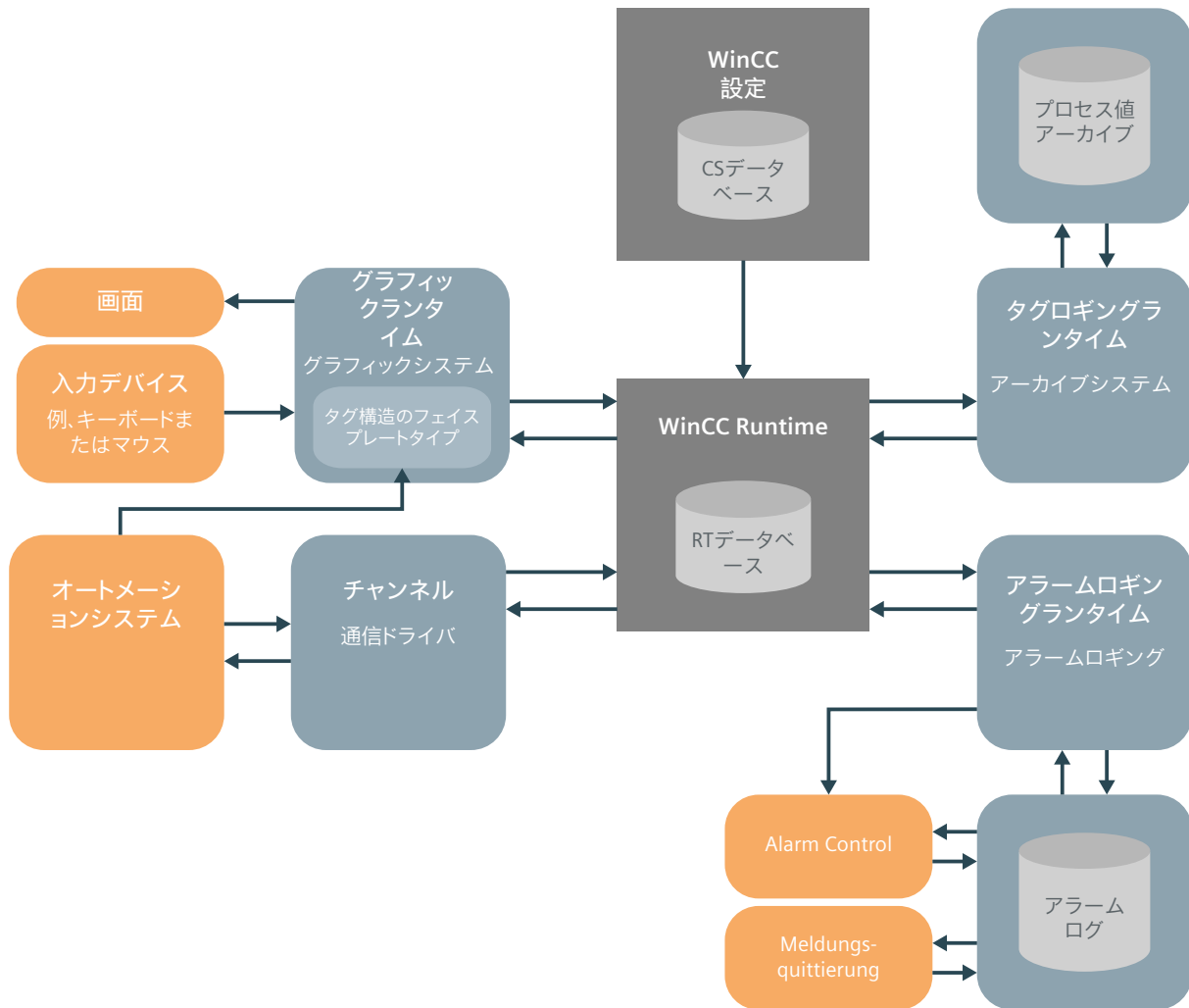
特定のステータスやプロセスにおける変化によって、メッセージシステムはメッセージを生成し、それらをランタイムのテーブルに出力します。メッセージは、危機的な状況を初期段階で認識し、ダウンタイムを回避するのに役立ちます。

メッセージシステムのコンポーネント

メッセージシステムは、設定コンポーネントおよびランタイムコンポーネントで構成されています。

- メッセージシステムの設定コンポーネントは"アラームロギング"エディタです。[アラームロギング]エディタで、主に以下のタスクを実行できます。
 - メッセージの作成
 - メッセージの準備
 - 制限値の設定
 - メッセージテキストとメッセージ状態の表示
 - メッセージの承認プロパティの定義
 - メッセージのアーカイブプロパティの定義
- アラームロギングランタイムは、メッセージシステムのランタイムコンポーネントです。アラームロギングランタイムは、主に次のタスクを実行するために使用されます:
 - 定義済みモニタリングの実行
 - メッセージ出力の制御
 - 確認応答の管理

WinCC システムでのメッセージシステムおよびメッセージアーカイブ



9.2 アラームロギングの起動

概要

次のステップは、[アラームロギング]エディタの開始方法を示しています。

"アラームロギング"エディタで、"Quick_Start"プロジェクトに必要な全てのビットメッセージおよびアナログメッセージを設定します。

必要条件

- "Quick_Start"プロジェクトが開いている。

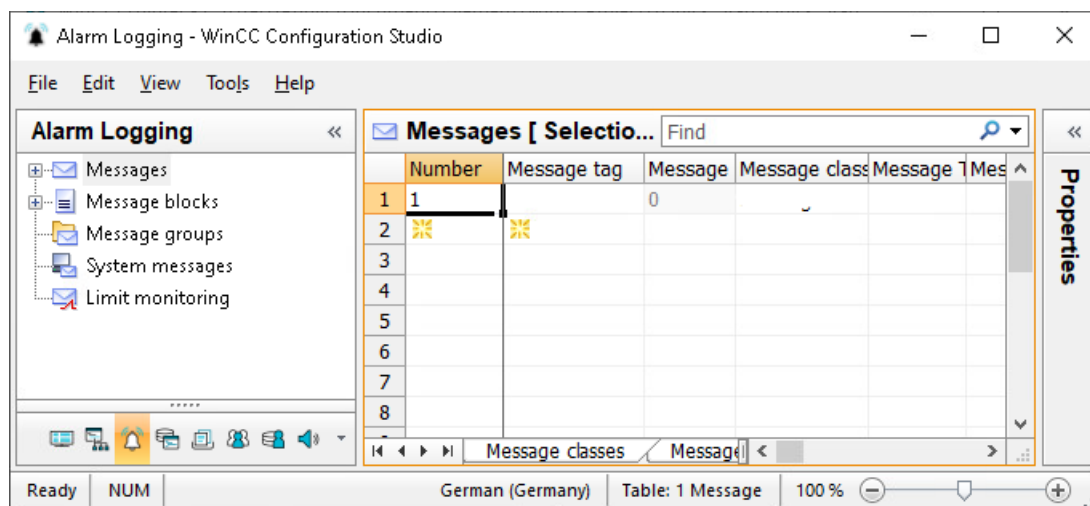
手順

1. [アラームロギング]エディタを起動します。
[アラームロギング]エディタが開きます。

結果

"アラームロギング"エディタが開いています。

次のステップで、メッセージブロックとメッセージクラスを定義します。



9.3 メッセージブロックの設定

概要

次のステップは、「Quick_Start」プロジェクトのメッセージに対するメッセージブロックの設定方法を示しています。

メッセージが、ランタイムのテーブルに表示されます。

メッセージブロック

それぞれのメッセージは、テーブルの列に表示される情報で構成されています。

これらの個別の情報が、メッセージブロックとして参照されます。それぞれのメッセージブロックは、テーブルの列に対応しています。

メッセージブロックは、3つのグループに細分されます。

- システムデータ(例: 日付、時刻、メッセージ番号、ステータス)を有するシステムブロック。
システムブロックは、事前定義されています。
- ユーザーテキストブロックには、説明テキスト(例: ロケーションおよび障害の原因に関する情報のあるテキスト)があります。
テキストは自由にカスタマイズできます。
- メッセージをプロセス値(例: 現在のフィルレベル、温度、速度)に接続するために使用するプロセス値ブロック。

メッセージブロックをランタイムに表示するためにプロパティを変更できます。

"Quick_Start"プロジェクトに使用されるユーザーテキストブロックの長さを変更します。

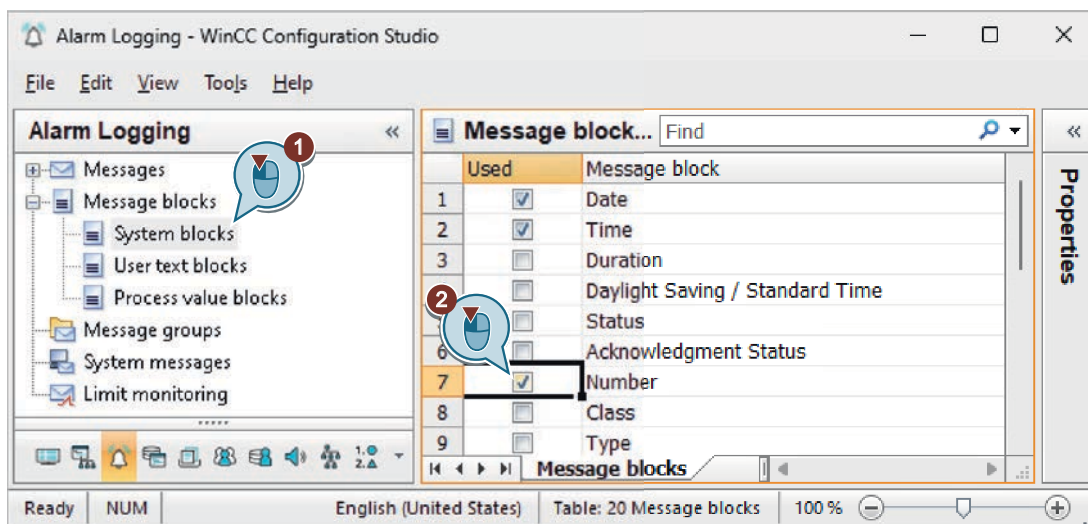
必要条件

- [アラームロギング]エディタが開いている。

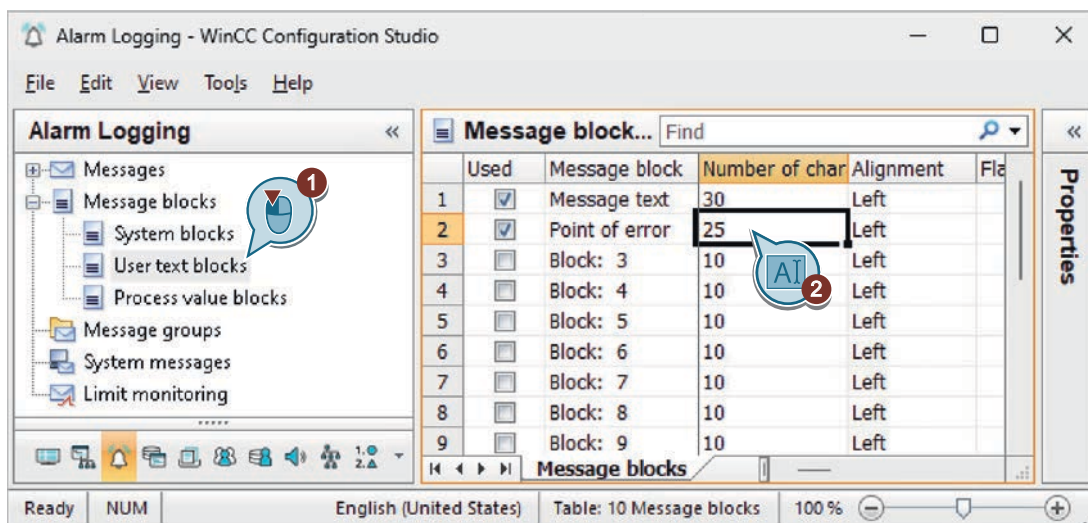
9.3 メッセージブロックの設定

手順

1. ナビゲーションエリアで、[メッセージブロック]フォルダにある[システムブロック]項目を選択します。
2. ランタイムに表示するシステムブロックを有効にします。
 - "日付"
 - "時間"
 - "番号"



3. [ユーザーテキストブロック]項目をナビゲーションエリアで選択し、[メッセージテキスト]および[エラーのポイント]ブロックを有効にします。
4. テーブルエリアまたは[プロパティ]ウィンドウのいずれかで、[文字数]フィールドのテキストの長さを変更します。
 - メッセージテキスト:30 文字
 - エラーのポイント:25 文字



結果

"Quick_Start"プロジェクトのメッセージに対するメッセージブロックが定義されています。

次のステップ

3つのディスクリートアラームを設定して、給水バルブのステータスの変化を監視します。

9.4 ビットメッセージの設定

概要

このセクションでは、[アラームロギング]エディタでのディスクリートアラームの設定方法を説明します。

概要

それぞれのディスクリートアラームは、バイオコンバータでの給水バルブのステータスに対応します。

- Valve_open (バルブが開いている)
- Valve_closed (バルブが閉じている)
- Valve_inop (バルブが故障している)

給水バルブのステータスが変化すると、対応するディスクリートアラームがトリガされ、ランタイムに表示されます。

新しい内部タグを作成して給水バルブの様々なステータスをシミュレーションできます。次いで、このタグを、作成したディスクリートアラームのメッセージタグとして設定します。

それぞれのディスクリートアラームに以下のプロパティを定義します。

メッセージタグ	メッセージタグは、プロセスにおけるステータスの変化にリンクしています。 プロセスでステータスの変化が発生すると、タグ値のビットが設定されます。タグ値に応じて、ディスクリートアラームがトリガされます。
メッセージビット	メッセージビットは、どのビットがディスクリートアラームをトリガするかを定義します。
メッセージテキスト	メッセージテキストは、例えば[バルブが開いている]などの、"Quick_Start"プロジェクトにおける給水バルブのステータスを説明します。
エラーの位置	エラーの位置はステータス変更の場所を示します。

9.4.1 ビットメッセージの作成

概要

次のステップは、[アラームロギング]エディタでのディスクリートアラームの作成方法を示しています。

"Quick_Start"プロジェクトでは、3つのディスクリートアラームをメッセージクラス[エラー]、メッセージタイプ[アラーム]で作成します。

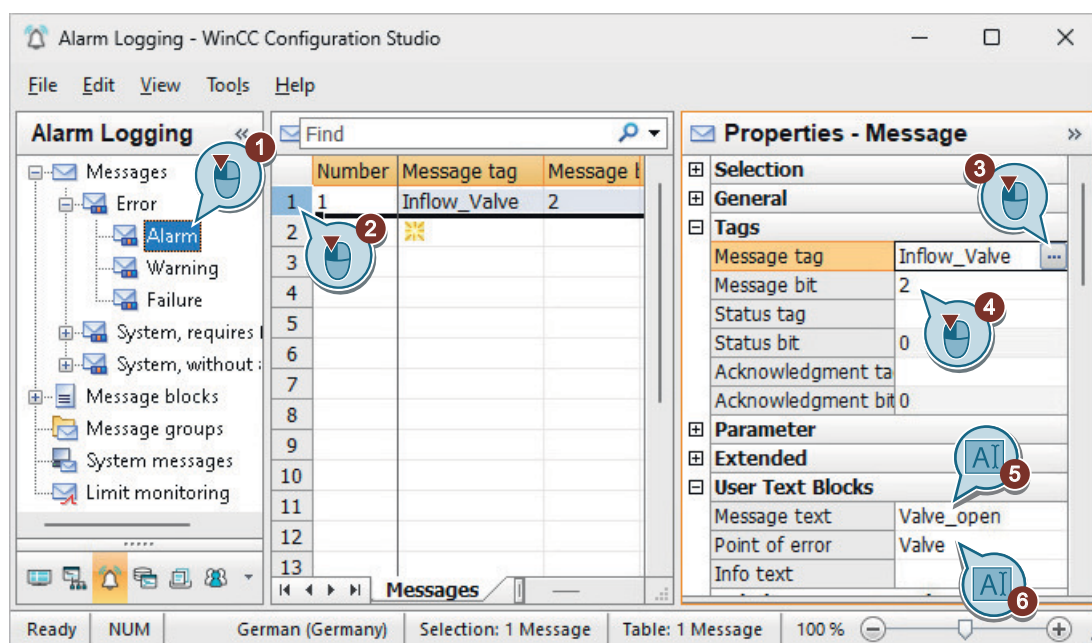
必要条件

- [アラームロギング]エディタが開いている。
 - 新しい内部タグをタグ管理で作成している。
 - 名前:"Inflow_Valve"
 - データタイプ:"符号なし 16 ビット値"
- 内部タグの作成 (ページ 43)に説明されている手順に従ってください。

9.4 ビットメッセージの設定

手順

1. ナビゲーションエリアの[メッセージ]フォルダで、[エラー]の下にある[アラーム]メッセージタイプを選択します。
 メッセージクラスとメッセージタイプの表示言語を変更できます。[表示]>[入力言語]メニューで、希望の言語を選択します。
 メッセージのプロパティを編集するには、[プロパティ]ウィンドウを開きます。
2. テーブルエリアで、先頭の行を選択し、次のプロパティを定義します。
 - メッセージタグ:"Inflow_Valve"
 - メッセージビット:2
 - メッセージテキスト:"Valve_open"
 - エラーのポイント: "Valve"



