

SIMATIC HMI

WinCC V7.5 SIMATIC HMI WinCC V7.5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

値のアーカイブと表示

7

プロセスアーカイブからの
値の出力

8

メッセージの設定

9

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

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

警告

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

商標

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

免責事項

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

目次

1	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	プロジェクトの作成	17
4.2	"Quick_Start"プロジェクトの作成	18
5	通信の設定.....	21
5.1	通信の設定	21
5.2	WinCCチャンネルとの接続確認.....	23
5.3	WinCCのタグ	25
5.4	チャンネルの追加	26
5.5	接続の作成	30
5.6	タググループのプロセスタグの作成	34
5.7	WinCCでのプロセスタグのスケーリング	40
5.8	内部タグの作成.....	42
6	プロセス画面の設定.....	47
6.1	プロセス画面の設定	47
6.2	グラフィックシステム	48
6.3	プロセス画面の作成	50
6.4	プロセス画面の編集	53
6.4.1	プロセス画面の編集	53
6.4.2	ライブラリからのグラフィックオブジェクトの挿入	55
6.4.3	[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	62
6.4.4	プロセス画面"SAMPLE.pdl"の編集	66
6.5	画面ダイナミックスの処理.....	68
6.5.1	画面ダイナミックスの処理.....	68
6.5.2	フィルレベルインジケータのダイナミック化	70
6.5.3	I/Oフィールドの挿入とダイナミック化(グラフィックシステム)	74
6.6	ユーザー定義メニューおよびツールバーの作成.....	79
6.6.1	カスタマイズされたメニューおよびツールバーの使用	79
6.6.2	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	80

6.6.3	画面変更のためのカスタマイズされたメニューの作成.....	84
6.6.4	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.....	91
6.7	ランタイムプロパティの定義(グラフィックシステム).....	96
6.8	プロジェクトの有効化(グラフィックシステム).....	99
6.9	プロジェクトのテスト(グラフィックシステム).....	102
6.10	ランタイムシステムダイアログの使用	108
7	値のアーカイブと表示.....	115
7.1	値のアーカイブと表示.....	115
7.2	アーカイブシステム	116
7.3	タグロギングの開始	118
7.4	タイマの設定	120
7.5	プロセス値アーカイブの作成.....	122
7.6	プロセス値アーカイブの編集.....	125
7.7	プロセス画面の設定	128
7.7.1	プロセス画像の設定(アーカイブシステム).....	128
7.7.2	[トレンド]ウィンドウの設定	129
7.7.3	[テーブル]ウィンドウの設定	137
7.8	ユーザー定義メニューのカスタマイズ(アーカイブシステム).....	142
7.9	ランタイムプロパティの定義(アーカイブシステム).....	145
7.10	プロジェクトの有効化とテスト(アーカイブシステム).....	147
8	プロセスアーカイブからの値の出力.....	151
8.1	プロセスアーカイブからの値の出力.....	151
8.2	レポートシステム	152
8.3	ページレイアウトの作成	155
8.4	ページレイアウトの編集	158
8.4.1	ページレイアウトの編集	158
8.4.2	ページレイアウトのプロパティの確立	161
8.4.3	ログの内容の決定	165
8.4.4	ヘッダーの編集.....	170
8.4.5	フッターの編集.....	174
8.5	印刷ジョブの編集	177
8.5.1	印刷ジョブの編集	177
8.5.2	印刷ジョブの定義と編集	178
8.6	ランタイムプロパティの定義(レポートシステム).....	182

8.7	プロジェクトの有効化とテスト(レポートシステム).....	185
8.8	ログの印刷	188
9	メッセージの設定	191
9.1	メッセージの設定	191
9.2	メッセージシステム	192
9.3	アラームロギングの開始	194
9.4	メッセージブロックの設定.....	196
9.5	ビットメッセージの設定	199
9.5.1	ビットメッセージの設定	199
9.5.2	ビットメッセージの作成	200
9.6	アナログメッセージの設定.....	204
9.6.1	アナログメッセージの設定.....	204
9.6.2	限界値の設定	205
9.6.3	トレンドウィンドウでのメッセージの表示	209
9.7	メッセージステータスの色の定義.....	214
9.8	プロセス画面の設定	217
9.8.1	プロセス画像の設定(アラームロギング).....	217
9.8.2	アラームメッセージウィンドウの設定	218
9.8.3	スライダオブジェクトの挿入とダイナミック化.....	223
9.8.4	スケールの挿入.....	226
9.8.5	I/Oフィールドの挿入とダイナミック化(アラームロギング)	230
9.9	ユーザー定義メニューのカスタマイズ(アラームロギング).....	235
9.10	ランタイムプロパティの定義(アラームロギング)	238
9.11	プロジェクトの有効化(アラームロギング)	241
9.12	プロジェクトのテスト(アラームロギング)	245
	用語集.....	251
	索引	263

WinCC「スタートガイド」へようこそ

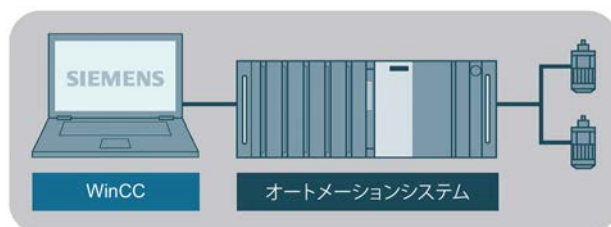
「スタートガイド」は、WinCCバージョン7に関する概要ドキュメントです。

SIMATIC HMI WinCC V7

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によって、マシンで実行されるプロセスを操作、監視できます。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

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
- 操作対象装置のシステムの動作ステータスを示すメッセージ

「スタートガイド」は以下のセクションで構成されています。

- プロジェクトを作成
- 通信の設定
- プロセス画像の設定
- 値のアーカイブと表示
- プロセスアーカイブからの値の出力
- メッセージの設定

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。

これには、設定ステップの実行に必要なすべてのプログラムも含まれています。

WinCC V7の技術情報およびトレーニング

WinCC V7に関する詳細情報は、インターネットで参照できます。

- オンラインサポートのSIMATIC WinCC V7
(<https://support.industry.siemens.com/cs/ww/en/view/93906404>):
"WinCC V7"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

メッセージの設定 (ページ 191)

プロセスアーカイブからの値の出力 (ページ 151)

値のアーカイブと表示 (ページ 115)

プロセス画面の設定 (ページ 47)

通信の設定 (ページ 21)

プロジェクトの作成 (ページ 17)

アイコン

はじめに

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

異なる指示ステップを図示できるように、本ドキュメントの作成時に以下のアイコンを使用しました。

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ながらドラッグします(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

はじめに

WinCCはモジュラシステムです:

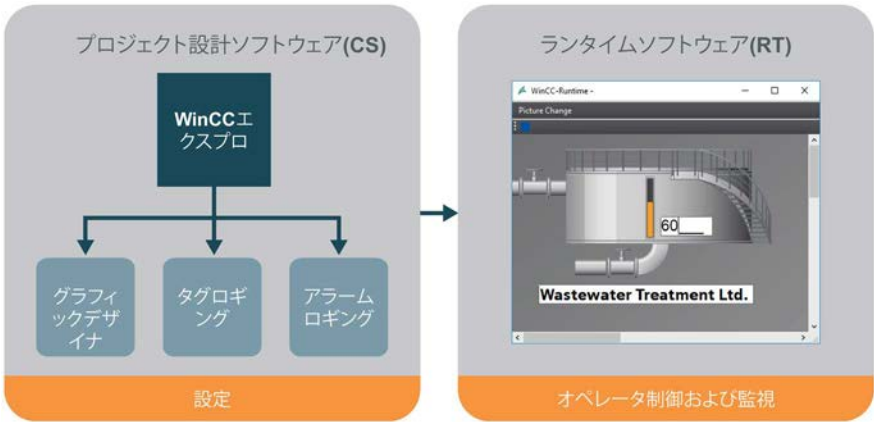
- WinCCを使用して、プロセスを視覚化し、グラフィカルユーザーインターフェースを設定します。
- プロセスのオペレータ制御および監視用ユーザーインターフェースを使用します。

WinCCの機能

- WinCCを使用すると、プロセスを視覚化できます。
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を使用すると、プロセスを制御できます。
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を使用すると、プロセスを監視できます。
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を使用すると、プロセスをアーカイブできます。
WinCCを使用することで、メッセージおよびプロセス値を印刷したり、電子的にアーカイブしたりすることができます。
結果として、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エクスプローラは設定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の特定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。

特に、WinCC Runtimeには、次のようなタスクがあります。

- 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
- 画面への画像の表示
- オートメーションシステムとの通信
- 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

4.1 プロジェクトの作成

はじめに

この章では、WinCCに関する情報とWinCC Explorerで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。

一般的手順

プロジェクトはWinCC Explorerで作成します。WinCC ExplorerはWinCCの設定コンポーネントです。このコンポーネントを使用してプロジェクトを管理します。

4.2 "Quick_Start"プロジェクトの作成

はじめに

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 V7が最低限の"標準"バージョンとして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。



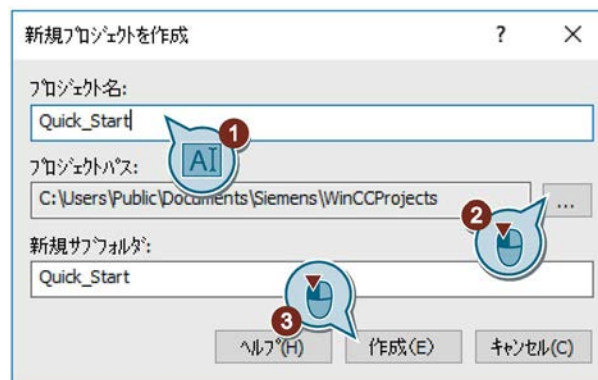
2. [WinCCexplorer]ダイアログボックスから[シングルユーザープロジェクト]プロジェクトタイプを選択します。

ダイアログボックスが自動的に開かなかった場合、WinCCexplorerのメニューバーで[ファイル] > [新規]を選択します。



[新規プロジェクトの作成]ダイアログボックスが開きます。

3. プロジェクト情報を入力します。



[新規サブフォルダ]や[プロジェクトパス]のフィールドで何も変更しなければ、標準設定が適用されます。

4.2 "Quick_Start"プロジェクトの作成

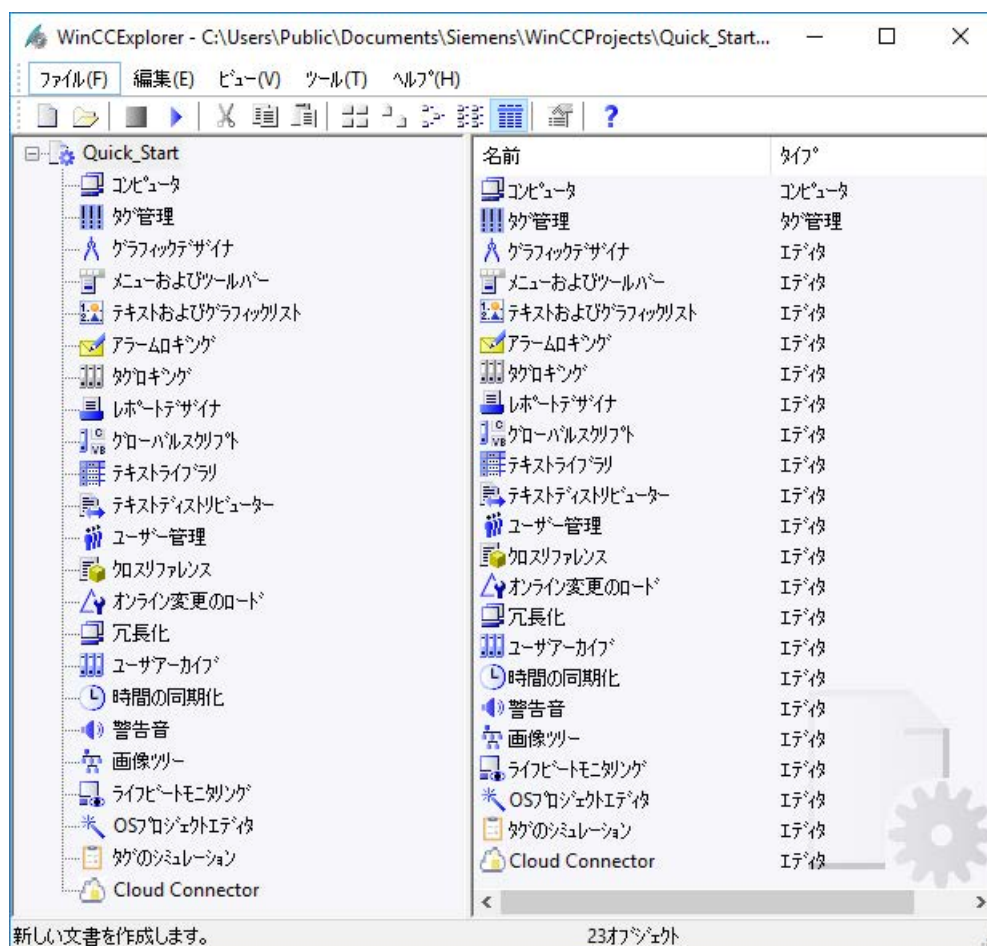
結果

"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。

必要なエディタやフォルダと共に、プロジェクトの構造がWinCCエクスプローラの左ペインに表示されます。

右部分ペインには、エディタやフォルダに属する項目が表示されます。



通信の設定

5.1 通信の設定

はじめに

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

また、このセクションでは、タグの作成方法とスケール方法についても説明します。

原理

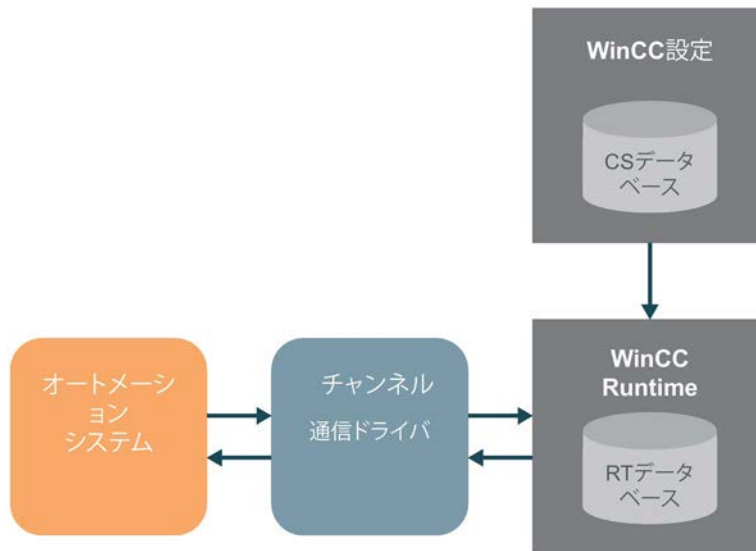
[WinCC Configuration Studio]エディタを使用して、通信を設定します。

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



WinCCプロジェクト"Quick_Start"

"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、リニアスケーリングと同様に、実践の練習だけを意図しています。

5.2 WinCCチャンネルとの接続確認

はじめに

チャンネルと接続によって、オートメーションシステムをWinCCに接続する方法を明確に指定します。

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、オートメーションシステムからプロセスタグにプロセス値を提供できます。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。

WinCCでは、WinCCからASへ値を転送するチャンネルも使用できます。これらのチャンネルによって、プロセスを制御します。

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。

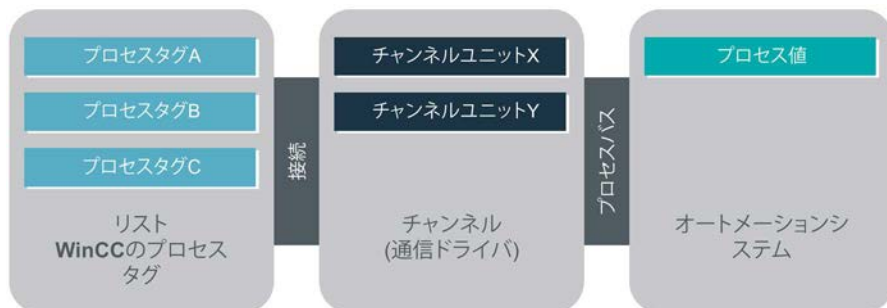
さらに、このチャンネルユニットは、特定のタイプのASにアクセスするために使用されます。

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イ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。



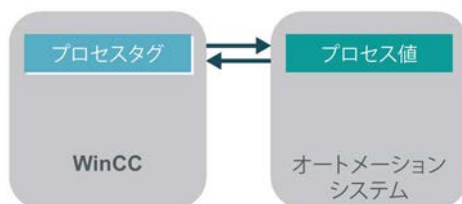
5.3 WinCCのタグ

はじめに

WinCCのタグは、プロセスの値または内部値を表します。
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。
WinCCは、すべての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

WinCCとASの間でのデータ交換のための接続リンクが、外部タグです。
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には内部タグもあります。
これらのタグはプロセスリンクを持たず、WinCC内で値を転送するだけです。

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。

5.4 チャンネルの追加

はじめに

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。

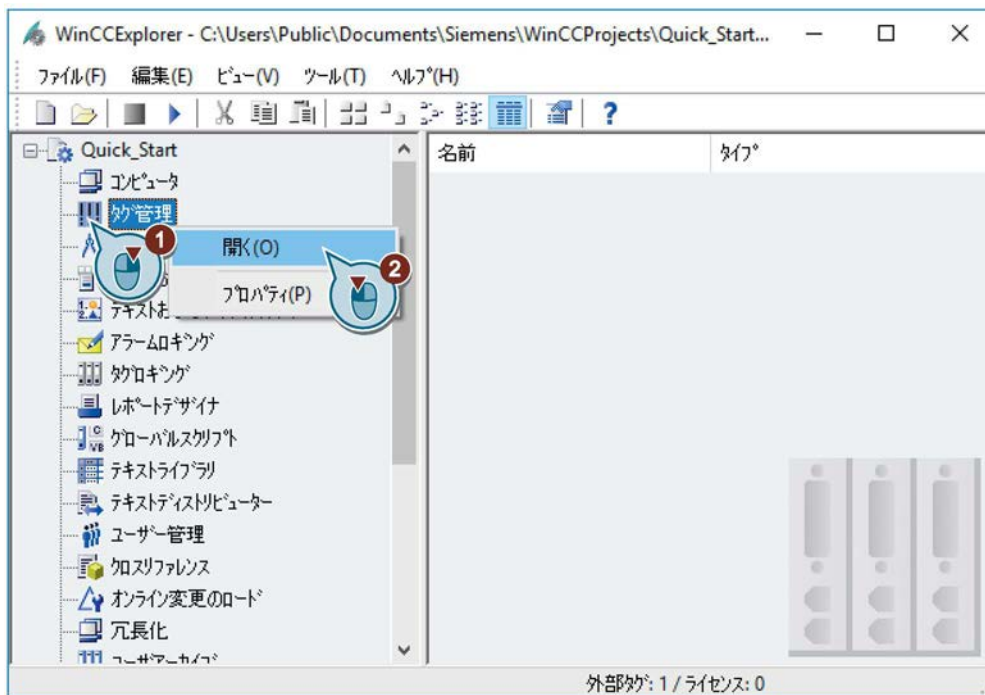
WinCCは、このチャンネルによってオートメーションシステムと通信します。
これは、プロセス値をWinCCのプロセスタグにも提供します。

必要条件

- "Quick_Start"プロジェクトが開いている。

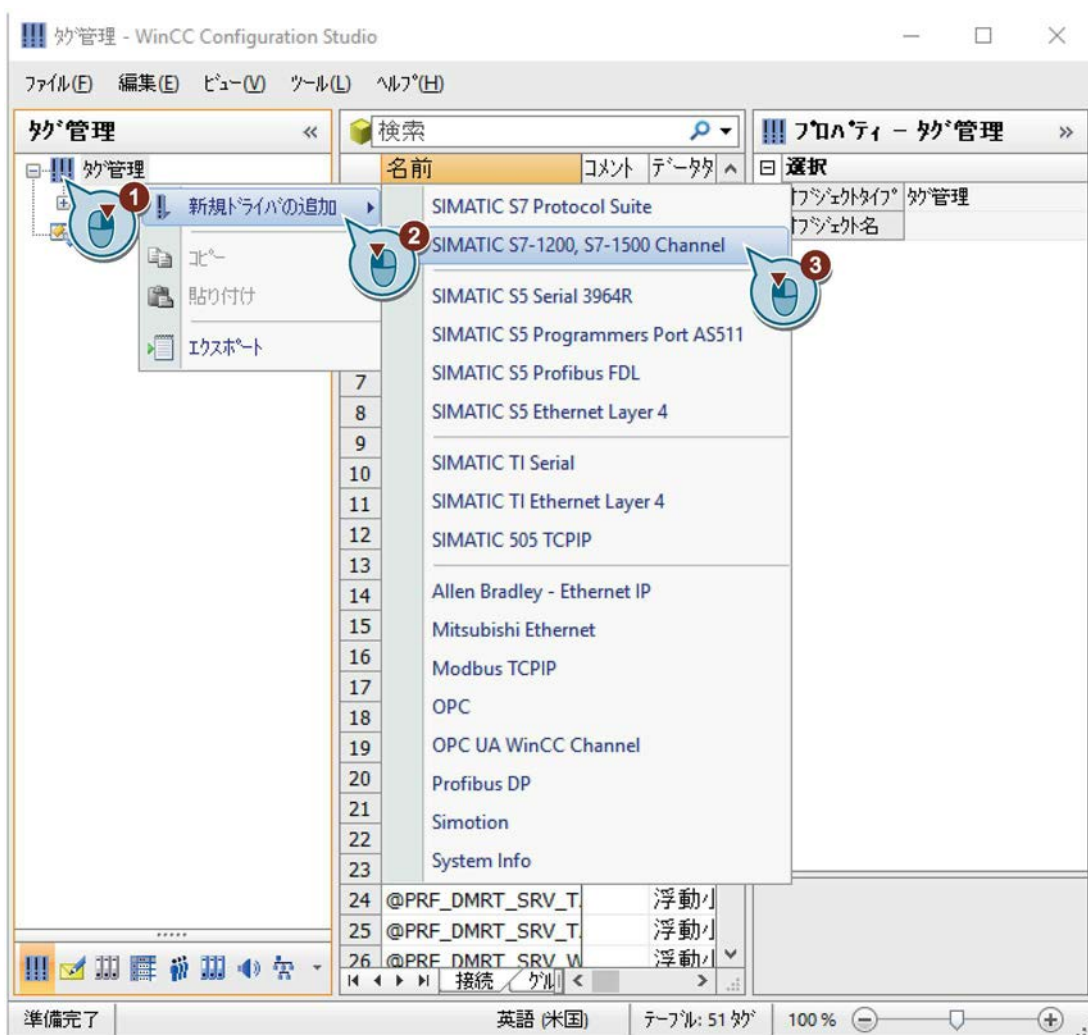
手順

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。

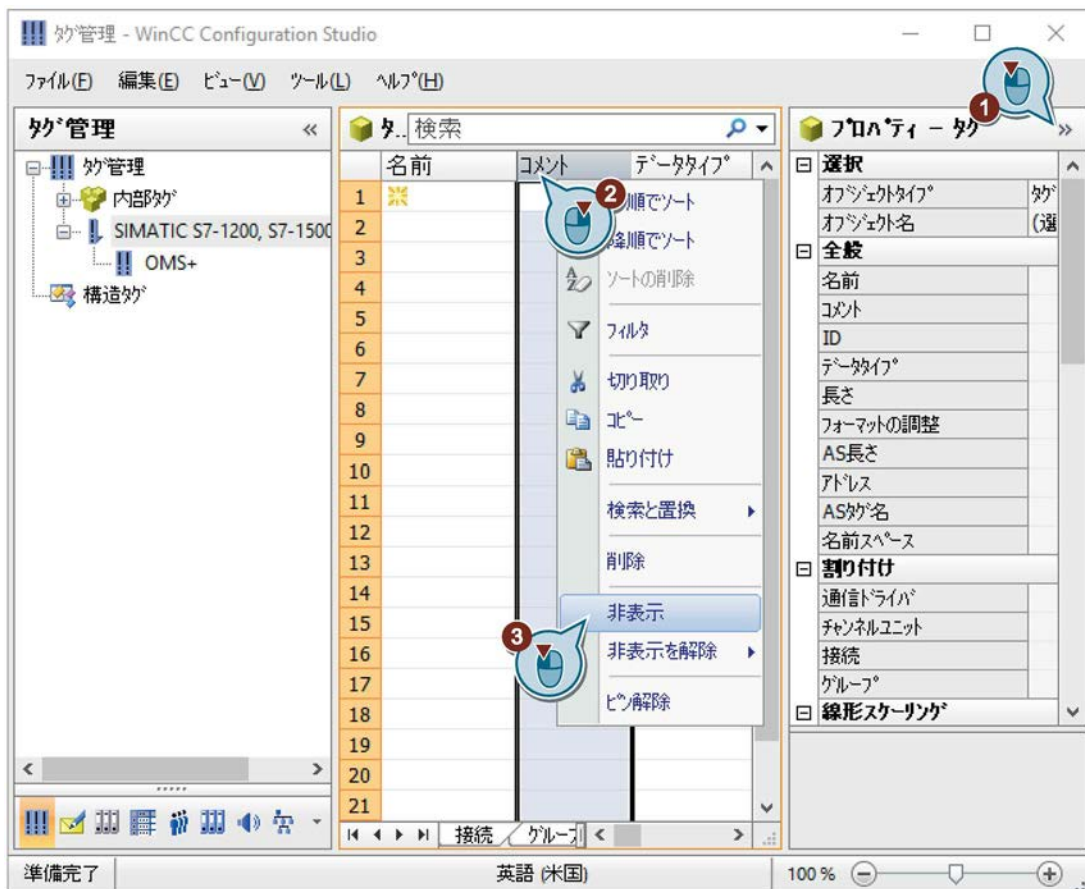
2. "SIMATIC S7-1200、S7-1500チャンネル"を挿入します。



5.4 チャンネルの追加

- 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。

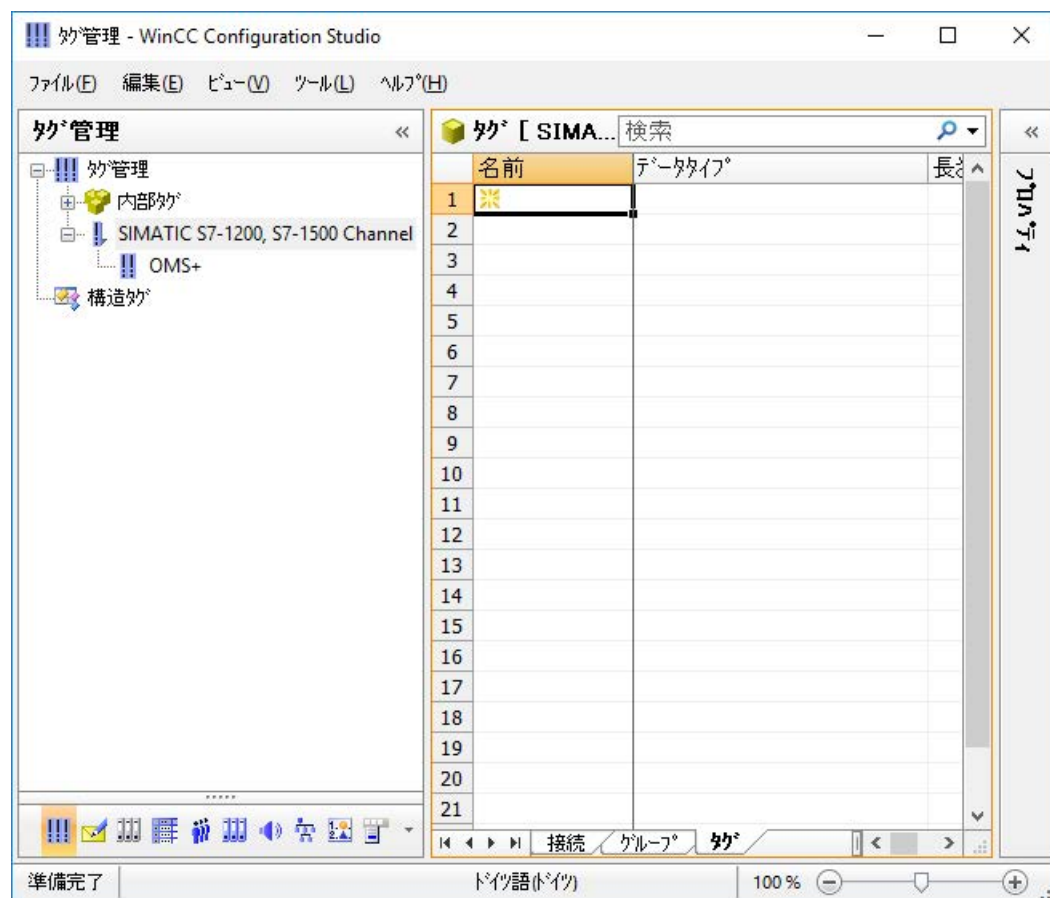
[プロパティ]ウィンドウを再度表示するには、 ボタンをクリックします。
- WinCC Configuration Studioのエディタで、データ領域にある列を非表示および表示することができます。



結果

"SIMATIC S7-1200、S7-1500チャンネル"がタグ管理に表示されます。

次のステップで、チャンネルユニット"OMS+"の下でオートメーションシステムへの接続を作成します。



5.5 接続の作成

はじめに

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。

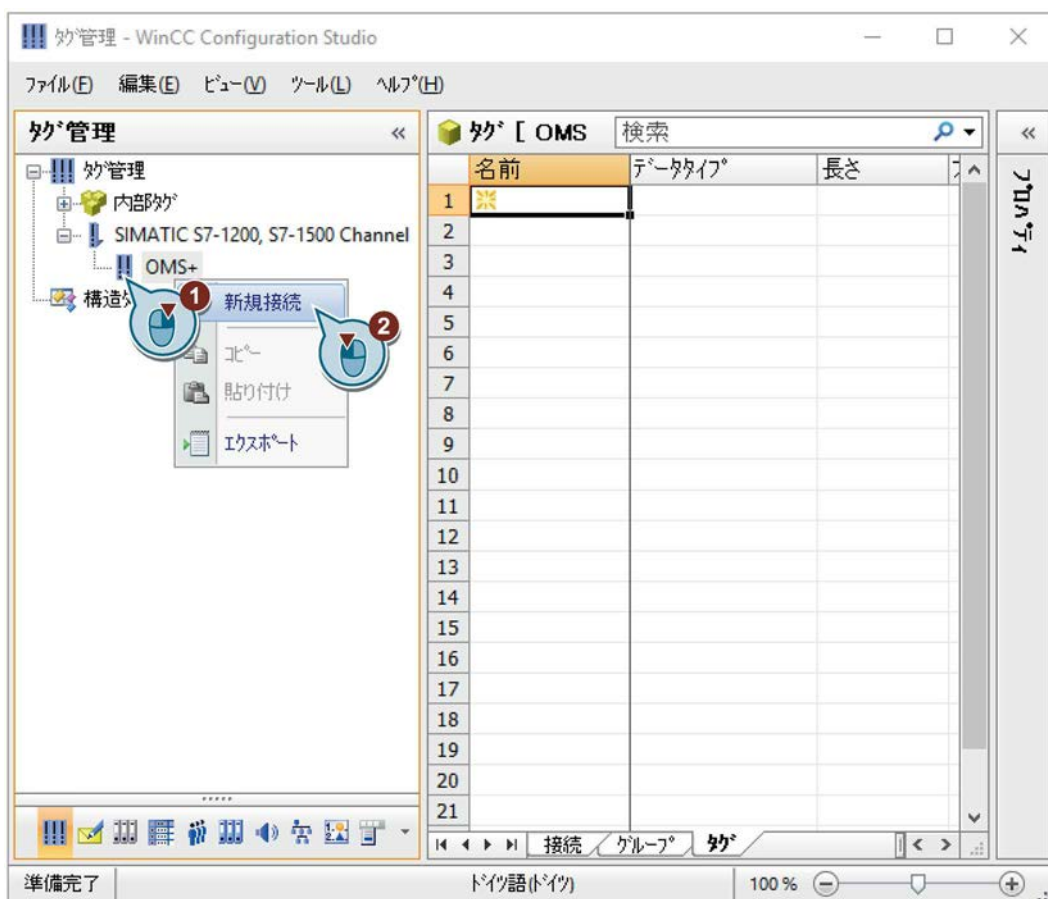
ランタイムには、この接続を通してデータ交換が行なわれます。

必要条件

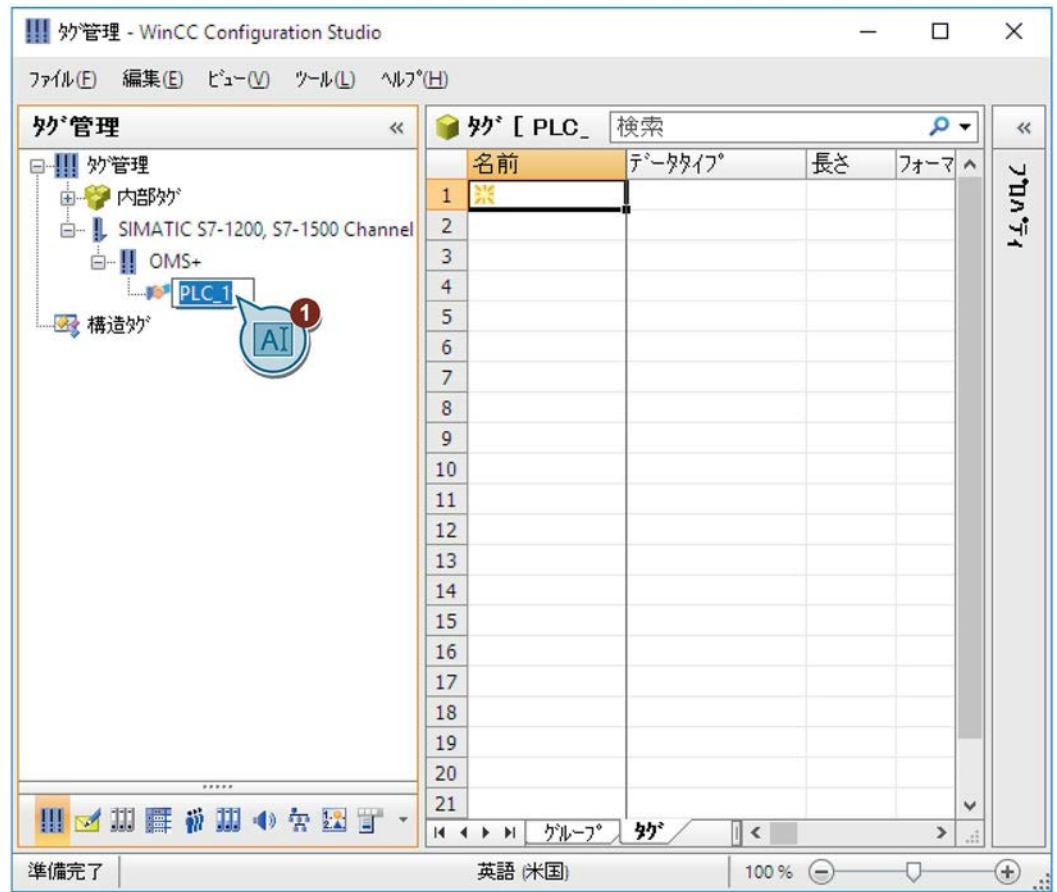
- "SIMATIC S7-1200、S7-1500チャンネル"がタグ管理に追加されていること。

手順

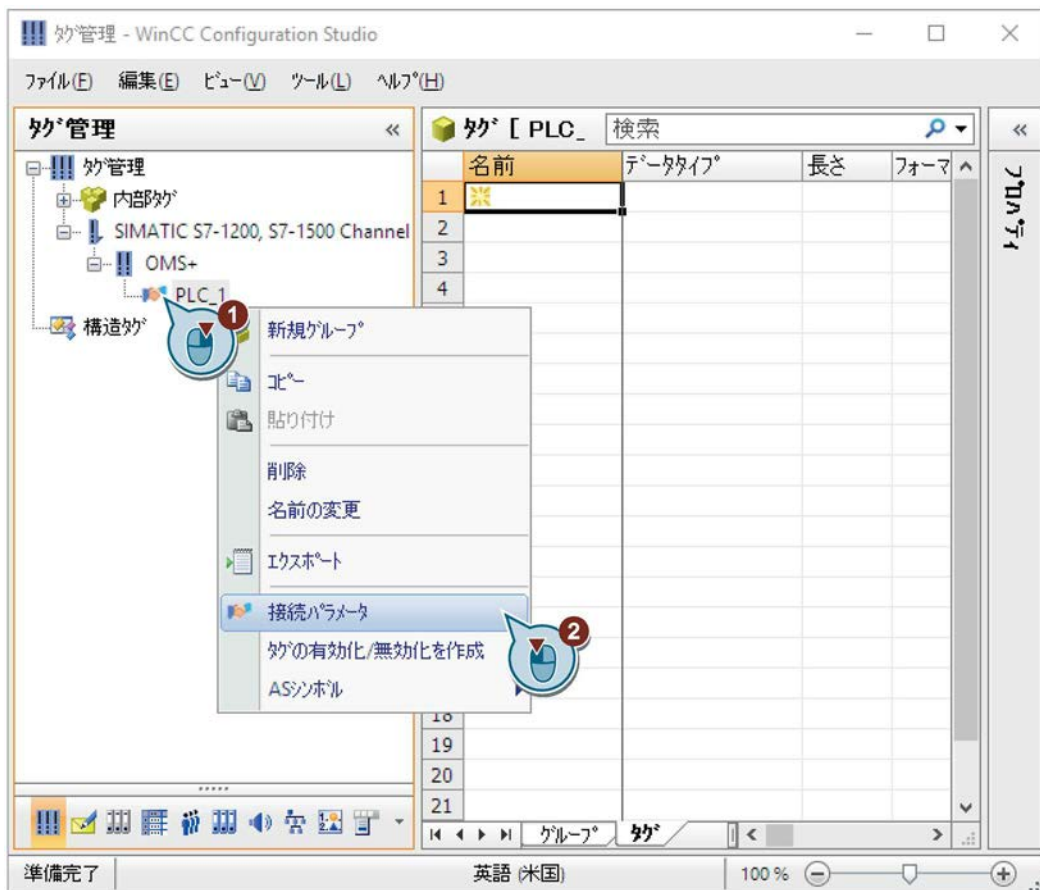
1. チャンネルユニット"OMS+"の下で、新しい接続を作成します。



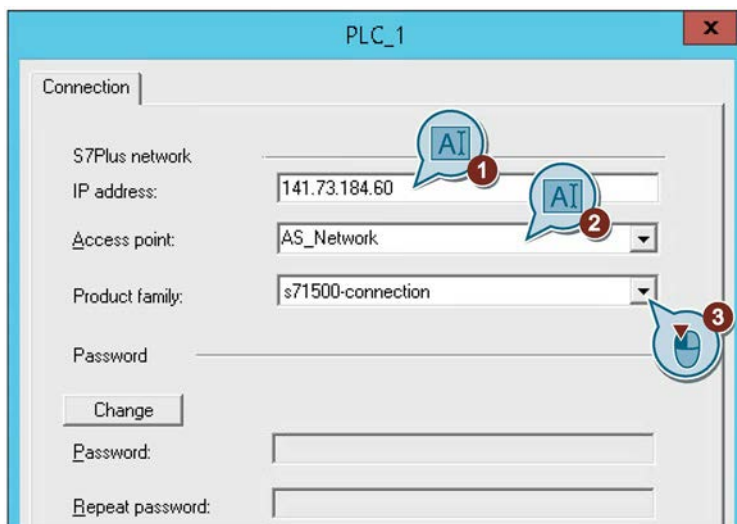
2. 接続名として"PLC_1"と入力します。



3. [接続]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。

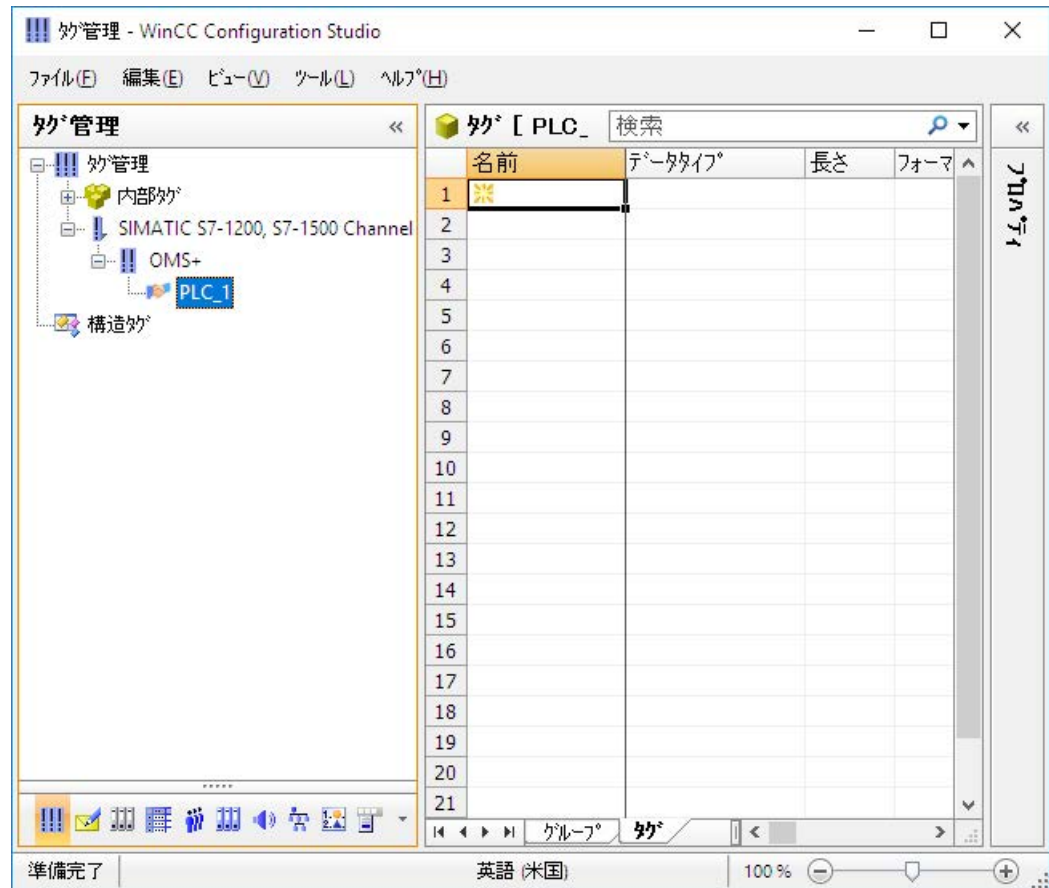


5. [OK]を押してダイアログを閉じます。

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。



5.6 タググループのプロセスタグの作成

はじめに

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で使用す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。

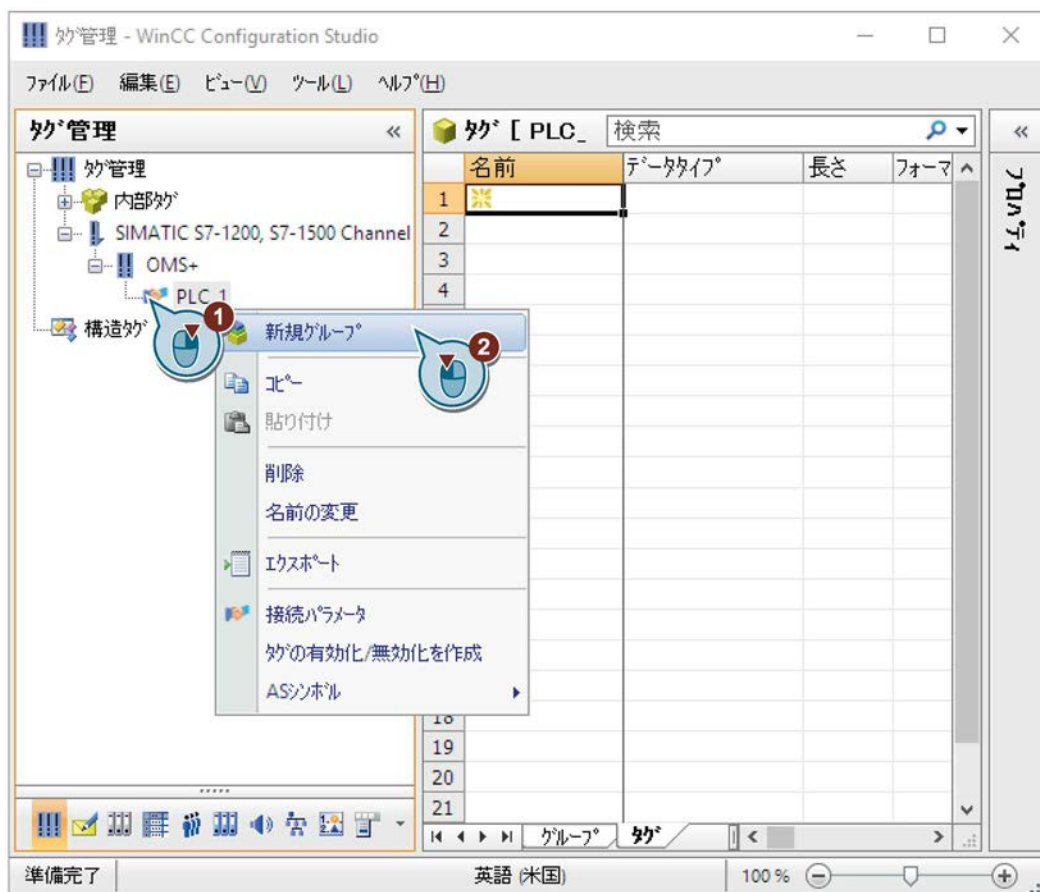
アドレス指定のタイプは、通信パートナーのタイプに依存します。

必要条件

- "PLC_1"接続がタグ管理で作成されていること。

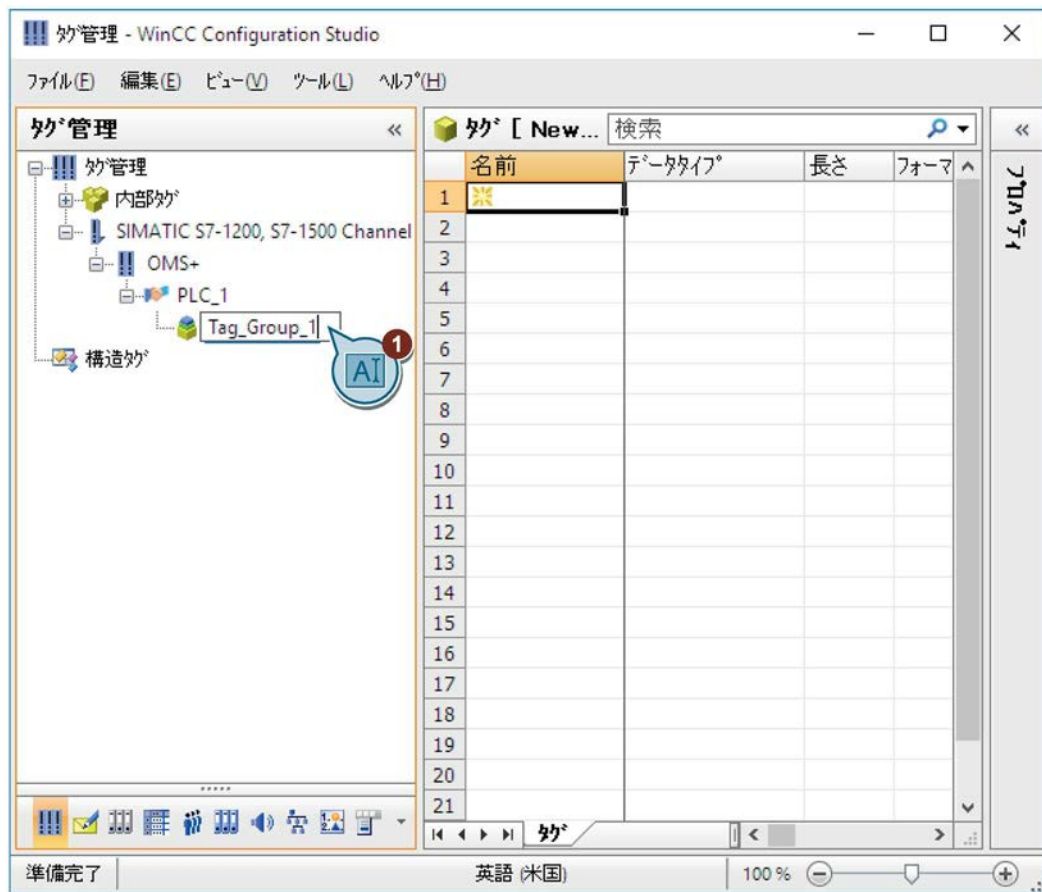
手順

1. タググループを作成します。



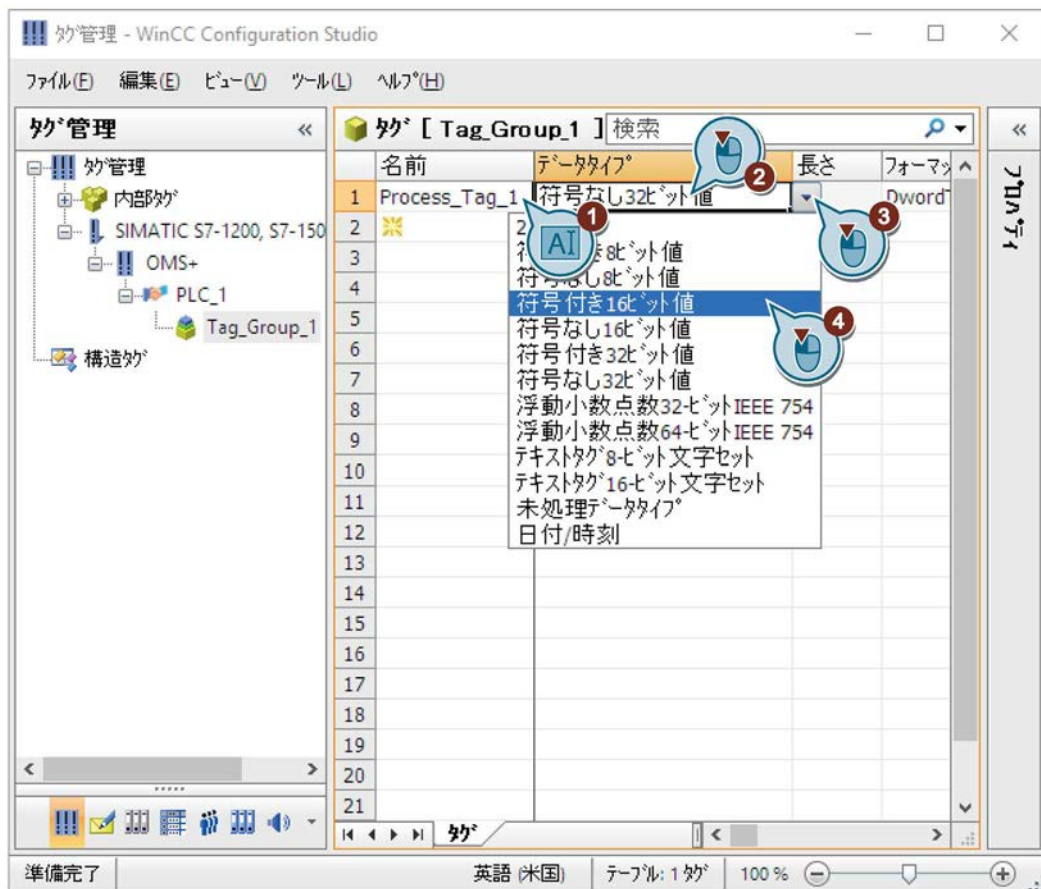
5.6 タググループのプロセスタグの作成

2. タググループ名として"Tag_Group_1"と入力します。



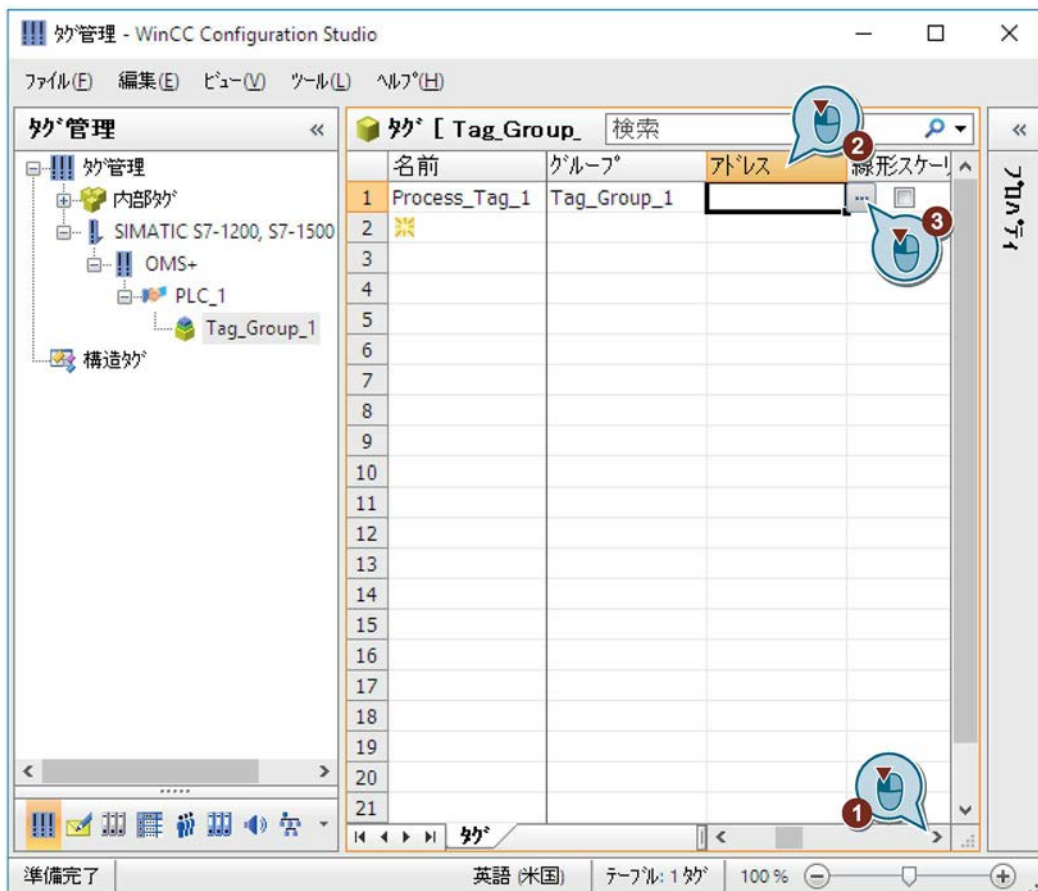
5.6 タググループのプロセスタグの作成

3. プロセスタグ名として、"Process_Tag_1"を入力して、タググループでプロセスタグを作成します。
4. "符号付き16ビット値"データタイプを選択します。



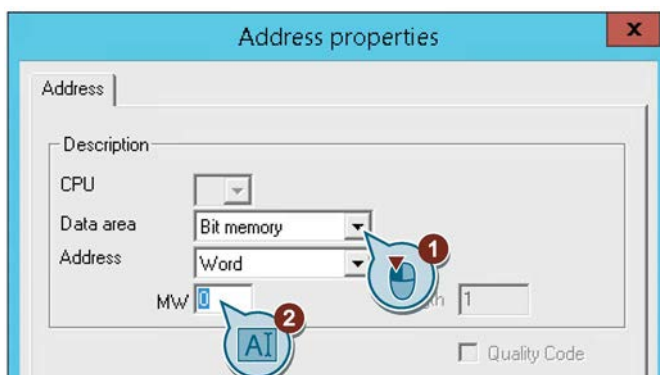
5.6 タググループのプロセスタグの作成

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

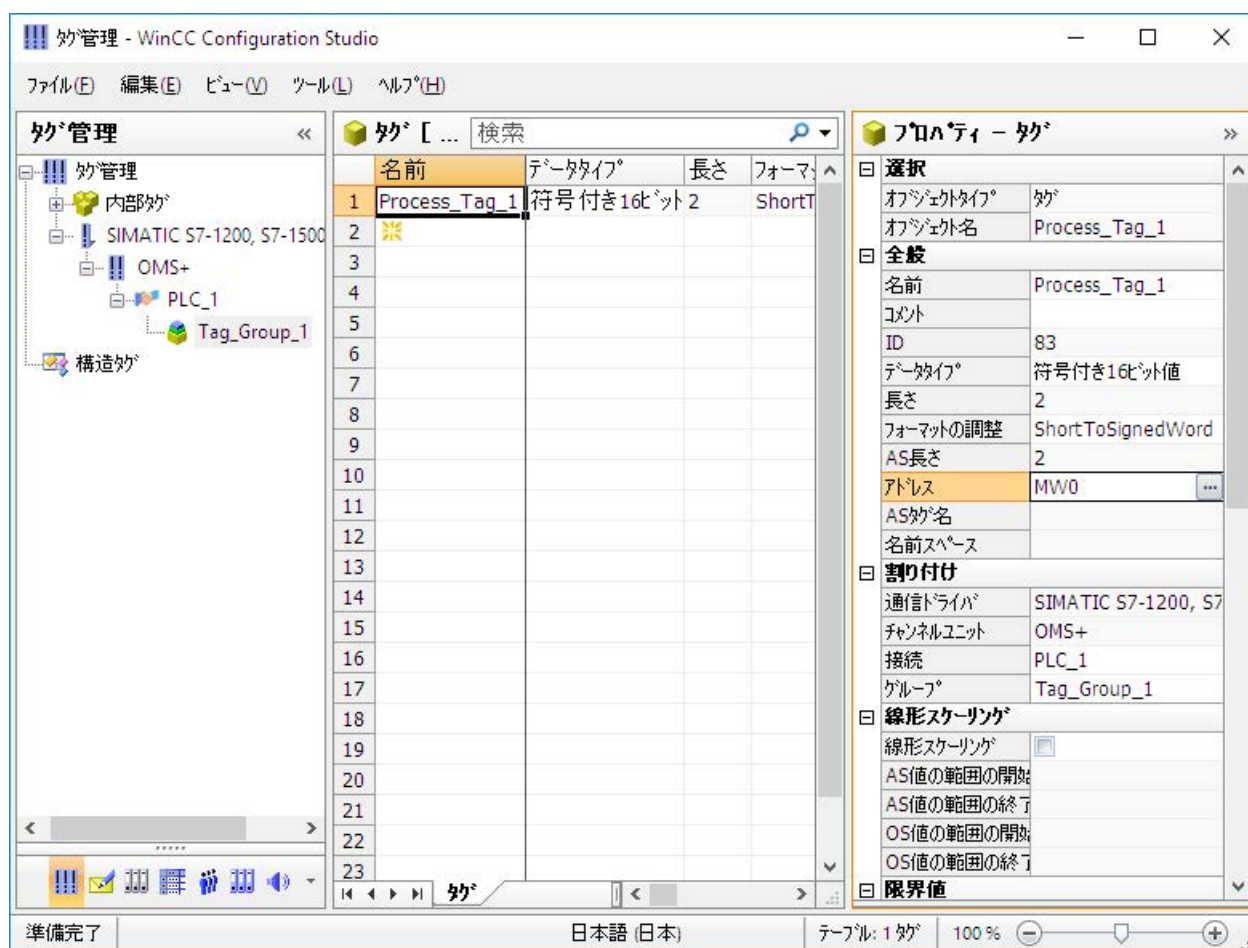
プロセスタグ"Process_Tag_1"が作成されます。[タグ管理]エディタのテーブルエリアに、プロセスタグが表示されます。

[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。

WinCCでプロセス値をスケールするために、次のステップでリニアスケールングのプロパティを指定します。



5.7 WinCCでのプロセスタグのスケーリング

はじめに

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。

リニアスケーリングについて、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。

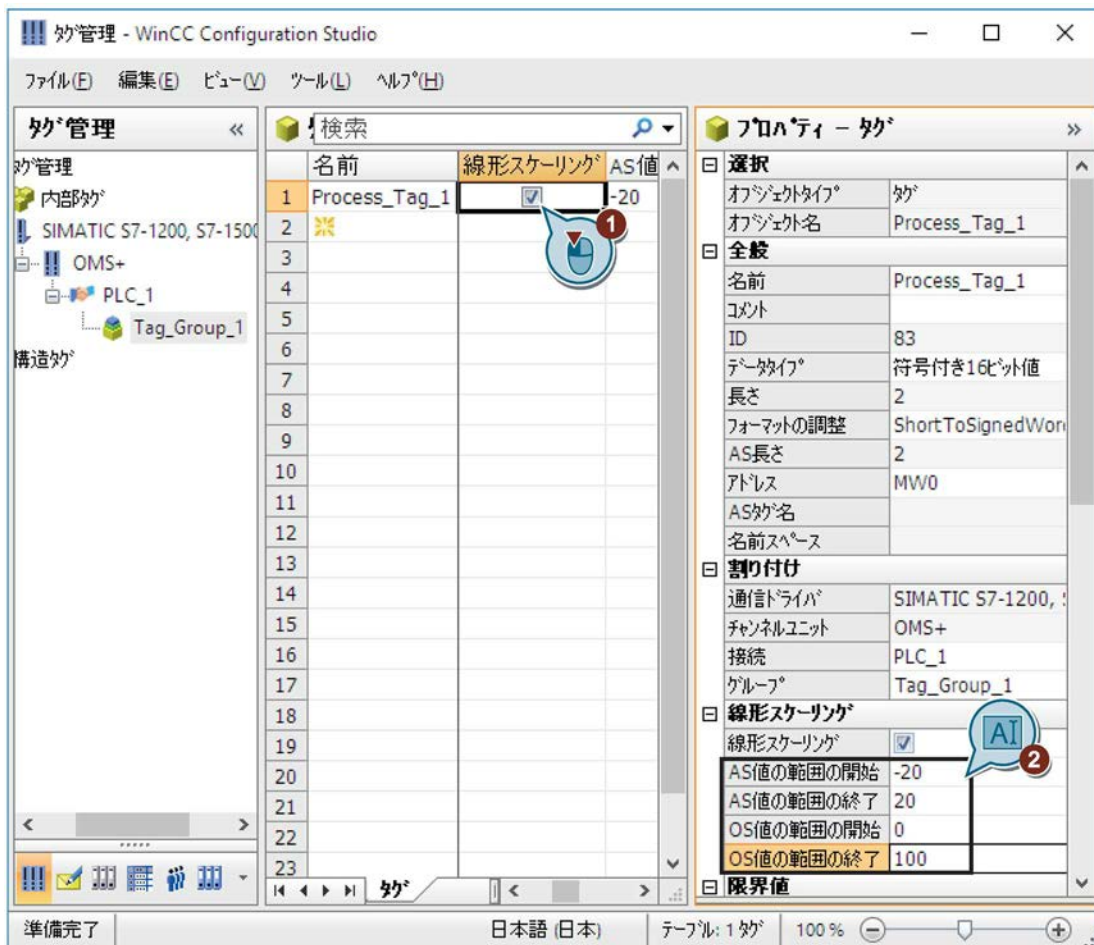
タグ管理では、抵抗値は[AS値の範囲]列の設定に対応しています。温度は"OS値の範囲"列の設定に対応しています。

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。

次のステップで、"Quick_Start"プロジェクトの内部タグを作成します。

5.8 内部タグの作成

はじめに

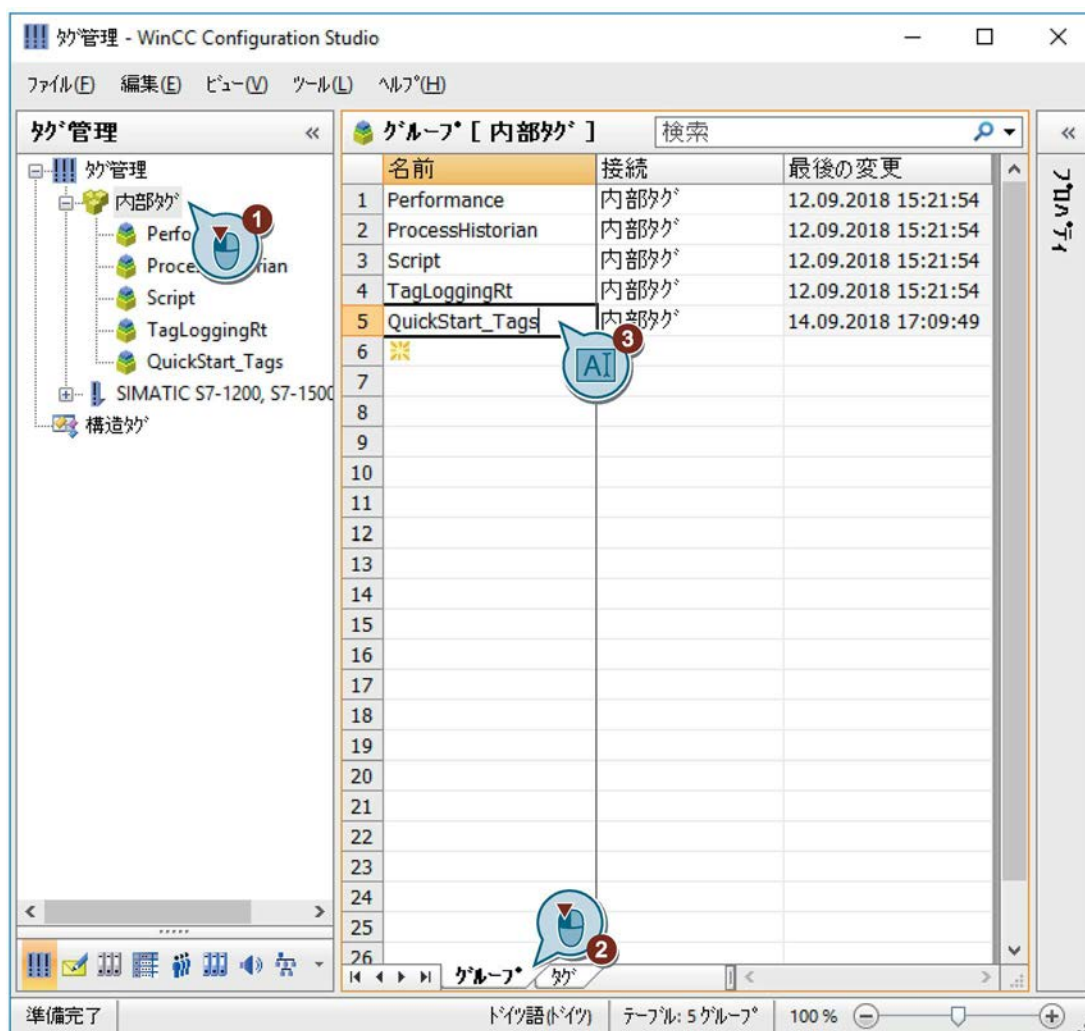
次のステップは、タググループでの内部タグの作成方法と、そのプロパティの決定方法について示しています。
内部タグは、WinCC内で値を転送するために使用します。

必要条件

- "Quick_Start"プロジェクトが開いている。

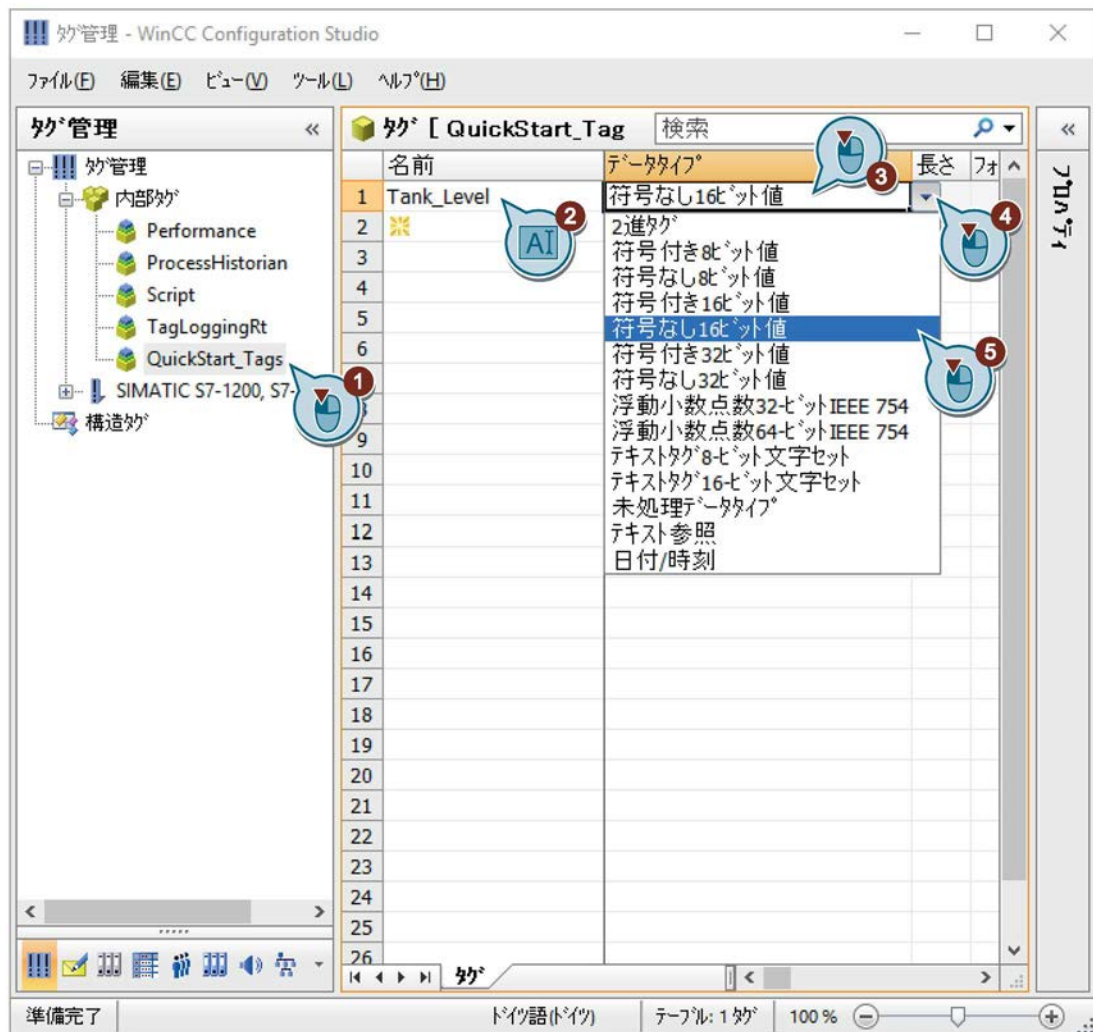
手順

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。



5.8 内部タグの作成

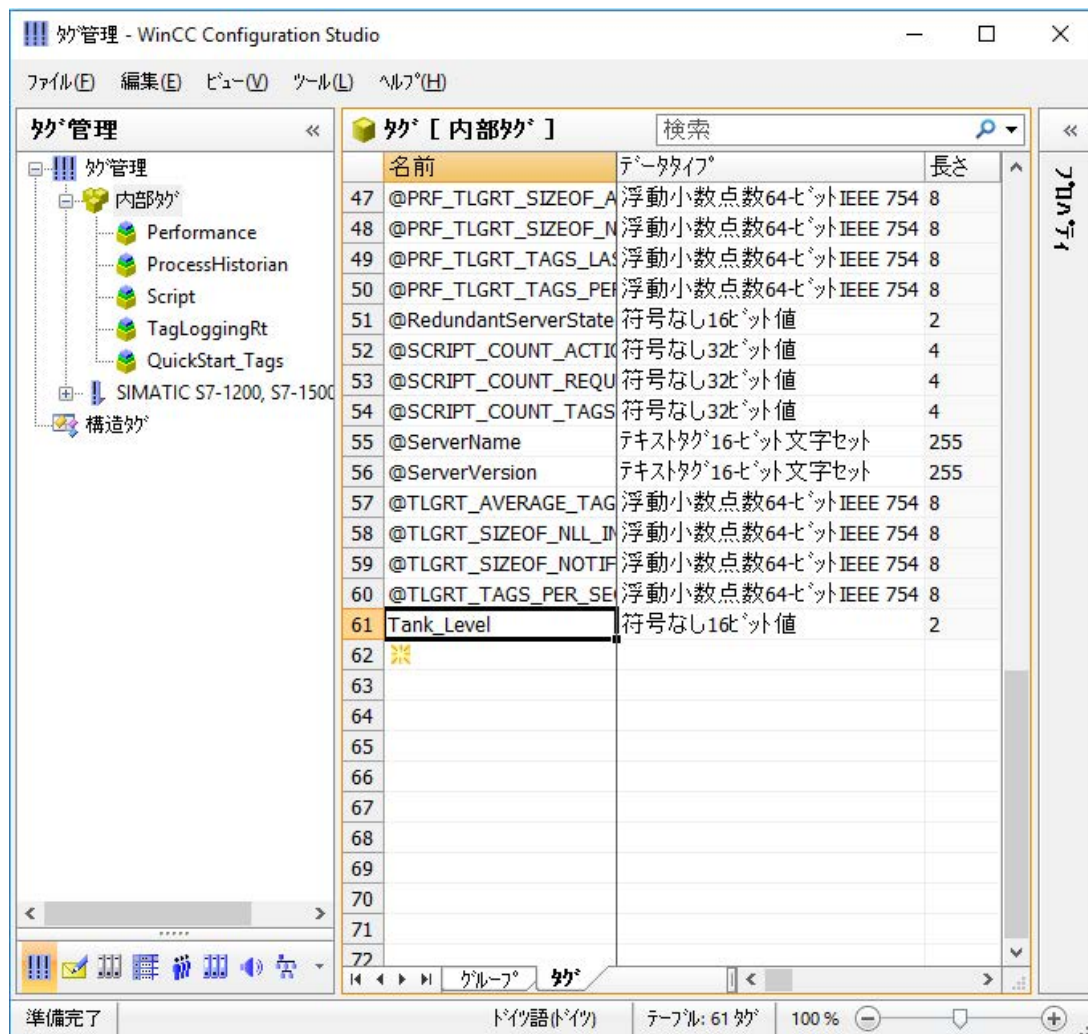
- 内部タグの名前として"**Tank_Level**"と入力して、内部タグをタググループで作成します。
- データタイプ[符号なし16ビット値]を選択します。



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。



5.8 内部タグの作成

プロセス画面の設定

6.1 プロセス画面の設定

はじめに

この章では、グラフィックシステムに関する情報やWinCCエクスプローラでの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。

それぞれのプロセス画像は、複数のオブジェクトで構成されています。

- 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。

プロジェクトは、たいていいくつかのプロセス画像から構成されます。
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。

このセクションでは、水処理システムを表すプロセス画像を作成します。
実習には、2番目のプロセス画像を作成することが関連しています。

このプロセス画像に必要なすべてのオブジェクトは、WinCCにあります。

6.2 グラフィックシステム

はじめに

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、以下の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。

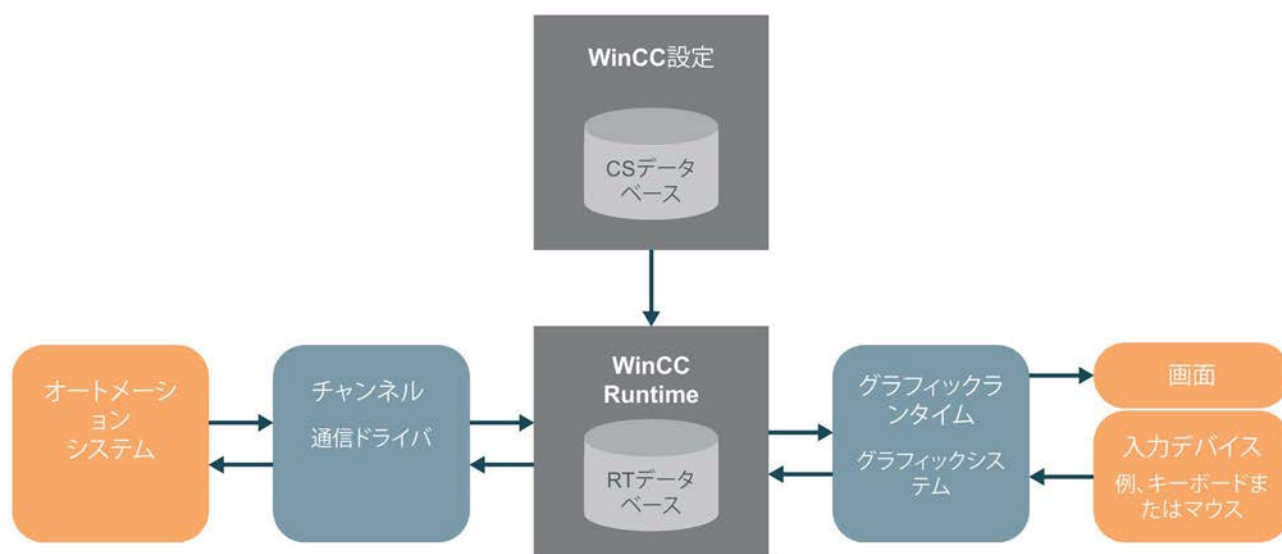
- [グラフィックデザイナー]エディタは、グラフィックシステムの設定コンポーネントです。