3. 2つの追加のディスクリートアラームを作成します。
 それぞれのテーブルエリアで次に利用可能な行の[番号]列をクリックします。
 メッセージの番号を入力します。

4. [プロパティ - メッセージ]エリアで、2 番目のディスクリートアラームの次のプロパティを定義します。
 - メッセージタグ:"Inflow_Valve"
 - メッセージビット:3
 - メッセージテキスト:"Valve_closed"
 - エラーのポイント: "Valve"
5. 3 番目のディスクリートアラームに以下のプロパティを定義します。
 - メッセージタグ:"Inflow_Valve"
 - メッセージビット:4
 - メッセージテキスト:"Valve_inop"
 - エラーのポイント: "Valve"

結果

"Quick_Start"プロジェクトのプロパティを持つディスクリートアラームを定義しました。作成されたディスクリートアラームは、[アラームロギング]エディタのテーブルウィンドウで表示されます。

ランタイムでのディスクリートアラームの出力はメッセージの設定でコントロールされています。例えば、内部タグ[nflow_Valve]の値で右から 2 番目のビットが設定されている場合、ディスクリートアラーム"Valve_open"がトリガされます。

Number	Message tag	Message	Message class	Message	Message text
1	Inflow_Valve	2	Error	Alarm	Valve_open
2	Inflow_Valve	3	Error	Alarm	Valve_closed
3	Inflow_Valve	4	Error	Alarm	Valve_inop
4	✖	✖			
5					
6					
7					
8					
9					
10					
11					
12					
13					

9.4 ビットメッセージの設定

次のステップ

アナログアラームを2つ設定して、内部タグ「Tank_Level」の動作を監視できるようにします。

9.5 アナログメッセージの設定

概要

この章では、"アラームロギング"エディタでアナログメッセージの設定方法を説明します。

概要

アナログアラームは、ランタイムでの制限値違反を表示します。

"Quick_Start"プロジェクトでのアナログメッセージの設定は、以下のステップから構成されています。

1. 監視するタグの定義
監視するタグを[アラームロギング]の[制限値モニタリング]で定義します。
2. 制限値の設定

"Quick_Start"プロジェクトで、内部タグ"Tank_Level"の動作を監視します。タグ値は、バイオコンバータの充填レベルをシミュレーションします。

タグの上限値および下限値を定義します。

[アラームロギング]エディタは、それぞれの定義済み制限値に対してアナログメッセージを作成します。

- 上限値は、バイオコンバータで許容される最大の水容量を定義します。
上限値が違反される場合、バイオコンバータは溢れてしまいます。
対応するアナログメッセージが、ランタイムで表示されます。
- 下限値は、バイオコンバータで必要とされる最小の水容量を定義します。
下限値が違反される場合、バイオコンバータの充填レベルは危険なレベルに低下しています。
対応するアナログメッセージが、ランタイムで表示されます。

9.5.1 制限値の設定

概要

次のステップは、内部タグ"Tank_Level"の制限値の定義方法を示しています。

1つのタグに対して、制限値はいくつでも設定できます。"アラームロギング"エディタは、それぞれの定義済み制限値に対してアナログメッセージを作成します。

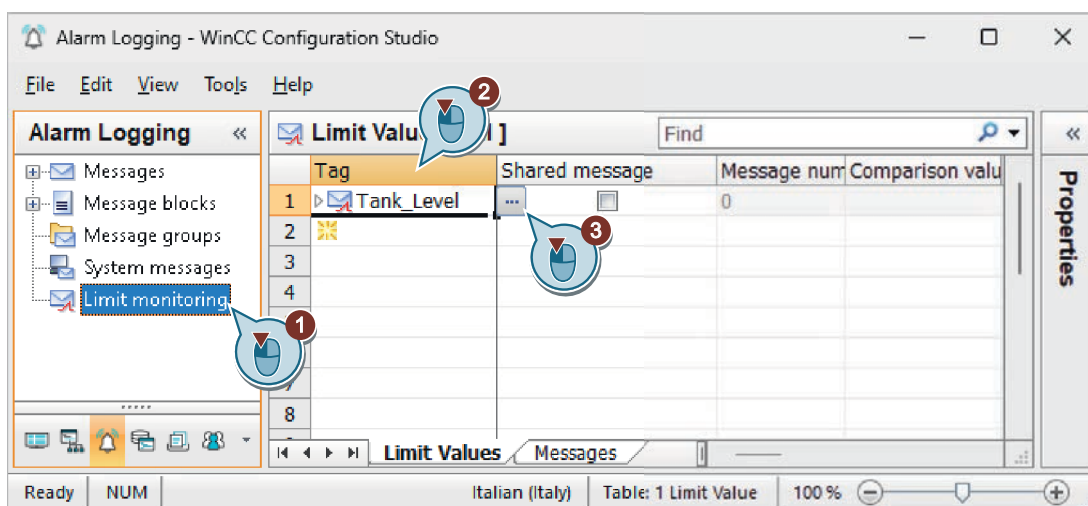
"Quick_Start"プロジェクトで、内部タグ"Tank_Level"の上限値と下限値を定義します。

必要条件

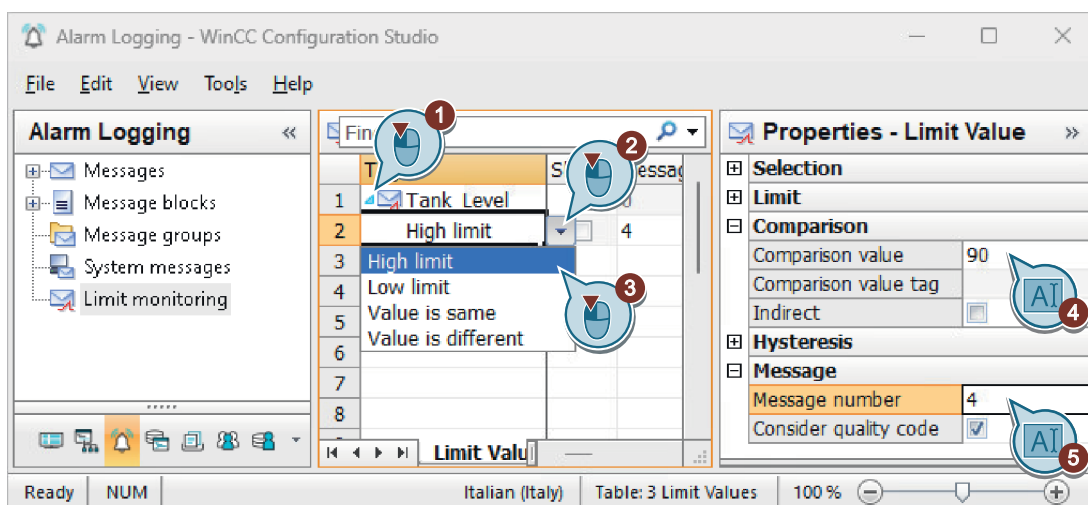
- [アラームロギング]エディタが開いている。
- 内部タグ"Tank_Level"が設定されている。

手順

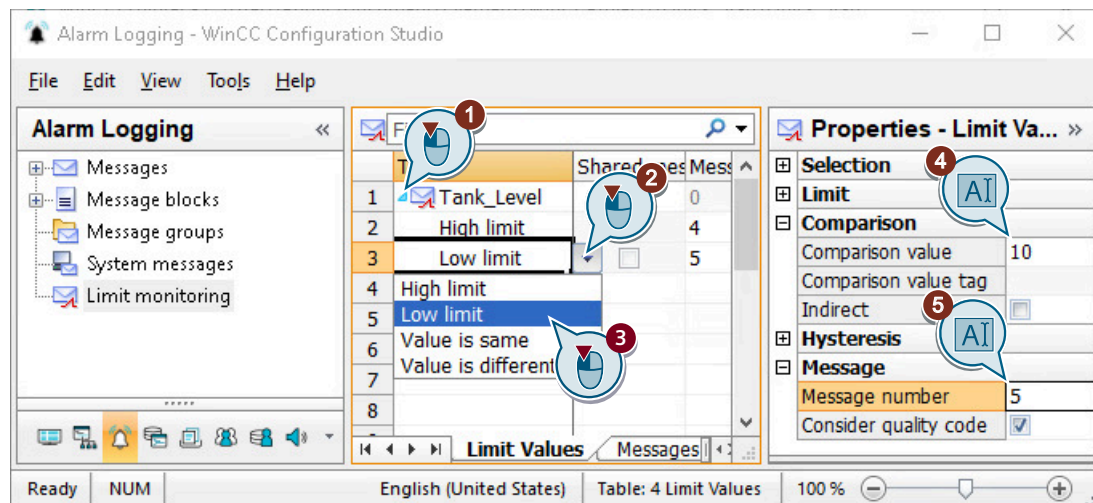
1. ナビゲーションエリアで、[制限値モニタリング]フォルダを選択します。
2. テーブルエリアで[タグ]列の先頭の空き行をクリックします。
内部タグ"Tank_Level"を選択します。



3. エントリを開くには、タグ名の前にある矢印をクリックします。
テーブルペインのタグの下で[上限]エントリを選択します。
4. 上限値のプロパティを[プロパティ - 制限値]エリアまたはテーブルエリアに入力します。
新しい、未使用のメッセージ番号を使用します。例、"4"。



5. 2番目の制限値のデータを入力するには、タグの下の次の行で[下限]エントリを選択します。
6. 下限値のプロパティを[プロパティ - 制限値]エリアまたはテーブルエリアに入力します。新しい、未使用のメッセージ番号を使用します。例、"5"。



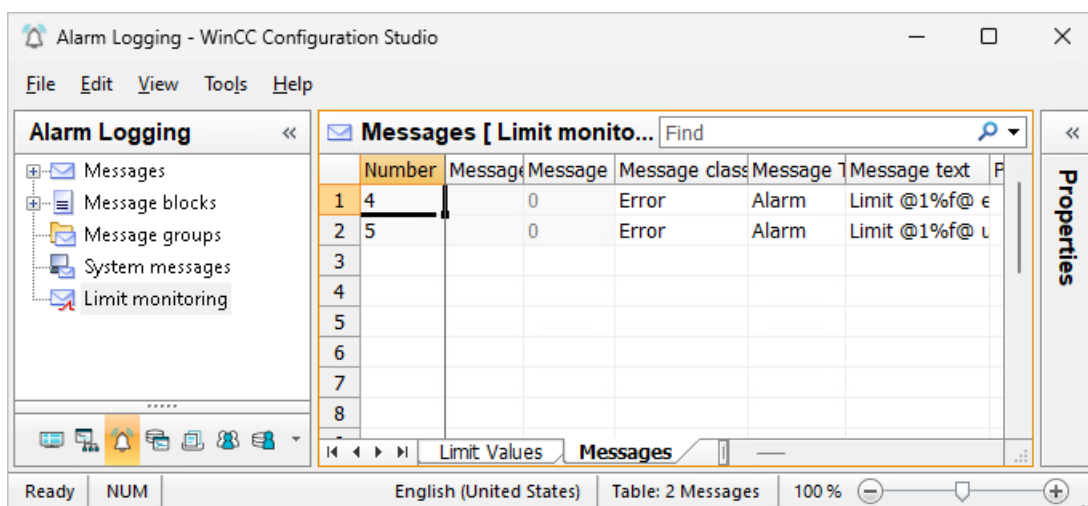
結果

内部タグ"Tank_Level"の制限値が定義されています。

- タグ値が 90 より大きい場合、アナログメッセージ[上限値]がトリガされ、ランタイムで表示されます。
このメッセージは、バイオコンバータの充填レベルを超えたことを示しています。
- タグ値が 10 より小さい場合、アナログメッセージ[下限値]がトリガされ、ランタイムで表示されます。
このメッセージは、充填レベルが許容レベルを下回っていることを示しています。

[制限値モニタリング]エントリをナビゲーションペインで選択し、[メッセージ]タブをクリックして作成済みのアナログアラームの概要を取得します。

9.5 アナログメッセージの設定



次のステップ

WinCC OnlineTrendControl での制限値違反の表示を設定します。

9.5.2 トレンドウィンドウでのメッセージの表示

概要

次のステップは、WinCC OnlineTrendControl でツールヒントとして制限値モニタリングのメッセージを表示する方法を示しています。

[タンクレベル]タグのトレンドを作成し、[アラームの表示]オプションを有効にします。

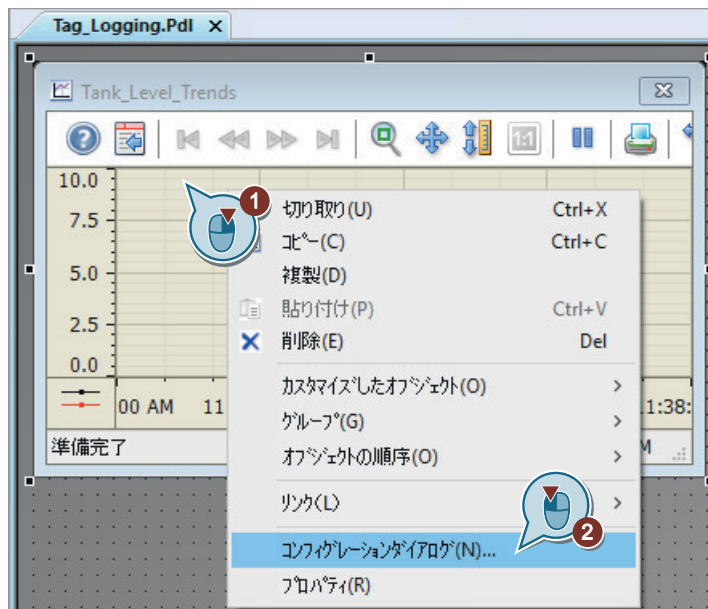
割り付けられたメッセージが、ランタイムで制限値違反のあるトレンド値に対するシンボルおよびツールヒントとして表示されます。

必要条件

- グラフィックデザイナーで「Tag_Logging.Pdl」画像が開かれている。
 - アラームロギングで制限値のモニタリングを設定している。
- 次に説明されている手順に従ってください。制限値の設定 (ページ 191)

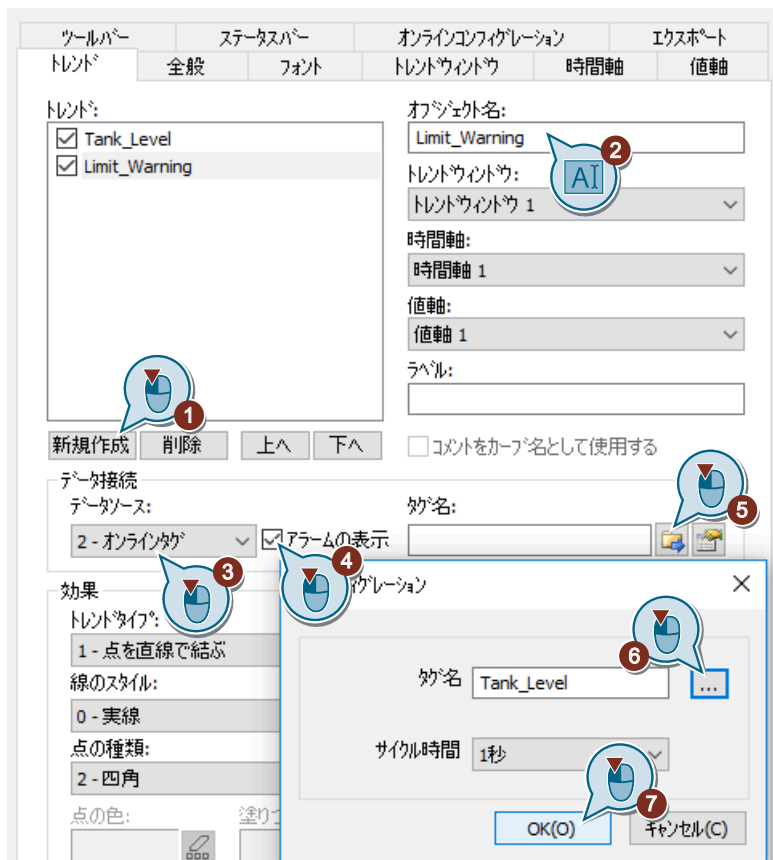
手順

1. WinCC OnlineTrendControl [Tank_Level_Trends]を選択します。
2. ショートカットメニューを使用して、[WinCC オンライントレンドコントロールのプロパティ]設定ダイアログを開きます。



9.5 アナログメッセージの設定

3. [Limit_Warning]トレンドを内部タグ[Tank_Level]に対して作成し、[アラームの表示]オプションを選択します。
タグ選択ダイアログでタグを表示するには、[データソース]エリアで[WinCC タグ]オプションの選択が必要な場合があります。