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。

WinCCシステムのグラフィックシステム



6.3 プロセス画面の作成

はじめに

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。

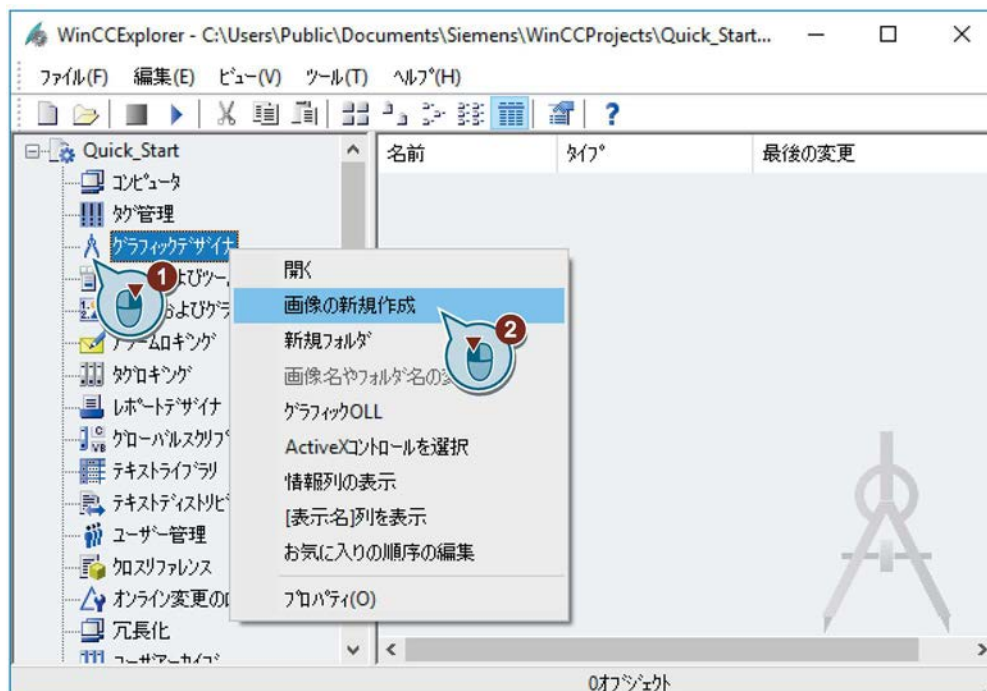
- すぐにプロセス画像を編集する場合は、[グラフィックデザイナー]エディタを使用して作成します。
- 処理する前に必要なプロセス画像をすべて作成しておく場合は、WinCCエクスプローラを使用することを推奨します。

必要条件

- "Quick_Start"プロジェクトが開いている。

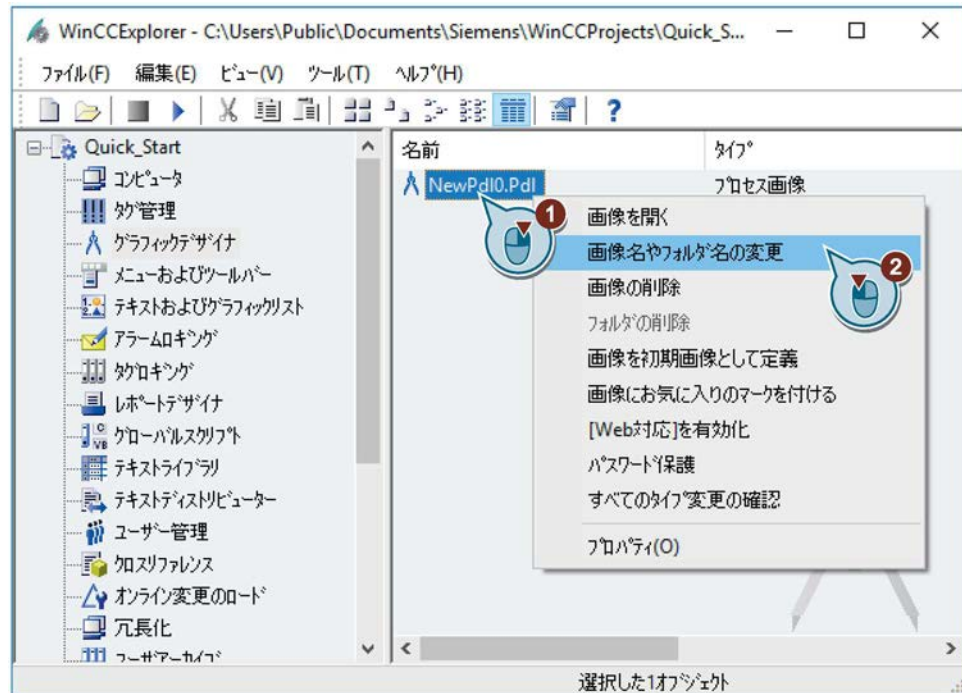
手順

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。

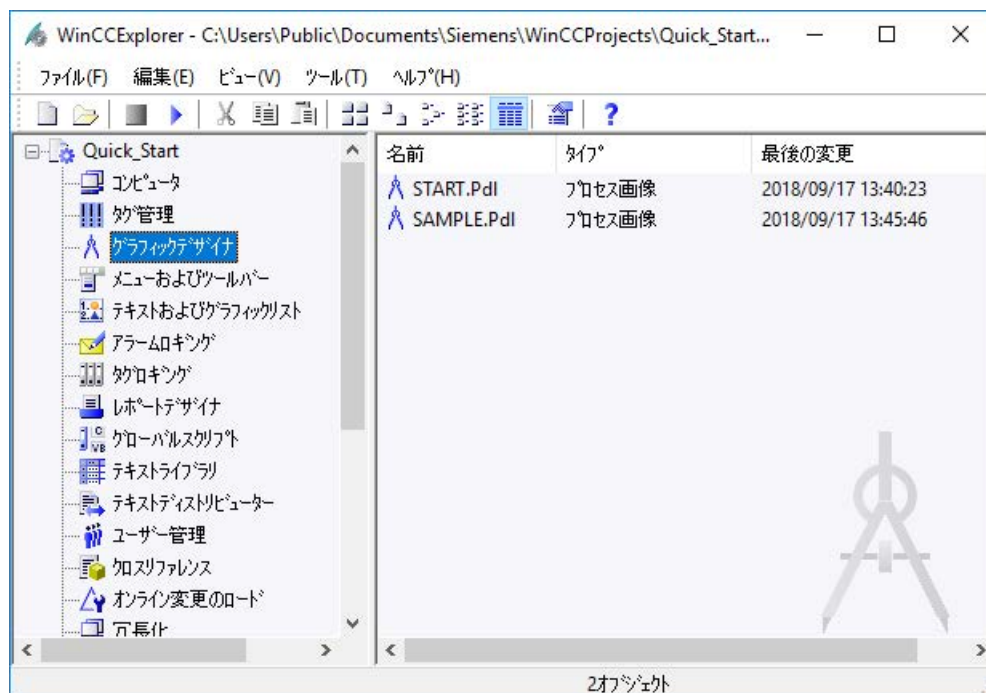
6.3 プロセス画面の作成

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。

水処理工場をグラフィカルに表示するために、次のステップでプロセス画像"START.pdl"に複数のオブジェクトを追加します。



6.4 プロセス画面の編集

6.4.1 プロセス画面の編集

はじめに

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"画像を操作して、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、プロセス画像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は、ドラッグおよびドロップでプロセス画像に追加できます。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。その他のコントロールはリンクできます。

6.4 プロセス画面の編集

ライブラリを使用すると、非常に効率のよい方法でピクチャを作成できます。
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。

"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からグラフィックをインポートすることも可能です。

6.4.2 ライブラリからのグラフィックオブジェクトの挿入

はじめに

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。

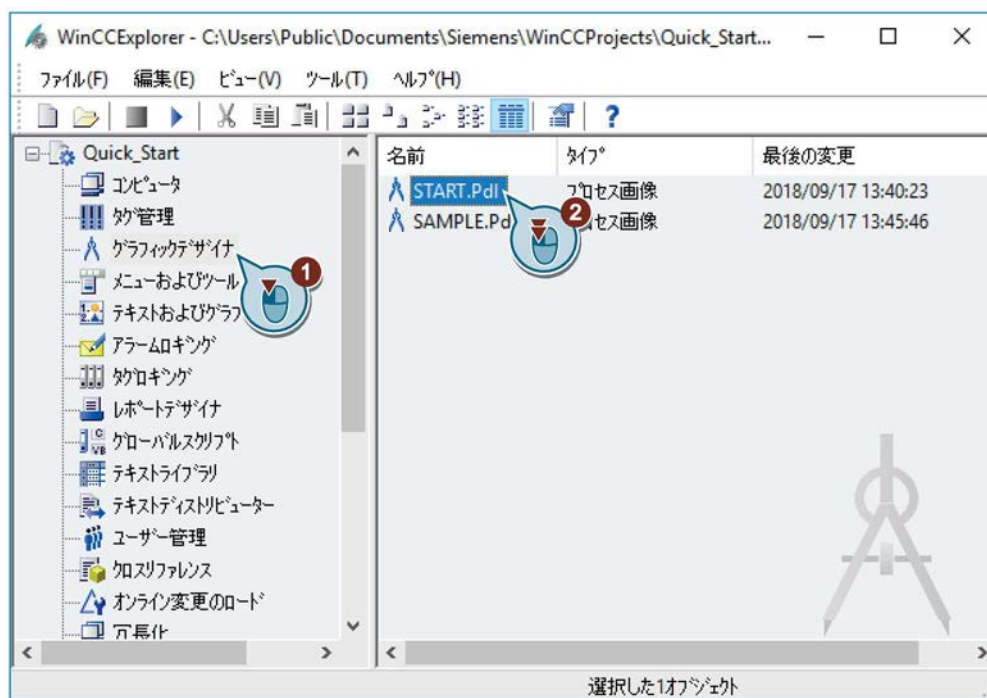
"Quick_Start"プロジェクトでは、**[グローバルSVGライブラリ]**エリアのグラフィックオブジェクトだけが必要です。

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描写します。

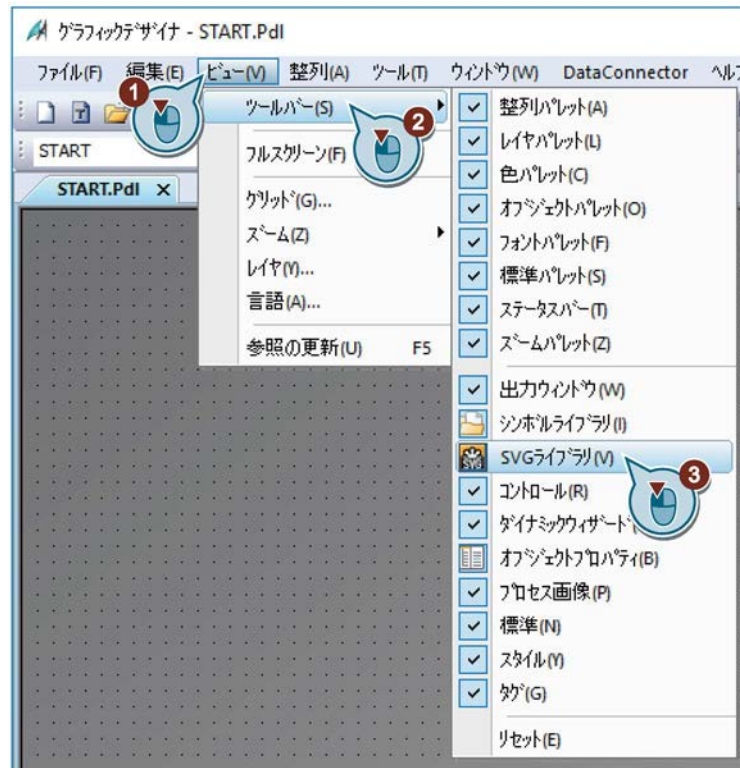
6.4 プロセス画面の編集

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。



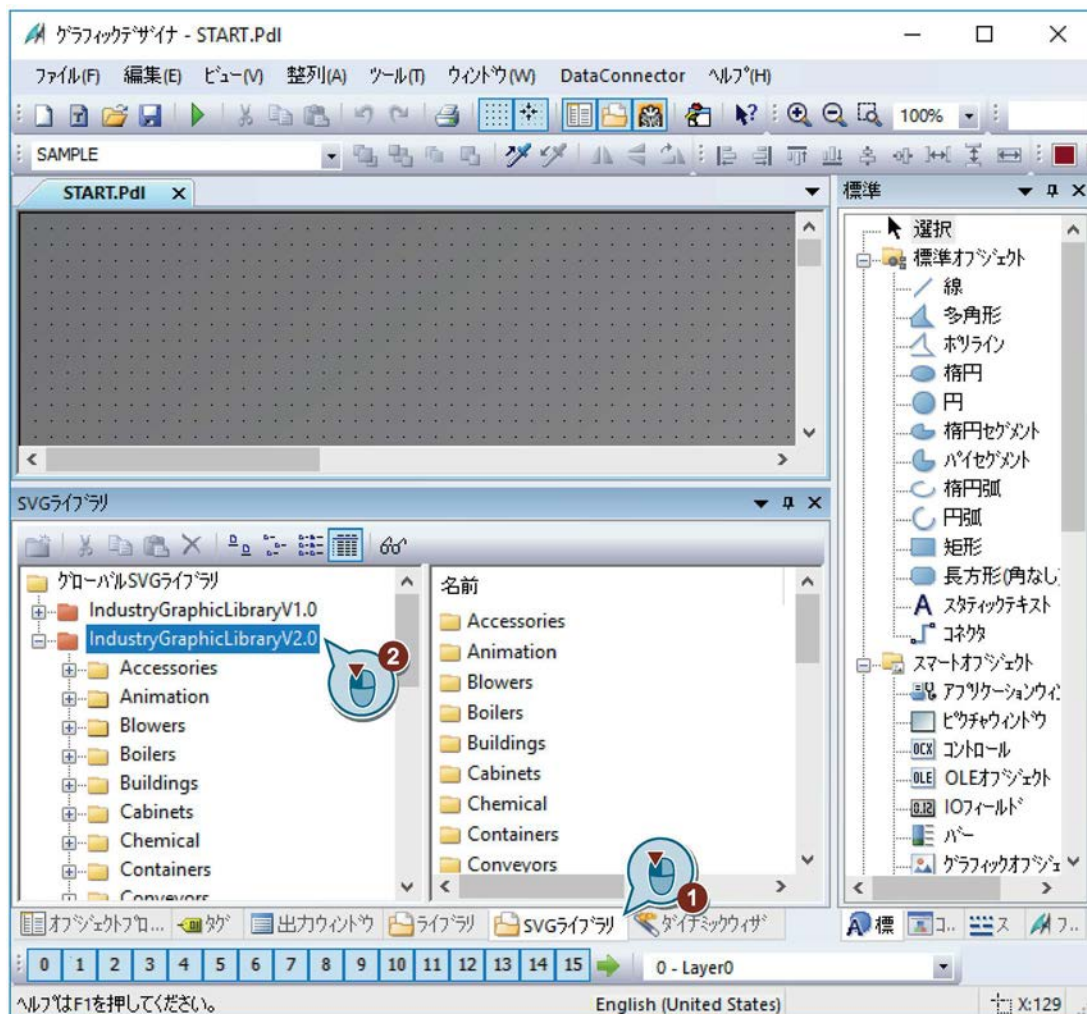
2. [SVGライブラリ]ウィンドウが表示されない場合、ビューを有効にします。



[SVGライブラリ]ウィンドウが開きます。

6.4 プロセス画面の編集

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。

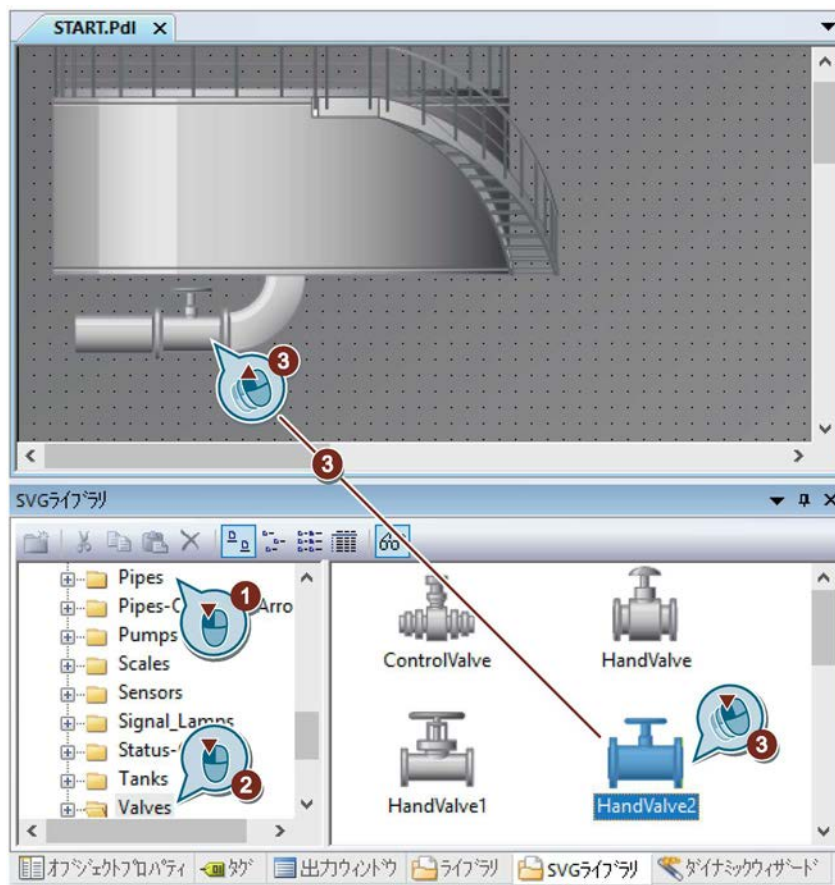
 と  ボタンを使用して表示されたシンボルのサイズを変更することができます。

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。

6.4 プロセス画面の編集

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)。

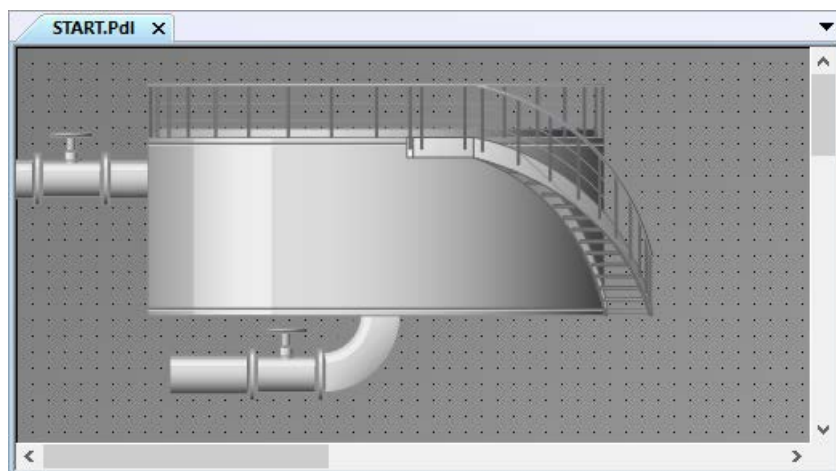
マウスまたはカーソルキーを使用して、プロセス画像上でオブジェクトの画像を移動します。



結果

水処理システムが描画されました。

表示されたプロセスにラベルを付けるために、次のステップでプロセス画像に
"スタティックテキスト"オブジェクトを挿入します。



6.4 プロセス画面の編集

6.4.3 [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

はじめに

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。

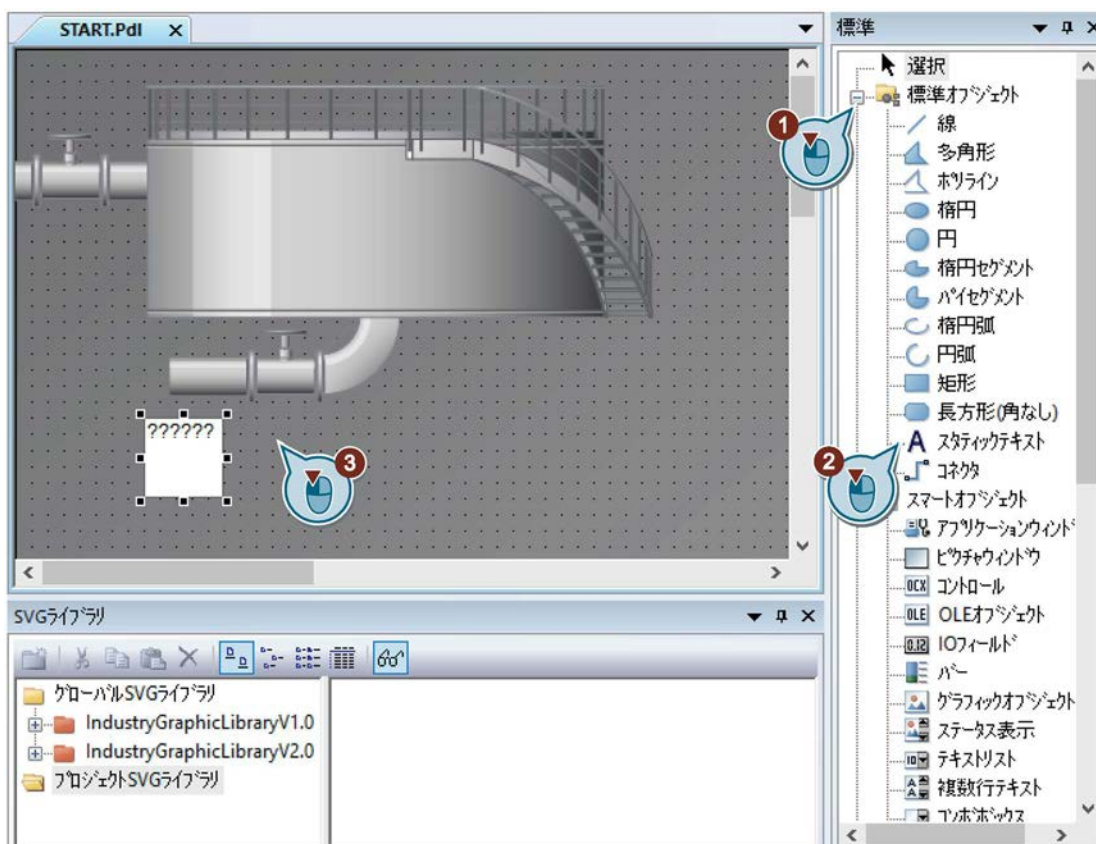
"Quick_Start"プロジェクトでは、表示されたプロセスにラベル付けするスタティックテキストが必要です。複数の図を作成する場合は、表示されたプロセスに一意のラベルを付けることが非常に重要です。

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。

手順

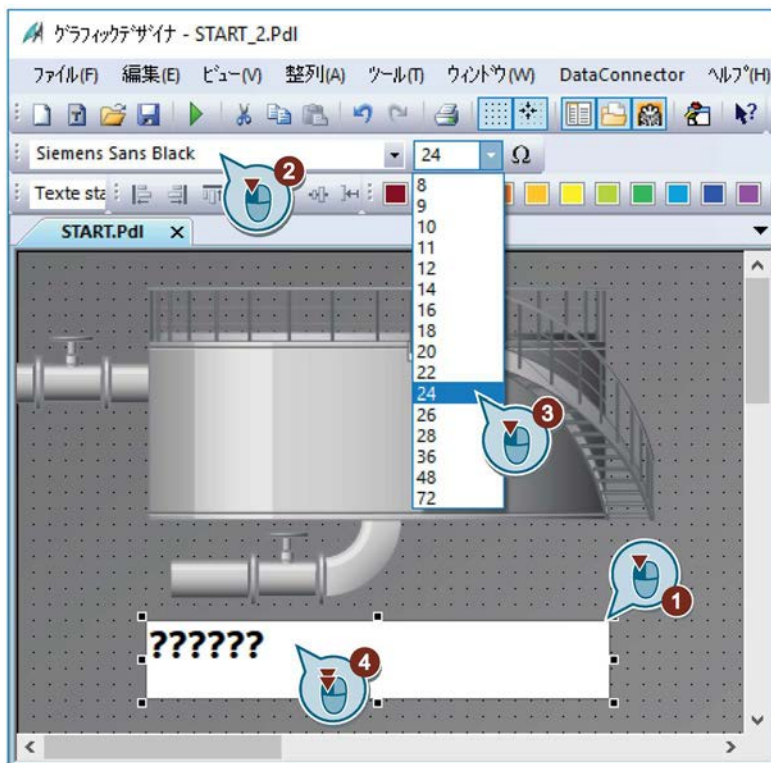
1. [スタティックテキスト]オブジェクトを挿入するには、[標準]ウィンドウでオブジェクトをクリックしてから、プロセス画像をクリックします。



テキストフィールドが、プロセス画像に表示されます。

6.4 プロセス画面の編集

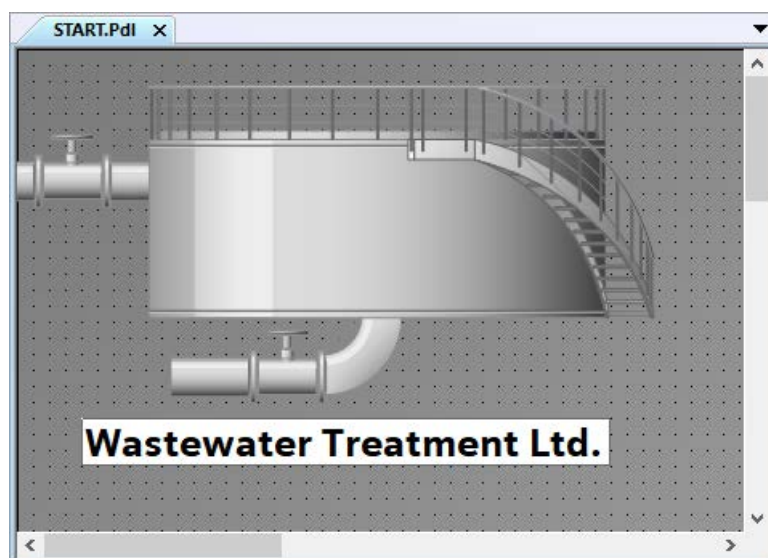
2. マウスを使用してテキストボックスをドラッグし、フォントタイプとサイズを設定します。



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. ツールバーボタンを使用して、プロセス画像"START.pdl"を保存します。
6. プロセス画像"START.pdl"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。



6.4 プロセス画面の編集

6.4.4 プロセス画面"SAMPLE.pdl"の編集

はじめに

次のステップで、プロセス画像"SAMPLE.pdl"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

- プロセス画像"SAMPLE.pdl"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。

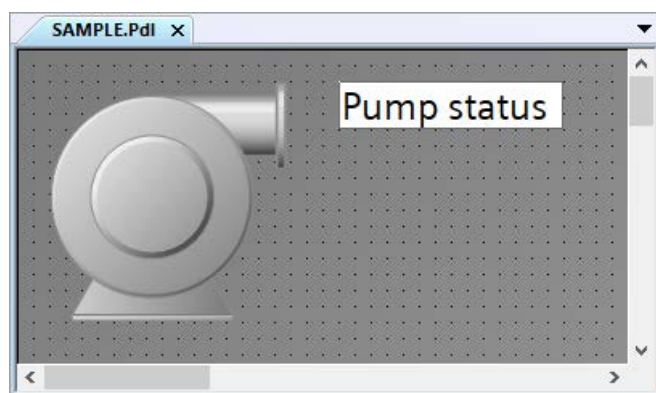
手順

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、任意のプロセスを表示します。
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. プロセス画像"SAMPLE.pdl"を閉じます。

結果

プロセス画像"SAMPLE.pdl"が編集されました。

"START.pdl"プロセス画像で、システムをダイナミックにするために、次のステップでエントリフィールドを挿入し、内部タグによってそれをバイオコンバータの画像とリンクさせます。



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 55)

6.5 画面ダイナミックスの処理

6.5.1 画面ダイナミックスの処理

はじめに

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にします。

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、転送された値に従っ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制御および監視を実行します。

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。

6.5.2 フィルレベルインジケータのダイナミック化

はじめに

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

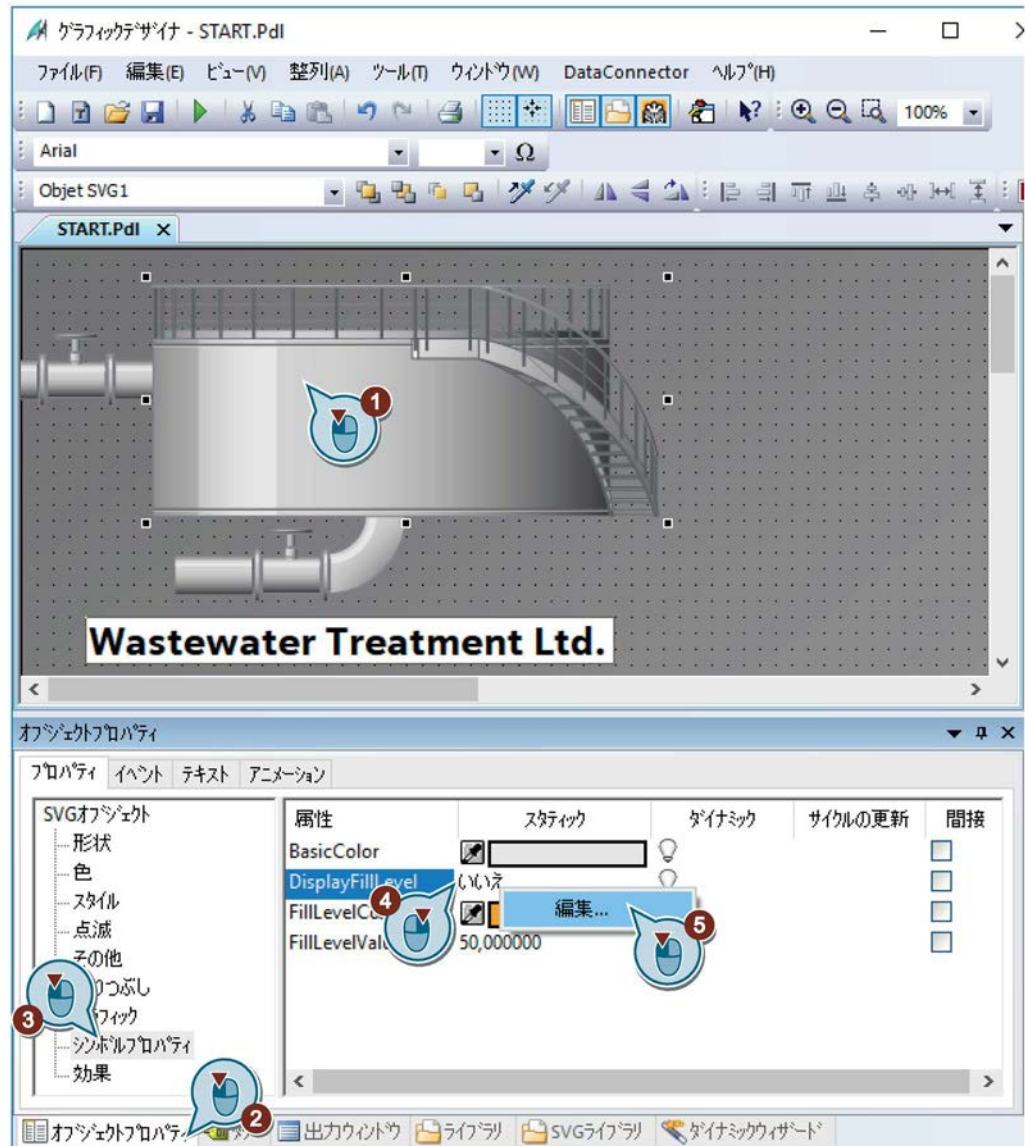
更新サイクルは、フィルレベルインジケータを更新する時間間隔を決定します。

必要条件

- プロセス画像"START.pdl"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画像として挿入されていること。

手順

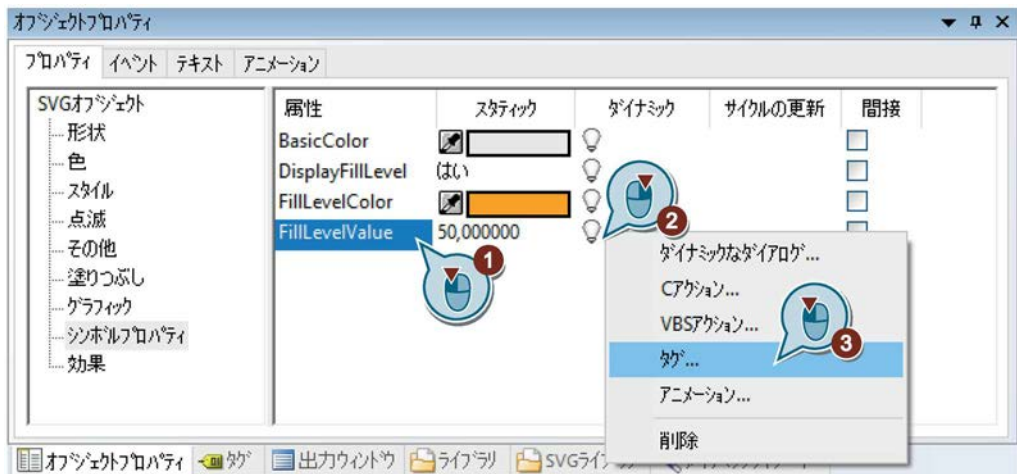
1. "START.pdl"プロセス画像を開き、[Digester]オブジェクトをクリックします。
2. [シンボルプロパティ]グループを[オブジェクトプロパティ]ウィンドウで開き、[DisplayFillLevel]属性を[なし]から[あり]に変更します。



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。

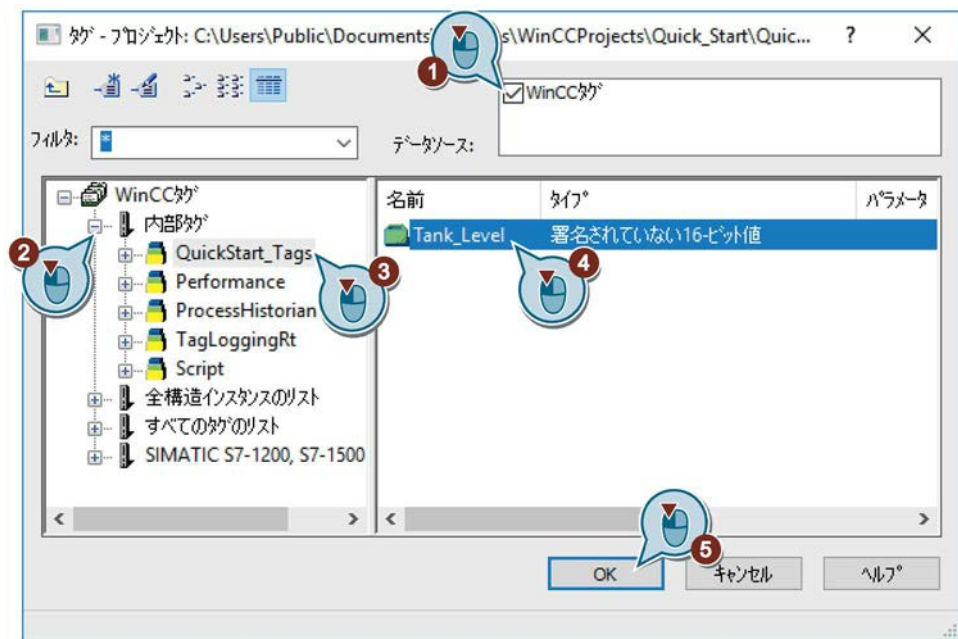
6.5 画面ダイナミックスの処理

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。



4. 内部タグ"Tank_Level"を選択します。

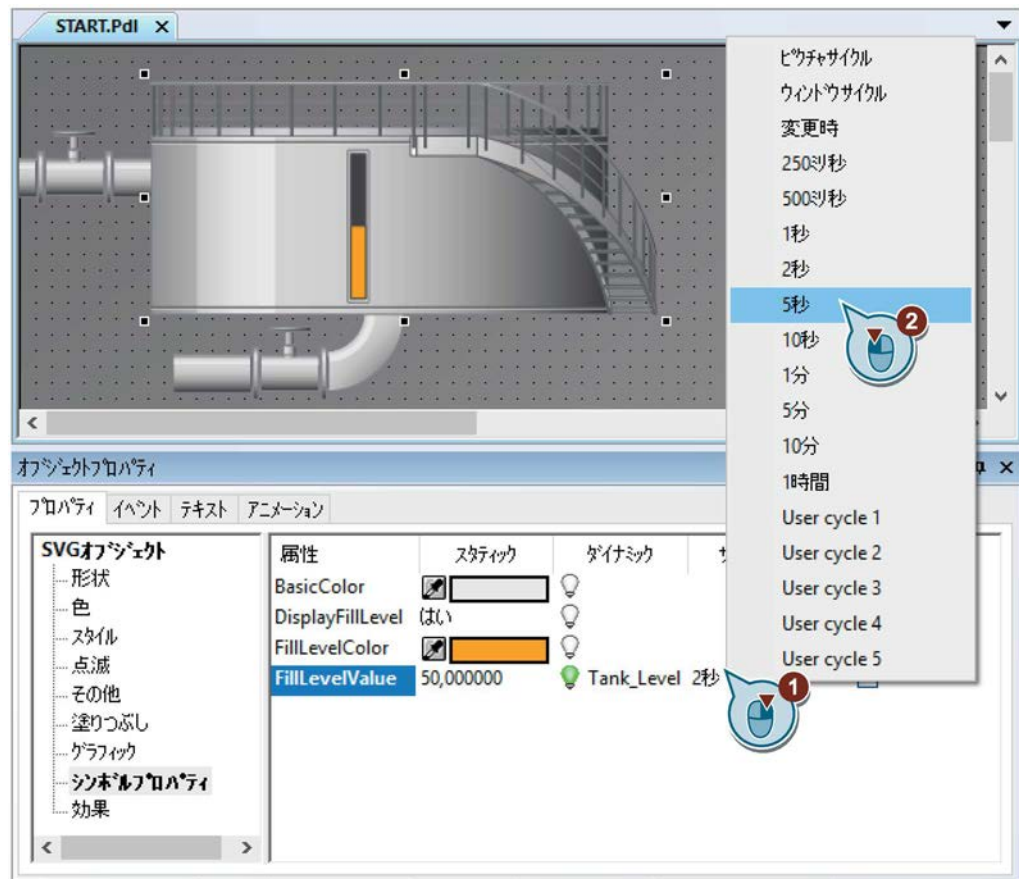
必要な場合、"データソース"エリアで"WinCCタグ"オプションを選択します。



"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。

5. 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。

値を入力または出力するために、次のステップでプロセス画像"START.pdl"にI/Oフィールドを追加します。

6.5.3 I/Oフィールドの挿入とダイナミック化(グラフィックシステム)

はじめに

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。
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画像の間に間接的接続が作成されます。

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。

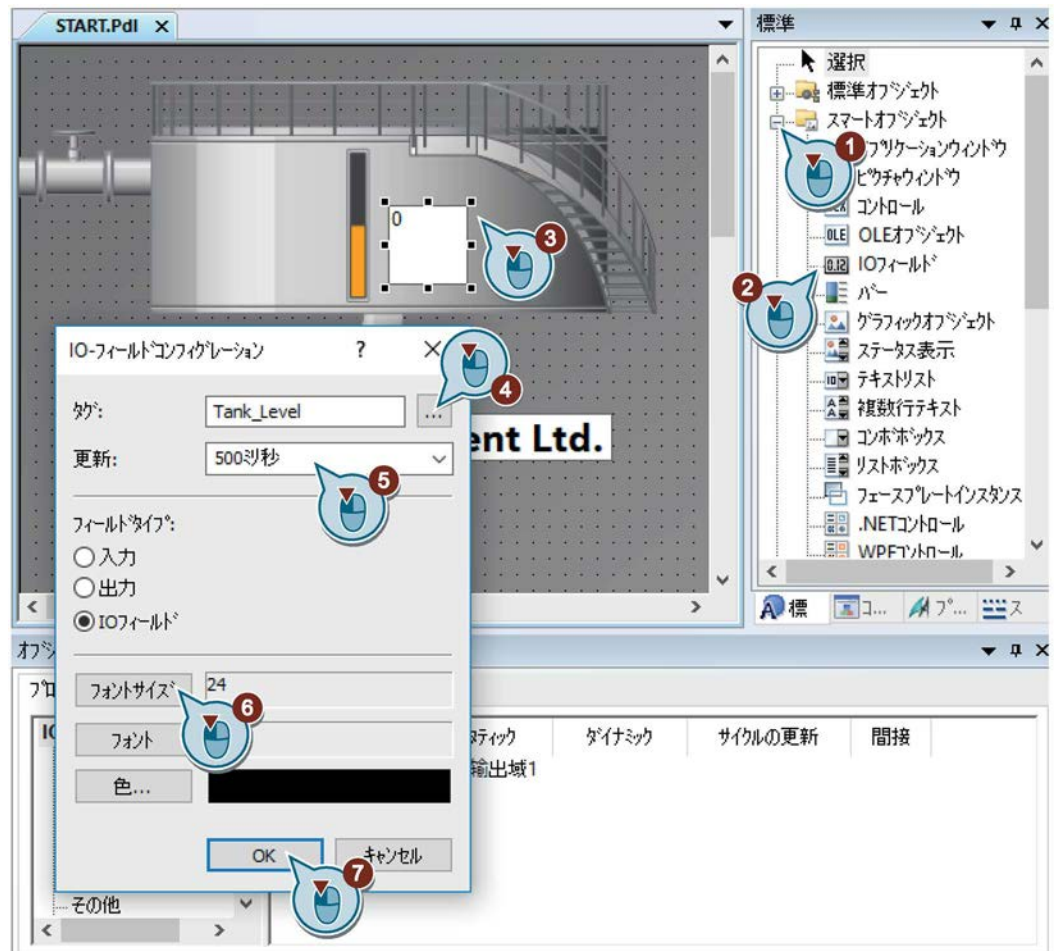
手順

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。



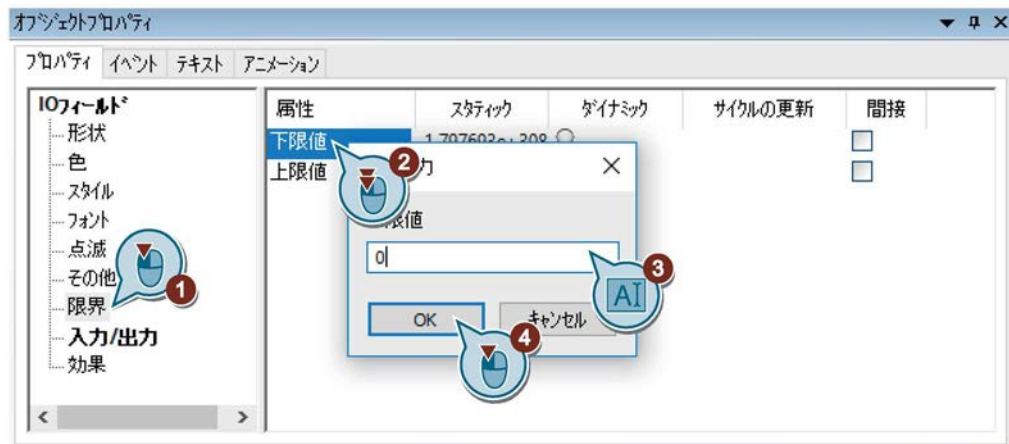
I/Oフィールドのショートカットメニューから"設定ダイアログ"を選択することで、"I/Oフィールド設定"ダイアログを再度開きます。

"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。

6.5 画面ダイナミックスの処理

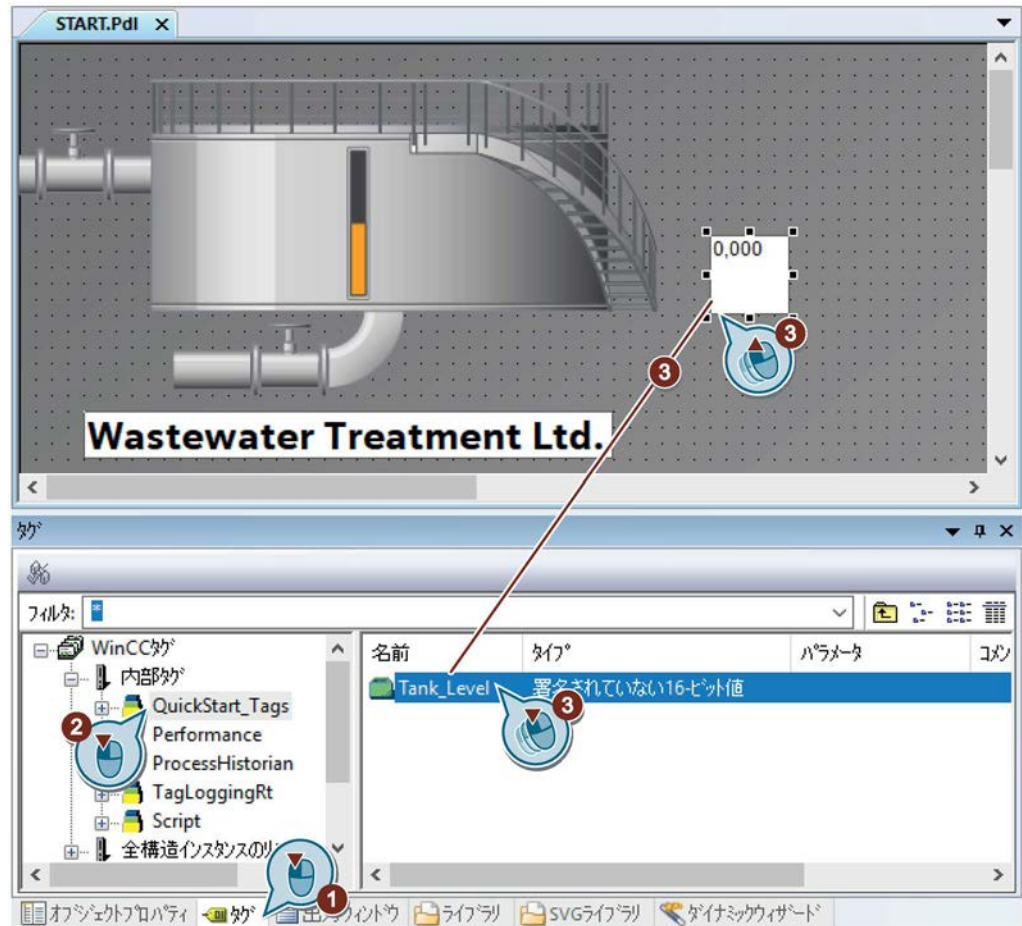
3. [制限]グループを[オブジェクトプロパティ]ウィンドウで開き、
[下限値]を0に設定します。



4. ステップ3と同様に、[上限値]を100に設定します。
5. プロセス画像"START.pdl"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。

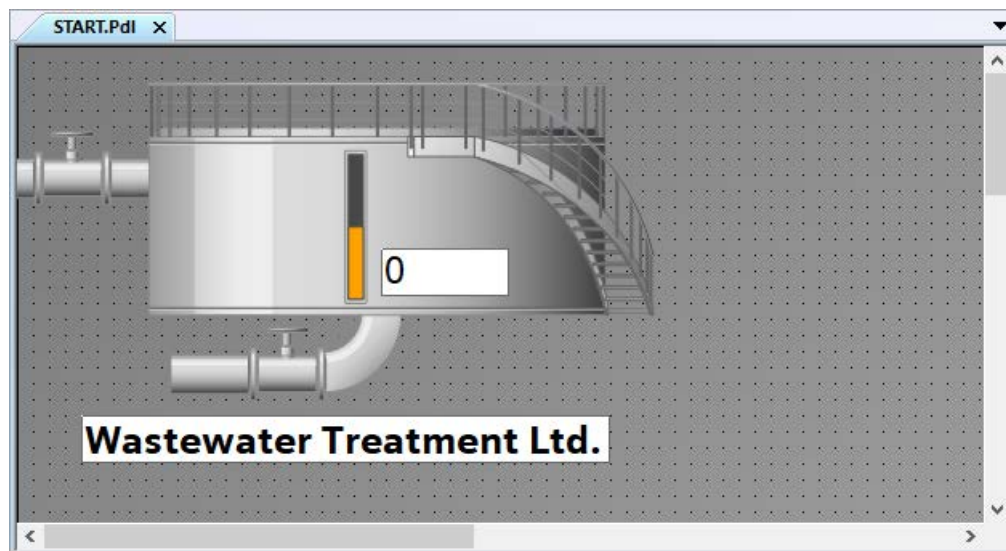
ランタイムにI/Oフィールドの値を入力すると、この値は内部タグによってバイオコンバータのグラフィック画像に転送されます。充填レベルインジケータが、入力された値に従って変化します。

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。

ランタイムの設定済み画像間の変更を有効にするため、次のステップでカスタマイズされたメニューを指定します。

"**START.pdl**"プロセス画像のダイナミック化を確認するために、WinCC Runtimeのプロパティを指定し、"**Quick_Start**"プロジェクトを起動します。



6.6 ユーザー定義メニューおよびツールバーの作成

6.6.1 カスタマイズされたメニューおよびツールバーの使用

はじめに

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"コンピュータのプロパティ"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。

カスタマイズされたメニューおよびツールバーは、次のように設定できます。

- オーソリゼーションの割り付け
ログインしたユーザーに必要な権限がない場合、この方法で設定したエレメントは自動的に無効になります。
- メニューコマンドとシンボルの非表示化または無効化
ランタイム中のユーザー変更など、新しい設定ファイルに変更されたファンクションのスコープを保存する場合、設定ファイルを交換することもできます。

6.6 ユーザー定義メニューおよびツールバーの作成

6.6.2 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

はじめに

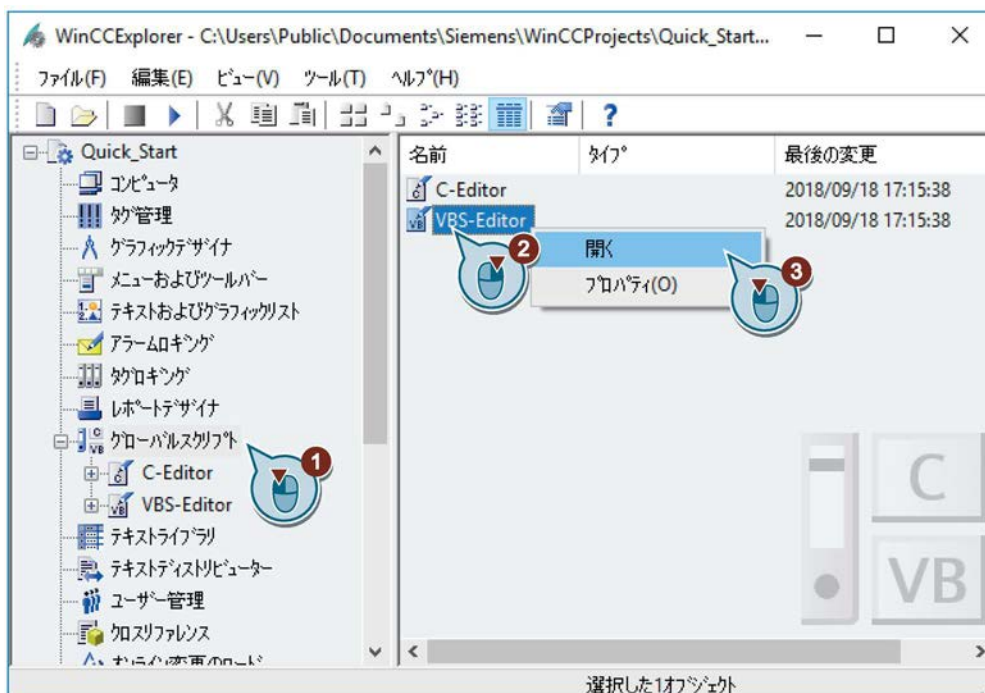
次のステップは、グローバルスクリプトのモジュールでのプロシージャの作成方法を示しています。

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。

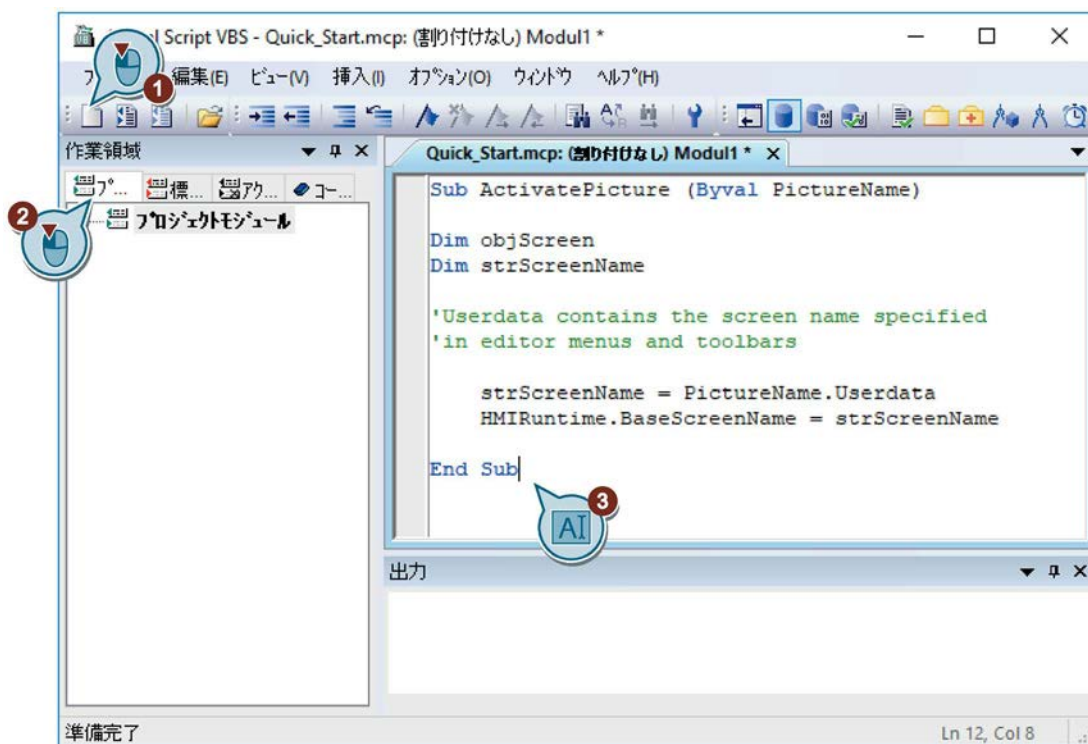
- **ActivatePicture(ByVal PictureName):**
"PictureName"パラメータと一緒に転送される画像に対して、画像の変更を実行します。
- **StopRuntime(ByVal Item):**
ランタイムを終了します。

手順

1. VBSエディタを開きます。

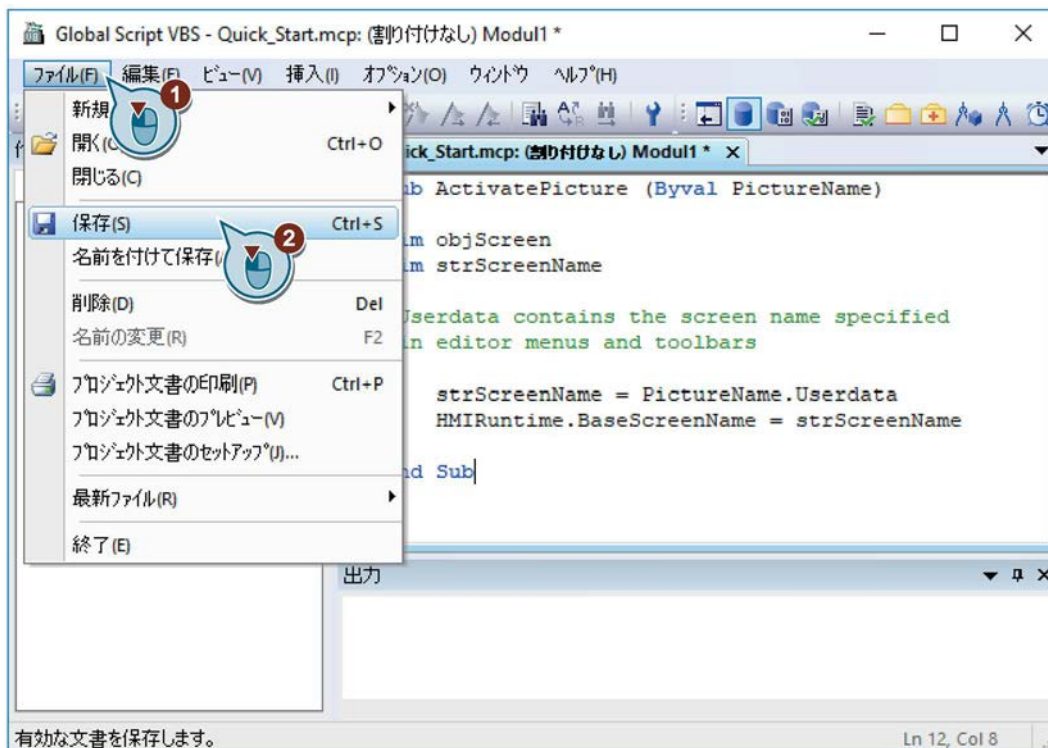


2. [プロジェクトモジュール]タブを選択し、以下のプロシージャコードを書き込みます。

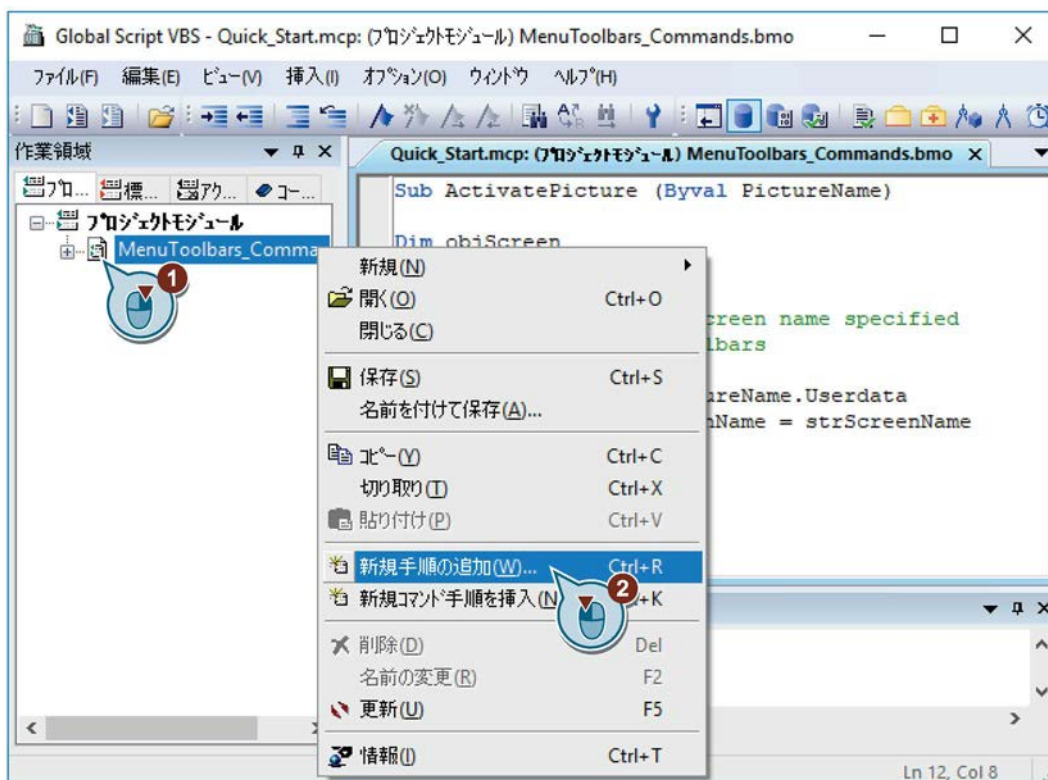


6.6 ユーザー定義メニューおよびツールバーの作成

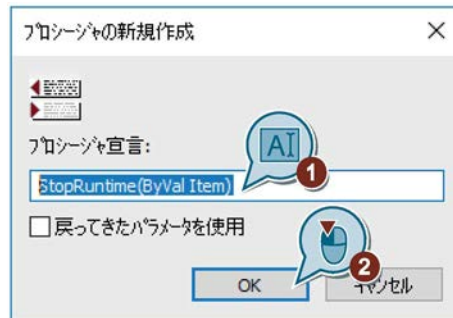
3. モジュールを"MenuToolbars_Commands.bmo"ファイルとして保存します。



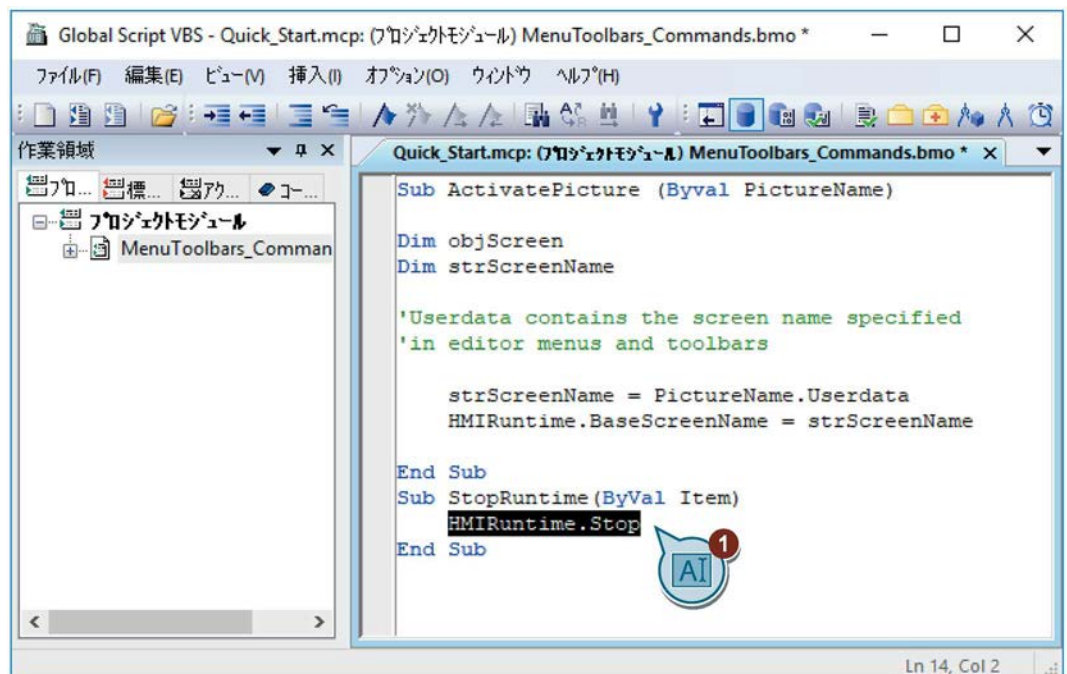
4. 新しいプロシージャを挿入します。



5. 名前"StopRuntime(ByVal項目)"を入力します。



6. 以下のプロシージャコードを書き込みます。



7. モジュールを保存します。

8. VBSエディタを閉じます。

結果

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

次に、画像"START.pdl"および"SAMPLE.pdl"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。

6.6 ユーザー定義メニューおよびツールバーの作成

6.6.3 画面変更のためのカスタマイズされたメニューの作成

はじめに

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。

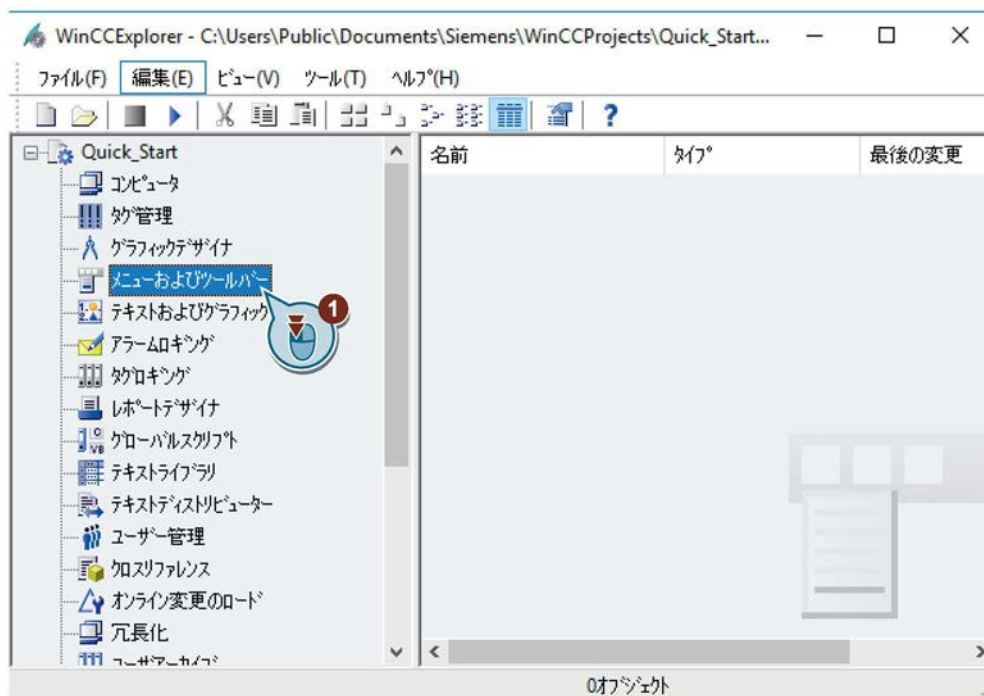
変更するプロセス画像の名前を、"ユーザーデータ"ボックスに入力します。

必要条件

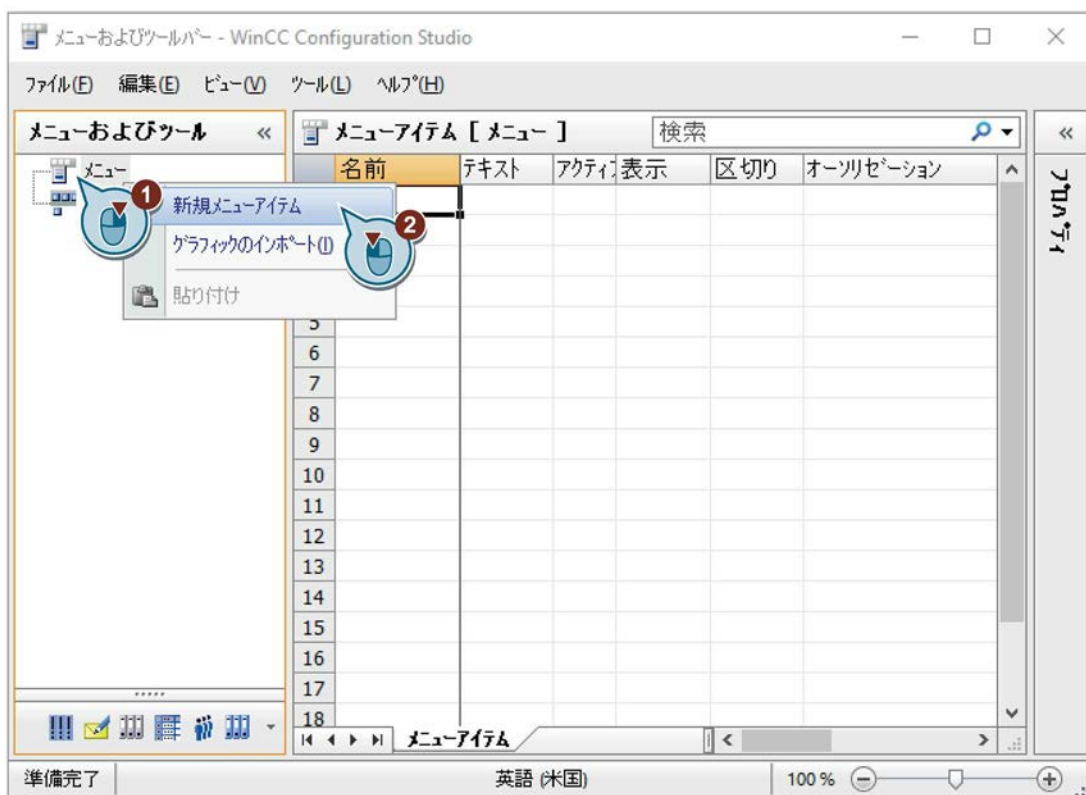
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。

手順

1. [メニューとツールバー]エディタを開きます。



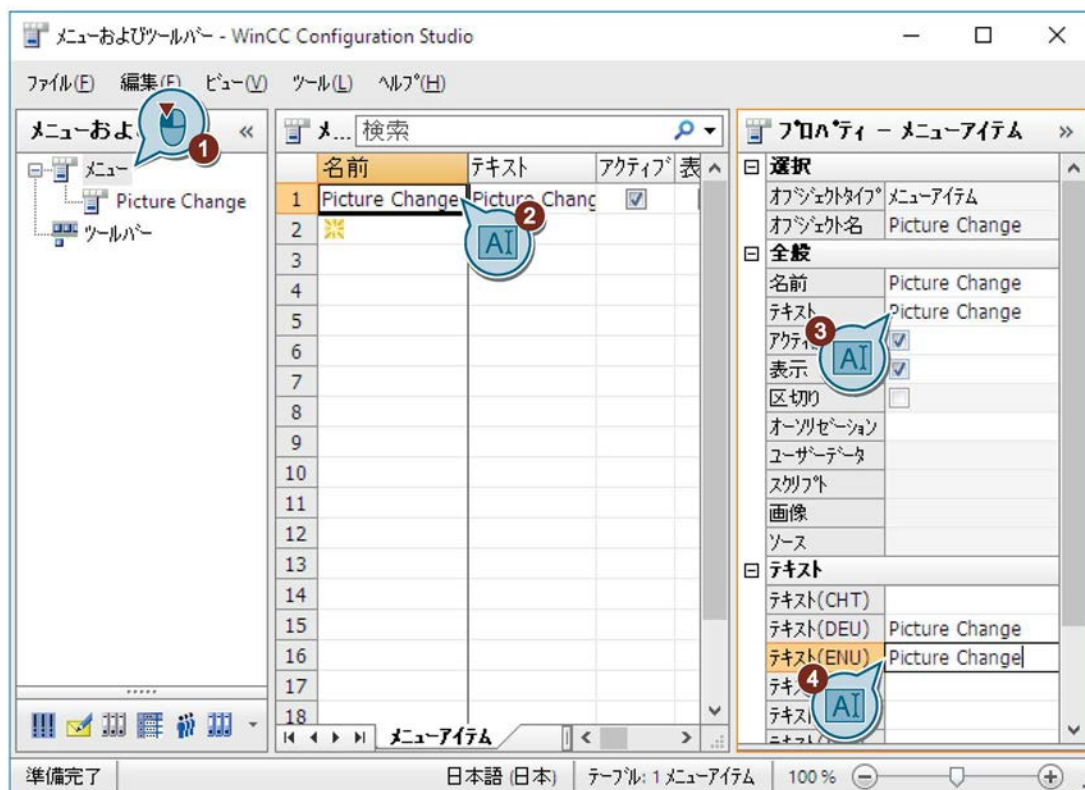
2. 新しいメニュー項目を作成します。



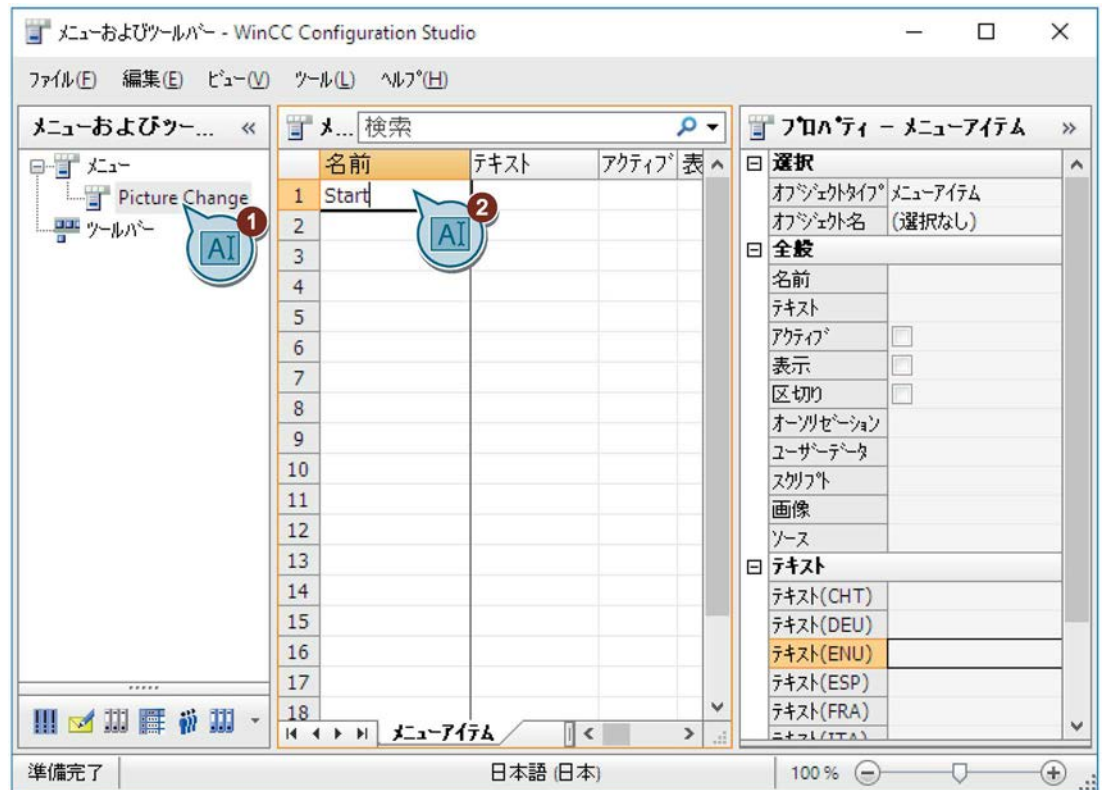
6.6 ユーザー定義メニューおよびツールバーの作成

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。

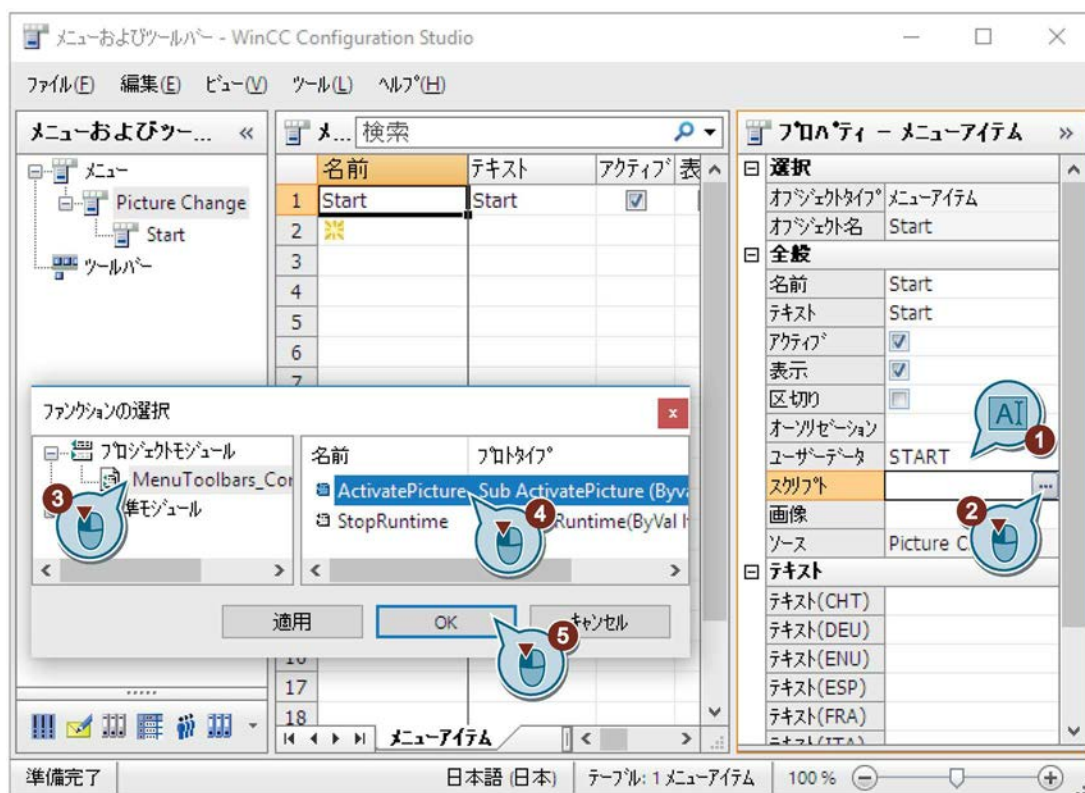


4. [スタート]メニューコマンドを作成します。



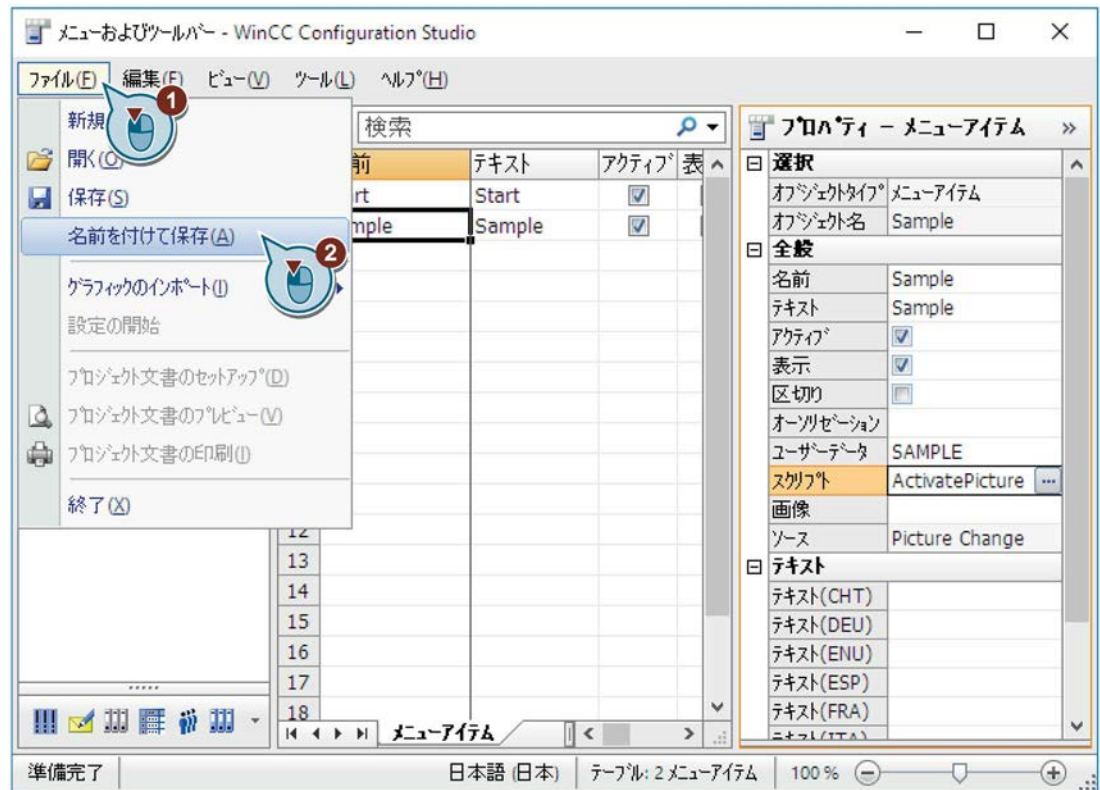
6.6 ユーザー定義メニューおよびツールバーの作成

5. "START.pdl"画像に変更するために、[スタート]メニューコマンドを設定します。



6.6 ユーザー定義メニューおよびツールバーの作成

6. 同様に、"SAMPLE.pdl"画像に変更するために、[サンプル]メニューコマンドを作成します。
7. 設定ファイルに"ActivatePicture"という名前を付けて保存します。