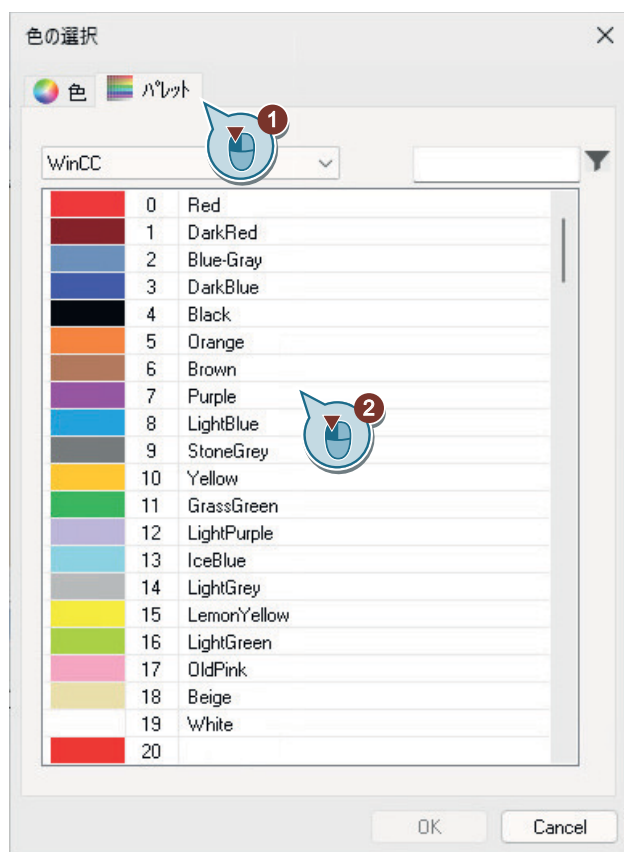


4. 下限のトレンドの色変更を設定します。



事前定義された色を適用したい場合は、[色選択]ダイアログで色パレットを選択します。

9.5 アナログメッセージの設定



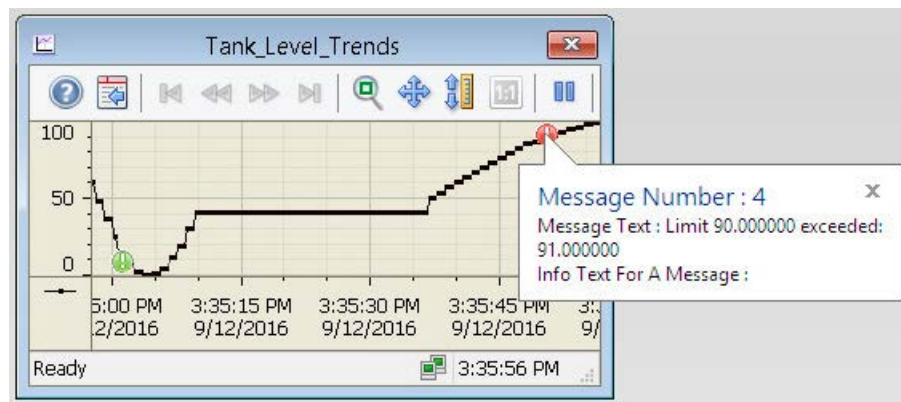
5. 上限および不特定のステータスの値のトレンドの色変更を設定します。

結果

[Limit_Warning]トレンドを内部タグ[Tank_Level]に対して設定し、それを制限値モニタリングに関連付けます。

制限値の違反は、ランタイムで表示されます。

- 赤色のシンボルは、制限値が違反されていることを示します。
- ツールヒントはメッセージのメッセージテキストを表示します。
- トレンドは色を変更します。
- 制限値内に値が戻ると、シンボルは緑色になります。
メッセージが引き続き表示されることがあります。



次のステップ

ランタイムでメッセージステータスを表示する色を定義します。

9.6 メッセージステータスの色の定義

概要

次のステップは、異なるメッセージステータスに対して表示色を定義する方法を示しています。

WinCC では、メッセージステータスの基本タイプが 3 つに分けられています。

- メッセージの原因が存在する限り、メッセージは[着信]です。
- メッセージの原因が存在しなくなると、メッセージは[発信]になります。
- ユーザーがメッセージを確認すると、メッセージは"確認済み"になります。

それぞれのメッセージの現在のステータスは、ランタイムで異なる色で表示されます。個別のメッセージステータスの表示色は、"アラームロギング"エディタで決定します。

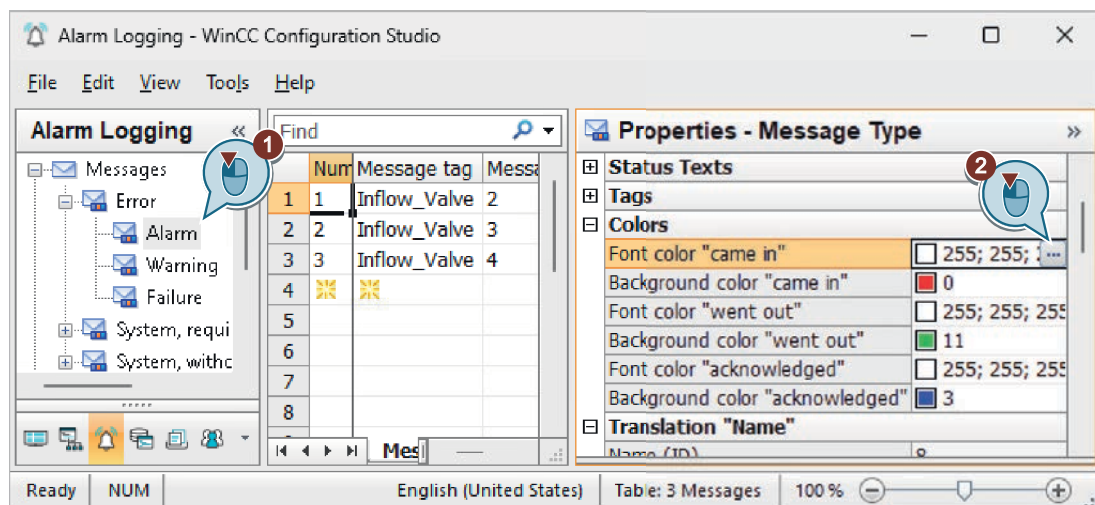
"Quick_Start"プロジェクトで、3 つの基本ステータスに対して異なるテキスト色と背景色を定義します。この定義は、メッセージクラス[エラー]のメッセージタイプ[アラーム]全体に適用されます。これは、設定を「Quick_Start」プロジェクトの全てのメッセージに適用します。

必要条件

- [アラームロギング]エディタが開いている。

手順

1. ナビゲーションエリアの[エラー]の下にある、[アラーム]メッセージタイプのフォルダを選択します。
2. [プロパティ]エリアで、メッセージタイプの色を編集します。



3. メッセージステータス[着信]の以下のプロパティを定義します。
 - フォント色:白色
 - 背景色:赤色
4. メッセージステータス[発信]の以下のプロパティを定義します。
 - フォント色:白色
 - 背景色:緑色
5. メッセージステータス「承認済み」の以下のプロパティを定義します。
 - フォント色:白色
 - 背景色:青色
6. "アラームロギング"エディタを閉じます。

結果

メッセージステータス[着信]、[発信]および[確認済み]の表示色が定義されています。
ランタイム中に、メッセージはステータスに応じてそれぞれの色で表示されます。

次のステップ

ランタイムのテグルでメッセージを出力するには、[グラフィックデザイナー]エディタのプロセス画像を設定します。

9.7 プロセス画面の設定(メッセージシステム)

概要

このセクションでは、メッセージの出力を表示するプロセス画像の設定方法を説明します。

原理

プロセス画像を[グラフィックデザイナー]エディタで設定します。

このため、以下のオブジェクトを使用します。

- WinCC AlarmControl
[WinCC AlarmControl]オブジェクトは、アラームウィンドウを作成するために使用します。メッセージは、アラームウィンドウのテーブルに表示されます。出力はランタイムで行なわれます。
- スライダオブジェクト
「Quick_Start」プロジェクトで、スライダオブジェクトを使用して、アナログ値を内部タグ「Tank_Level」に転送できるようにします。
転送した値が定義済みの制限値に違反している場合、対応するアナログアラームがトリガされます。
- I/O フィールド
[I/O フィールド]オブジェクトを、「Quick_Start」プロジェクトの内部タグ「Inflow_Valve」とリンクします。
I/O フィールドに2進数値を入力します。これらの値は、ランタイムでタグ「Inflow_Valve」に割り当てられます。
特定のビットがタグ値に設定されている場合、対応するディスクリートアラームがトリガされます。

9.7.1 アラームメッセージウィンドウの設定

概要

次のステップは、メッセージウィンドウの設定方法を示しています。

[グラフィックデザイナー]エディタでメッセージウィンドウを設定します。

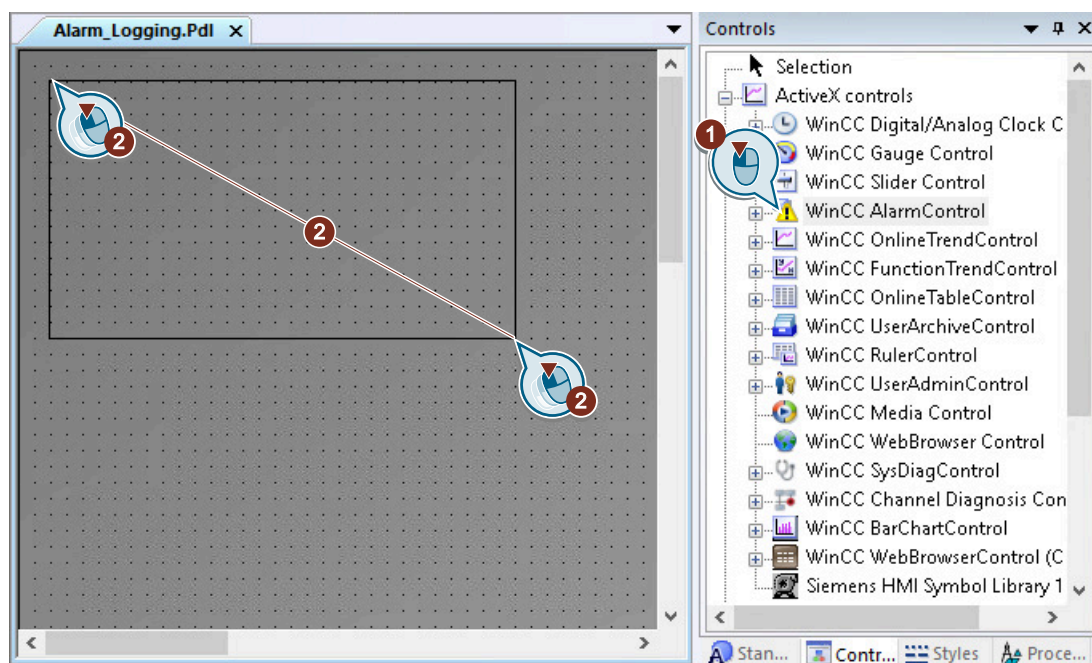
このため、新しいプロセス画像を作成します。"WinCC AlarmControl"オブジェクトをプロセス画像に挿入します。このオブジェクトは、メッセージの表示のために事前に設定されています。

WinCC AlarmControl のプロパティを使用して、メッセージウィンドウの列としてどのメッセージブロックを表示するかを定義します。

ランタイムで、メッセージはこれらのメッセージブロックで構成されます。

手順

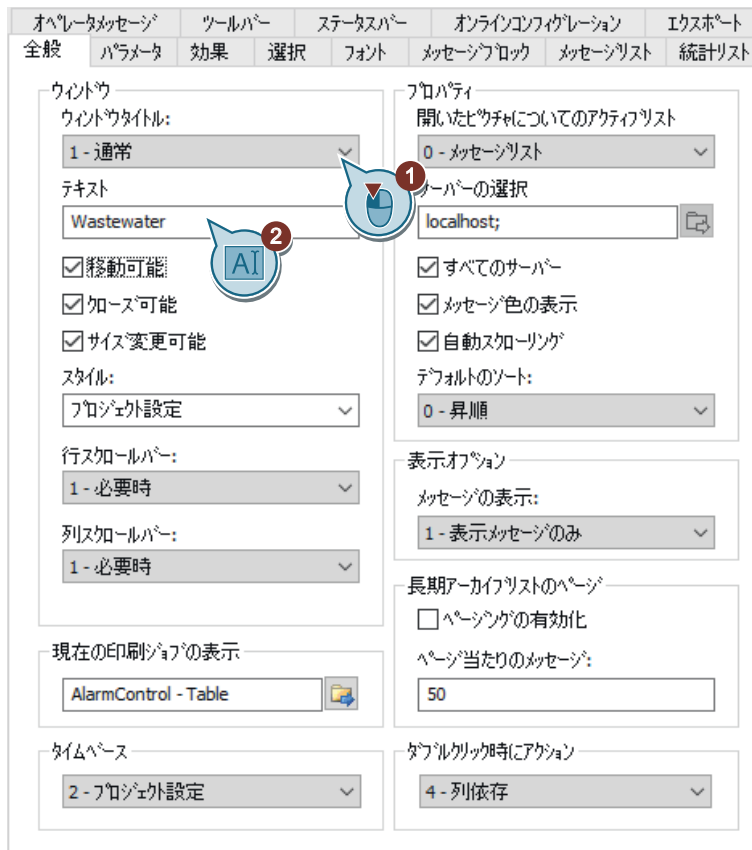
1. 「Alarm_Logging.Pdl」という名前の新しいプロセス画像を作成し、[グラフィックデザイナー]エディタで開きます。
2. "WinCC AlarmControl"オブジェクトをプロセス画像に挿入します。



[WinCC AlarmControl プロパティ]ダイアログが開きます。

9.7 プロセス画面の設定(メッセージシステム)

3. メッセージウィンドウの名前を指定します。



4. [メッセージブロック]タブをクリックします。
 [プロジェクト設定の適用]設定が有効になっていることを確認します。
 これにより、アラームロギングからのメッセージブロックの設定が適用されます。

オペレータメッセージ	ツールバー	ステータスバー	オンラインコンフィグレーション	エクスポート
全般	パラメータ	効果	選択	フォント
			メッセージブロック	メッセージリスト
				統計リスト

使用可能なメッセージブロック:

- 日付
- 時刻
- 継続時間
- 夏時間/標準時間
- ステータス
- 確認ステータス
- 番号
- クラス
- タイプ
- コントローラ/CPU番号
- タグ

表示

- テキストとしてのコンテンツ
- アイコンとしてのコンテンツ
- テキストとしてのタイトル
- アイコンとしてのタイトル

オブジェクト名:

Date

ラベル: 日付 テキストID: 14

配置: 0 - 左揃え 長さ(文字数): 8

メッセージ色の表示

選択可能な選択ダイアログ

点滅オン

0 - 標準

書式

時間の形式:

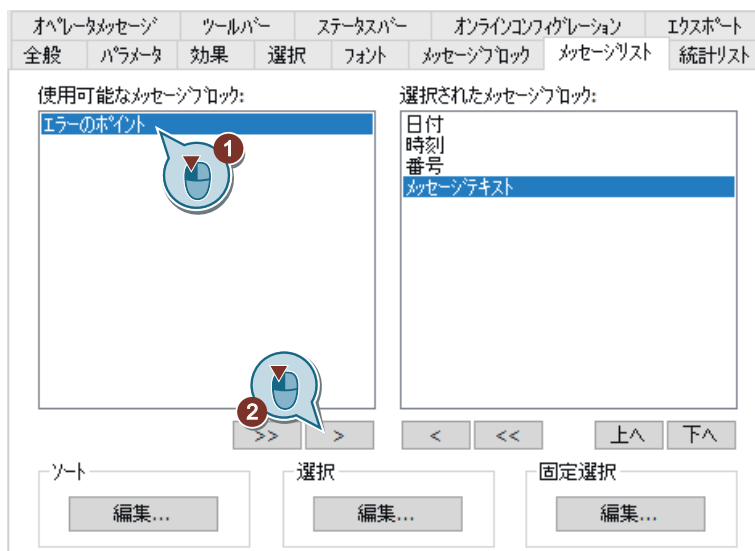
日付の形式: dd/MM/yy

日付の表示

プロジェクト設定の適用

9.7 プロセス画面の設定(メッセージシステム)

- [メッセージリスト]タブをクリックします。
ユーザーテキストブロック[メッセージテキスト]および[エラーのポイント]を[選択されているメッセージブロック]フィールドに移動します。



ユーザーテキストブロック[メッセージテキスト]および[エラーのポイント]は、メッセージウィンドウに表示されます。

- [OK]を押してダイアログを閉じます。
必要な場合に、メッセージウィンドウを拡大します。

結果

[水処理]メッセージウィンドウが設定されました。

「Quick_Start」プロジェクトに対して設定したメッセージはランタイム中にこのウィンドウで表示されます。

メッセージがトリガされるかどうかは、内部タグ"Tank_Level"と"Inflow_Valve"の値により異なります。メッセージの表示色は、メッセージステータスに応じて変化します。