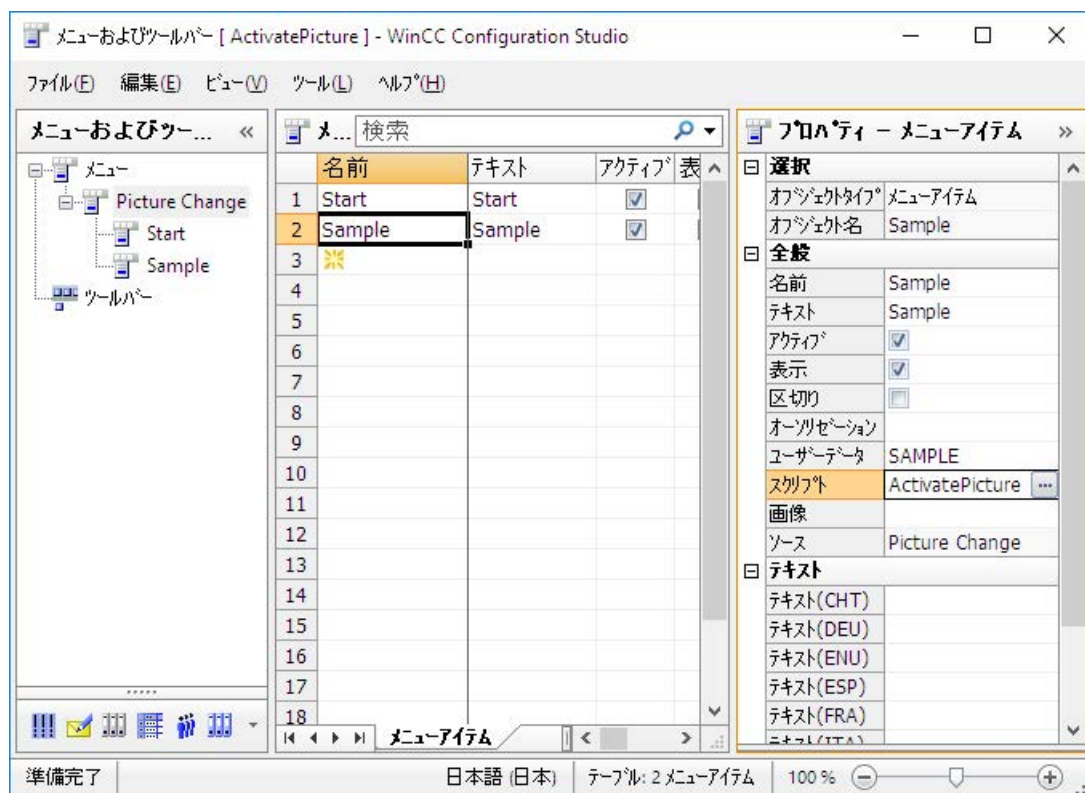


6.6 ユーザー定義メニューおよびツールバーの作成

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。



6.6.4 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

はじめに

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

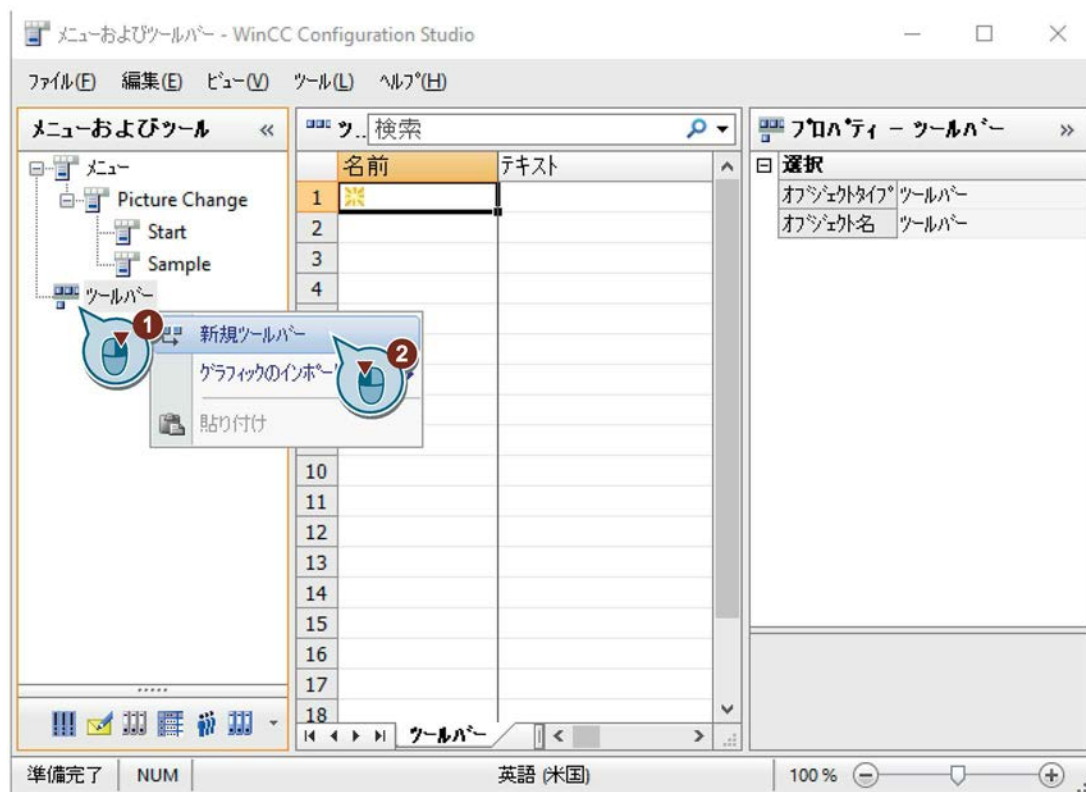
- [メニューとツールバー]エディタが開いていること。
- "StopRuntime(ByVal項目)"プロシージャが作成されていること。
- グラフィックファイルが"ランタイムの終了"のシンボルとしてWinCCプロジェクトフォルダにコピーされていること。例:

C:\Users\Public\Documents\Siemens\WinCCProjects\Quick_Start\GraMT\mtimages

6.6 ユーザー定義メニューおよびツールバーの作成

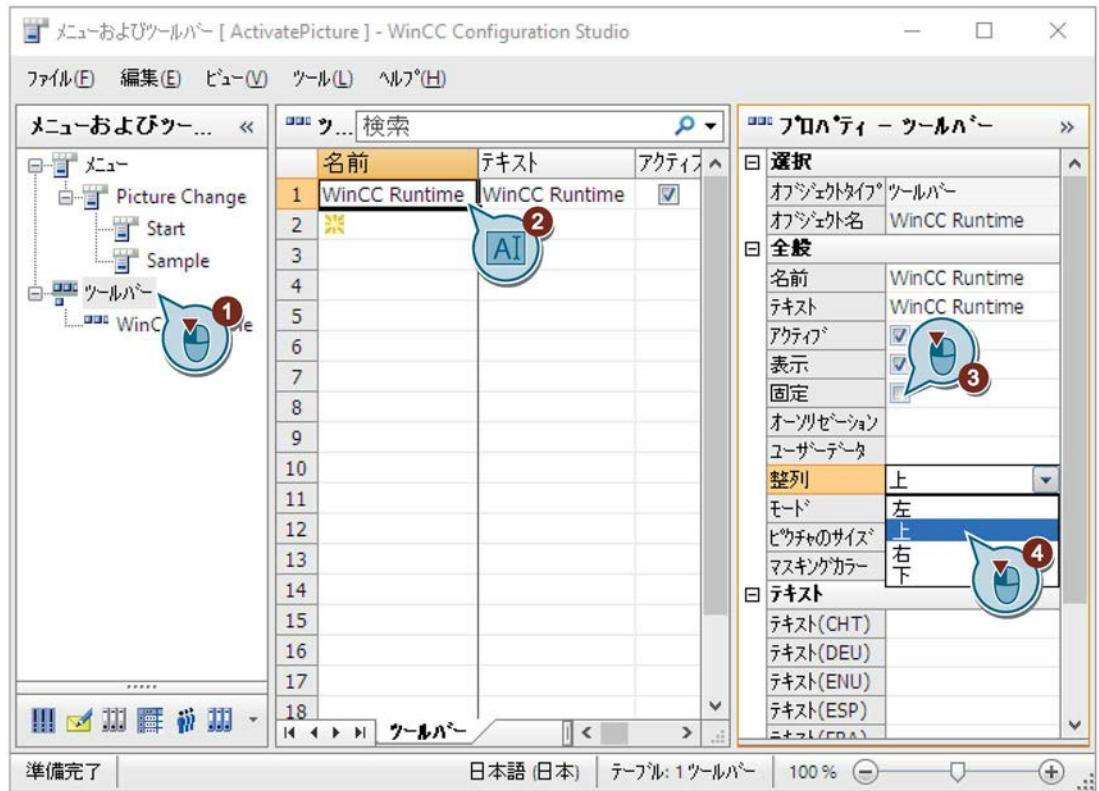
手順

1. 新しい"WinCC Runtime"ツールバーを作成します。



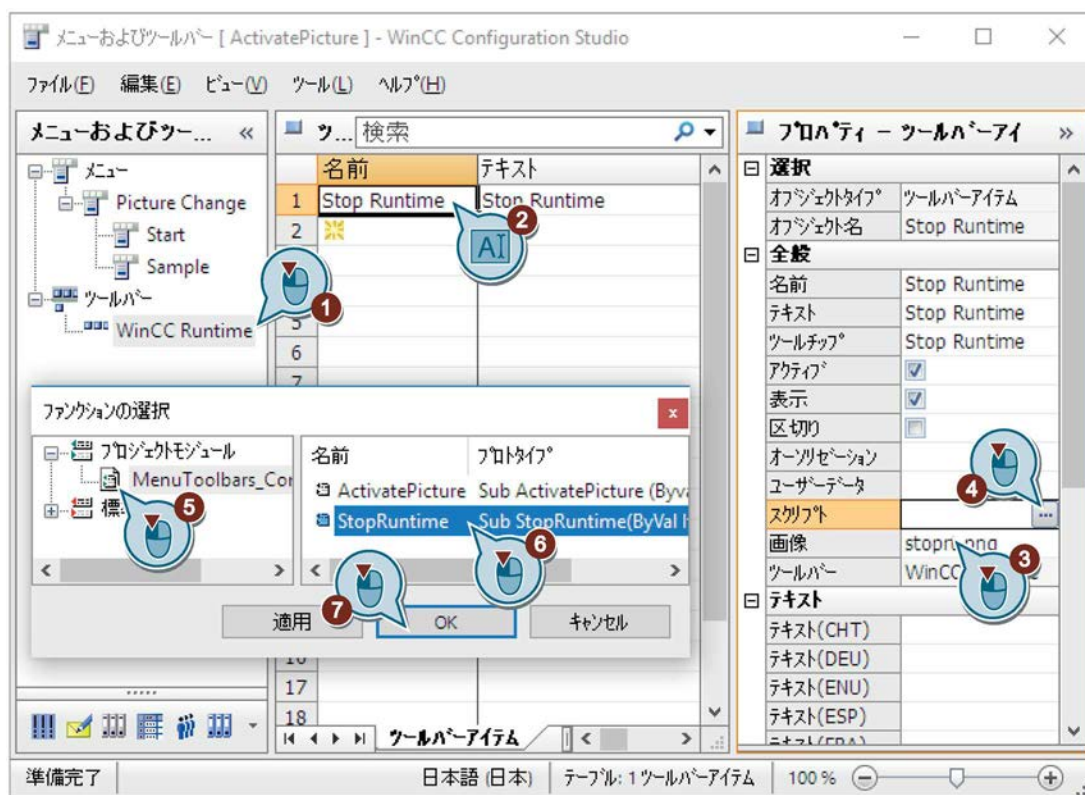
6.6 ユーザー定義メニューおよびツールバーの作成

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。

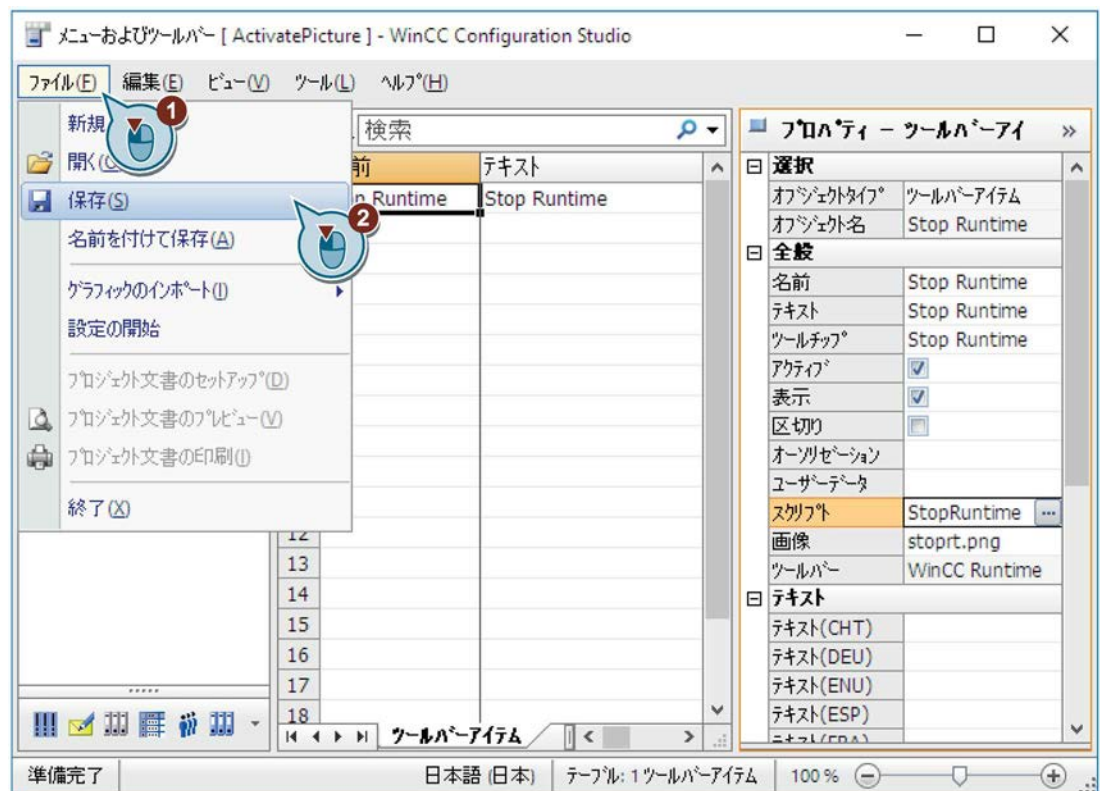


6.6 ユーザー定義メニューおよびツールバーの作成

3. [ランタイムの停止]エレメントを作成し、ランタイムを閉じるアイコンを設定します。



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を持つツールバーが作成され、設定が保存されます。

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を選択します。

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。

6.7 ランタイムプロパティの定義(グラフィックシステム)

はじめに

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エクスプローラで、WinCC Runtimeのプロパティを定義します。

このセクションでは、プロジェクトの起動時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設定します。

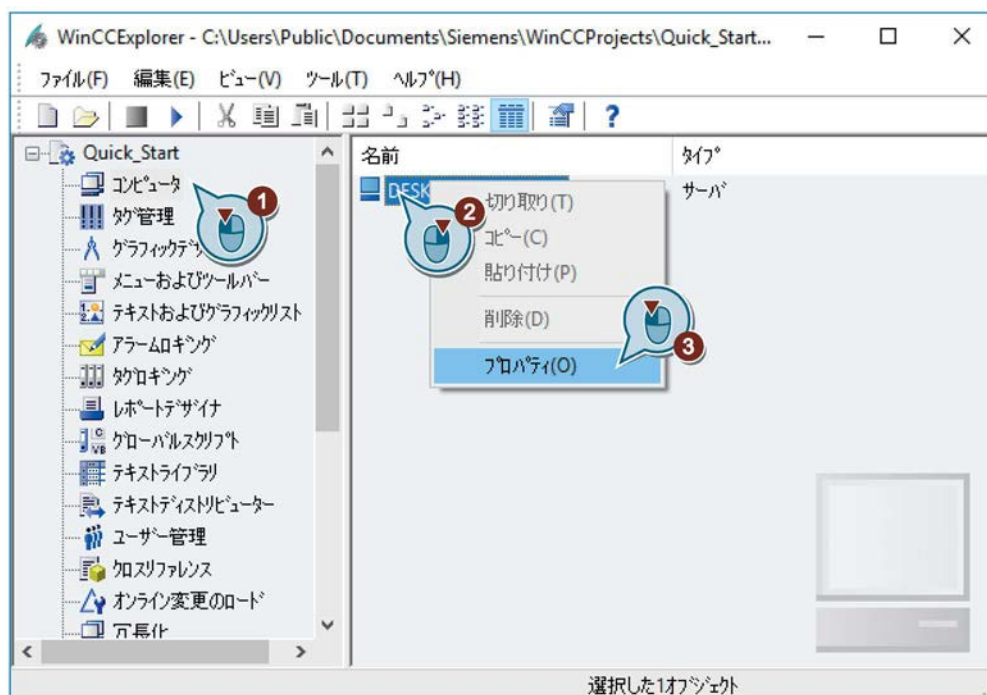
作成した設定ファイルをプロセス画像でユーザー定義メニューおよびツールバーを表示するために読み込みます。

必要条件

- "Quick_Start"プロジェクトが開いている。

手順

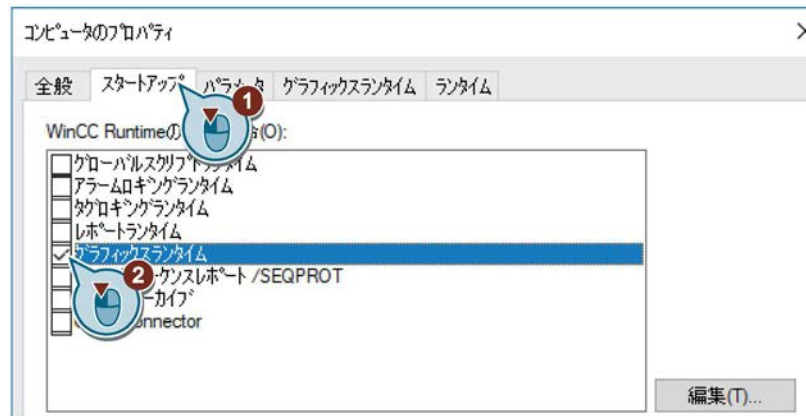
1. [コンピュータのプロパティ]ダイアログを開きます。



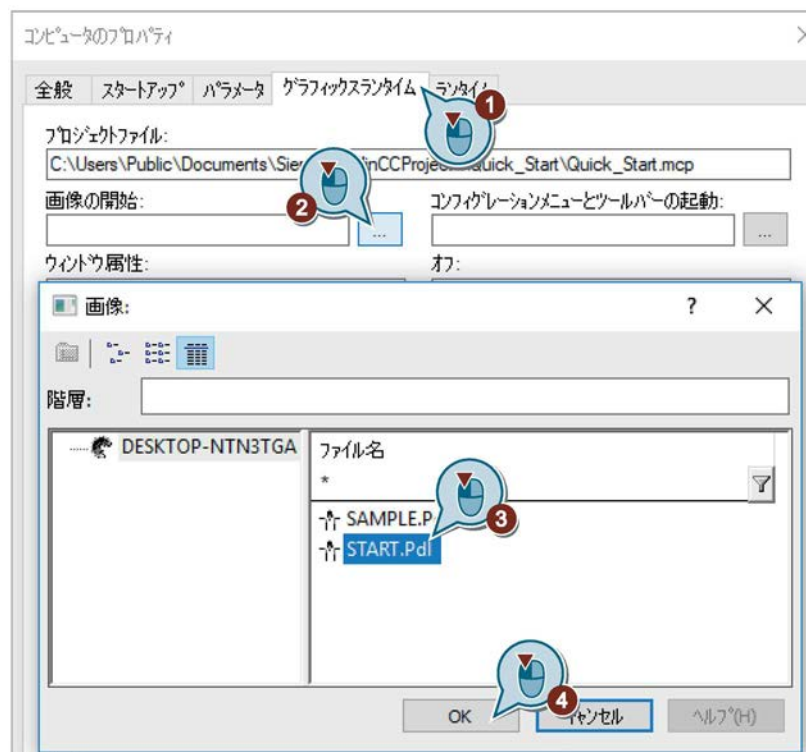
[コンピュータのプロパティ]ダイアログが開きます。

6.7 ランタイムプロパティの定義(グラフィックシステム)

2. "スタートアップ"タブをクリックして、"グラフィックスランタイム"アプリケーションを有効にします。



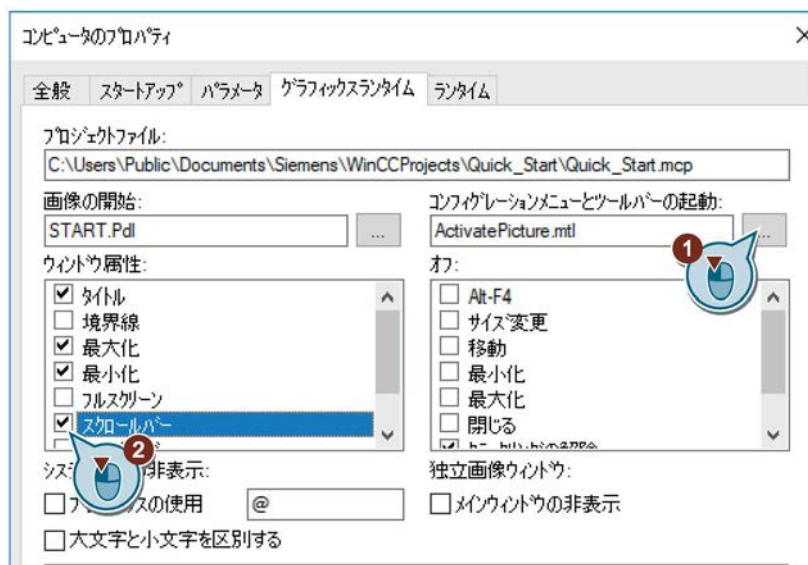
3. [グラフィックスランタイム]タブで、開始画像としてプロセス画像"START.pdl"を開きます。



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。

6.7 ランタイムプロパティの定義(グラフィックシステム)

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。



6. [OK]をクリックします。

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。

次のステップで、"Quick_Start"プロジェクトを有効にします。

6.8 プロジェクトの有効化(グラフィックシステム)

はじめに

次のステップは、"Quick_Start"プロジェクトを有効化する方法や、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。

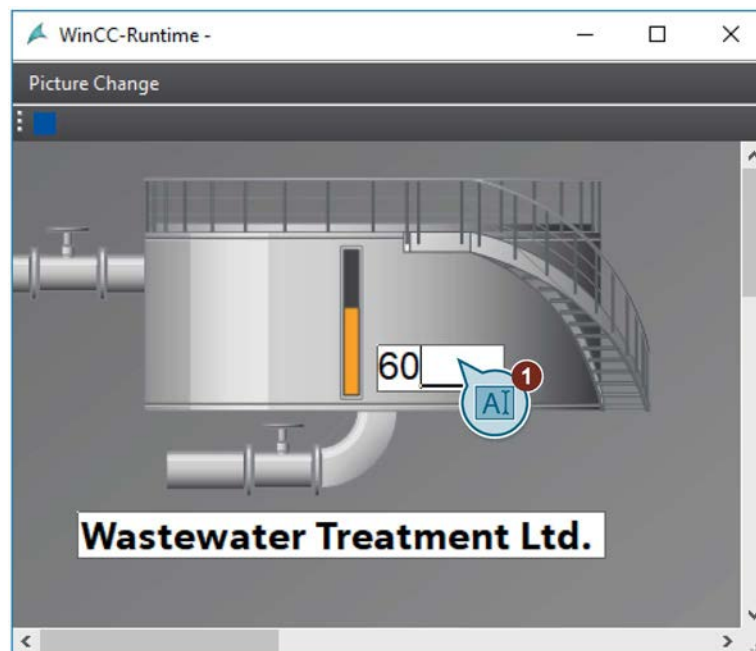
必要条件

- "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。

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。



6.8 プロジェクトの有効化(グラフィックシステム)

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]コマンドをクリックします。



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]コマンドをクリックします。

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。

ツールバーが「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 91)」セクションでシンボルと関連付けたグラフィックファイルを表示します。



結果

"Quick_Start"プロジェクトが有効になり、WinCC Runtimeが開始します。

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。

内部タグ[Tank_Level]をシミュレートして、"Quick_Start"プロジェクトをテストするために、次のステップで[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 91)

6.9 プロジェクトのテスト(グラフィックシステム)

はじめに

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。

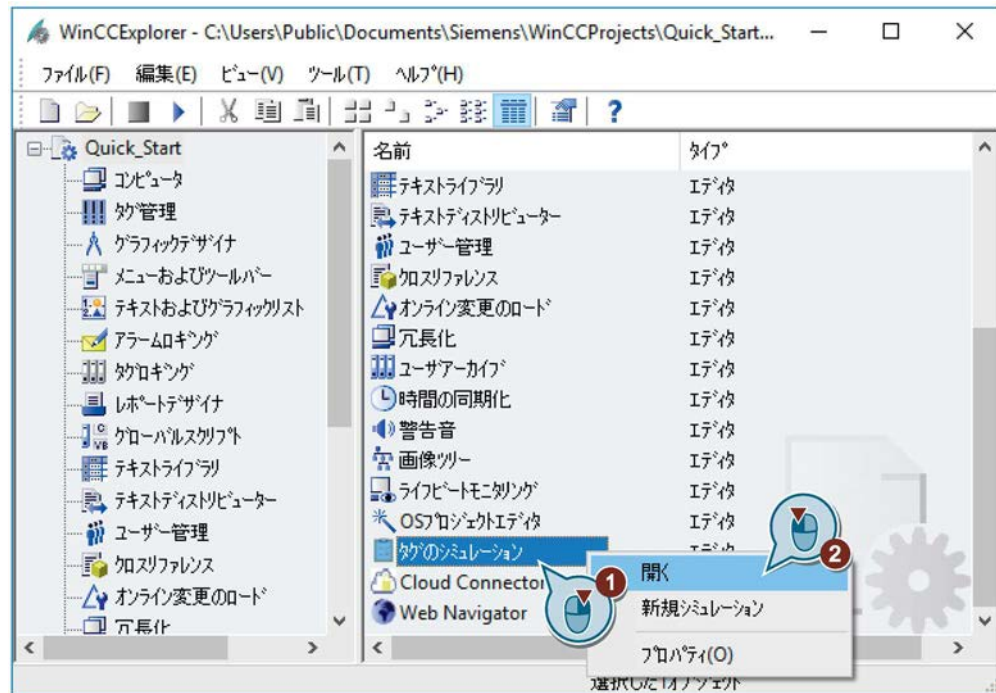
WinCCタグシミュレータは、内部タグ"Tank_Level"に異なる値を割り付けます。バイオコンバータのグラフィック描写に内部タグを接続すると、バイオコンバータの充填レベルインジケータはタグ値に従って変化します。

必要条件

- ランタイムプロパティが定義されている。

手順

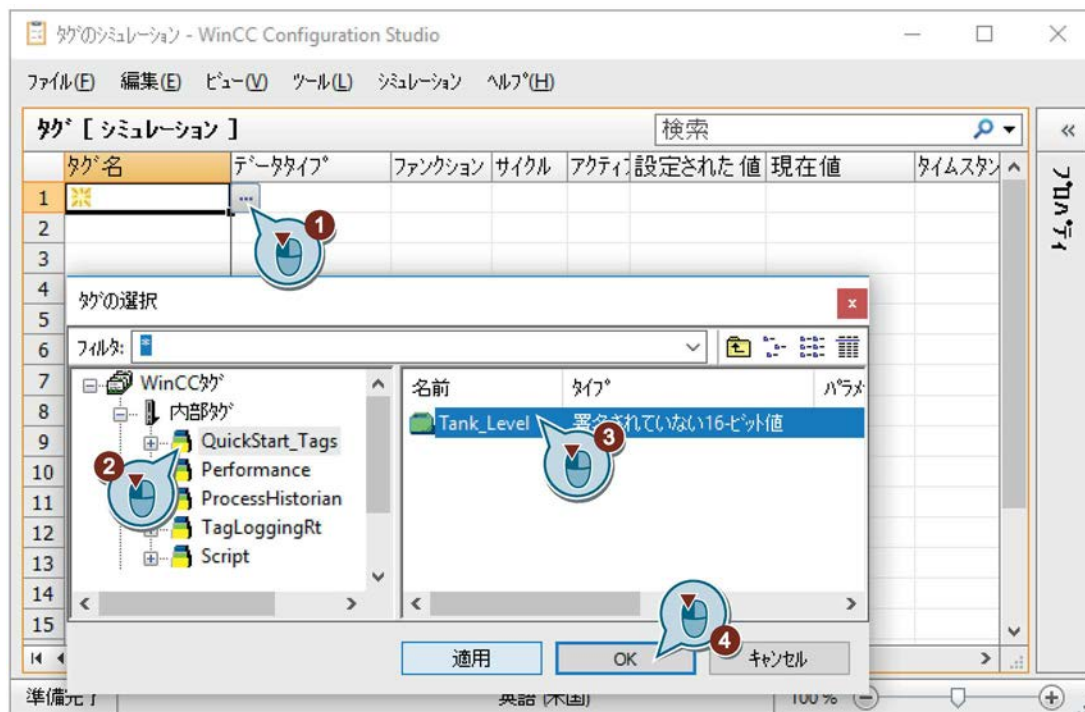
1. [タグシミュレーション]エディタを起動します。



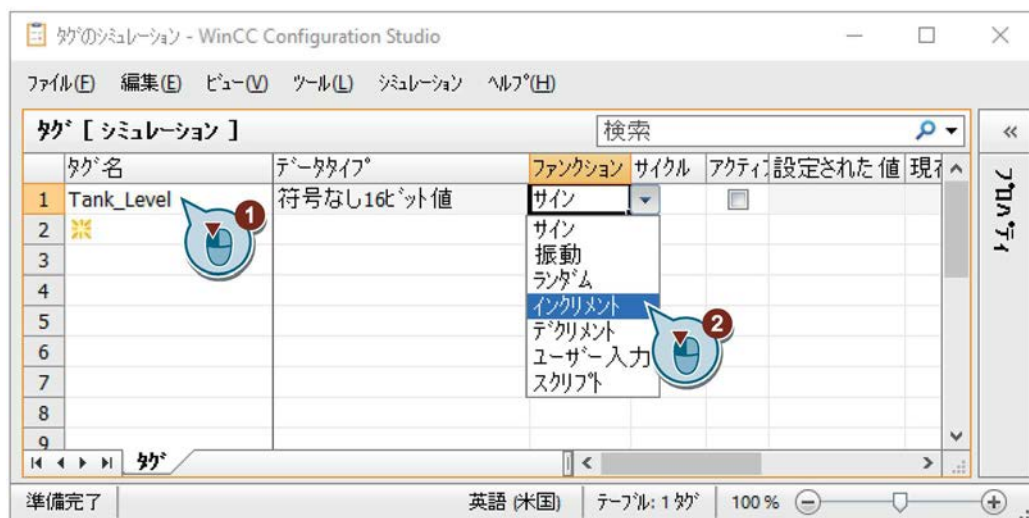
[タグシミュレーション]エディタが開きます。

6.9 プロジェクトのテスト(グラフィックシステム)

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。

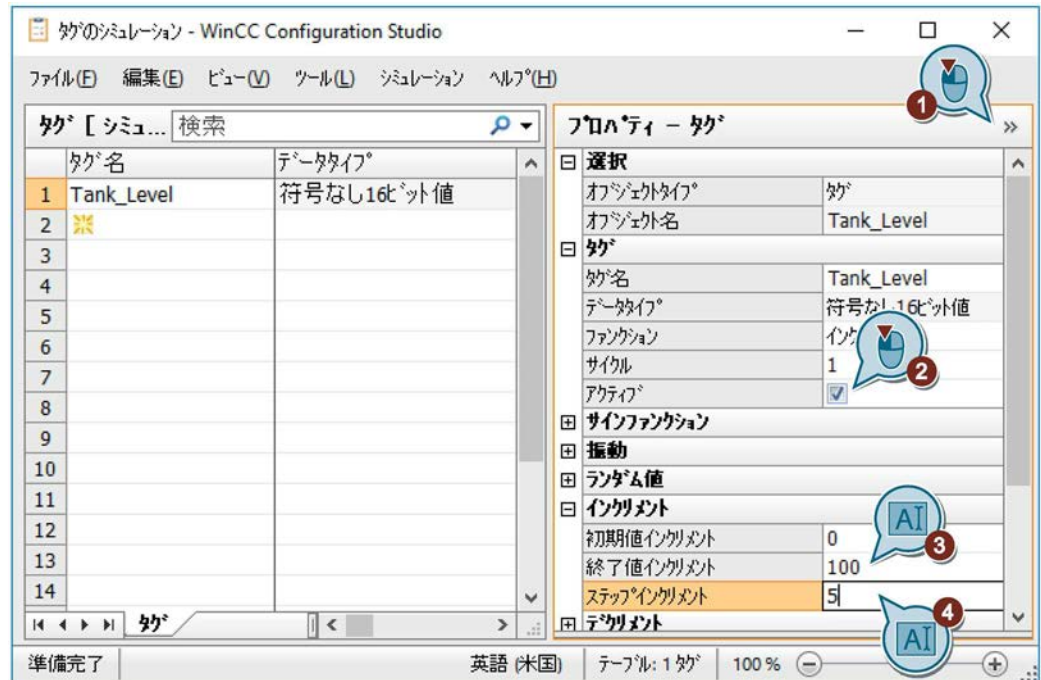


3. シミュレーションタイプを選択します。



6.9 プロジェクトのテスト(グラフィックシステム)

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。

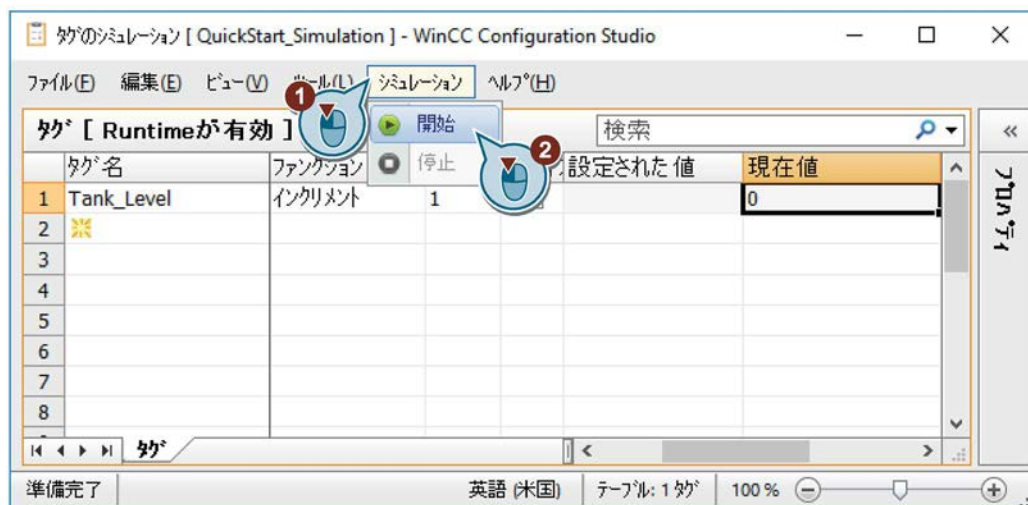


5. シミュレーションを"QuickStart_Simulation"として保存します。



6.9 プロジェクトのテスト(グラフィックシステム)

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。

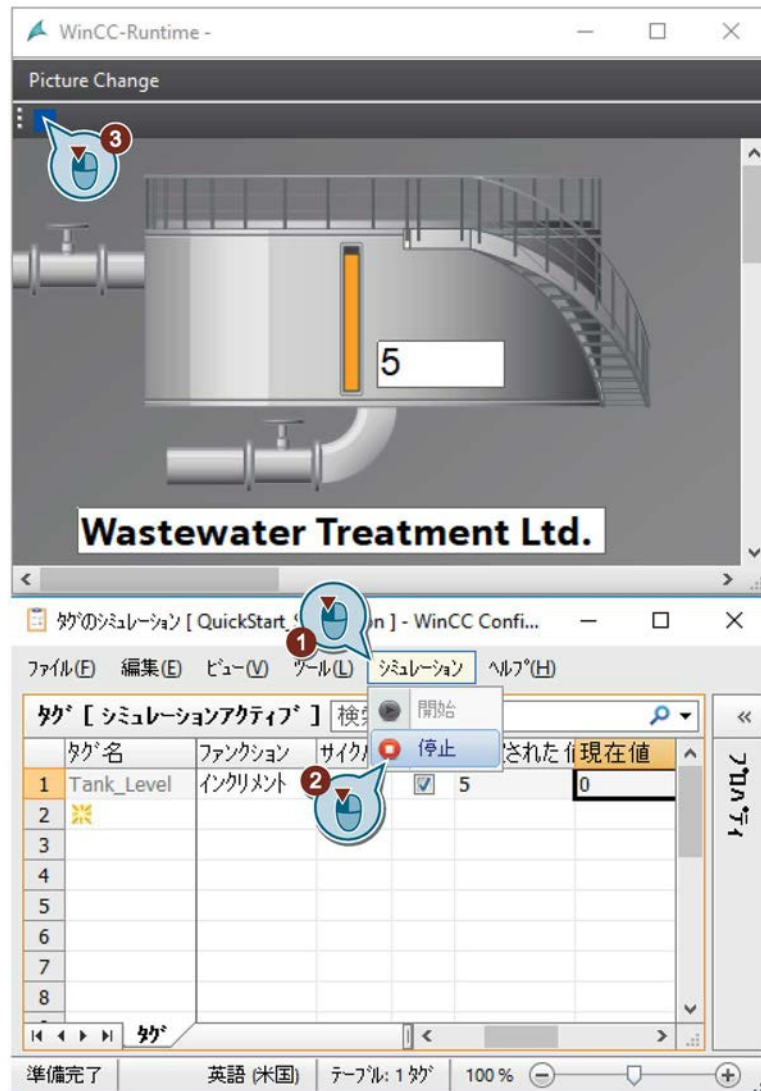


8. [ラベルシミュレーション]エディタとランタイムウィンドウを、両方を同時に監視できるように位置調整します。

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。

6.10 ランタイムシステムダイアログの使用

はじめに

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

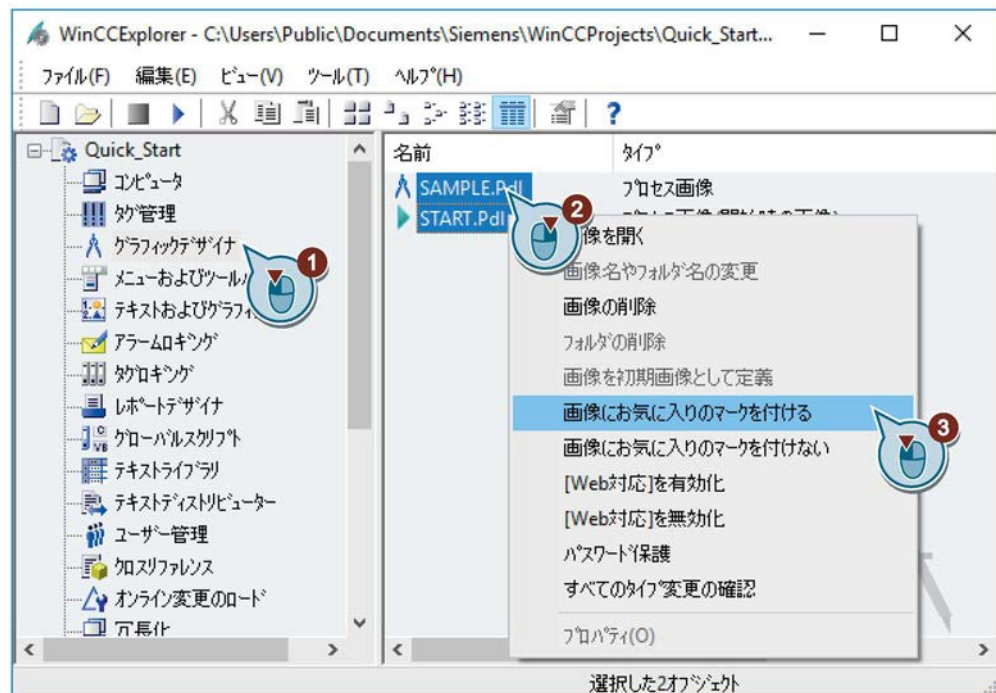
- "START.pdl"と"SAMPLE.pdl"プロセス画像が作成されている。
- "START.pdl"プロセス画像が開始画像として定義されている。

手順

1. WinCCエクスプローラで、"START.pdl"と"SAMPLE.pdl"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選択します。

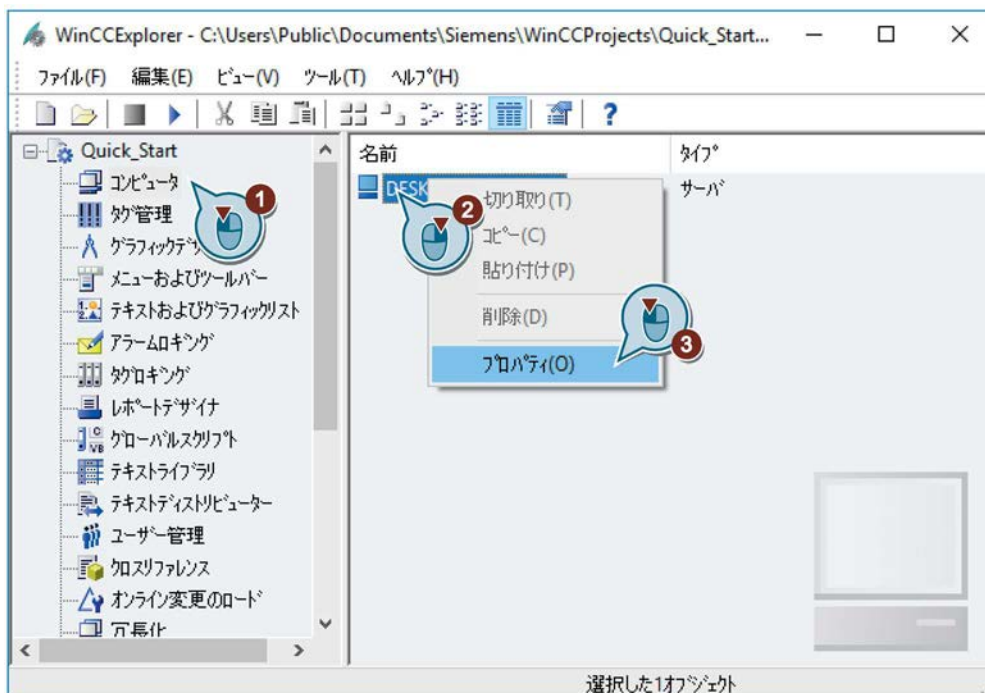


"START.pdl"と"SAMPLE.pdl"プロセス画像がWinCCエクスプローラの右ペインにお気に入りとして表示されます。

名前	タイプ
★ SAMPLE.Pdl	プロセス画像 (お気に入り)
★ START.Pdl	プロセス画像 (お気に入り、開始画像)

6.10 ランタイムシステムダイアログの使用

3. [コンピュータのプロパティ]ダイアログを開きます。



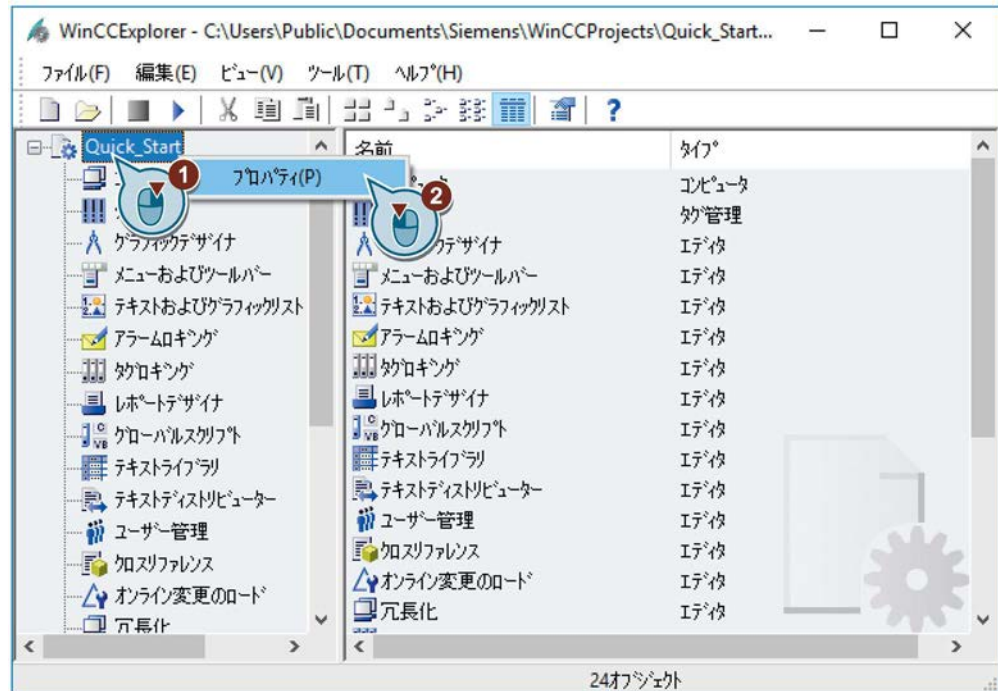
[コンピュータのプロパティ]ダイアログが開きます。

4. [ランタイム]タブをクリックし、[ランタイムシステムダイアログの有効化]チェックボックスを選択します。



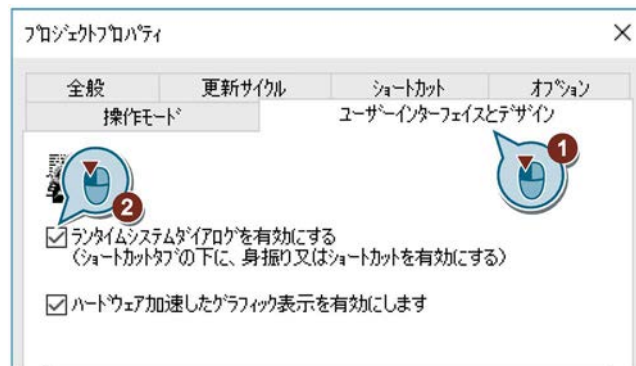
5. [OK]をクリックしてダイアログボックスを閉じます。

6. [プロジェクトプロパティ]ダイアログを開きます。



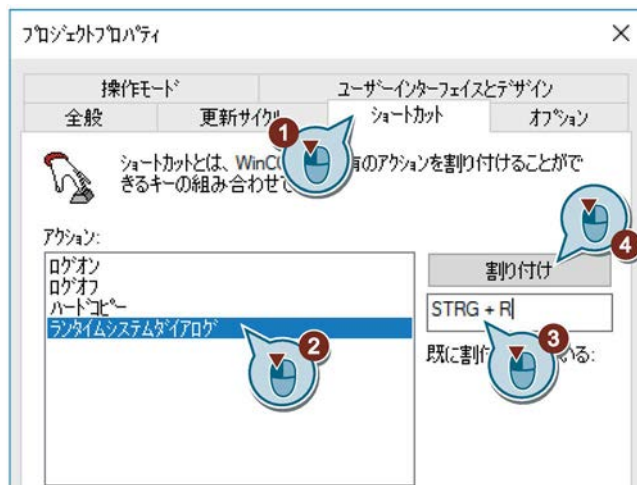
[プロジェクトプロパティ]ダイアログが開きます。

7. [ユーザーインターフェースおよびデザイン]タブをクリックし、[ランタイムシステムダイアログの有効化]チェックボックスを選択します。



6.10 ランタイムシステムダイアログの使用

8. [ショートカット]タブをクリックし、アクション"ランタイムシステムダイアログ"にキーボードショートカットを割り当てます(例、<Ctrl+R>)。

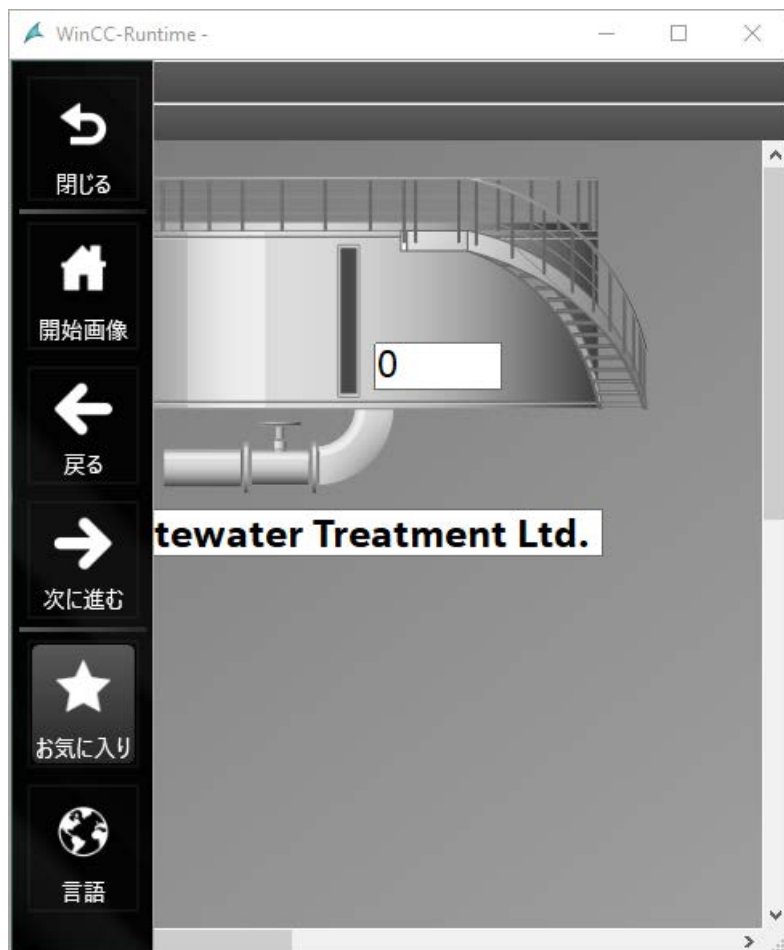


9. [OK]をクリックしてダイアログボックスを閉じます。
10. "Quick_Start"プロジェクトを有効にします。

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。

11. 割り当てられたショートカットキーを使ってランタイムシステムダイアログを起動します(例、<Ctrl+R>)。

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。



6.10 ランタイムシステムダイアログの使用

12. [お気に入り]および[SAMPLE]プロセス画像をクリックします。



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。

値のアーカイブと表示

7.1 値のアーカイブと表示

はじめに

この章では、アーカイブシステムに関する情報やプロセス値アーカイブで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"プロジェクトでは、プロセス値は必要ありません。

このプロジェクトで、内部タグ[Tank_Level]を選択します。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化を監視します。

7.2 アーカイブシステム

はじめに

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、プロセス画像および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、以下の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。

- 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。
 - プロセス値アーカイブと圧縮アーカイブの設定
 - 取得サイクルとアーカイブサイクルの定義
 - 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

アーカイブ

プロセス値は、アーカイブデータベースのハードディスク、またはタグロギングランタイムのメインメモリに保存できます。

データ量を削減するために、既にアーカイブされたプロセス値を圧縮できます。

アーカイブ期間

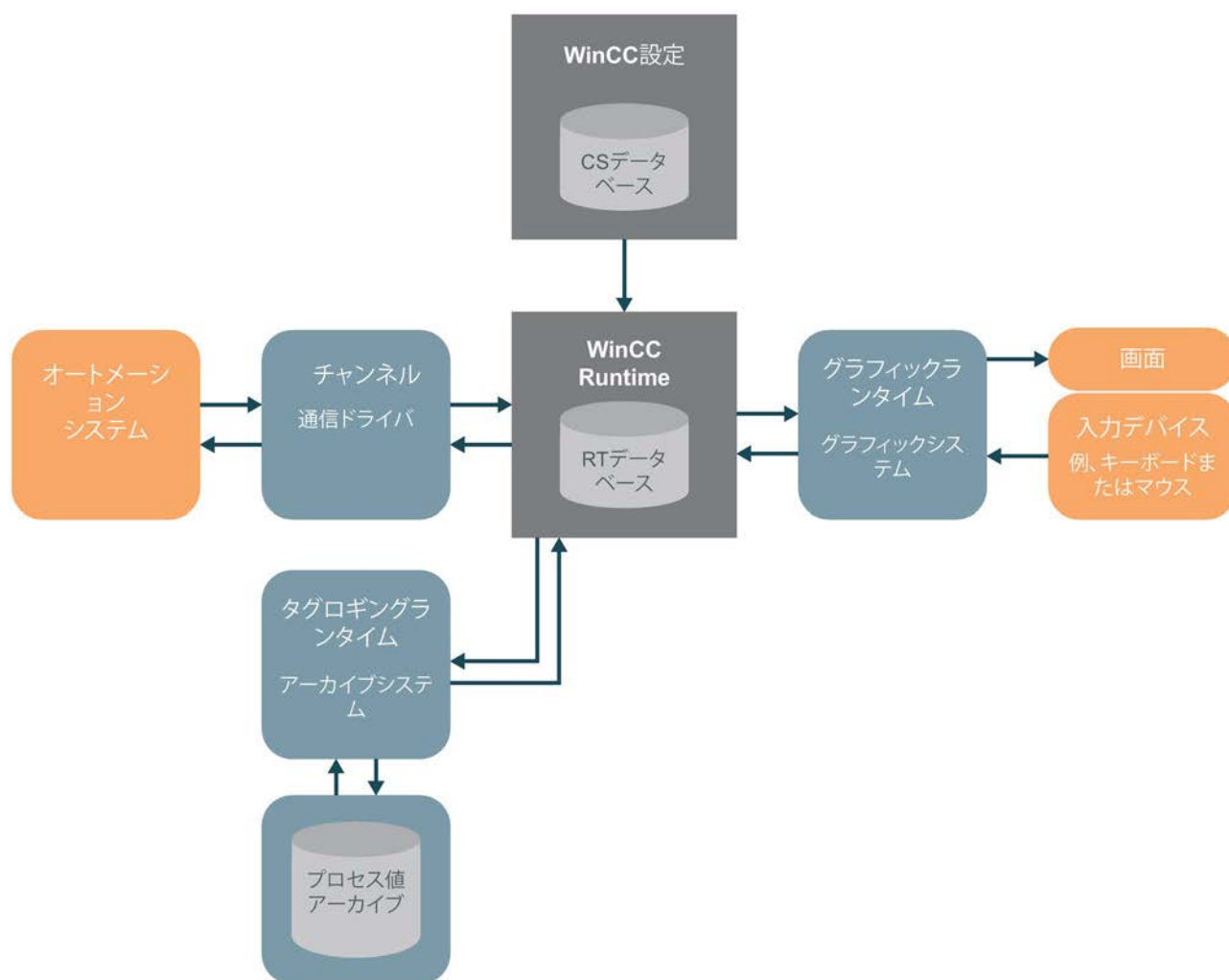
アーカイブサイクルとイベントを使用すると、アーカイブ時間をコントロールできます。

例えば、プロセス値のアーカイブを、一定のタイムサイクルで実施、またはプロセス値が特定の量または特定の割合で変化した場合のみ実施することができます。

ライセンシング

WinCC基本システムで、追加のライセンスなしで512のアーカイブ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



7.3 タグロギングの開始

はじめに

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。

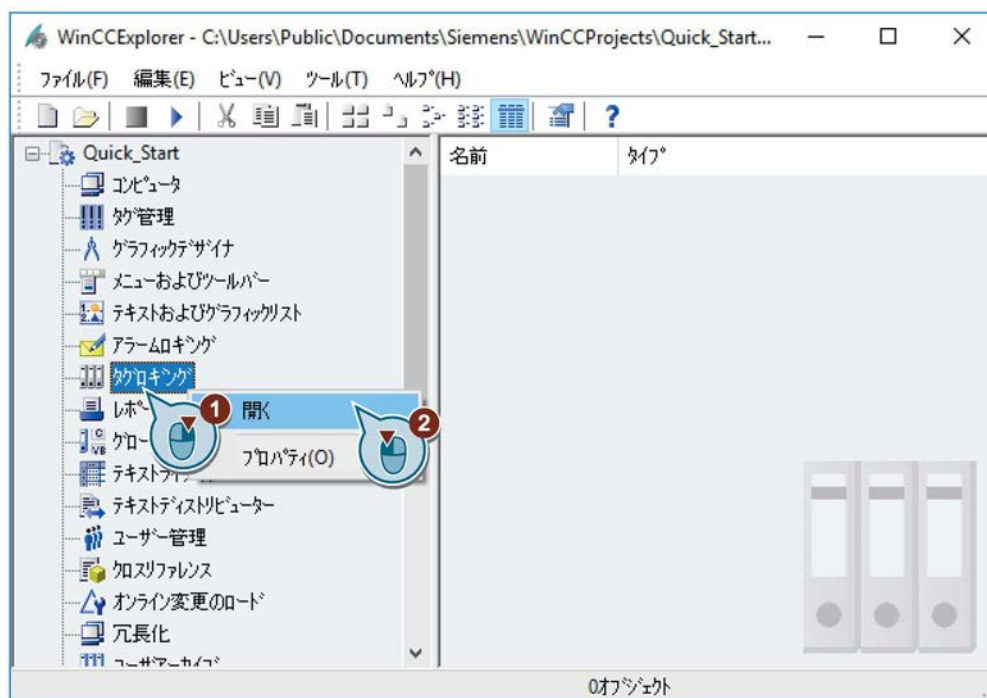
"タグロギング"エディタで、取得サイクルやアーカイブサイクルの時間と共に、プロセス値アーカイブを設定します。

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。



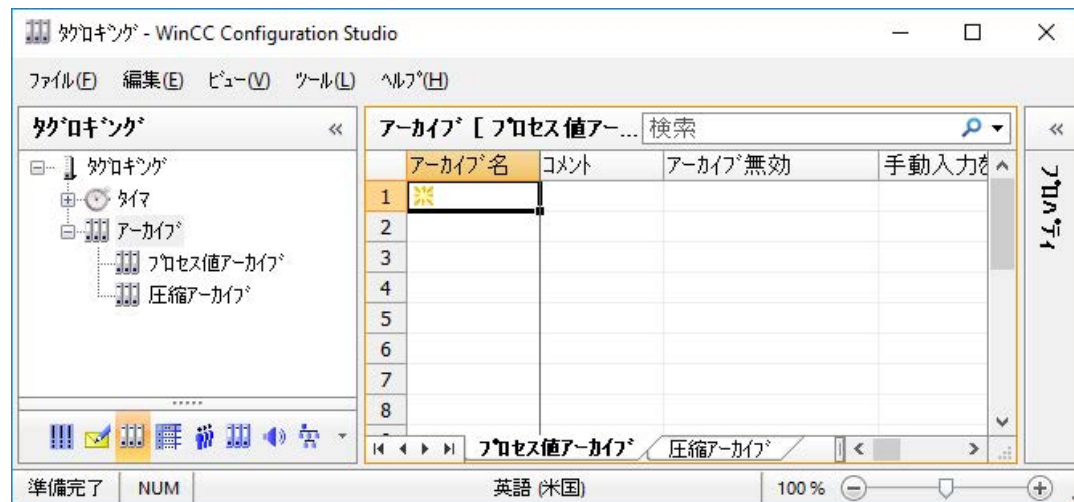
"タグロギング"エディタが開きます。

結果

"タグロギング"エディタが開いています。

次のステップでは、このエディタを使って、取得やアーカイブサイクル時間を設定します。設定した時間に基づき、タグ値の取得とアーカイブの時間間隔を定義します。

プロセス値アーカイブをエディタで作成できます。



7.4 タイマの設定

はじめに

"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、基本時間を1秒に、時間係数を5秒に設定すると、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。

内部タグをアーカイブするために、次のステップでアーカイブを作成します。

7.5 プロセス値アーカイブの作成

はじめに

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

- [タグロギング]エディタが開いている。
- [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。

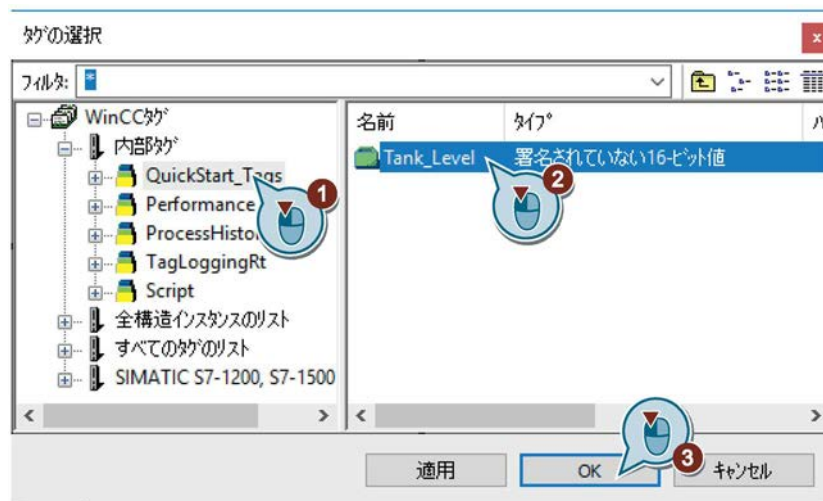
テーブルエリアで[タグ]タブを選択します。

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。

7.5 プロセス値アーカイブの作成

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。

次のステップで、新しいプロセス値アーカイブを編集します。



7.6 プロセス値アーカイブの編集

はじめに

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。

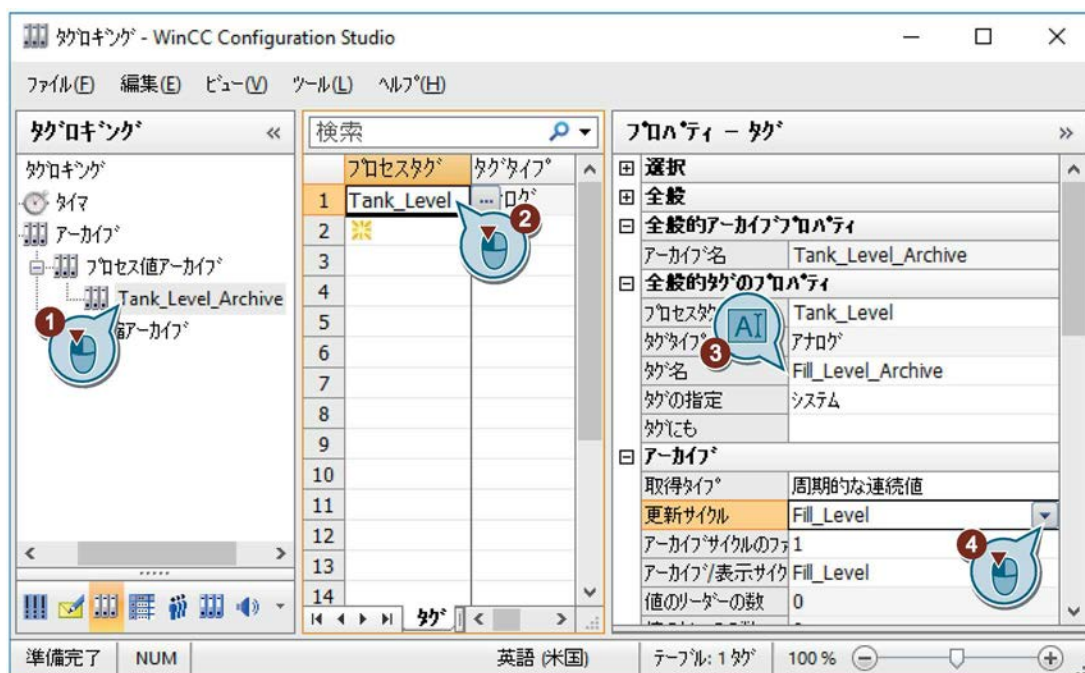
必要条件

- [タグロギング]エディタが開いている。
- プロセス値アーカイブ"**Tank_Level_Archive**"が作成されている。

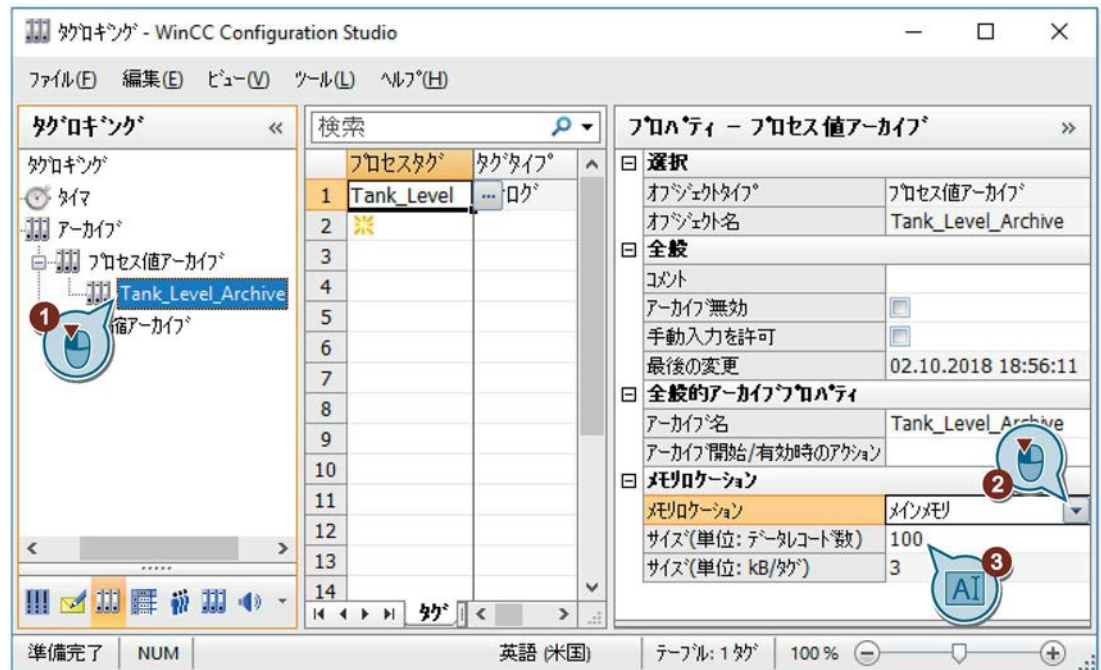
7.6 プロセス値アーカイブの編集

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。
必要の場合は、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。
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。

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、次のステップで"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。

7.7 プロセス画面の設定

7.7.1 プロセス画像の設定(アーカイブシステム)

はじめに

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。

7.7.2 [トレンド]ウィンドウの設定

はじめに

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。

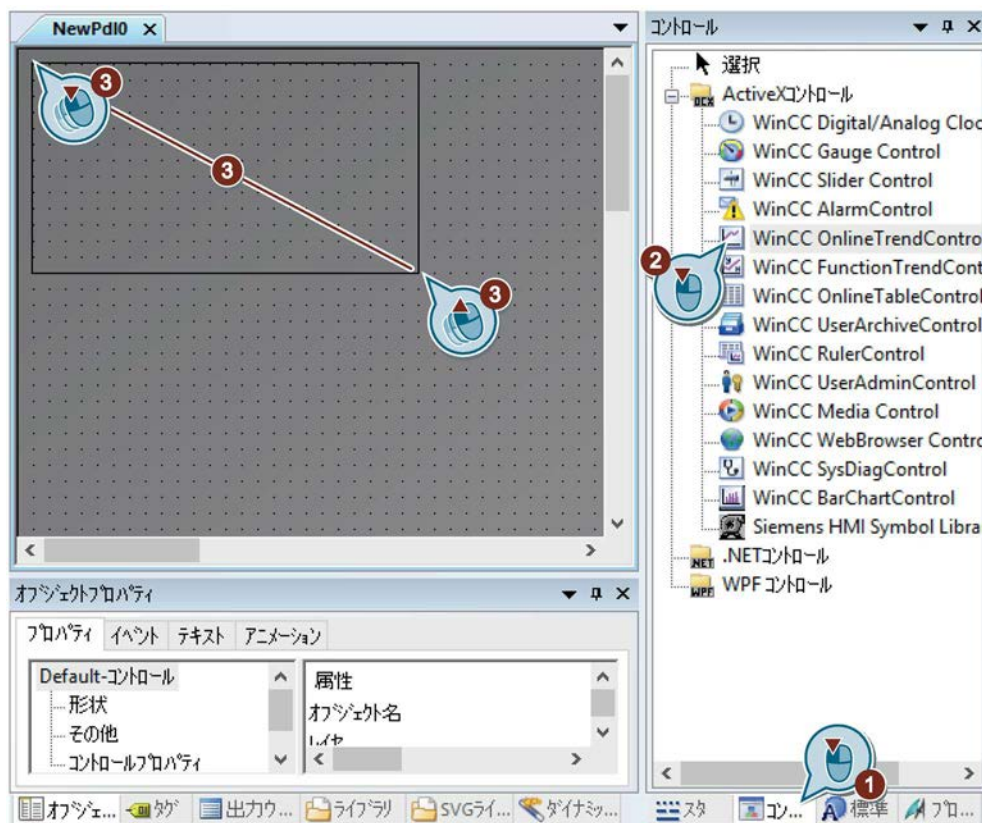
これで、ランタイムでアーカイブタグに保存された値がトレンドとして出力されます。

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. [制御]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。

3. [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	エクスポート
トレンド	全般	トレンドウィンドウ	時間軸

時間軸:

☒ 時間軸 1

新規作成 削除 上へ 下へ

オブジェクト:

時間軸 1

トレンドウィンドウ:

トレンドウィンドウ 1

ラベル:

配置:

0 - 下揃え

☒ 最新の情報の更新

時間範囲

設定:

2 - 測定点の数

測定点の数: 120

開始時間:

19.10.2018 14:37:42

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

日付の形式:

自動

☒ 日付の表示

色:

☐ トレンド色を使用

5. 値軸の設定を定義します。

ツールバー		ステータスバー		オンラインエンジニアリング		エクスポート	
トレンド		全般		トレンドウィンドウ		時間軸	
値軸: ☒ 値軸 1				オブジェクト名: 値軸 1			
新規作成 削除 上へ 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			
				スケーリング: 0 - 線形			
値範囲 開始: 0 終了: 10				☒ 自動			
効果 小数点以下の桁数: 2 ☒ 自動 ☐ 指数表記				色: [黒色] ☐ トレンド色を使用			
ユーザースケーリング ☐ 使用				エリア名 ☐ 表示			

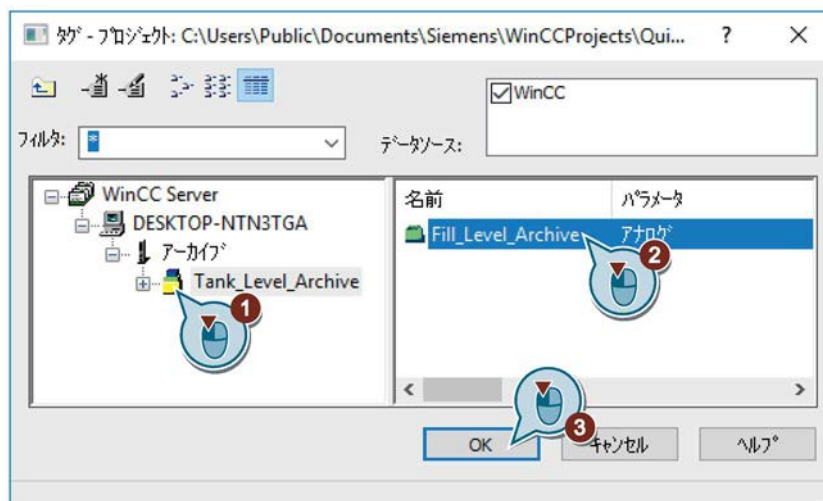
6. トレンド名前"Tank_Level"を入力し、をクリックします。



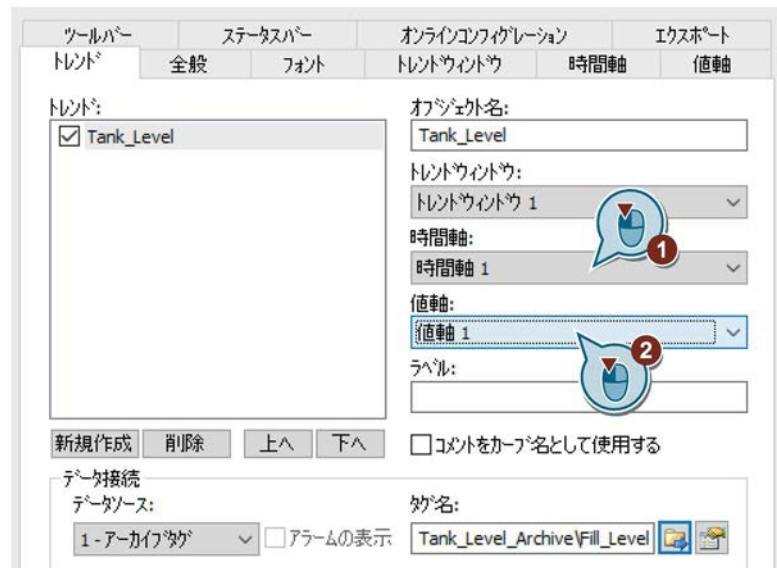
タグの選択用ダイアログが開きます。

7. [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。



8. 時間軸と値軸を選択します。



9. [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。

10. プロセス画像"TagLogging.pdl"を保存します。

別の手順

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. [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。
3. 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。
4. 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。