メッセージの内容は、メッセージウィンドウの以下のメッセージブロックによって構成されます。

- システムブロック:日付、時刻および番号
- ユーザーテキストブロック:メッセージテキストおよびエラーのポイント

	日付	時刻	番号	メッセージテキスト	エラーのポイント
91	04/10/18	04:43:48 午	91	TEXT	TEXT
92	04/10/18	04:43:49 午	92	TEXT	TEXT
93	04/10/18	04:43:50 午	93	TEXT	TEXT
94	04/10/18	04:43:51 午	94	TEXT	TEXT
95	04/10/18	04:43:52 午	95	TEXT	TEXT
96	04/10/18	04:43:53 午	96	TEXT	TEXT
97	04/10/18	04:43:54 午	97	TEXT	TEXT
98	04/10/18	04:43:55 午	98	TEXT	TEXT
99	04/10/18	04:43:56 午	99	TEXT	TEXT
100	04/10/18	04:43:57 午	100	TEXT	TEXT

次のステップ

スライダオブジェクトを設定して、内部タグ「Tank_Level」にアナログ値を供給できるようにします。

9.7.2 スライダオブジェクトの挿入とダイナミック化

概要

次のステップは、スライダオブジェクトを挿入し、ダイナミック化する方法を示しています。

スライダオブジェクトは、タグ値の表示や変更のために使用します。スライダオブジェクトのプロセスタグへの接続によって、AS を制御できます。

「Quick_Start」プロジェクトで、「Alarm_Logging.Pdl」プロセス画像にスライダオブジェクトを挿入します。

内部タグ「Tank_Level」への接続により、スライダオブジェクトをダイナミック化します。

ランタイムにスライダオブジェクトを使用する場合、内部タグ"Tank_Level"に値が割り当てられます。割り当てられた値が定義済みの制限値のいずれかに違反している場合、対応するアナログアラームがアラームウィンドウに表示されます。

9.7 プロセス画面の設定(メッセージシステム)

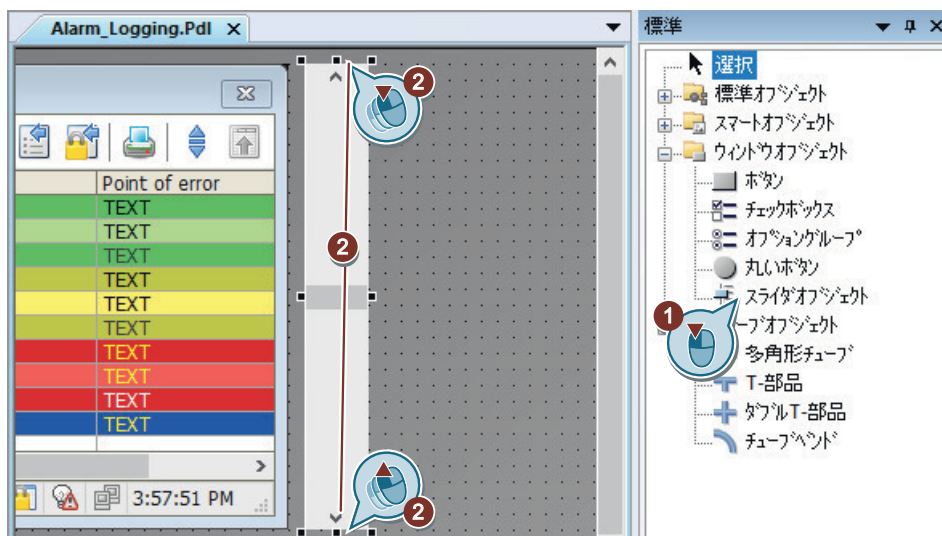
「Quick_Start」プロジェクトに対して、スライダオブジェクトのプリセットプロパティを変更します。[オブジェクトプロパティ]ウィンドウで、スライダオブジェクトの新しい名前を入力し、その高さを定義します。

必要条件

- 「Alarm_Logging.Pdl」プロセス画像が開いている。
- 内部タグ"Tank_Level"が作成されている。

手順

1. 「Alarm_Logging.Pdl」プロセス画像に、スライダオブジェクトを挿入します。



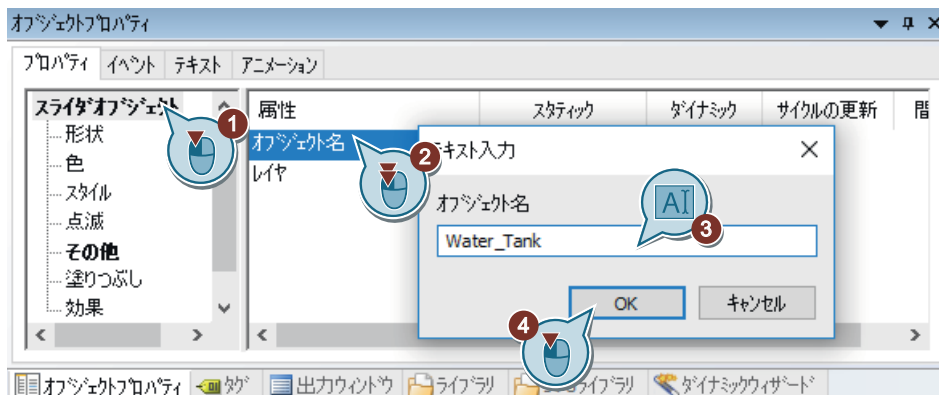
[スライダ設定]ダイアログが開きます。

2. スライダオブジェクトを内部タグ"Tank_Level"にリンクさせます。

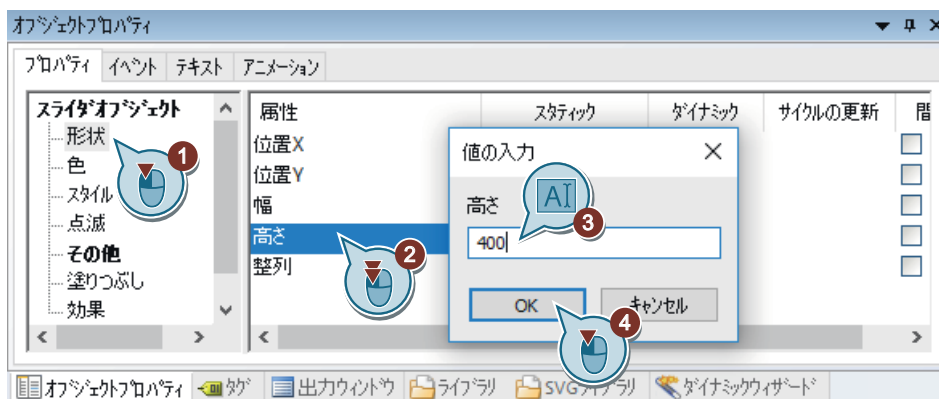


9.7 プロセス画面の設定(メッセージシステム)

- [オブジェクトプロパティ]ウィンドウに切り替えます。
スライダオブジェクトの名前として、"Water_Tank"を入力します。



- スライダオブジェクトの高さとして、[400]と定義します。



結果

スライダオブジェクト"Water_Tank"が挿入され、ダイナミック化されています。

スライダオブジェクトのダイナミック化により、内部タグ"Tank_Level"への値の転送が可能になります。このタグに対して制限値モニタリングが設定されているので、設定した値が違反している場合は対応するメッセージがトリガされます。

次のステップ

スライダオブジェクトで値を設定できるようにするために、スケールを挿入します。

9.7.3 スケールの挿入

概要

次のステップは、[グラフィックデザイナー]エディタのライブラリからスケールを挿入する方法を示しています。

「Quick_Start」プロジェクトで、「Alarm_Logging.Pdl」プロセス画像にスケールを挿入します。

このスケールによって、スライダオブジェクト「Water_Tank」が想定できる値を表示します。スケール上のラインは、スライダオブジェクトの操作ステップに対応します。

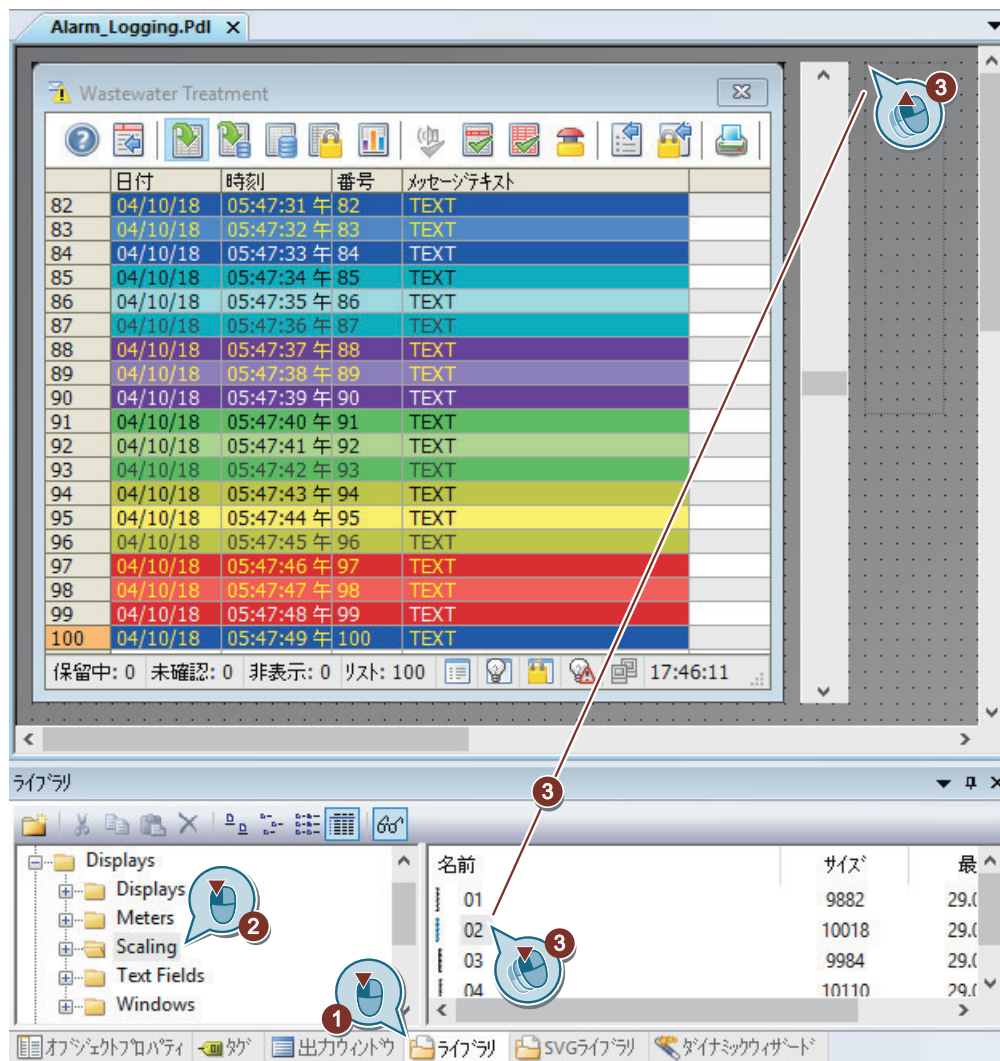
必要条件

- 「Alarm_Logging.Pdl」プロセス画像が開いている。
- スライダオブジェクト"Water_Tank"が挿入されている。

9.7 プロセス画面の設定(メッセージシステム)

手順

1. グローバルライブラリに切り替えます。
2. スケール"02"を挿入します。



スケールは、プロセス画像「Alarm_Logging.Pdl」に表示されます。

3. [オブジェクトプロパティ]に切り替えます。
スケールの高さとして、[400]と定義します。

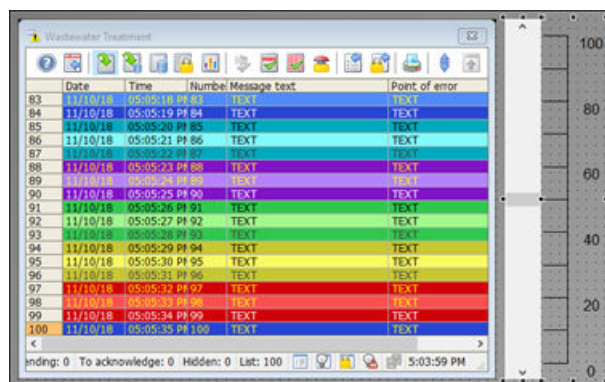


9.7 プロセス画面の設定(メッセージシステム)

4. スケールのフォントサイズとして、"16"を設定します。



5. スケールとスライダオブジェクトを同じ水平線に並べます。



結果

スケールが、「Alarm_Logging.Pdl」プロセス画像に挿入されています。

ランタイムで、このスケールによってスライダオブジェクト「Water_Tank」の値を設定します。

次のステップ

I/O フィールドを挿入して、内部タグ「Inflow_Valve」を2進数値と共に供給できるようにします。

9.7.4 I/O フィールド(メッセージシステム)の挿入とダイナミック化

概要

次のステップは、I/O フィールドを挿入し、ダイナミック化する方法を示しています。

「Quick_Start」プロジェクトで、「Alarm_Logging.Pdl」プロセス画像に I/O フィールドを挿入します。

内部タグ「Inflow_Valve」への接続により、I/O フィールドをダイナミック化します。

このタグにステータスを保存することにより、I/O フィールドの値のバイナリ出力形式を定義します。

I/O フィールドは、ランタイム中にバイナリ値を内部[Inflow_Valve]タグに転送するために使用されます。

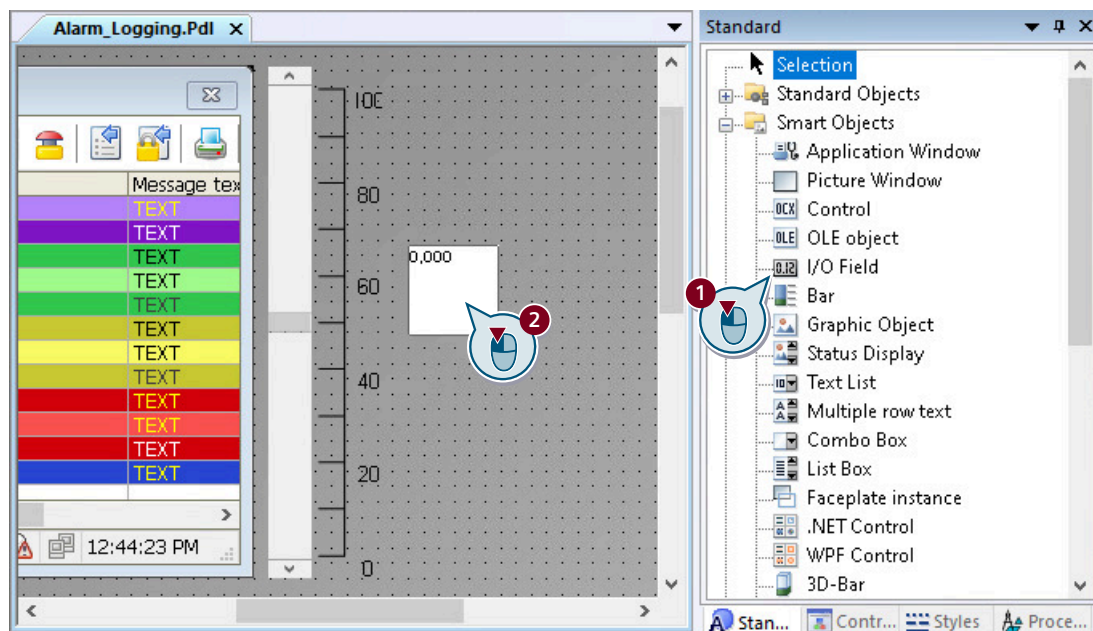
必要条件

- 「Alarm_Logging.Pdl」プロセス画像が開いている。
- 内部タグ"Inflow_Valve"が作成されている。

9.7 プロセス画面の設定(メッセージシステム)

手順

1. [I/O フィールド]スマートオブジェクトを挿入するには、[標準]ウィンドウでオブジェクトをクリックしてから、プロセス画像をクリックします。

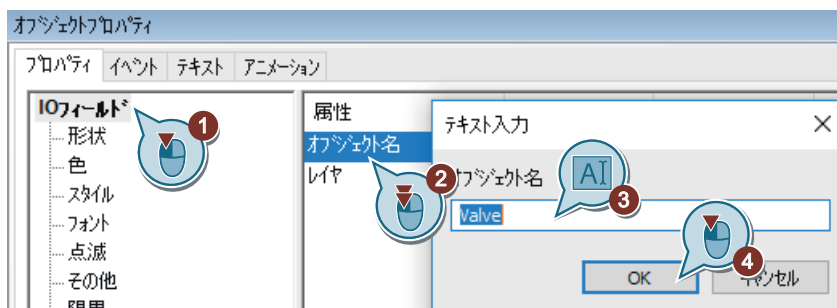


[I/O フィールド設定]ダイアログが開きます。

2. I/O フィールドを内部タグ"Inflow_Valve"にリンクします。



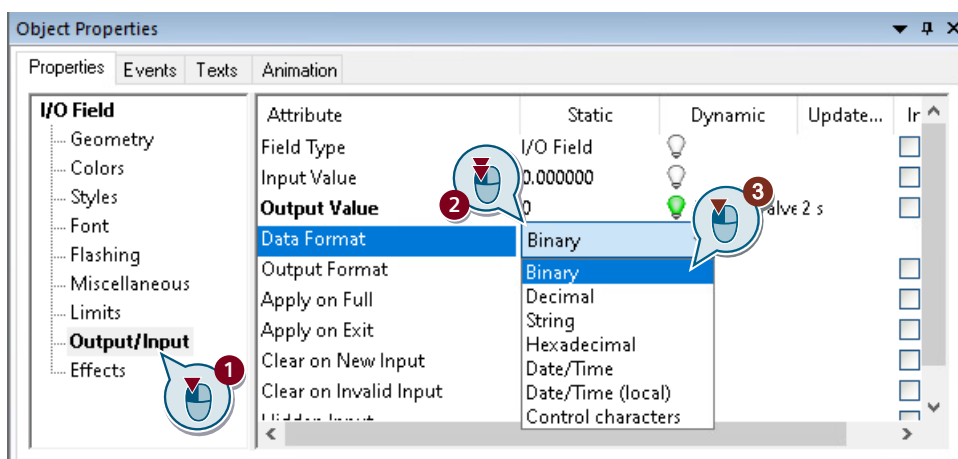
- [オブジェクトプロパティ]ウィンドウに切り替えます。
I/O フィールドの名前として、"バルブ"と入力します。



- I/O フィールドの以下のフォントプロパティを定義します。
 - フォントサイズ:28
 - 太字:はい
 - X 整列:右揃え
 - Y 整列:中央揃え

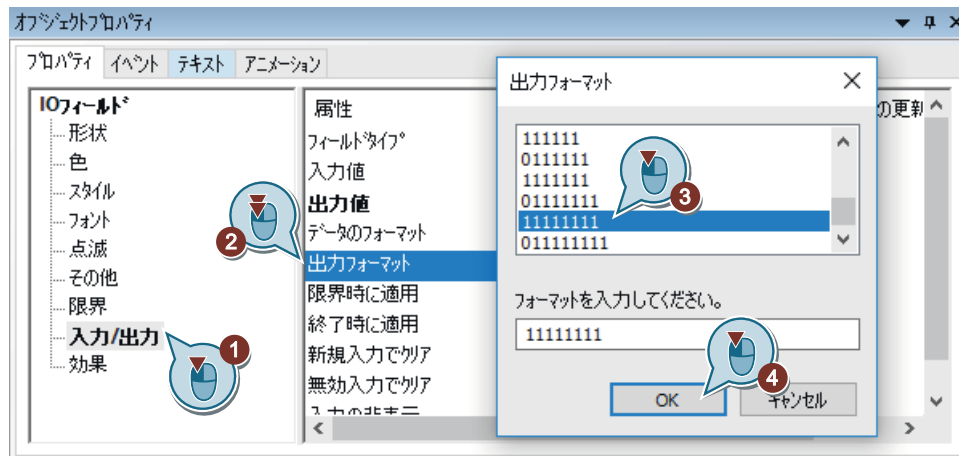


- I/O フィールドの出力フォーマットとして、[2 進数]を定義します。



9.7 プロセス画面の設定(メッセージシステム)

6. [出力フォーマット]プロパティに対して、位置の数を 6 から 8 へ増やします。"11111111"。



7. I/O フィールドを拡大し、「Alarm_Logging.Pdl」プロセス画像を保存します。
 8. [グラフィックデザイナー]を閉じます。

結果

I/O フィールド"バルブ"が設定されています。

設定済み I/O フィールドに 2 進数値を入力します。これらの値は、内部タグ「Inflow_Valve」に転送されます。

タグ値で特定のビットが設定されると、対応するディスクリートアラームがトリガされ、メッセージウィンドウに表示されます。例えば、2 番目のビットがタグ値で設定されると、ディスクリートアラーム"Valve_closed"がトリガされます。

	日付	時刻	番号	メッセージテキスト
82	08/10/18	05:07:01 午	82	TEXT
83	08/10/18	05:07:02 午	83	TEXT
84	08/10/18	05:07:03 午	84	TEXT
85	08/10/18	05:07:04 午	85	TEXT
86	08/10/18	05:07:05 午	86	TEXT
87	08/10/18	05:07:06 午	87	TEXT
88	08/10/18	05:07:07 午	88	TEXT
89	08/10/18	05:07:08 午	89	TEXT
90	08/10/18	05:07:09 午	90	TEXT
91	08/10/18	05:07:10 午	91	TEXT
92	08/10/18	05:07:11 午	92	TEXT
93	08/10/18	05:07:12 午	93	TEXT
94	08/10/18	05:07:13 午	94	TEXT
95	08/10/18	05:07:14 午	95	TEXT
96	08/10/18	05:07:15 午	96	TEXT
97	08/10/18	05:07:16 午	97	TEXT
98	08/10/18	05:07:17 午	98	TEXT
99	08/10/18	05:07:18 午	99	TEXT
100	08/10/18	05:07:19 午	100	TEXT

保留中: 0 未確認: 0 非表示: 0 リスト: 100 17:05:44

次のステップ

- 「Alarm_Logging.Pdl」 ページをユーザー定義メニューに追加します。
- WinCC Runtime のプロパティを定義します。

9.8 ユーザー定義メニューのカスタマイズ(メッセージシステム)

概要

次のステップは、[アラームロギング]メニュー項目をカスタマイズされた[画像変更]メニューに追加する方法を示しています。

"アラームロギング"メニューエントリをプロシージャ"ActivatePicture(ByVal PictureName)"と接続させます。

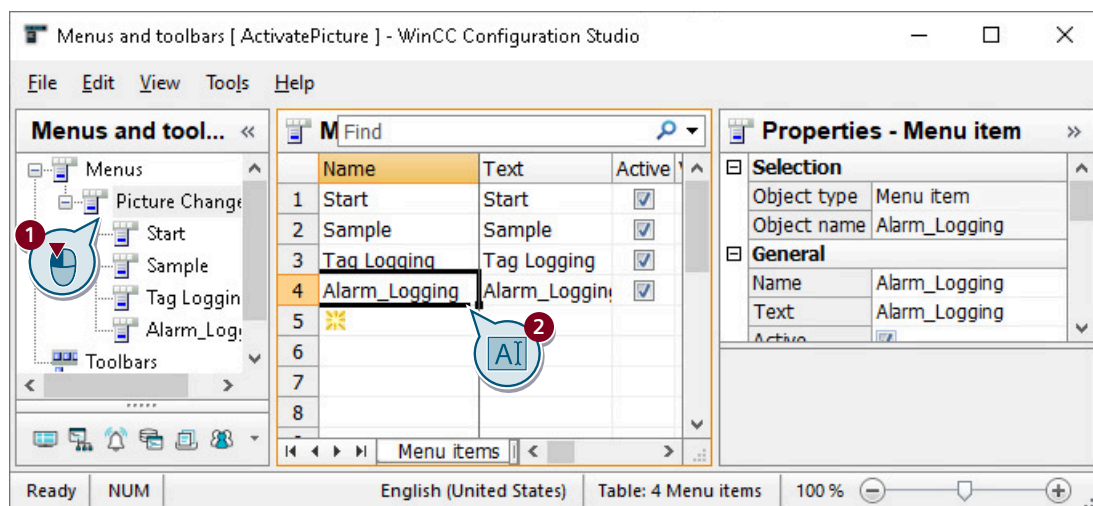
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

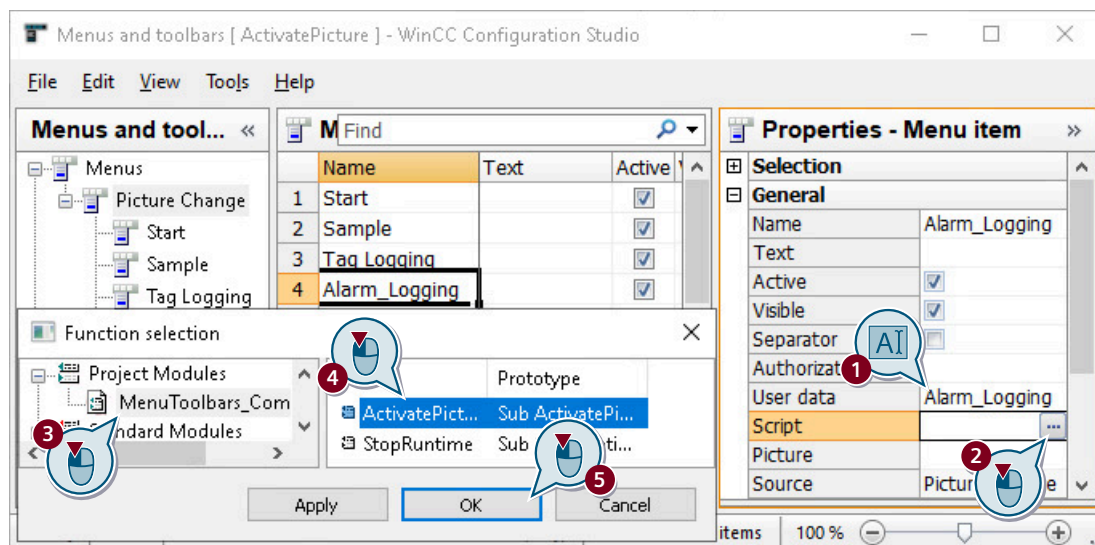
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- ユーザー定義の[画像変更]メニューが「START.Pdl」、「SAMPLE.Pdl」、「Tag_Logging.Pdl」のプロセス画像用に作成されていること。

手順

1. 保存された設定[ActivatePicture.mtl]を[メニューとツールバー]エディタで開きます。
2. [アラームロギング]メニューコマンドを作成します。



3. 「Alarm_Logging.pdl」画像に変更するために、[アラームロギング]メニューコマンドを設定します。



4. [メニューとツールバー]エディタで変更内容を保存します。
5. "メニューとツールバー"エディタを閉じます。

結果

[アラームロギング]メニュー項目を[画像変更]メニューに追加しました。
ランタイムでメニューエントリを使用して、「START.Pdl」、「SAMPLE.Pdl」、
「Tag_Logging.Pdl」および「Alarm_Logging.Pdl」のプロセス画像に切り替えられるように
します。

9.9 ランタイムプロパティの定義(メッセージシステム)

概要

次のステップは、WinCC Runtime のプロパティの定義方法を示しています。

このセクションでは、プロジェクトの起動時にアラームロギングランタイムが実行されるように WinCC Runtime をセットアップします。

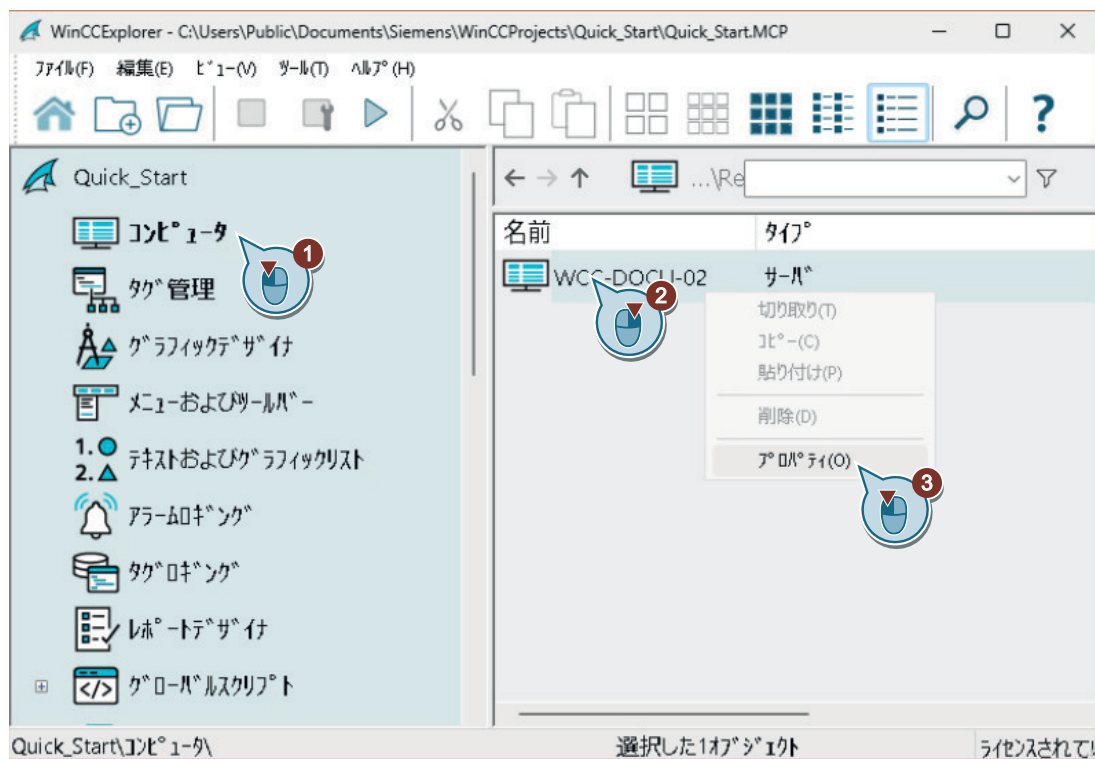
「Alarm_Logging.Pdl」プロセス画像を、ランタイムウィンドウの開始画像として設定します。

必要条件

- "Quick_Start"プロジェクトが開いている。
- 「Alarm_Logging.Pdl」プロセス画像が作成されている。

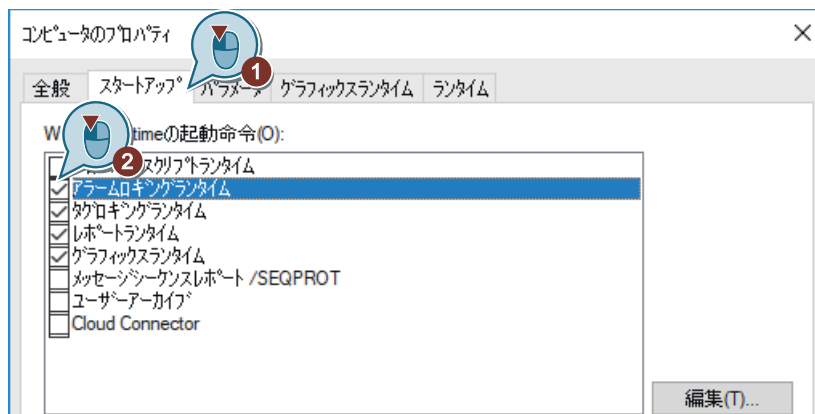
手順

1. [コンピュータ]エディタを開きます。



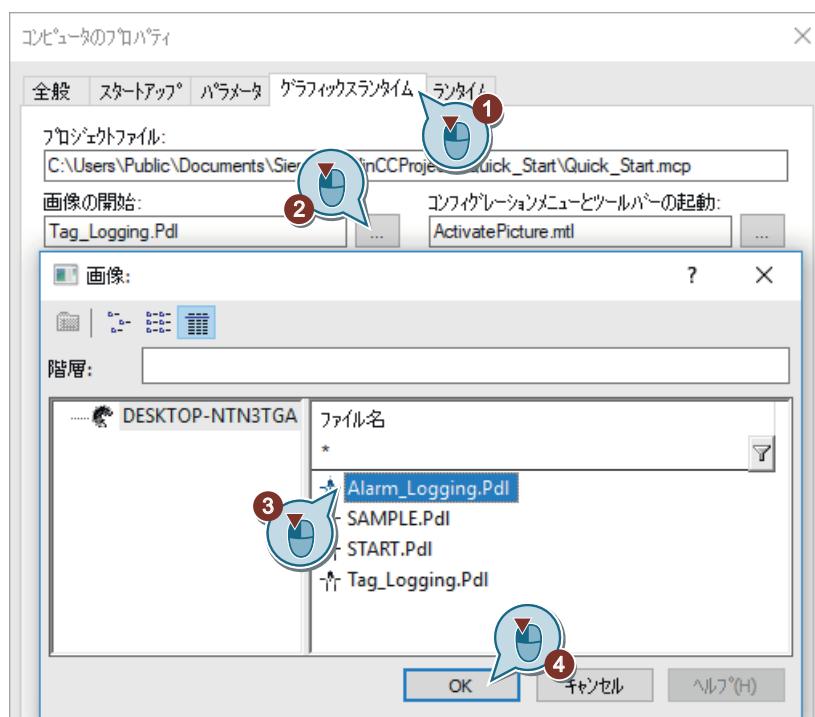
[コンピュータ]エディタが WinCC Configuration Studio で開きます。

2. ナビゲーションエリアでコンピュータ名を選択します。
[WinCC Runtime の起動時のプロセス]タブで[アラームロギングランタイム]アプリケーションを有効にします。



3. [プロパティ-コンピュータ]エリアの[グラフィック]で、「Alarm_Logging.Pdl」プロセス画像をランタイムウィンドウの初期画像として設定します。

9.9 ランタイムプロパティの定義(メッセージシステム)



結果

WinCC Runtime のプロパティが定義されています。

「Quick_Start」プロジェクトを有効にするときに、タグロギングランタイムが実行され、「Alarm_Logging.Pdl」プロセス画像が表示されます。

次のステップ

「Quick_Start」プロジェクトを有効にします。

9.10 プロジェクトの有効化(アラームロギング)

概要


次のステップは、「Quick_Start」プロジェクトを有効にする方法や、ランタイムでの「Alarm_Logging.Pdl」プロセス画像の操作方法を示しています。