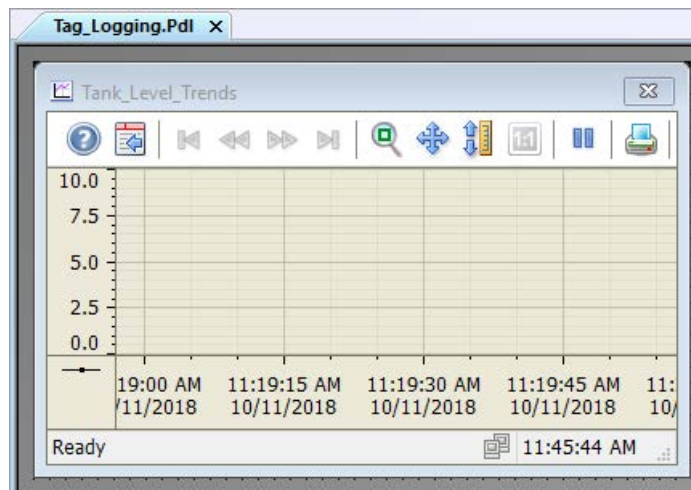
作成されたコントロールには、選択した[Fill_Level_Archive]アーカイブタグのデータ
接続のあるトレンドが含まれています。
5. 上記手順で説明したように、ステップ3から5および8に進みます。

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。

アーカイブタグに保存された値をランタイムのテーブルとして出力するには、次のステップで"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。



7.7.3 [テーブル]ウィンドウの設定

はじめに

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、プロセス画像"Tag_Logging.pdl"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。

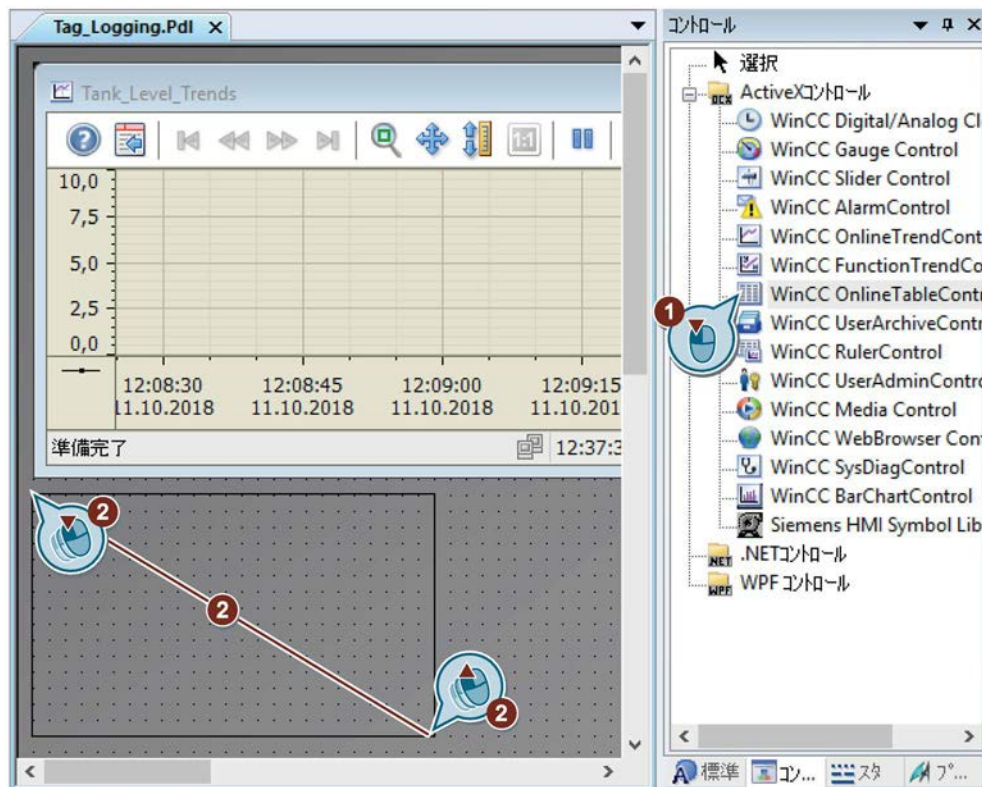
これで、ランタイムでアーカイブタグに保存された値がテーブルとして出力されます。

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。
- プロセス画像"Tag_Logging.pdl"が開いている。

手順

1. "Tag_Logging.pdl"プロセス画像に、[WinCC OnlineTableControl]オブジェクトを挿入します。



"グラフィックデザイナ"エディタの作業エリアに、
[テーブル]ウィンドウが表示されます。

[WinCC OnlineTableControlのプロパティ]ダイアログが開きます。

2. [テーブル]ウィンドウに"Tank_Level_Tables"という名前を入力します。

The screenshot shows the configuration window for a table in SIMATIC HMI WinCC. The 'Text' field is highlighted with a red circle and the number 2, indicating where to enter the table name 'Tank_Level_Tables'. Other settings like 'Units', 'Precision', and 'Edit' are also visible.

時間列	値列	オペレータメッセージ	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	ツールバー	ステータスバー

ウィンドウ
単位:
1 - 通常

テキスト:
Tank_Level_Tables

☒ 移動可能
☒ クローズ可能
☒ サイズ変更可能

スタイル:
プロジェクト外設定

行スクロールバー:
1 - 必要時

列スクロールバー:
1 - 必要時

精度
係数: 0 x 時間単位: 同一

ピクチャを開く
☒ 更新開始
☒ アーカイブからのデータのロード

列の色の使用
☒ フォント色
☐ 背景色

編集
☒ 編集を許可する

現在の印刷ジョブを表示
OnlineTableControl - Table

時間ベース
2 - プロジェクト外設定

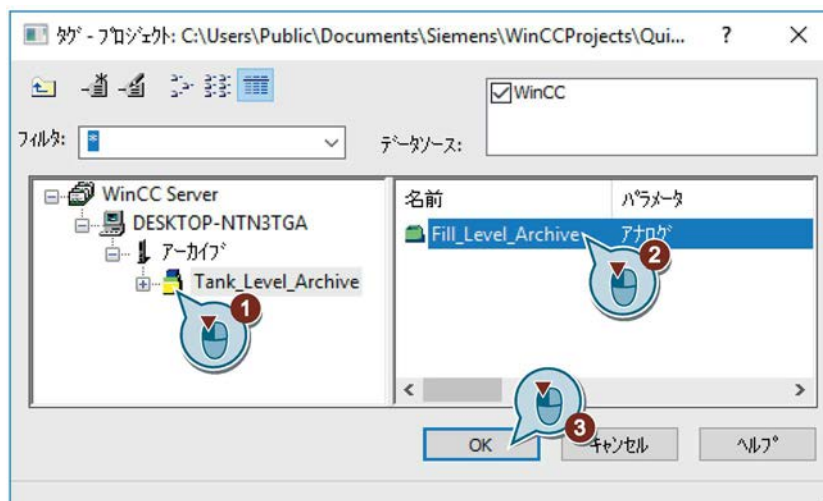
3. [値]列に"Tank_Level"という名前を入力し、をクリックします。



アーカイブタグ選択のダイアログが開きます。

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。



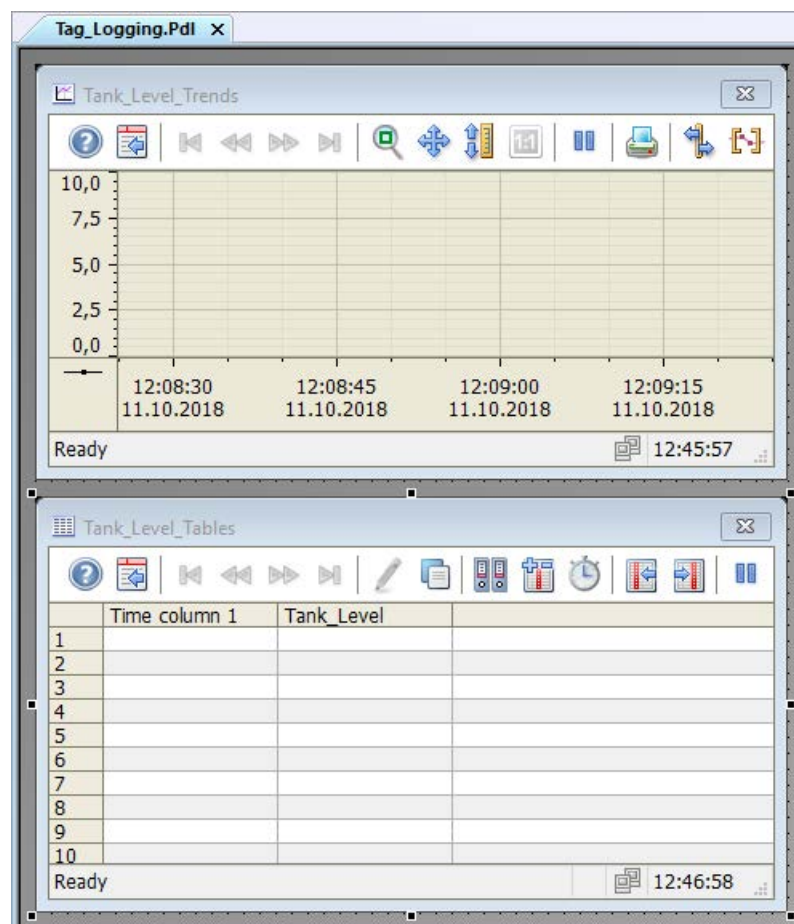
5. [OK]をクリックして、[WinCC OnlineTableControlのプロパティ]ダイアログを閉じます。
6. プロセス画像"Tag_Logging.pdl"を保存します。
7. グラフィックデザイナーを閉じます。

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。

ランタイムで出力値を表示するために、次のステップでWinCC Runtimeのプロパティを定義します。



7.8 ユーザー定義メニューのカスタマイズ(アーカイブシステム)

はじめに

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。

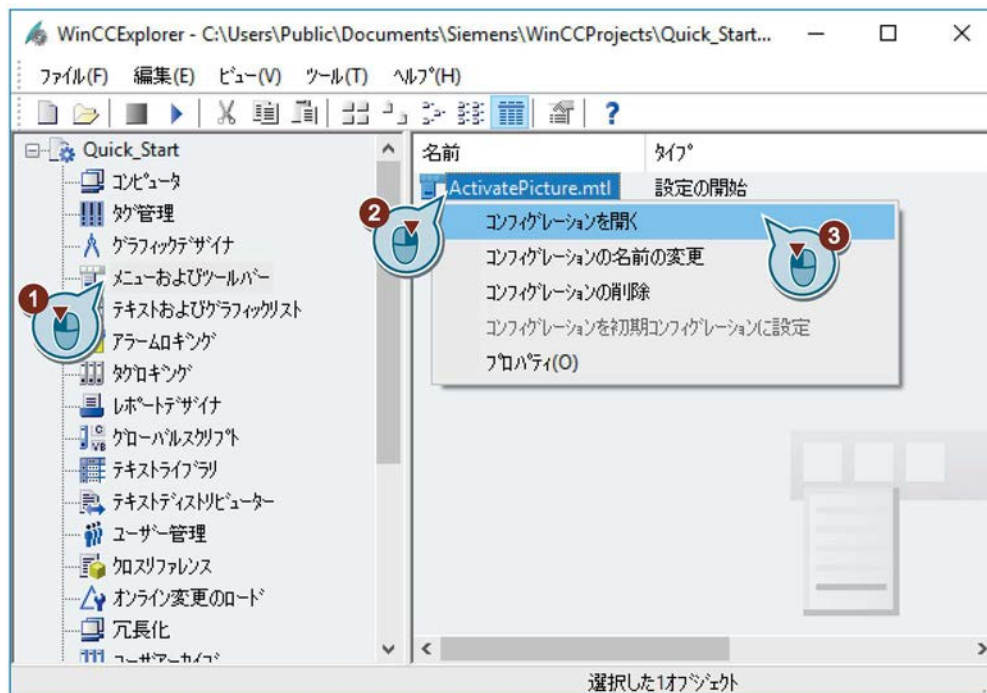
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- カスタマイズされた[画像変更]メニューがプロセス画像"START.pdl"および"SAMPLE.pdl"用に作成されていること。

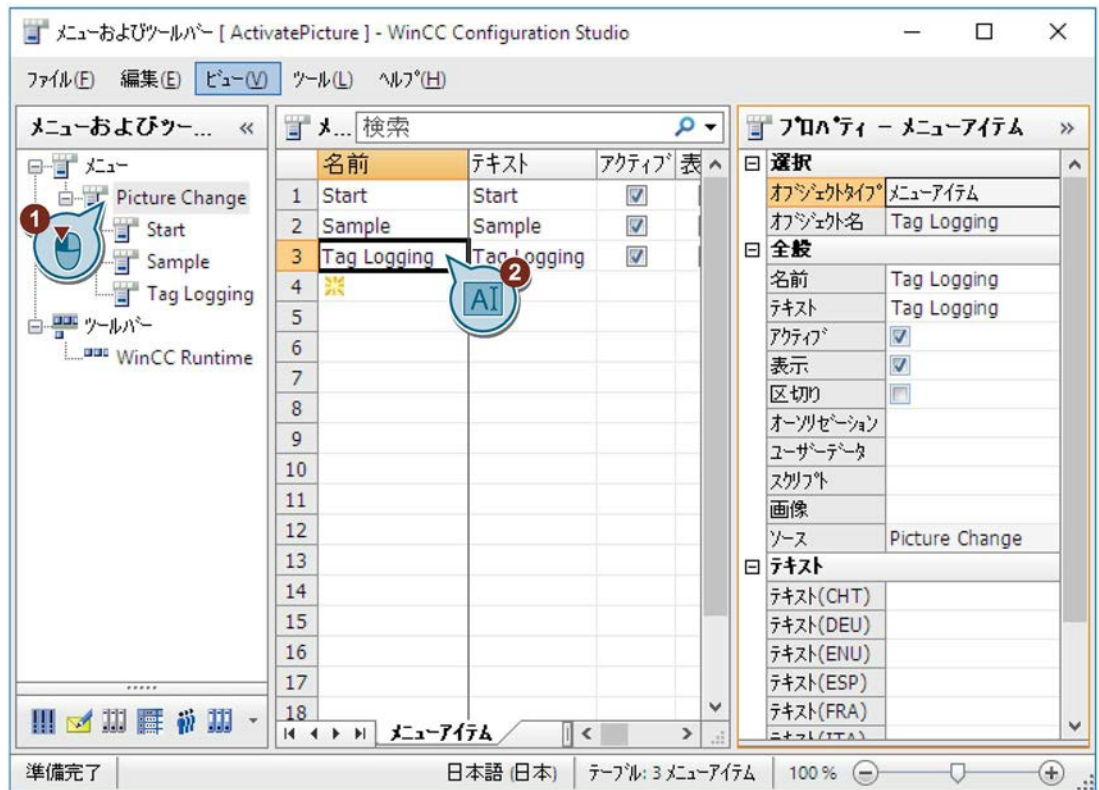
手順

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。



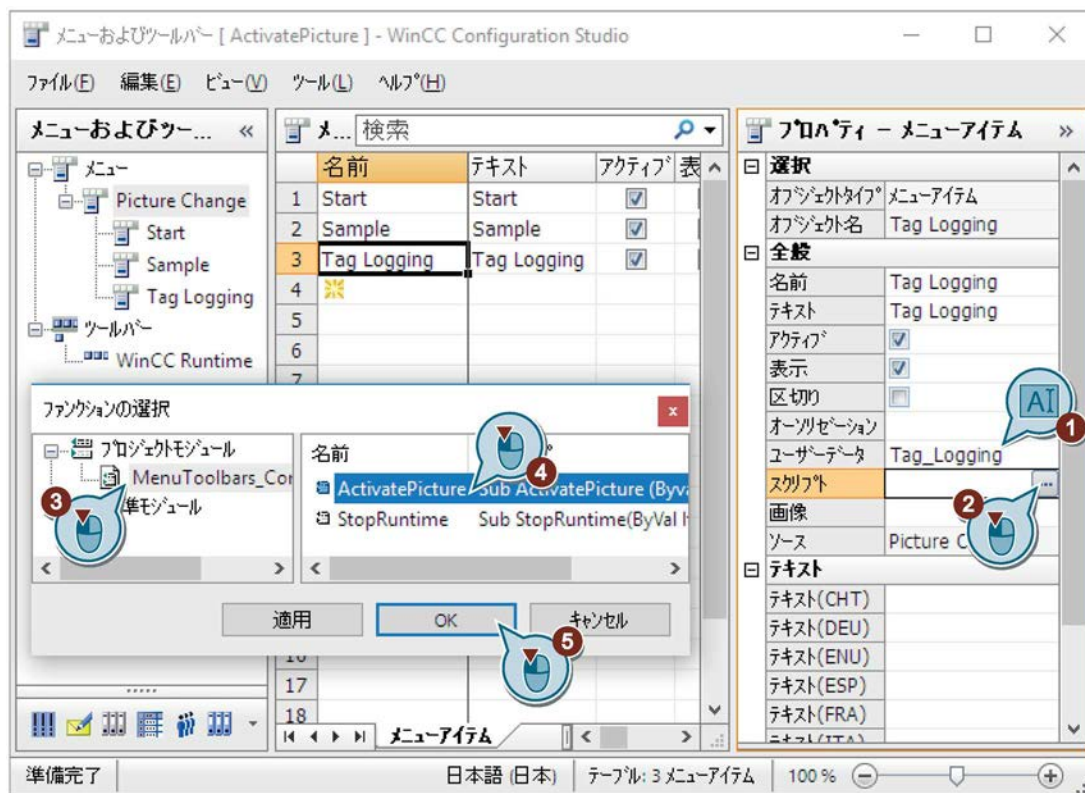
7.8 ユーザー定義メニューのカスタマイズ(アーカイブシステム)

2. [タグロギング]メニューコマンドを作成します。



7.8 ユーザー定義メニューのカスタマイズ(アーカイブシステム)

3. "Tag_Logging.pdl"画像に変更するために、[タグロギング]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

[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。

メニューエントリを使用して、ランタイムでプロセス画像"START.pdl"、
"SAMPLE.pdl"および"Tag_Logging.pdl"に切り替えられます。

7.9 ランタイムプロパティの定義(アーカイブシステム)

はじめに

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。

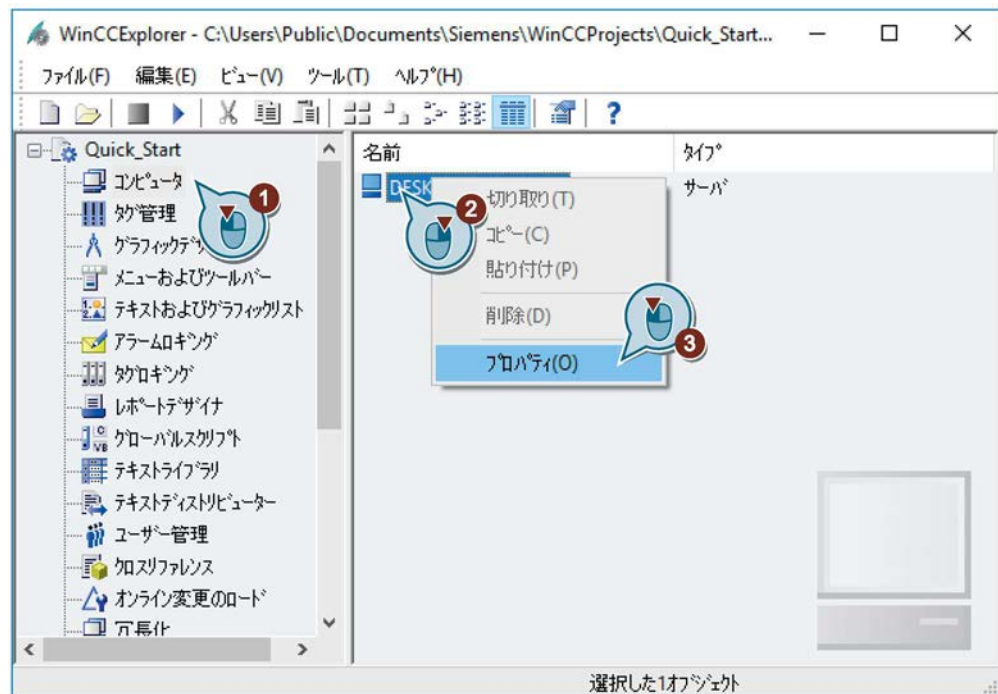
このセクションでは、プロジェクトの起動時にタグロギングランタイムが実行されるようにWinCC Runtimeをセットアップします。"Tag_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

必要条件

- "Quick_Start"プロジェクトが開いている。
- プロセス画像"Tag_Logging.pdl"が作成されている。

手順

1. [コンピュータのプロパティ]ダイアログを開きます。



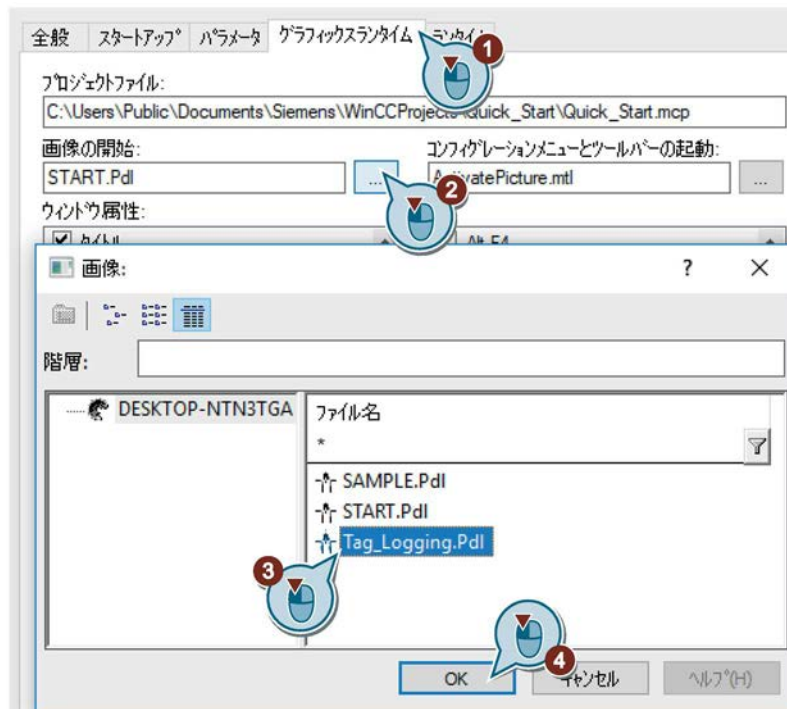
[コンピュータのプロパティ]ダイアログが開きます。

7.9 ランタイムプロパティの定義(アーカイブシステム)

2. [スタートアップ]タブをクリックして、"タグ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. [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"Tag_Logging.pdl"を定義します。



4. [OK]をクリックして[コンピュータのプロパティ]ダイアログを終了します。

結果

WinCC Runtimeのプロパティが定義されています。"Quick_Start"プロジェクトの起動時に、タグロギングランタイムが実行され、プロセス画像"Tag_Logging.pdl"が表示されます。次のステップで、"Quick_Start"プロジェクトを有効化して、テストします。

7.10 プロジェクトの有効化とテスト(アーカイブシステム)

はじめに

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"に値を割り当てます。

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。

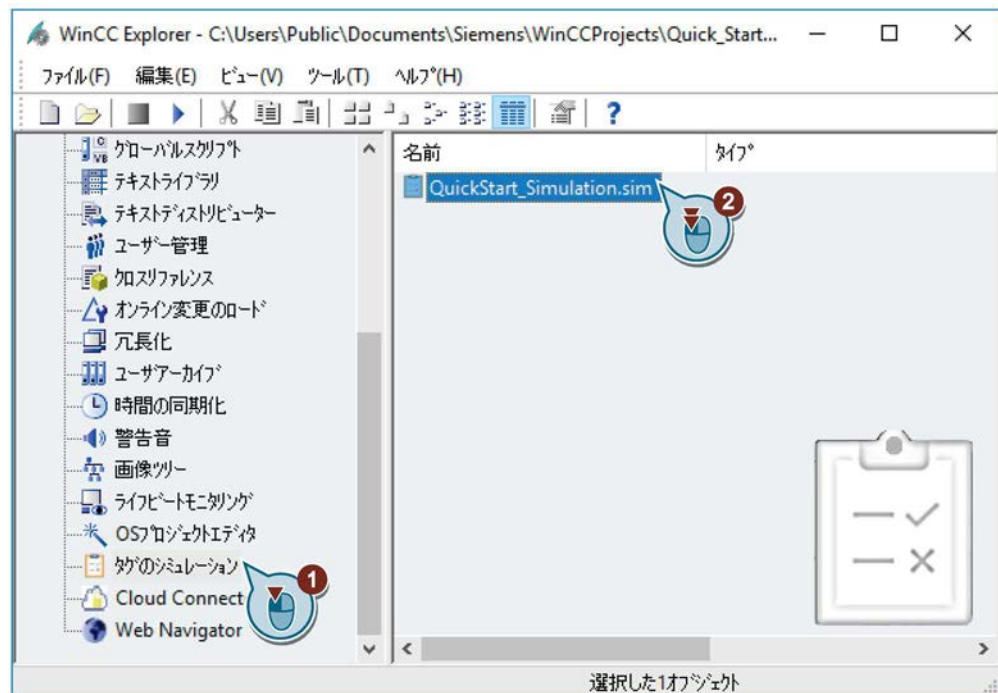
タグロギングランタイムがアーカイブ値を読み取り、[トレンド]ウィンドウおよび[テーブル]ウィンドウに転送します。値は、トレンドおよびテーブルとして出力されます。

必要条件

- ランタイムプロパティが定義されている。

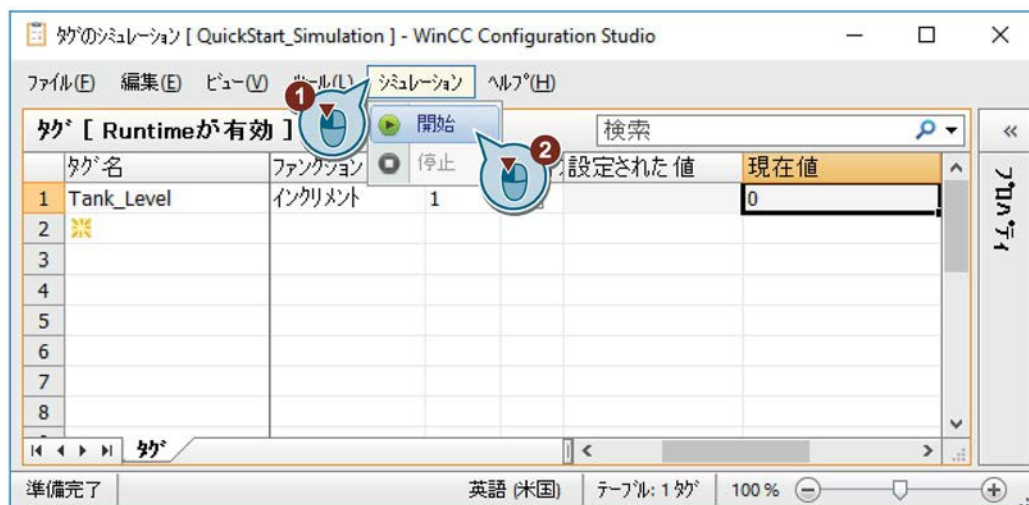
手順

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。

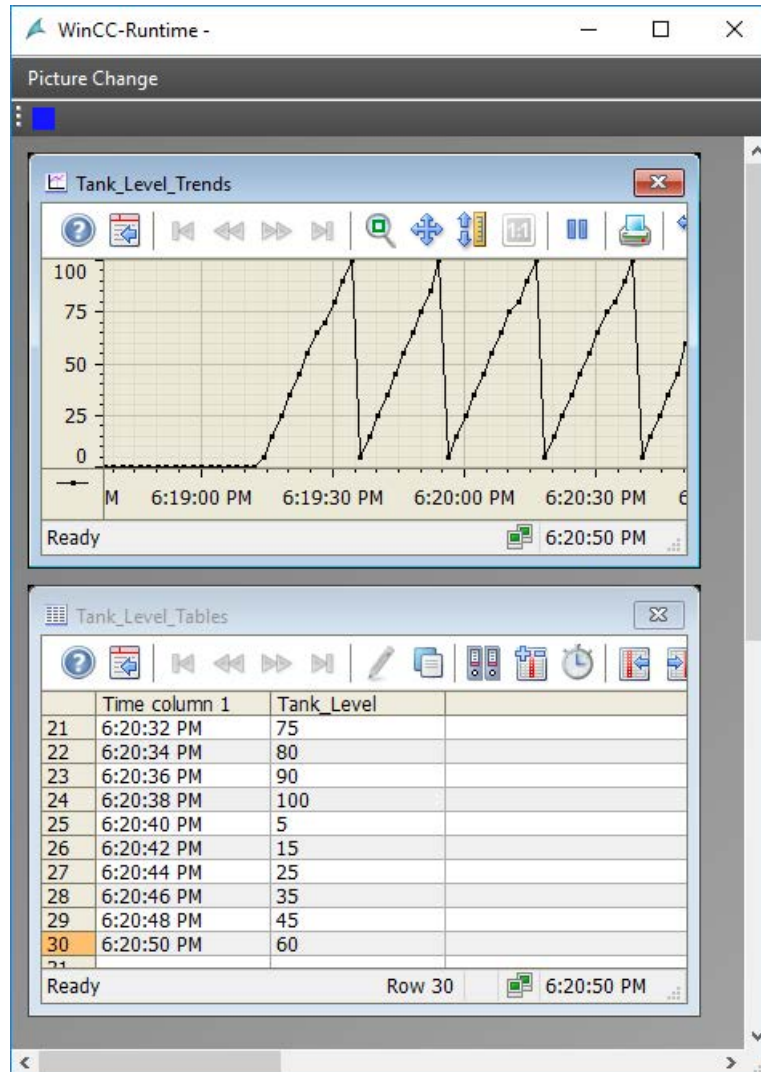


7.10 プロジェクトの有効化とテスト(アーカイブシステム)

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. [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。

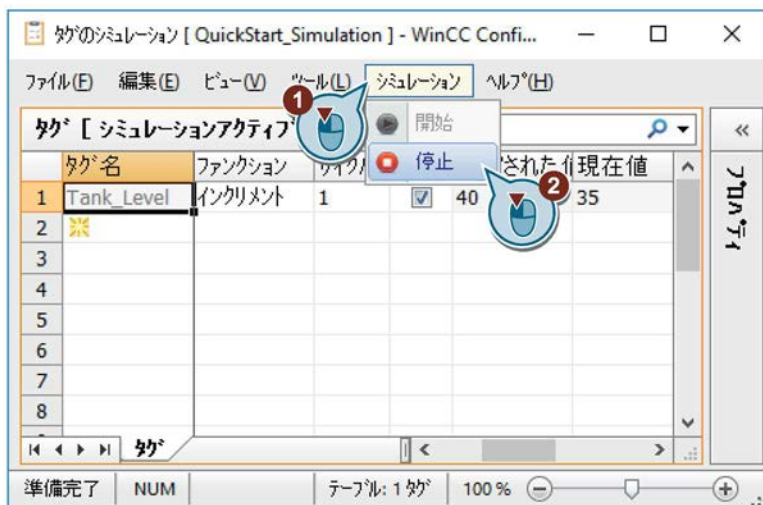


4. プロセス画像"Tag_Logging.pdl"のシミュレーション値の出力を監視します。



7.10 プロジェクトの有効化とテスト(アーカイブシステム)

5. シミュレーションを終了します。



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。

プロセスアーカイブからの値の出力

8.1 プロセスアーカイブからの値の出力

はじめに

この章では、ロギングシステムに関する情報やプロセス値アーカイブからの値をログする方法を説明します。

原理

レポートシステムは、設定データやランタイムデータを文書化します。

- 設定データには、使用したタグを持つテーブルをはじめ、オブジェクトを持つ工場画像、作成済みのユーザーグループなどがあります。
- ランタイムデータには、メッセージアーカイブまたはメッセージリストからのメッセージをはじめ、プロセス値アーカイブあるいは圧縮アーカイブからのプロセス値などがあります。

文書化されたデータは、レポートまたはログとして保存されます。

レポートやログには、ページレイアウトまたはラインレイアウトが使用されます。レイアウトで、外観やレポート出力のためのデータ提供を設定します。

ログやレポートの出力は、印刷ジョブによって制御されます。次のパラメータは、主に印刷ジョブで定義されます。

- 時間コントロール
- 出力媒体
- 出力の範囲

"Quick_Start"プロジェクトで、プロセス値アーカイブ"**Tank_Level_Archive**"からの値を文書化します。これらの値をログに出力します。

ログに対して、新しいページレイアウトを作成し、印刷ジョブのパラメータを決定します。

印刷ジョブを、[テーブル]ウィンドウの設定ボタンにリンクさせます。ランタイムで、レポートの出力はこのボタンによってトリガされます。

アーカイブ値は、[テーブル]ウィンドウの現在のビューに表示される出力ログで文書化されます。

8.2 レポートシステム

はじめに

レポートシステムはWinCCのサブシステムであり、設定データやランタイムデータを文書化するために使用されます。

設定データはレポートとして出力されます。設定データはタグ、ファンクション、あるいはプロジェクトで使用されるグラフィックスです。

ランタイムデータはログとして出力されます。ログには、以下のランタイムデータを含めることができます。例えば、

- メッセージシーケンスレポート:時系列順に発生したすべてのメッセージ
- アーカイブレポート:特定のメッセージアーカイブからのメッセージ
- メッセージレポート:メッセージリストからの現在のメッセージ
- 特定のプロセス値や圧縮アーカイブからの値
- WinCCから起動しない他のアプリケーションからのデータ

WinCCログにこのようなデータを統合するために様々なログオブジェクトが使用できます。

レポートシステムのコンポーネント

レポートシステムは、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。

設定コンポーネント

レポートシステムの設定コンポーネントは"レポートデザイナー"エディタです。

[レポートデザイナー]エディタには、[レイアウト]と[印刷ジョブ]の2つのコンポーネントがあります。

これらのコンポーネントには、事前に定義された標準レイアウトや編集可能な印刷ジョブが含まれています。[レポートデザイナー]エディタでは、新しいレイアウトや印刷ジョブを作成できます。

レイアウトを作成および編集するため、[レポートデザイナー]エディタには、2つの追加ツールである、ページレイアウトエディタおよび行レイアウトエディタが用意されています。

ランタイムコンポーネント

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、主に次のタスクを実行するために使用されます:

- アーカイブまたはコントロールからの文書化する値の読み取り
- プリンタ出力の制御

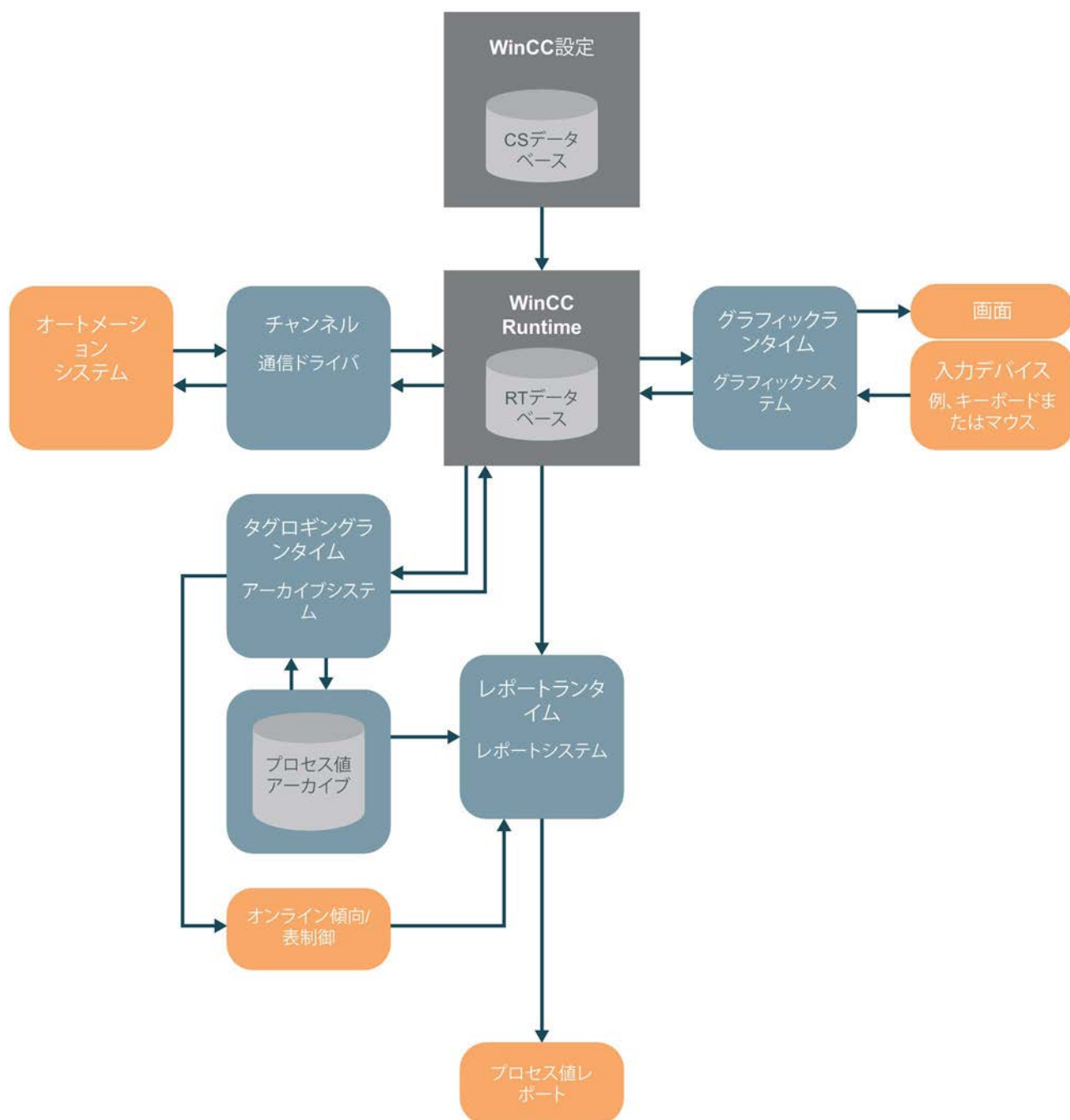
出力媒体

[レポートデザイナー]エディタは、以下のようなレポートやログを出力できます。

- プリンタへ
- ファイルへ
- 画面へ

8.2 レポートシステム

WinCCシステムのロギング



8.3 ページレイアウトの作成

はじめに

次のステップは、WinCCエクスプローラでのページレイアウトの作成方法、名前の変更方法を示しています。

ページレイアウトは、データを出力する異なるオブジェクトを含むテンプレートです。どの情報やデザイン機能がレポートまたはログに表示されるかを、ページレイアウトのオブジェクトによって決定します。