"Quick_Start"プロジェクトを有効にすると、WinCC Runtime が開始します。

開始画像として、「Alarm_Logging.Pdl」プロセス画像が表示されます。

ランタイムのメッセージウィンドウ

ランタイムで、メッセージウィンドウを操作、監視します。ツールバーのボタンによってメッセージウィンドウを使用します。

 ボタンによって、メッセージリストを表示できます。メッセージリストは、現在保留中のメッセージを含みます。


メッセージウィンドウのメッセージの表示色は、メッセージステータスによって変化します。

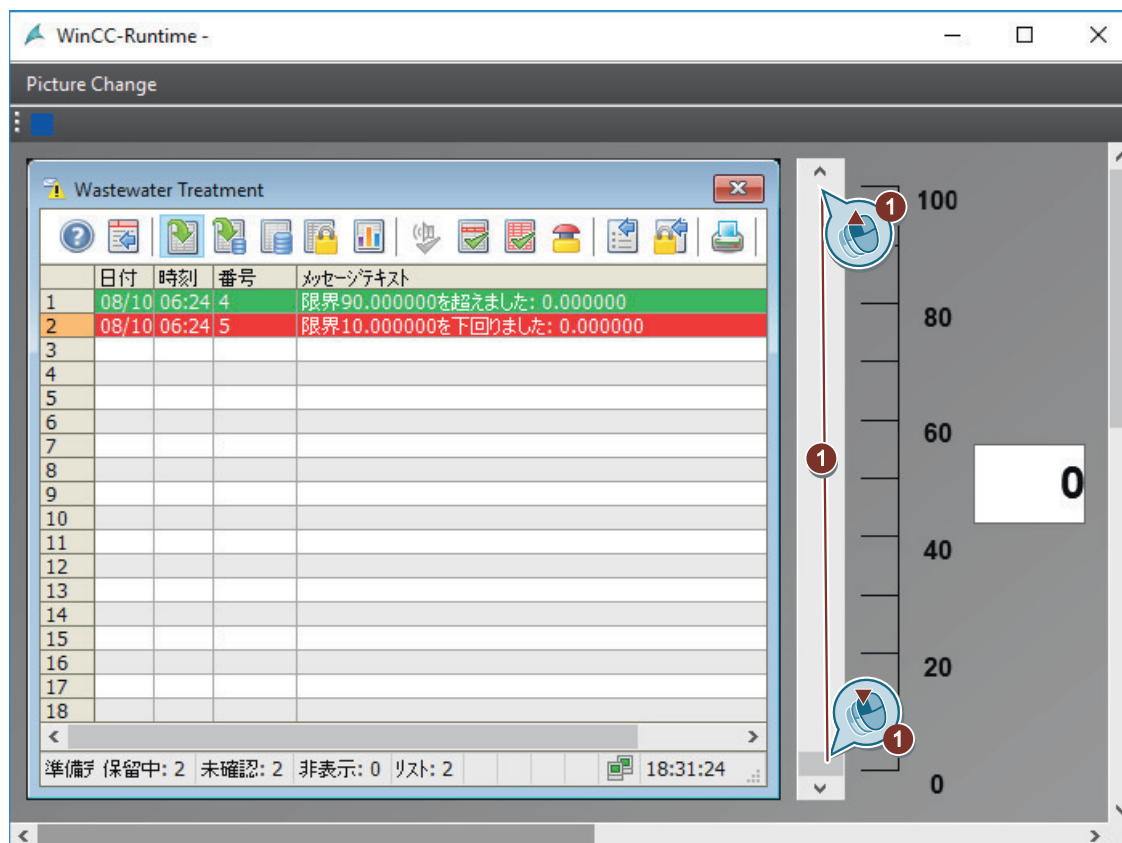
必要条件

- "Quick_Start"プロジェクトが開いている。
- ランタイムプロパティが定義されている。

9.10 プロジェクトの有効化(アラームロギング)

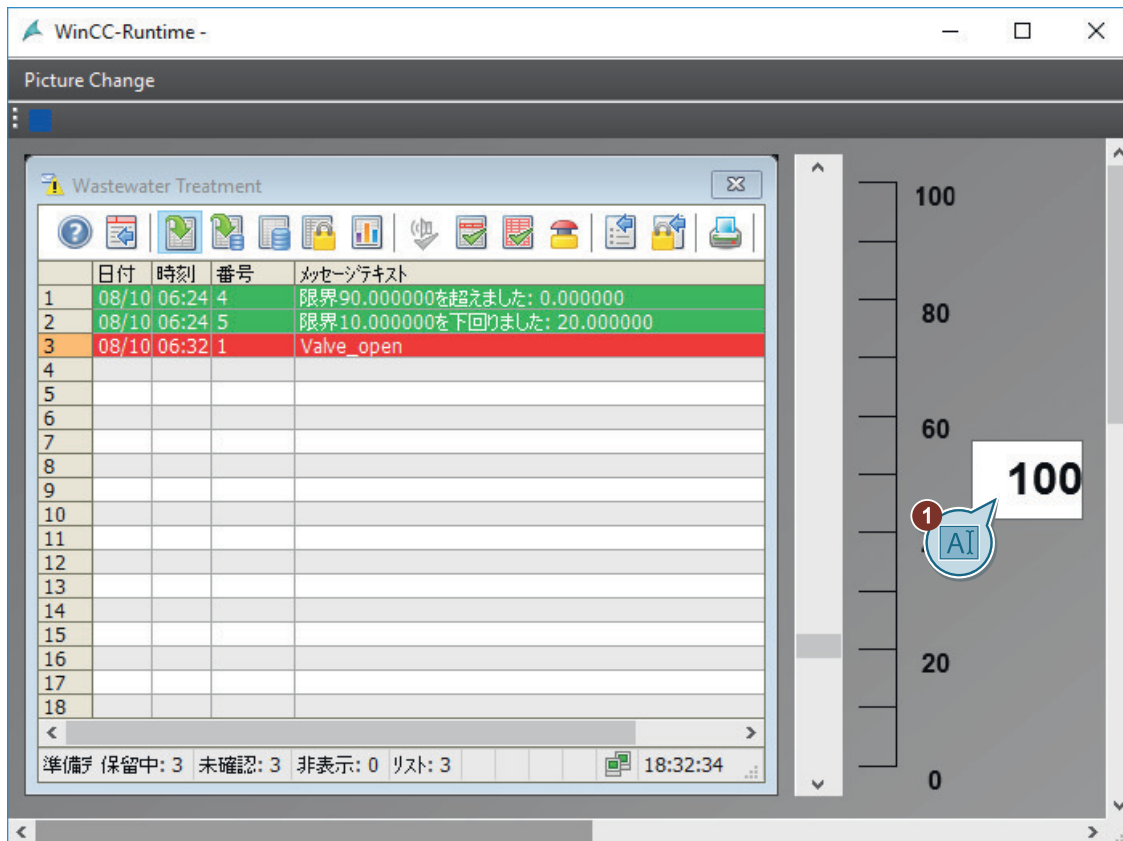
手順

1. WinCC エクスプローラの  ツールバーボタンを使って"Quick_Start"プロジェクトを有効にします。
WinCC Runtime が起動します。
「Alarm_Logging.Pdl」プロセス画像がランタイムウィンドウに表示されます。
2. [Water_Tank]スライダオブジェクトのコントローラを動かします。



内部タグ「Tank_Level」に、スライダ位置に対応する値が割り付けられます。
この値が設定されている下限値(10)を下回った場合、アナログメッセージ「下限値」がトリガされます。
上限値(90)が違反される場合、アナログアラーム「上限値」がトリガされます。


3. [バルブ] I/O フィールドに値[100]を入力し、<Return>で入力内容を確定します。

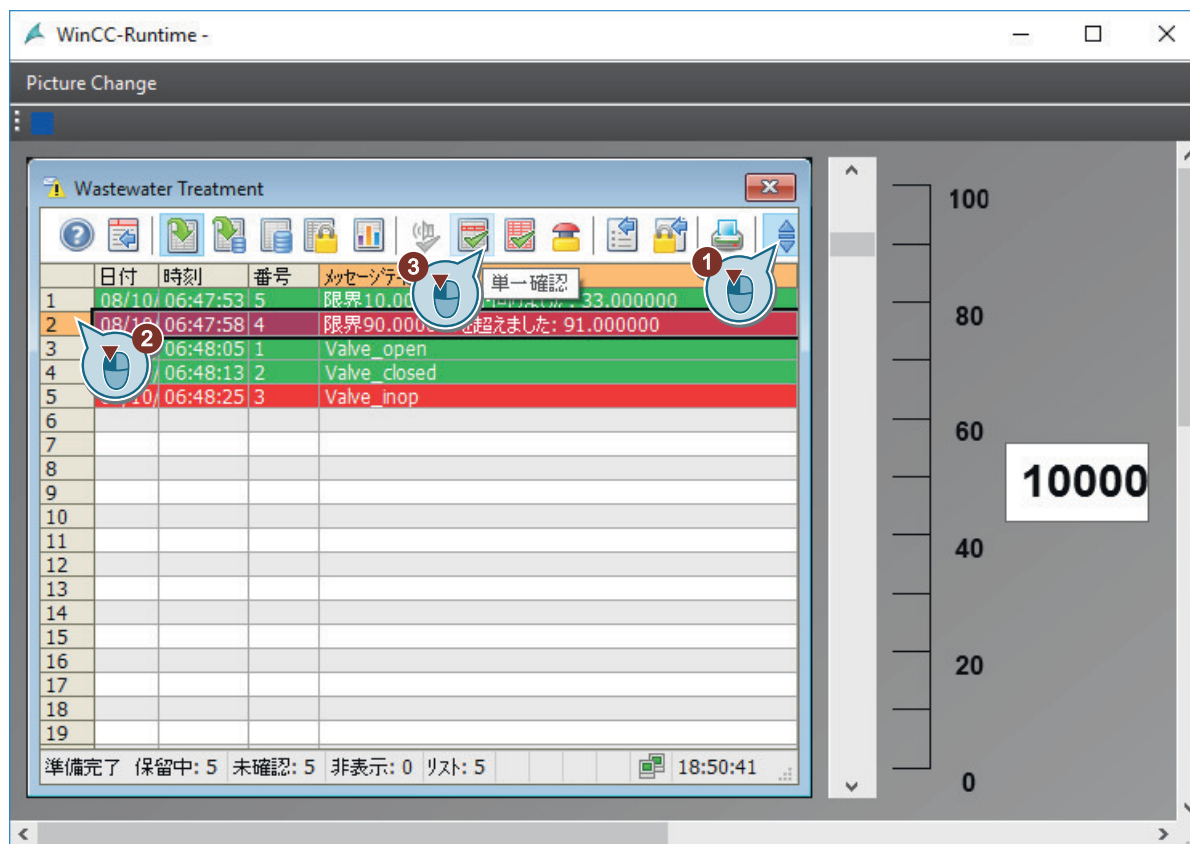


タグ値の 2 番目のビットが設定されます。
ディスクリットアラーム"Valve_open"が表示されます。

4. I/O フィールドに値[1000]を入力し、<Return>で入力内容を確定します。
タグ値の 3 番目のビットが設定されます。
ディスクリットアラーム"Valve_closed"が表示されます。
5. I/O フィールドに値[10000]を入力し、<Return>で入力内容を確定します。
タグ値の 4 番目のビットが設定されます。
ディスクリットアラーム"Valve_inop"が表示されます。

9.10 プロジェクトの有効化(アラームロギング)

6. メッセージウィンドウのツールバーにある  ボタンをクリックし、[自動スクロール]機能を無効にします。
このようにして、メッセージを選択することができます。
7. アナログアラーム[上限値]をクリックし、アナログアラームを確認します。
メッセージステータスの表示色は変化します。



結果

"Quick_Start"プロジェクトが有効化されています。

プロジェクトの開始画像として、「Alarm_Logging.Pdl」プロセス画像が表示されます。

内部タグ"Tank_Level"および"Inflow_Valve"が、手動入力による値と共に提供されます。これらのタグは、アラームロギングによってモニタリングされます。

タグ値によって、対応するメッセージがトリガされ、アラームウィンドウに表示されます。

次のステップ

内部タグ[Tank_Level]および[Inflow_Valve]を値と共に自動的に供給するには、[タグシミュレーション]エディタを使用します。

9.11 プロジェクトのテスト(メッセージシステム)

概要

次のステップは、WinCC タグシミュレータによって「Quick_Start」プロジェクトをテストする方法を示しています。

WinCC タグシミュレータは、ランタイムの内部タグ「Tank_Level」および「Inflow_Valve」に値を割り付けます。

これらのタグに対してモニタリングが設定されているため、タグ値が定義されている制限に到達するとメッセージがトリガされます。

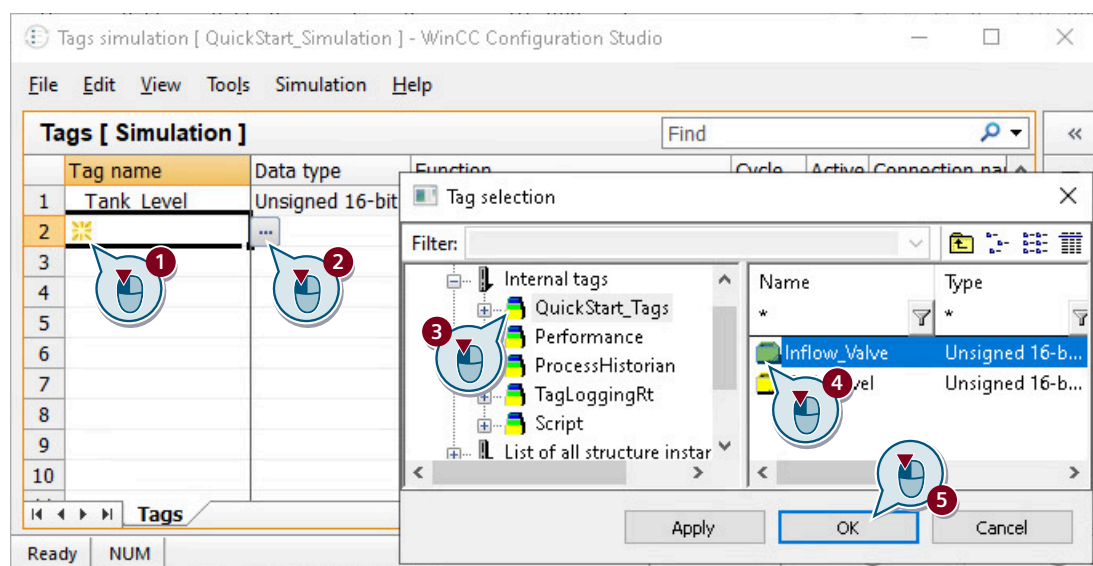
メッセージが、メッセージウィンドウに表示されます。メッセージステータスは、異なる表示色によってマークされます。

必要条件

- ランタイムプロパティが定義されている。

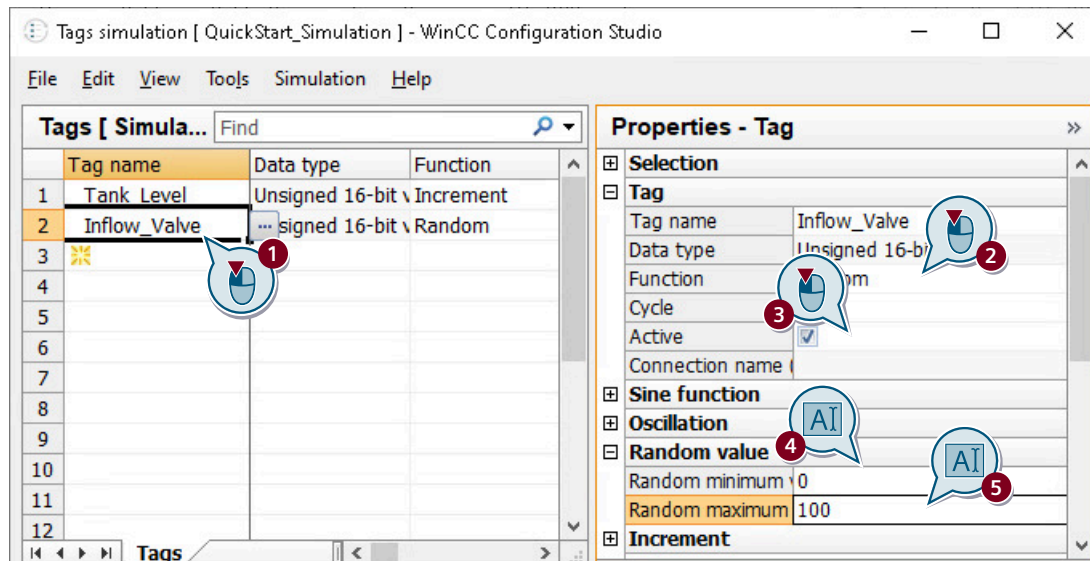
手順

- 「QuickStart_Simulation.sim」設定を[タグシミュレーション]エディタで開きます。
- タグ選択ダイアログを開き、内部タグ[Inflow_Valve]をタググループ[QuickStart_Tags]から選択します。



9.11 プロジェクトのテスト(メッセージシステム)

3. シミュレーションタイプを選択し、シミュレーションタイプのプロパティを定義します。



4. メニューバーで[ファイル] > [保存]コマンドを使用して設定を保存します。
5. ▶ ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。
6. [シミュレーション] > [起動]を使用して、[タグシミュレーション]エディタでシミュレーションを起動します。
7. 「Alarm_Logging.Pdl」プロセス画像で、シミュレーション値の出力を監視します。
シミュレーション値の変化が、メッセージウィンドウにどのように影響するかを監視します。

9.11 プロジェクトのテスト(メッセージシステム)

結果

WinCC タグシミュレータによって、"Quick_Start"プロジェクトがテストされます。

テストは、モニタリングされているタグが値と共に継続的に提供されたときの、アラームロギングの動作を示します。

WinCC OnlineTrendControl は、[Inflow_Valve]タグの 2 番目のトレンドを表示します。感嘆符をクリックすることで、ツールヒントの形式で、このタグのトリガされたメッセージを表示します。

検索、発見、フィルタリング

概要

WinCC エクスプローラには、オブジェクトやテキストを検索するための便利なオプションが用意されています。

検索と発見

1. 検索機能を開くには、WinCC エクスプローラで次の操作を選択してください
 - プロジェクトエクスプローラのメニューバーで検索アイコンを選択します

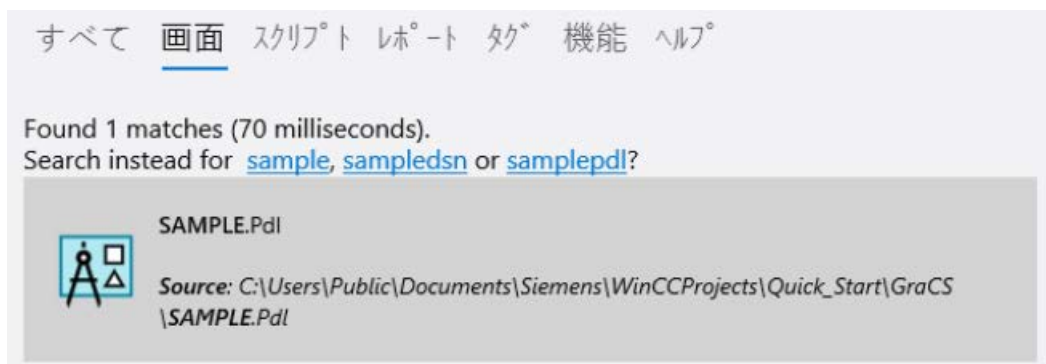


または

- スタートページで[検索]エントリを選択します。
- 検索語句を入力するテキストボックスを含むスタートページが表示されます。



2. 検索したい文字列を入力します。
ワイルドカード(* ?)はサポートされていません。
3. 検索シンボルをクリックします。
その文字列を含む結果が表示されます。
4. 表示される結果を絞り込むには、定義済みの領域のいずれかを選択します。



さらに検索するための提案も表示されます。クリックするだけで、検索用のテキストボックスに直接転送できます。

5. エディタまたは結果の情報を開くには、結果をクリックします。これにより、たとえばグラフィックデザイナー、スクリプトエディタ、情報システムなどが開きます。
6. プロジェクトに戻るには、[戻る]アイコンをクリックします。



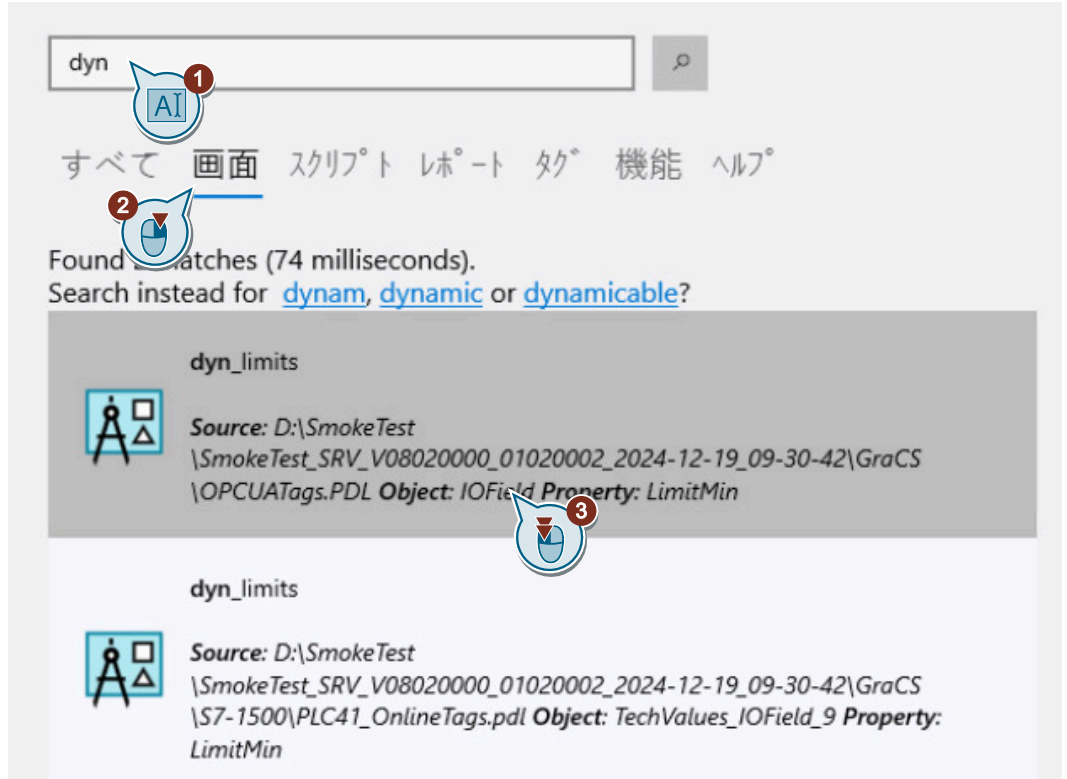
例:ダイナミック化に使用されるタグ

この例では、タグを使用してダイナミック化を行う画像を示します。

検索機能を開きます。



1. テキストボックスにタグ名を入力します。
名前の一部のみを入力すると、この文字列を含むすべてのタグの結果が表示されます。
2. [画面]エリアを選択します。
検索したタグが使われている画像がすべて表示されます。



以下の情報が得られます。

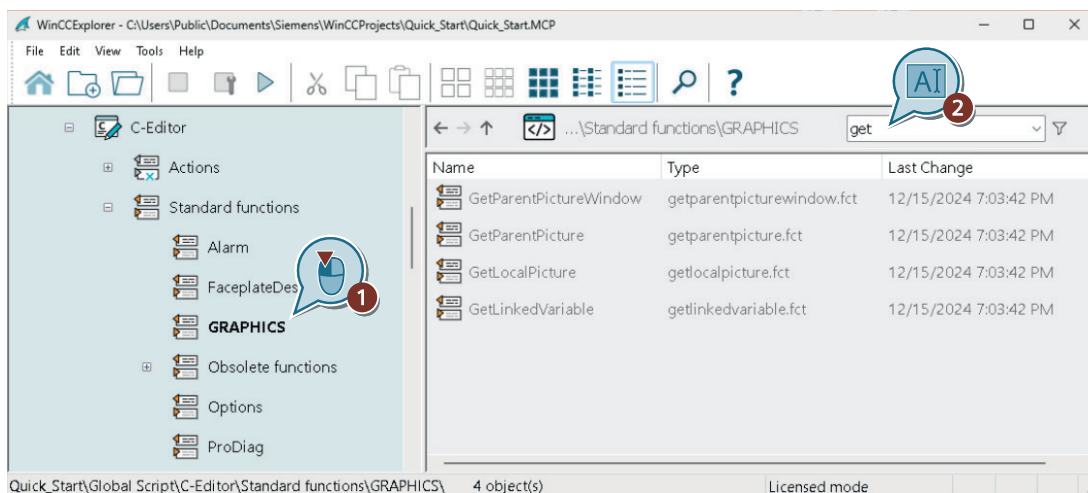
- プロジェクトの名前
 - 画面の名前
 - タグがダイナミック化に使用されるオブジェクト
 - 検索内のタグによってダイナミック化されたプロパティ。
- タグが複数のオブジェクトで使用されている場合、それに応じて複数の結果が返されます。

3. 画像をダブルクリックします。
グラフィックデザイナーで、画像が開きます。動的プロパティは、[オブジェクトプロパティ]ウィンドウで選択されます。

フィルタリングで検索

WinCC エクスプローラプロジェクトウィンドウには、画像やスクリプトなど、選択したエディタで作成され使用可能なオブジェクトが表示されます。異なるサイズのタイルやリスト表示を選択できます。

特定のオブジェクトのみを表示するには、フィルタ機能を使用します。



1. エディタを選択します。
利用可能なオブジェクトが表示されます。
2. フィルタのテキストボックスに文字列を入力します。
指定した文字列を含む名前を持つオブジェクトのみが表示されます。

用語集

[グラフィックデザイナー]エディタのライブラリ

[グラフィックデザイナー]エディタのライブラリは、グラフィックオブジェクトを保存、管理する多用途のツールです。

シンボルライブラリでは、プロセス画像の作成に使用するカスタマイズオブジェクトおよびコントロールを管理します。

SVG ライブラリでは、SVG オブジェクトを管理します。

ライブラリは、2つの領域に分割されています:

- グローバルライブラリ
- プロジェクトライブラリ

Configuration Studio

[WinCC Configuration Studio]には、[タグ管理]、[アラームロギング]および[タグロギング]エディタなどの、WinCC の設定に必要なほとんどのエディタが含まれています。

I/O フィールド

I/O フィールドは、タグ値を表示、変更するために使用する入力/出力フィールドです。

WinCC AlarmControl

[WinCC AlarmControl]オブジェクトは、メッセージイベントの表示に使用される[メッセージ]ウィンドウです。

WinCC OnlineTableControl

"WinCC OnlineTableControl"オブジェクトは、プロセス値アーカイブからのプロセス値を表示するテーブルウィンドウとして使用します。

WinCC OnlineTrendControl

"WinCC OnlineTrendControl"オブジェクトは、プロセス値アーカイブからのプロセス値を表示するトレンドウィンドウとして使用します。

WinCC Runtime

WinCC Runtime は、プロセスモードでプロジェクトを実行します。

WinCC Runtime は、プロセスのオペレータ制御および監視を有効にします。

WinCC エクスプローラ

WinCC エクスプローラは、WinCC のプロジェクト設計ソフトウェアの核となっています。

WinCC エクスプローラでは、プロジェクト構造全体が表示され、プロジェクトが管理されます。

WinCC タグシミュレータ

WinCC タグシミュレータによって、まだ開発段階であるプロジェクトをテストできます。

[タグシミュレーション]エディタを使用することで、シミュレーションを設定および有効化することができます。

WinCC のプロジェクト設計ソフトウェア

プロジェクト設計ソフトウェアは、WinCC の一部です。

プロジェクト設計ソフトウェアの中核には、WinCC エクスプローラと WinCC Configuration Studio があります。

アーカイブサイクル

アーカイブサイクルは、プロセス値がアーカイブデータベースに保存される時間間隔です。

アーカイブサイクルは、常に設定された取得サイクルの整数倍です。

アーカイブサイクルは、WinCC Runtime の起動時またはユーザーが定義した時点に開始します。開始ポイントを指示すると、値の時間差アーカイブやアーカイブ負荷の分散が可能になります。

アーカイブタグ

アーカイブする値は、アーカイブタグに保存されます。

アナログアラーム

アナログアラームは、制限値を超過しているか、達していないことを示します。

アナログアラームは、定義されている制限値を超過しているか、達していないときに、トリガされます。

アニメーション

アニメーションでは、複数のオブジェクトプロパティのダイナミック化をタグまたはスクリプトと結合します。

これにより、複数のプロパティ変更をタグ値または値範囲に割り付けて、オブジェクトのサイズを拡大したり、同時に点滅させたりすることができます。

アラームロギング

[アラームロギング]エディタは、アラームロギングの設定コンポーネントです。

このエディタは、メッセージを設定するために使用します。

アラームロギングランタイム

アラームロギングランタイムは、アラームロギングのランタイムコンポーネントです。アラームロギングランタイムは、主に次のタスクを実行するために使用されます:

- 定義済みモニタリングの実行
- メッセージ出力の制御
- 承認の管理

グラフィックデザイナー

[グラフィックデザイナー]エディタは、グラフィックシステムの設定コンポーネントです。

このエディタは、プロセス画像を設定するために使用します。

グラフィックランタイム

グラフィックランタイムは、グラフィックシステムのランタイムコンポーネントです。

グラフィックランタイムは、ランタイム中に画像を表示し、プロジェクトの起動時にすべての入力と出力を管理します。

システムブロック

システムブロックは、メッセージブロックに属します。

システムブロックは、システムデータ(日付、時刻、メッセージ番号、ステータスなど)を含みます。

システムブロックは、事前定義されています。

シングルユーザープロジェクト

「シングルユーザープロジェクト」は、1台のPCでのみ実行されます。他のコンピュータはこのプロジェクトにアクセスできません。

プロジェクトは、データを処理するサーバー、およびオペレータステーションとして動作する1台のコンピュータ上で実行されます。

スタティックオブジェクト

スタティックオブジェクトは、ランタイム中に変化しません。

スタティックオブジェクトには、長方形、円、線、あるいはコネクタなどがあります。

スタティックテキスト

"スタティックテキスト"オブジェクトは、表示したプロセスまたは内容をマークするために使用するスタティックオブジェクトです。

スライダオブジェクト

スライダオブジェクトは、タグ値の表示や変更のために使用します。

スライダオブジェクトは、タグへの接続によって動的に作成されます。スライダオブジェクトのプロセスタグへの接続によって、ASを制御できます。

ダイナミックオブジェクト

ダイナミックオブジェクトは、個々のプロセス値に従って変更されます。

ダイナミックオブジェクトの例としては、バーがあります。バーの長さは、現在の温度値に応じて変化します。

タググループ

タググループは、タグ管理のコンポーネントです。

タググループは、タグを明確な構造に整理するために使用します。

タグロギング

[タグロギング]エディタは、アーカイブシステムの設定コンポーネントです。

このエディタは、アーカイブを設定するために使用します。

タグロギングランタイム

タグロギングランタイムは、アーカイブシステムのランタイムコンポーネントです。タグロギングランタイムは、主に次のタスクを実行するために使用されます:

- プロセス値のプロセス値アーカイブへの書き込み
- プロセス値アーカイブからの、アーカイブ済みプロセス値の読み取り

タグ管理

"タグ管理"コンポーネントの下で、WinCC と AS の間の通信を設定します。

"タグ管理"で、全てのタグやチャンネルを作成、管理します。

チャンネル

チャンネルは、特殊な通信ドライバです。

チャンネルによって、WinCC とオートメーションシステム(AS)の間の通信が可能になります。

チャンネル経由で、WinCC のプロセスタグに AS からのプロセス値が提供されます。

チャンネルユニット

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イスとして動作します。

さらに、このチャンネルユニットは、特定のタイプの AS にアクセスするために使用されます。

ディスクリートアラーム

ディスクリートアラームはプロセスのステータスの変化を示し、PLC によって起動されます。

フェイスプレートタイプ/フェイスプレートインスタンス

フェイスプレートタイプは、再利用可能なテンプレートを作成し、プロセス画像でフェイスプレートインスタンスとして使用する特別なプロセス画像です。

インターフェースタグまたは構造タグで、フェイスプレートタイプのオブジェクトをダイナミック化します。

フェイスプレートタイプでの変更が、すべてのフェイスプレートインスタンスに自動的に適用されます。

フェイスプレートタイプは[グラフィックデザイナー]エディタで設定されます。

プロジェクト

プロジェクトは、WinCC のユーザーインターフェースの設定の基本です。

プロジェクト内で、プロセスを操作、監視する必要がある全てのオブジェクトを作成、編集します。

プロセスタグ

プロセスタグは、WinCC と AS 間のデータ交換用リンクを形成します。

WinCC の各プロセスタグは、接続された AS のいずれかのメモリ中の特定のプロセス値に対応します。

プロセス画像

プロセス画像はプロジェクトの中心的要素です。

プロセス画像はプロセスを表現し、このプロセスに関する操作や監視を可能にします。

[グラフィックデザイナー]エディタを使用して、プロセス画像を設定できます。

プロセス値

プロセス値は、AS によって提供される値です。

プロセス値アーカイブ

プロセス値アーカイブは、AS からのプロセス値を保存するアーカイブです。

プロセス値ブロック

メッセージをプロセス値(例: 現在のフィルレベル、温度、速度)に接続するために使用するプロセス値ブロック。

ページレイアウト

ページレイアウトでは、外観やレポート出力のためのデータ提供を設定します。

ボタン

ボタンは、プロセス操作を有効にします。

ボタンを使って、メッセージの確認やランタイムのナビゲーションができます。

メインメモリ

コンピュータのメインメモリは、データオブジェクトを保存し、後でそこから取り出すメモリです。

メインメモリは、作業メモリとも呼ばれます。

メッセージクラス

メッセージクラスには、類似の動作を行なうメッセージが含まれます。

メッセージクラスによって、個別のメッセージを一元管理できます。

メッセージタグ

メッセージタグは、プロセスにおけるステータスの変化にリンクしています。

プロセスでステータスの変化が発生すると、タグ値のビットが設定されます。タグ値に応じて、ディスクリートアラームがトリガされます。

メッセージの表示色

表示色は、メッセージの現在のステータスを示します。

個別のメッセージの表示色は、"アラームロギング"エディタで決定します。

メッセージビット

[メッセージビット]プロパティによって、ディスクリートアラームをいつトリガするかを定義できます。

メッセージブロック

メッセージの内容はメッセージブロックで構成されています。

各メッセージブロックは、WinCC アラームコントロールの表形式の表示の中の 1 列に対応します。

メッセージレポート

[メッセージ]ウィンドウの現在のメッセージリストにある全てのメッセージは、メッセージレポートで文書化されます。

ユーザーテキストブロック

ユーザーテキストブロックは、メッセージブロックに属します。

ユーザーテキストブロックには、説明テキスト(例:ロケーションおよび障害の原因に関する情報)が含まれています。

ユーザーテキストブロック内のテキストは自由にカスタマイズできます。

ユーザーテキストブロックの長さが、このブロックに入力できる文字の数を決定します。文字列の最大長は 255 文字です。

ランタイム

このオプションが有効な場合は、プロジェクトはランタイム中です。

リニアスケールリング

リニアスケールリングの使用時には、プロセスタグの値の範囲を WinCC のプロセスタグの値の特定の範囲にマップできます。

プロセス値そのものは修正されません。

レポートデザイナー

[グラフィックデザイナー]エディタは、レポートシステムの設定コンポーネントです。

このエディタを使用して、ページや行のレイアウトの設定、さらに印刷ジョブの設定を行います。

レポートランタイム

レポートランタイムは、レポートシステムのランタイムコンポーネントです。

レポートランタイムは、アーカイブまたはコントロールから文書化するデータを取得し、印刷出力をコントロールします。

印刷ジョブ

印刷ジョブは、プロジェクト文書やランタイム文書を出力します。

印刷ジョブでは、出力媒体、印刷する範囲、印刷の開始時間、その他の出力パラメータを設定します。

開始画像

開始画像は、[ランタイム]ウィンドウでプロジェクトが起動したときに表示されるプロセス画像です。

更新サイクル

更新サイクルは、ランタイムの表示を更新する時間間隔を決定します。

構造タグ/構造タイプ

構造タイプは、タグの特定のコンパイルーションを作成するために、WinCC で使用されるスキームを示します。

構造タグは、構造タイプにリンクされたタグです。

構造タイプを使用して、複数のタグを構造タグ要素として、1つのステップで同時に作成することができます。この操作で、内部タグとプロセスタグを作成できます。

コントロールシステムから直接タグ構造を読み込み、WinCC タグ管理にインポートすることもできます。

取得サイクル

取得サイクルは、プロセスタグのプロセス値を読み出す間隔を決定します。

取得サイクルは、WinCC Runtime が起動するとすぐに開始します。

取得サイクルおよびアーカイブサイクルの時間

取得サイクルおよびアーカイブサイクルの時間は、2つのアーカイブの間の時間間隔を決定します。

制限値モニタリング

制限値モニタリングは、WinCC のアドインです。

制限値モニタを用いると、タグに任意数の制限値を設定できます。これらの制限値のいずれかに違反すると、対応するメッセージがトリガされ、ランタイム中に表示されます。

接続

接続は、定義された個々の AS へのインターフェースを説明します。

データ交換は接続を介してランタイムで行なわれます。

接続は、チャンネルユニットの下で設定されます。

操作可能なオブジェクト

コントロール可能な画面エレメントを使用することで、オペレータはプロセスに対して能動的な影響を与えることができます。

そのようなオブジェクトとしては、ボタン、スライダオブジェクト、あるいは特定のプロセスパラメータへの入力に使用する I/O フィールド(入力/出力フィールド)などがあります。

直接タグ接続

直接タグ接続では、プロセス画像のダイナミックオブジェクトと1つのタグを接続します。

ランタイムでタグが値を取ると、この値はダイナミックオブジェクトに直接転送されます。オブジェクトの動的表示は、タグ値に従ってランタイムで変化します。

内部タグ

内部タグはプロセスリンクを持たず、WinCC 内部の値を転送するだけです。

索引

[

- [テーブル]ウィンドウ, 164
 - WinCC OnlineTableControl, 164
 - 設定, 164
- [トレンド]ウィンドウ, 156
 - WinCC OnlineTrendControl, 156
 - 設定, 156

A

- Alarm_Logging.Pdl, 202

I

- I/O フィールド, 66, 71, 202
 - ダイナミック化, 71, 215
 - フォントプロパティの定義, 215
 - 更新の定義, 71
 - 出力フォーマットの定義, 71, 215
 - 挿入, 71, 215
 - 入力制限, 71

M

- MPI, 26, 31

Q

- Quick_Start
 - シングルユーザープロジェクト, 18

S

- S7-1200、S7-1500 チャンネル, 26
- SAMPLE.Pdl, 51, 64
 - 作成, 51
 - 編集, 64
- START.Pdl, 51
 - 作成, 51
 - 編集, 64
- SVG オブジェクト
 - ダイナミック化, 67
- SVG ライブラリ, 56

T

- Tag_Logging.pdl, 156

W

- WinCC, 13
 - サブシステム, 13
 - プロジェクト設計ソフトウェア, 13
 - ランタイムソフトウェア, 13
- WinCC AlarmControl, 202
- WinCC OnlineTableControl, 156, 164
 - 挿入, 164
- WinCC OnlineTrendControl, 156
 - メッセージの表示, 194
 - 挿入, 156
- WinCC Runtime, 13
 - プロパティの指定, 87, 173, 222
 - 開始画像, 173
- WinCC エクスプローラ, 13
 - フィルタ, 235
- WinCC タグシミュレータ, 93, 137, 176, 229
 - シミュレーションタイプの定義, 229
 - 開始, 93, 137, 176, 229
- Windows オブジェクト, 55

ア

- アーカイブ, 143, 150
 - プロセス値アーカイブ, 143, 150
 - 値, 143
- アーカイブサイクル, 148
 - 時間の割り付け, 153
 - 時間の設定, 148
- アーカイブシステム, 144
 - タグロギング, 144
 - タグロギングランタイム, 144
- アーカイブタグ, 150
 - 作成, 150
 - 名前の変更, 153
- アイコン, 11
- アドレス指定, 35
 - プロセスタグ, 35
- アナログアラーム, 179
 - ランタイムでの表示, 194
 - 制限値の設定, 191
 - 設定, 191

アラームウィンドウ, 202
 WinCC AlarmControl, 202
 設定, 202
アラームロギング, 179, 180
 開始, 182
アラームロギングランタイム, 180
 有効, 222

ウ

ウィンドウ属性, 87
 有効, 87

エ

エディタ, 13, 24, 49
 アラームロギング, 13, 182
 グラフィックデザイナー, 49, 55
 タグロギング, 144, 146
 タグ管理, 24
 レポートデザイナー, 13

オ

オブジェクト, 47, 49, 56
 I/O フィールド, 202, 215
 WinCC AlarmControl, 202
 WinCC OnlineTableControl, 156, 164
 WinCC OnlineTrendControl, 156
 Windows オブジェクト, 55
 スケール, 211
 スタティックオブジェクト, 47
 スタティックテキスト, 61
 スマートオブジェクト, 55
 スライダオブジェクト, 202, 207
 ダイナミックオブジェクト, 47
 挿入, 55
 操作可能なオブジェクト, 47
 標準オブジェクト, 55
オブジェクトパレット, 55
 Windows オブジェクト, 55
 コントロール, 55
 スマートオブジェクト, 55
 標準オブジェクト, 55
オブジェクトプロパティ, 67, 71, 207, 211
 アニメーション化, 133
 ダイナミック化, 116, 133

ク

グラフィックシステム, 49
 グラフィックデザイナー, 49
 グラフィックランタイム, 49
グラフィックデザイナー, 49, 55
 オブジェクトパレット, 47
 プロセス画像の設定, 47
 ライブラリ, 55
グラフィックランタイム, 49
 有効, 87

コ

コントロール, 55
 WinCC AlarmControl, 202
 WinCC OnlineTableControl, 156, 164
 WinCC OnlineTrendControl, 156
コンポーネント, 13
 WinCC, 13
 アーカイブシステム, 144
 グラフィックシステム, 49
 メッセージシステム, 180

シ

システムダイアログ
 使用, 97
システムブロック, 183
シミュレーション, 93, 137
 WinCC タグシミュレータ, 93, 137, 176, 229
 シミュレーションタイプ, 93, 137, 176, 229
 開始, 93, 137, 229
 終了, 229
シミュレーションタイプ, 229

ス

スクリプト
 検索, 233
スケール, 41, 211
 サイズの定義, 211
 フォントサイズの定義, 211
 プロセスタグ, 41
 挿入, 211
スタティックテキスト, 61
 フォントサイズを変更する, 61
 挿入, 61

ステータス, 187

バルブ, 179

メッセージ, 200

スマートオブジェクト, 55

スライダオブジェクト, 202

サイズの定義, 207

ダイナミック化, 207

色の定義, 207

挿入, 207

名前の定義, 207

タ

ダイナミックオブジェクト, 47

ダイナミック化, 66, 67

I/O フィールド, 71, 215

スライダオブジェクト, 207

タイプ変換, 35

プロセスタグ, 35

タイマ, 148

作成, 148

設定, 148

タグ, 24

アーカイブタグ, 150

スケーリング, 41

プロセスタグ, 24, 35, 41

リンク, 67

検索, 233

作成, 35, 43

内部タグ, 24, 43

タググループ, 24

作成, 35

タグロギング, 144

開始, 146

タグロギングランタイム, 144

有効, 173

タグ管理, 24

タググループ, 35

プロセスタグ, 24, 35

内部タグ, 24, 43

タグ接続, 66

直接タグ接続, 66

チ

チャンネル, 23

SIMATIC S7-1200、S7-1500 チャンネル, 26

挿入, 26

チャンネルユニット, 23, 31

MPI, 26

挿入, 26

テ

ディスクリートアラーム, 186

作成, 187

データタイプ, 35, 43

プロセスタグ, 35

符号なし 16 ビット値, 43

符号付き 16 ビット値, 35

データ交換, 24

プロセスタグ, 24

テキストフィールド, 61

テスト, 93, 137

プロジェクト, 93, 137, 176, 229

フ

フィルタ, 235

フェイスプレートインスタンス

作成, 129

使用, 129

フェイスプレートタイプ

オブジェクトの挿入, 103

ポップアップ画面, 112, 113, 116, 127

作成, 103, 120

設定, 103, 120, 122, 127, 133

変更, 133

フォントサイズ, 61

変更, 61, 71, 211

プロジェクト, 17, 90, 93, 137

WinCC の起動, 18

シングルユーザープロジェクト, 18

テスト, 93, 137, 176, 229

作成, 18

有効, 90, 176, 225

プロジェクト設計ソフトウェア, 13

WinCC, 13

プロセス, 13

モニタリング, 13, 179

視覚化, 13, 47, 49

操作, 13, 21, 23, 24

プロセスタグ, 24

アドレス指定, 35

スケーリング, 41

タイプ変換, 35

データタイプ, 35

作成, 35

プロセス画像, 47

Alarm_Logging.Pdl, 202

SAMPLE.Pdl, 51

START.Pdl, 51

Tag_Logging.Pdl, 156
ダイナミック化, 66
作成, 51
設定, 47, 156, 202
編集, 55, 64
保存, 61, 64
名前の変更, 51
プロセス値, 21, 24, 143
アーカイブ, 143, 148
ウィンドウ, 143
シミュレーション, 93, 137, 176, 229
入力, 148
プロセス値アーカイブ, 143
サイズの定義, 153
メモリの場所の定義, 153
作成, 150
編集, 153
保存, 153
プロセス値ブロック, 183

ホ

ポップアップ画面
フェイスプレートタイプでの使用, 112, 113, 116, 127
作成, 112, 113, 116

マ

マニュアル類, 143

メ

メッセージ, 179
アナログアラーム, 179
ディスクリートアラーム, 179
メッセージステータス, 200
メッセージステータスの色の定義, 200
確認, 225
制限値モニタリング, 191
設定, 179
保存, 191
メッセージクラス, 183
定義, 183
メッセージシステム, 180
アラームロギング, 180
アラームロギングランタイム, 180
メッセージステータス, 187, 200
メッセージブロック, 183
システムブロック, 183
プロセス値ブロック, 183

ユーザーテキストブロック, 183
定義, 183
表示の有効化, 202
メッセージリスト, 225

ユ

ユーザーテキストブロック, 183

ラ

ライブラリ, 55, 56
グローバルライブラリ, 56
プロジェクトライブラリ, 56
開く, 56
ランタイム, 13
ランタイムウィンドウ, 87
ウィンドウ属性, 87
開始画像, 87, 173, 222
ランタイムコンポーネント, 13
アーカイブシステム, 144
グラフィックシステム, 49
メッセージシステム, 180
ランタイムシステムダイアログ, 97
ランタイムソフトウェア, 13
WinCC, 13

リ

リニアスケーリング, 41
例, 41

レ

レポート
検索, 233

画

画像
検索, 233

開

開く, 18, 51
プロジェクト, 18
プロセス画像, 51, 64
ライブラリ, 56

開始, 18

WinCC, 18

WinCC タグシミュレータ, 176

アラームログイン, 182

タグログイン, 146

開始画像, 87, 173

定義, 87, 222

確

確認, 225

メッセージ, 225

機

機能

検索, 233

更

更新サイクル

I/O フィールド, 71

構

構造タイプ

データタイプ, 106, 109

リンク, 122

作成, 106

構造タグ, 106, 109

作成, 109

構造タグ要素, 109

構造要素, 106

作

作成, 31, 35, 43, 156

[テーブル]ウィンドウ, 164

[トレンド]ウィンドウ, 156

アナログアラーム, 191

アラームウィンドウ, 202

タイマ, 148

タググループ, 35

ディスクリートアラーム, 187

プロジェクト, 17, 18

プロセスタグ, 35

プロセス画像, 51

プロセス値アーカイブ, 150

接続, 31

内部タグ, 43

実

実例, 56

スケール, 211

取

取得サイクル, 148

時間の割り付け, 153

時間の設定, 148

色

色, 200

メッセージステータスの色の定義, 200

制

制限値, 191

下限値, 191

上限値, 191

定義, 191

制限値モニタリング, 191

ランタイムでのメッセージの表示, 194

制限値の設定, 191

接

接続, 23, 31

作成, 31

設

設定, 21, 55, 156

[テーブル]ウィンドウ, 164

[トレンド]ウィンドウ, 156

アラームウィンドウ, 202

プロセス画像, 55, 156, 202

プロセス値アーカイブ, 150, 153

メッセージ, 179, 186, 191

取得サイクルおよびアーカイブサイクルの時間, 148

通信, 21

設定コンポーネント, 49, 144, 180

アーカイブシステム, 144

グラフィックシステム, 49

メッセージシステム, 180

値

- 値, 21
 - アーカイブ, 143
 - シミュレーション, 93, 137
 - プロセス値, 21, 24, 143
- 値の範囲, 41
 - プロセスタグのスケーリング, 41

通

- 通信, 21
 - チャンネル, 23, 26
 - チャンネルユニット, 23, 26
 - プロセスタグ, 24, 35
 - 接続, 23, 31
 - 設定, 21
- 通信ドライバ, 23
 - SIMATIC S7-1200、S7-1500 チャンネル, 26

貼

- 貼り付け, 26
 - I/O フィールド, 71, 215
 - SIMATIC S7-1200、S7-1500 チャンネル, 26
 - WinCC AlarmControl, 202
 - WinCC OnlineTableControl, 164
 - WinCC OnlineTrendControl, 156
 - スケール, 211
 - スタティックテキスト, 61
 - スライダオブジェクト, 207
 - チャンネル, 26
 - チャンネルユニット, 26
 - ライブラリからのオブジェクト, 56

内

- 内部タグ, 24
 - データタイプ, 43
 - 構造タイプ, 106, 109
 - 作成, 43

発

- 発見, 233

標

- 標準オブジェクト, 55
 - スタティックテキスト, 61

表

- 表示色, 200
 - メッセージステータスの色の定義, 200
- 表示要素, 71, 143, 207
 - 値, 71, 143, 207

符

- 符号なし 16 ビット値, 43
- 符号付き 16 ビット値, 35

編

- 編集, 55
 - スタティックテキスト, 61
 - プロセス画像, 55, 64
 - プロセス値アーカイブ, 153

保

- 保持, 176
- 保存, 64
 - プロセス画像, 61, 64
 - プロセス値アーカイブ, 153
 - メッセージ, 191

名

- 名前の変更, 51, 153
 - アーカイブタグ, 153
 - プロセス画像, 51

有

- 有効, 90
 - アラームロギングランタイム, 222
 - ウィンドウ属性, 87
 - グラフィックランタイム, 87
 - タグロギングランタイム, 173
 - プロジェクト, 90, 176, 225
 - リニアスケーリング, 41