既にWinCCは、大半のアプリケーションのために事前に設定されたレイアウトを提供しています。これらのレイアウトは、要件に合致するように、"レポートデザイナー"エディタで編集できます。ページレイアウトには、言語非依存レイアウトと言語依存レイアウトがあります。

"Quick_Start"プロジェクトに対して、"レポートデザイナー"エディタで新しいページレイアウトを作成します。このページレイアウトを、プロセス値アーカイブからの値が文書化されるメッセージレポートのテンプレートとして使用します。

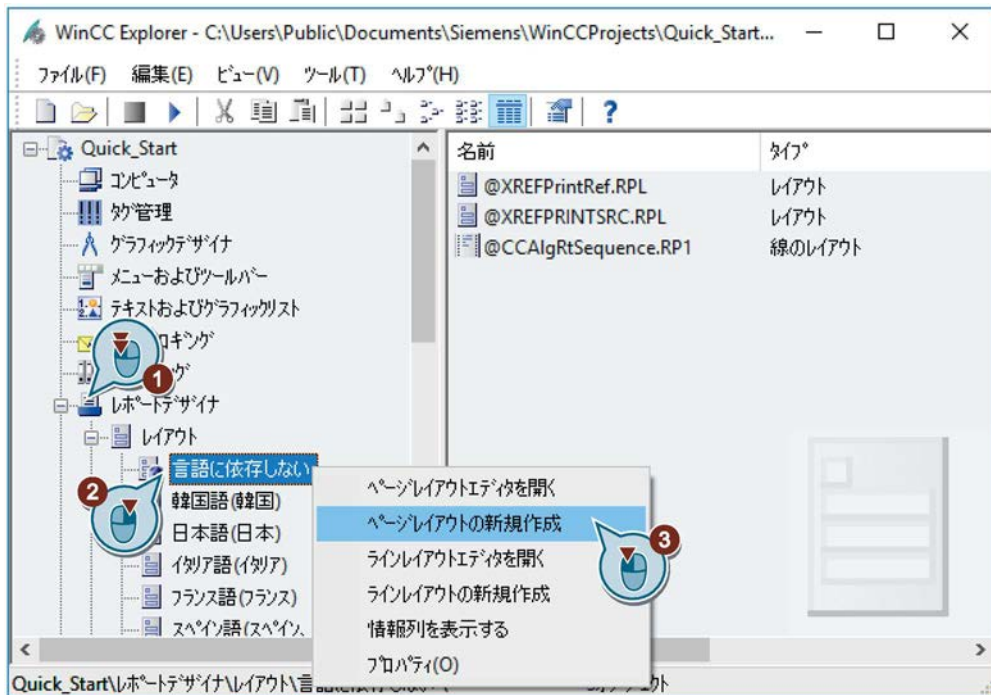
必要条件

- "Quick_Start"プロジェクトが開いている。

8.3 ページレイアウトの作成

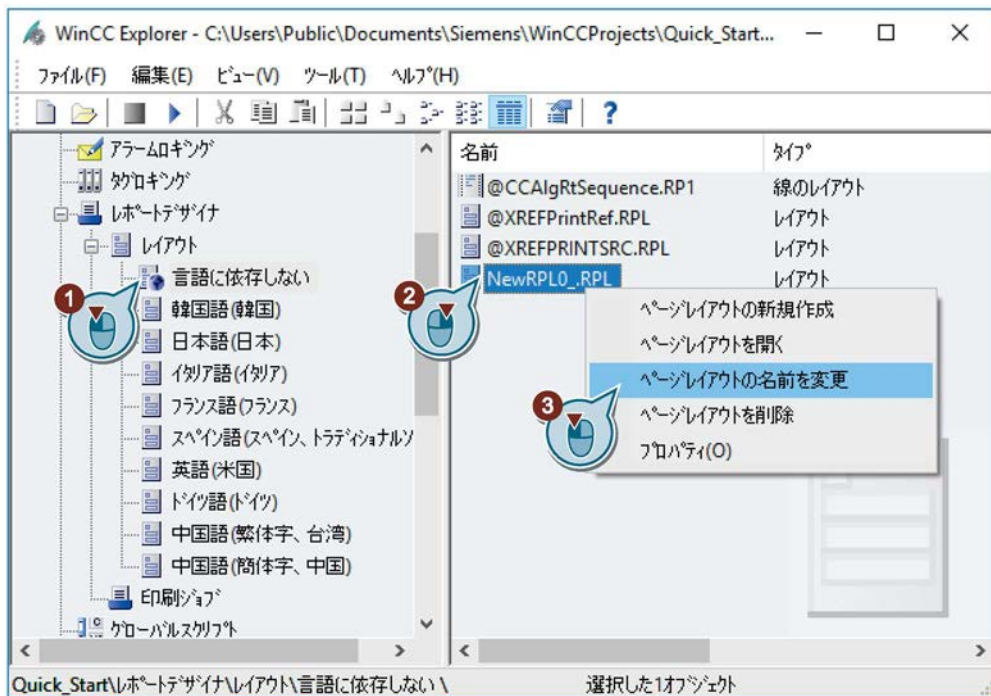
手順

1. WinCC Explorerで、新しい言語非依存ページレイアウトを作成します。

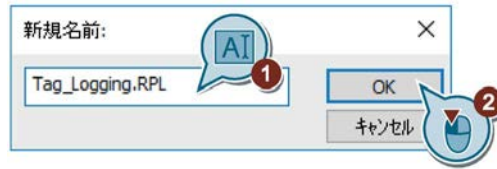


新しいレイアウトファイル"NewRPL0.RPL"は、[言語非依存]フォルダで作成されます。

2. [新しい名前]ダイアログを開いて、ページレイアウトの名前を変更します。



3. レイアウトファイルの名前として、"Tag_Logging.rpl"と入力します。



ページレイアウトの名前が変更されます。

結果

WinCCエクスプローラで、新しいページレイアウトが作成され、名前が変更されました。"Quick_Start"プロジェクトで、このページレイアウトを、プロセス値アーカイブからの値が文書化されるメッセージレポートのテンプレートとして使用します。

メッセージレポートの内容やデザイン機能を決定するために、次のステップでページレイアウトを編集します。

8.4 ページレイアウトの編集

8.4.1 ページレイアウトの編集

はじめに

このセクションでは、ページレイアウトエディタを使用したページレイアウトの編集方法を説明します。

レイアウトの編集は、以下のステップで行なわれます。

- 全体ページレイアウトのプロパティの確立
- カバーシートの出力の無効化
- ログの内容の決定
- ヘッダーとフッターの編集

原理

ページレイアウトについて、同じページレイアウトを持つ全てのページに適用されるプロパティを定義します。

"Quick_Start"プロジェクトで、ページレイアウト"Tag_Logging"のページに対してフォーマットと印刷余白を定義します。

ページレイアウトには3ページある場合があります。

- カバーシート

カバーシートは、ログまたはレポートの最初のページです。

カバーシートの出力は、ページレイアウトエディタで事前に設定されます。このセクションでは、[カバーシート]ページが出力されないように、事前に定義された設定を変更します。

- レポート内容

"レポートの内容"のページで、ログまたはレポートのセットアップや内容を定義します。レポート内容の出力は必須です。

プロセス値アーカイブからの値を文書化するために、"Quick_Start"プロジェクトでダイナミックオブジェクト"WinCC Control Runtime Printprovider"を使用します。

オブジェクト"WinCC Control Runtime Printprovider Table"および"WinCC Control Runtime Printprovider Picture"のどちらかを選べます。テーブルの完全な内容がテーブルに出力され、画像での出力がWinCCコントロールの現在の表示となります。両方の変数に対して、WinCCコントロールのボタンで可能なのは、ログ出力だけです。

- 最終ページ

最終ページは、ログまたはレポートの最終ページです。

最終ページの出力は、[レポートデザイナー]エディタで事前設定されていません。

"Quick_Start"プロジェクトでは、最終ページは出力しません。

スタティックレイヤとダイナミックレイヤ

ページレイアウトの各ページには、スタティックレイヤとダイナミックレイヤが含まれています。

- ヘッダーとフッターは、ページレイアウトのスタティックレベルで定義します。

例えば、スタティックレベルは、会社名、会社のロゴ、時刻やページ番号を出力するために使用します。

スタティックレベルでは、スタティックオブジェクトとシステムオブジェクトだけを挿入できます。

- ダイナミックレベルは、設定データやランタイムデータを出力するダイナミックオブジェクトを含んでいます。

ダイナミックレイヤでは、スタティックオブジェクトとダイナミックオブジェクトの両方を挿入できます。

8.4 ページレイアウトの編集

ページレイアウトのオブジェクト

オブジェクトパレットのオブジェクトは、ドラッグおよびドロップではページレイアウトに追加できません。オブジェクトパレットのオブジェクトを挿入するには、そのオブジェクトを選択して、ページレイアウトの作業エリアでマウスの左ボタンで一度クリックします。

"Quick_Start"プロジェクトで、ページ"レポートの内容"のヘッダーにシステムオブジェクト"プロジェクト名"を追加します。このオブジェクトは、プロジェクト名を表示するために使用されます。

フッターに、システムオブジェクト"ページ番号"を挿入します。システムオブジェクト"ページ番号"によって、ログにページ番号が付けられます。

8.4.2 ページレイアウトのプロパティの確立

はじめに

次のステップは、ページレイアウトエディタの起動方法や全体ページレイアウトのプロパティの定義方法を示しています。

ページレイアウトエディタは"レポートデザイナー"エディタのコンポーネントであり、ページレイアウトを作成、編集するために使用します。ページレイアウトエディタを使用できるのは、現在WinCCエクスプローラで開いているプロジェクトだけです。レイアウトは、特定のオブジェクトに対して保存されます。

このセクションでは、ページレイアウトエディタを使用して、[Tag_Logging]ページレイアウトを編集します。

レイアウトを編集することで、出力するログのセットアップや内容を決定します。

"Quick_Start"プロジェクトに対して、"Tag_Logging"ページレイアウトの以下のプロパティを定義します。

- ページサイズ

ページサイズは、レイアウトにおける全体エリアを示します。

ページサイズは、ログの出力フォーマットを決定します。

- 印刷余白

印刷余白は、印刷不可の余白領域を定義します。

この領域は、デフォルトではページレイアウトエディタでグレー表示され、編集できません。

- [カバースHEET]ページの出力

ページレイアウトの[オブジェクトプロパティ]ダイアログで、ログを最初のページ（カバー）と共に出力するかどうかを定義します。

このセクションでは、[カバースHEET]の出力を無効にします。

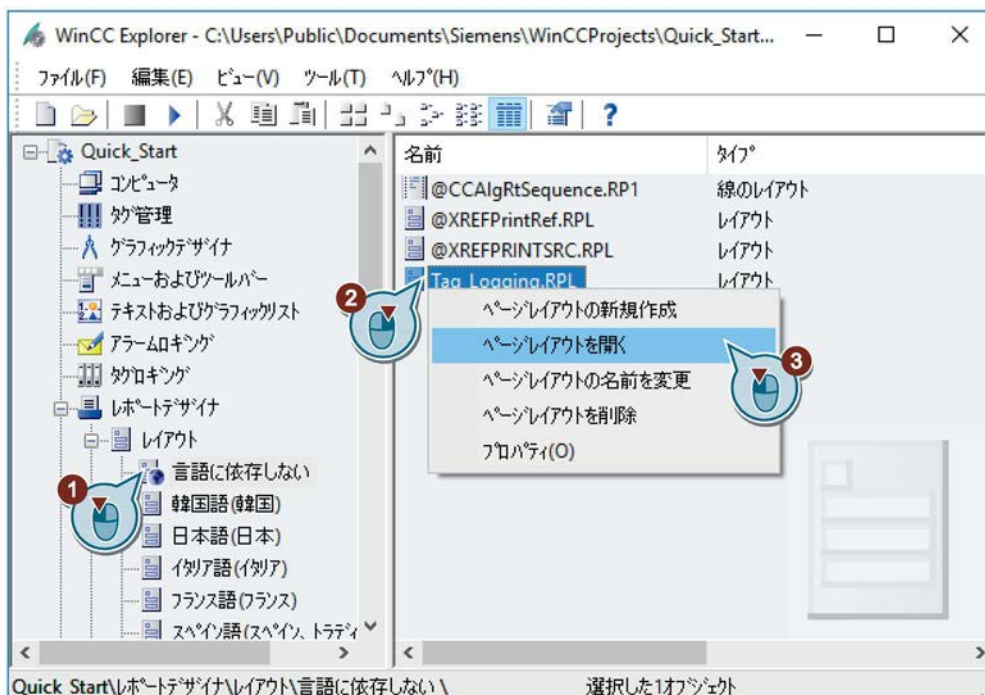
必要条件

- レイアウトファイル"Tag_Logging.rpl"が作成されている。

8.4 ページレイアウトの編集

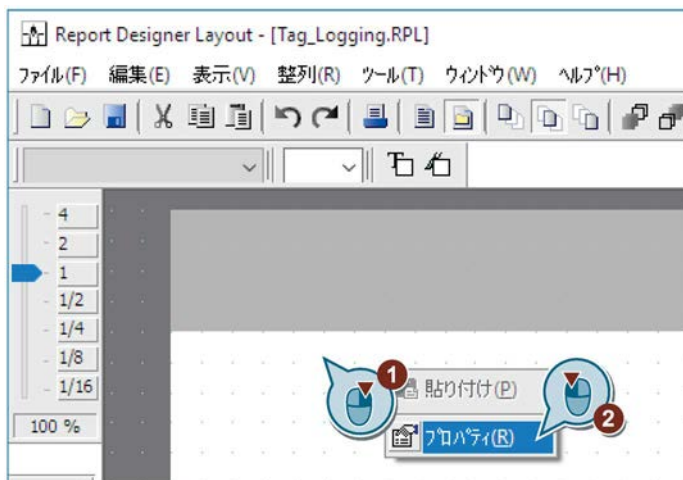
手順

1. ページレイアウトエディタで、"Tag_Logging.rpl"レイアウトファイルを開きます。



ページレイアウトエディタで、"Tag_Logging.rpl"レイアウトファイルが開きます。

2. ページレイアウトの[オブジェクトプロパティ]ダイアログを開きます。



[オブジェクトプロパティ]ダイアログが開きます。

3. "ジオメトリ"プロパティのパラメータに以下の値を定義します。

- 用紙サイズ:レター
- 左の印刷余白:2 cm
- 右の印刷余白:2 cm
- 上の印刷余白:2 cm
- 下部の印刷余白:2 cm



4. 単位を"cm"から"インチ"に変更するには、[ツール] >

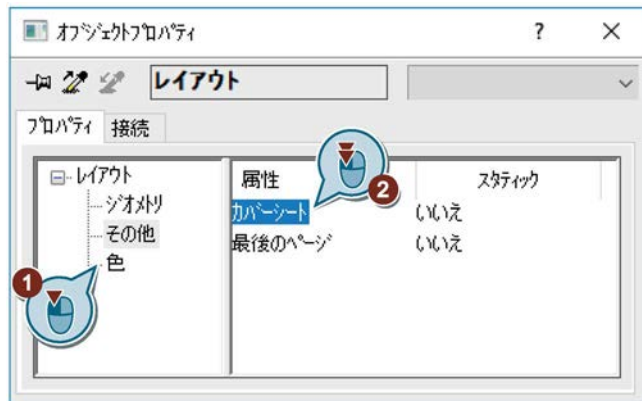
[設定]メニューで、[設定]ダイアログを開きます。

[座標]エリアの[単位]タブで単位"インチ(inch)"を選択します。

指示された値を"cm"から"inch"に変換します。次の規則が適用されます。1センチメートル = 0.3937インチ

8.4 ページレイアウトの編集

5. [カバーシート]ページの出力を無効化します。



6. [オブジェクトプロパティ]ダイアログを閉じます。

結果

ページフォーマットと、"Tag_Logging"ページレイアウトの印刷余白が定義されます。

これは、ページの印刷可能領域と印刷不可領域を定義します。"カバーシート"ページの事前に定義された出力が変更されます。

ログの内容を定義するために、次のステップでページレイアウトの"レポートの内容"ページを編集します。

8.4.3 ログの内容の決定

はじめに

次のステップは、ログの内容の定義方法を示しています。

ページレイアウトの"レポートの内容"ページで、メッセージレポートの内容を定義します。このために、ページレイアウトエディタのオブジェクトパレットのオブジェクトを使用します。

このセクションでは、プロセス値アーカイブからの値の表示に使用する[WinCCコントロールランタイム印刷プロバイダ - テーブル]オブジェクトを挿入します。

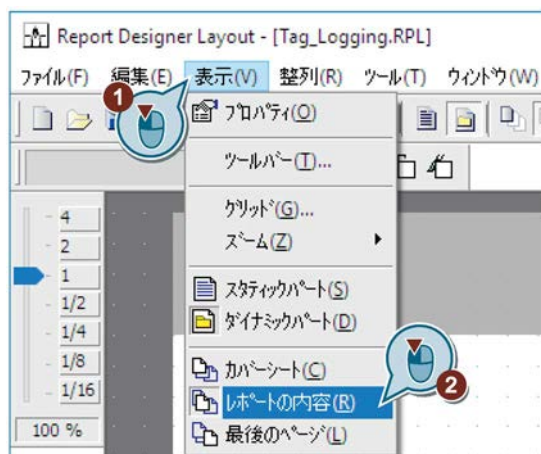
必要条件

- ページレイアウトエディタで、[Tag_Logging.rpl]レイアウトファイルが開いている。

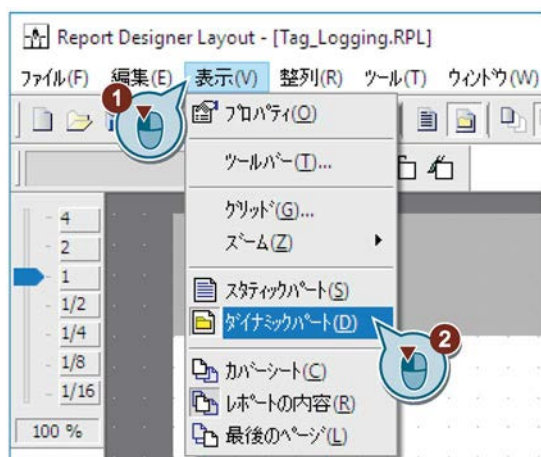
8.4 ページレイアウトの編集

手順

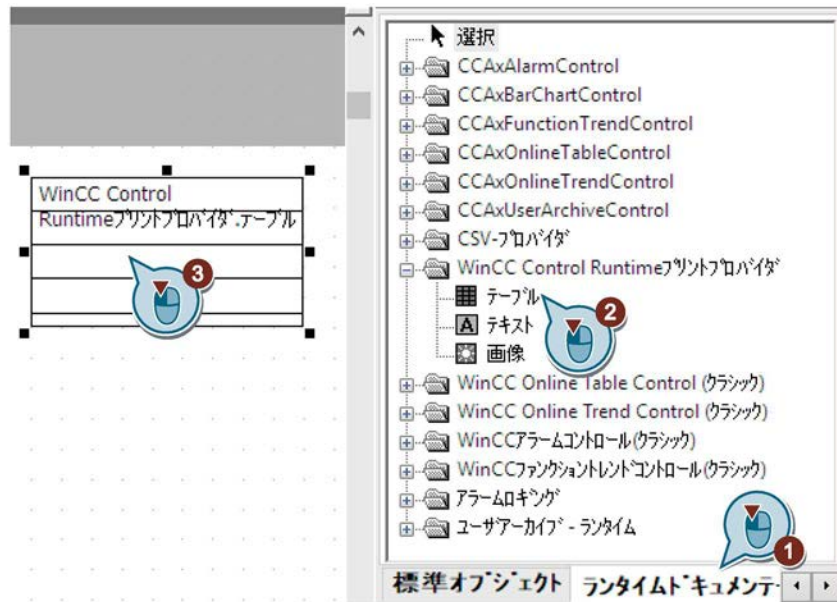
1. ページレイアウトの"レポートの内容"ページを開きます。



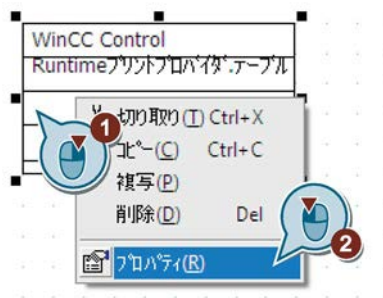
2. [レポートの内容]ページのダイナミックレベルへ変更します。



3. [Tag_Logging.rpl]ページレイアウトに、[WinCCコントロールランタイム印刷プロバイダ - テーブル]オブジェクトを挿入します。



4. [オブジェクトプロパティ]ダイアログを開きます。



8.4 ページレイアウトの編集

5. [プロパティ]タブをクリックして、"ジオメトリ"プロパティに次の値を定義します。

- 位置X:3 cm
- 位置Y:5 cm
- 幅:16 cm
- 高さ:18 cm
- 列:1



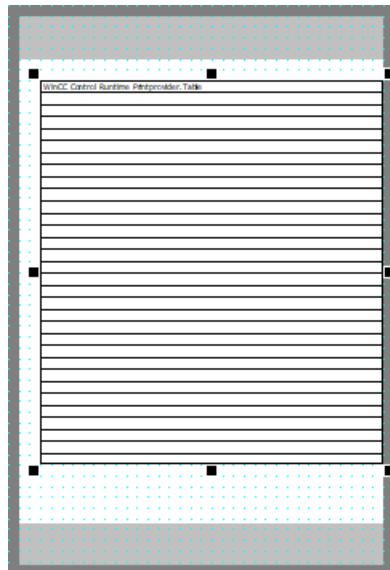
6. [オブジェクトプロパティ]ダイアログを閉じます。

結果

[WinCCコントロールランタイム印刷プロバイダ - テーブル]オブジェクトが挿入されます。

これによって、プロセス値アーカイブから値を読み取り、ログ"Tag_Logging.rpl"で文書化できるようになります。

ログにプロジェクト名を出力するために、次のステップで"レポートの内容"ページのヘッダーを編集します。



8.4 ページレイアウトの編集

8.4.4 ヘッダーの編集

はじめに

次のステップでは、"レポートの内容"ページのヘッダーの編集方法を示しています。

ヘッダーは、ページレイアウトのスタティックレベルにあります。

ヘッダーは、"カバーシート"ページ、"レポートの内容"ページおよび最終ページに対して、個別に定義されます。"Quick_Start"プロジェクトでは、"レポートの内容"ページのヘッダーだけを編集します。

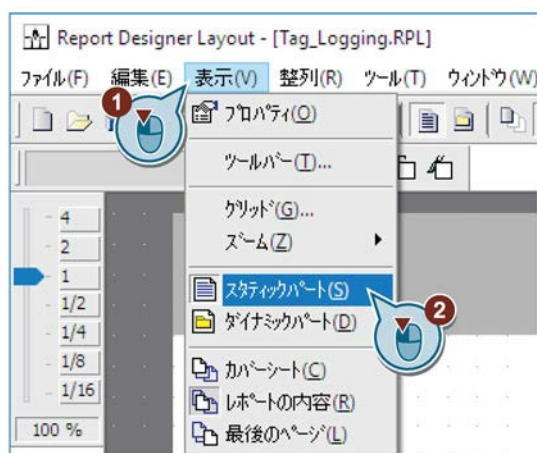
ヘッダーで、システムオブジェクト"プロジェクト名"を挿入します。このオブジェクトは、ログでのプロジェクト名の表示でプレースホルダになります。

必要条件

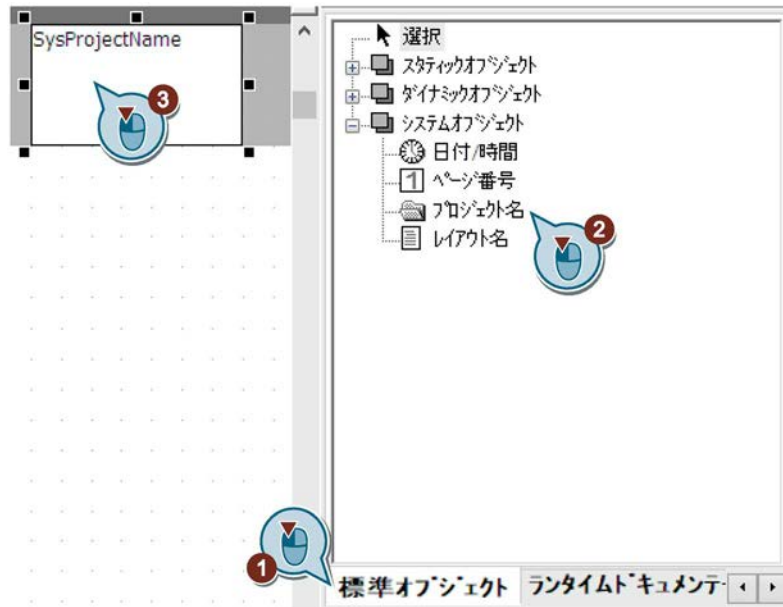
- ページレイアウトエディタで、[Tag_Logging.rpl]レイアウトファイルが開いている。

手順

1. [表示]メニューのページレイアウトの"レポートの内容"ページを開きます。
2. [レポートの内容]ページのスタティックレベルへ変更します。



3. ヘッダーに[プロジェクト名]システムオブジェクトを挿入します。



4. [オブジェクトプロパティ]ダイアログを開きます。



[オブジェクトプロパティ]ダイアログが開きます。

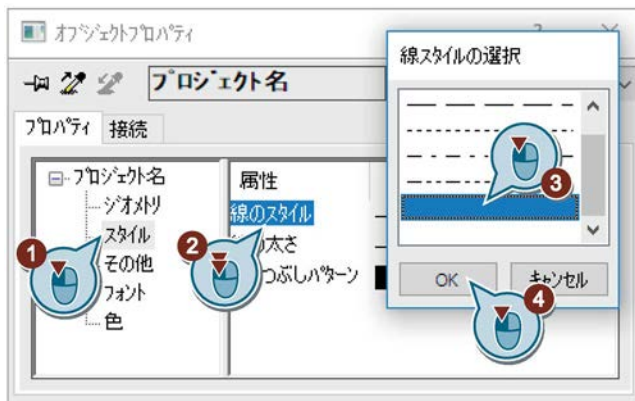
8.4 ページレイアウトの編集

5. "ジオメトリ"プロパティのパラメータに以下の値を定義します。

- 位置X:2 cm
- 位置Y:2 cm
- 幅:12 cm
- 高さ:1 cm



6. 挿入されたオブジェクトの線のスタイルとして、[線無し]を設定します。



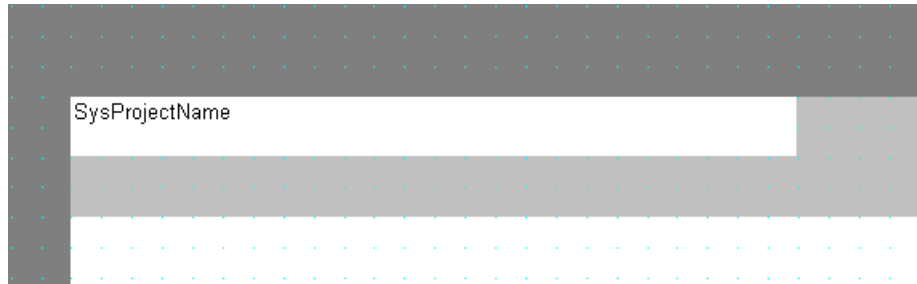
7. [オブジェクトプロパティ]ダイアログを閉じます。

結果

"レポート内容"ページのヘッダーにシステムオブジェクト"プロジェクト名"が挿入されます。

このオブジェクトは、ログでプロジェクト名を表示するために使用されます。

ログのページに番号を付けるために、次のステップでフッターを編集します。



8.4 ページレイアウトの編集

8.4.5 フッターの編集

はじめに

次のステップでは、"レポートの内容"ページのフッターの編集方法を示しています。

フッターは、ページレイアウトのスタティックレベルにあります。

フッターは、"カバーシート"ページ、"レポートの内容"ページおよび最終ページに対して、個別に定義されます。"Quick_Start"プロジェクトでは、"レポートの内容"ページのフッターだけを編集します。

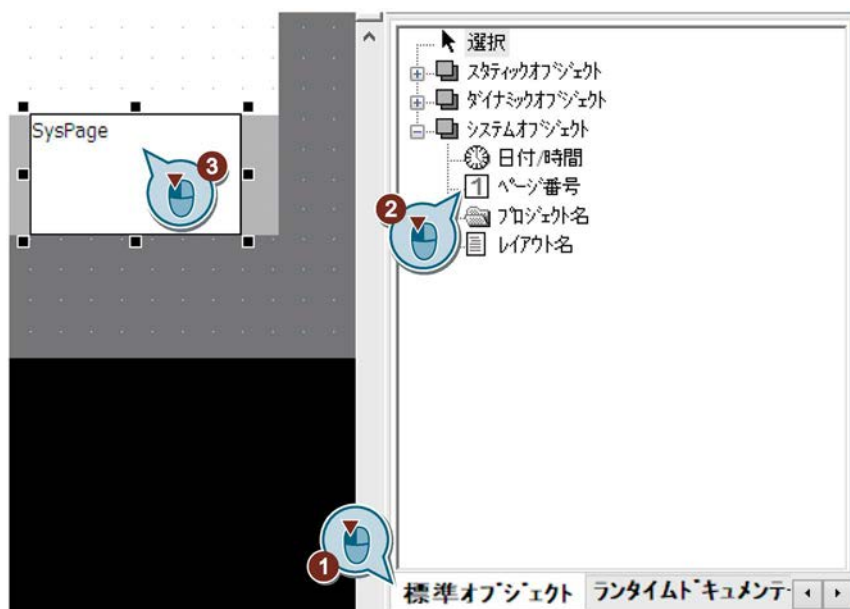
ページレイアウトエディタのオブジェクトパレットからシステムオブジェクト"ページ番号"を挿入して、フッターを編集します。このオブジェクトは、ログでのページ番号の表示でプレースホルダになります。

必要条件

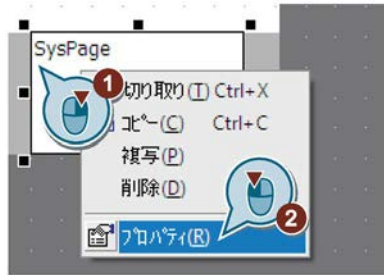
- ページレイアウトエディタで、[Tag_Logging.rpl]レイアウトファイルが開いている。

手順

1. ページレイアウトの"レポートの内容"ページを開きます。
2. [レポートの内容]ページのスタティックレベルへ変更します。
3. フッターに、[ページ番号]システムオブジェクトを挿入します。



4. [オブジェクトプロパティ]ダイアログを開きます。



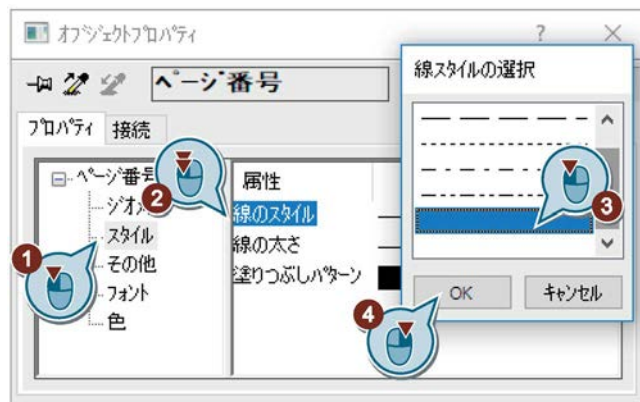
[オブジェクトプロパティ]ダイアログが開きます。

5. "ジオメトリ"プロパティのパラメータに以下の値を定義します。

- 位置X:18 cm
- 位置Y:24 cm
- 幅:1 cm
- 高さ:1 cm



6. 挿入されたオブジェクトの線のスタイルとして、[線無し]を設定します。



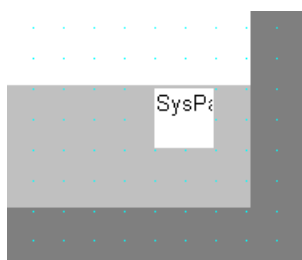
8.4 ページレイアウトの編集

7. [オブジェクトプロパティ]ダイアログを閉じます。
8. ツールバーの  ボタンで、[Tag_Logging.rpl]レイアウトファイルを保存します。
9. "レポートデザイナー"を閉じます。

結果

"レポート内容"ページのフッターにシステムオブジェクト"ページ番号"が挿入されます。

このオブジェクトによって、ログのページに番号が付けられます。



8.5 印刷ジョブの編集

8.5.1 印刷ジョブの編集

はじめに

このセクションでは、印刷ジョブの編集方法を説明します。

原理

プロジェクト文書やランタイム文書を出力するのに最も重要なのは、**WinCC**の印刷ジョブです。

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。

WinCCには、多数の事前に定義された印刷ジョブがあります。これらの印刷ジョブは、特定の**WinCC**アプリケーションと既に関連付けられています。したがって、システムの印刷ジョブは削除できません。必要であれば、システムの印刷ジョブの名前を変更することはできます。これらの事前に定義された印刷ジョブでは、一部の設定だけを変更できます。

印刷ジョブの設定

出力するために、各レイアウトは印刷ジョブと関連付けられています。

"Quick_Start"プロジェクトで、事前に定義された印刷ジョブを使用します。

この印刷ジョブを[Tag_Logging]ページレイアウトと接続します。これは、[Tag_Logging]ページレイアウトでログを出力します。

さらに、ログの出力のための出力媒体も定義します。

ログの印刷

印刷ジョブは、[テーブル]ウィンドウのツールバーのボタンによって実行されます。

8.5 印刷ジョブの編集

8.5.2 印刷ジョブの定義と編集

はじめに

次のステップは、ログの出力のための印刷ジョブの定義と編集の方法を示しています。

"Quick_Start"プロジェクトで、事前に定義された印刷ジョブ"@OnlineTableControl - Table"を使用します。この印刷ジョブの編集手順は、次のとおりです。

- 印刷ジョブのページレイアウトへのリンク
- 出力媒体の定義

この印刷ジョブは、"Tag_Logging"ページレイアウトとリンクしています。これによって、ログはこのページレイアウトで定義されたデザイン機能と設定を適用します。

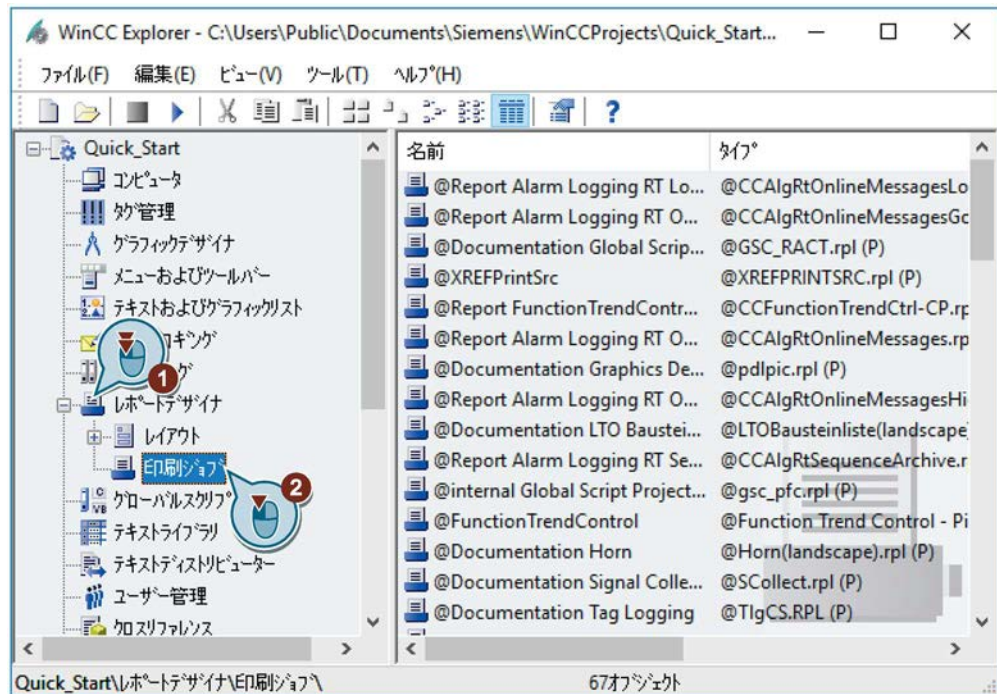
選択したプリンタを、ログの出力媒体として定義します。プリンタがない場合は、ログをファイルに出力できます。このトピックに関する詳細情報は、「プロジェクトのドキュメント」の「WinCCの操作」 > 「設定データとランタイムデータのドキュメント」 > 「プロジェクトのドキュメント」 > 「新しい印刷ジョブの作成方法」のセクションにあります。

必要条件

- "Quick_Start"プロジェクトが開いている。
- ページレイアウト"Tag_Logging.rpl"が作成されている。

手順

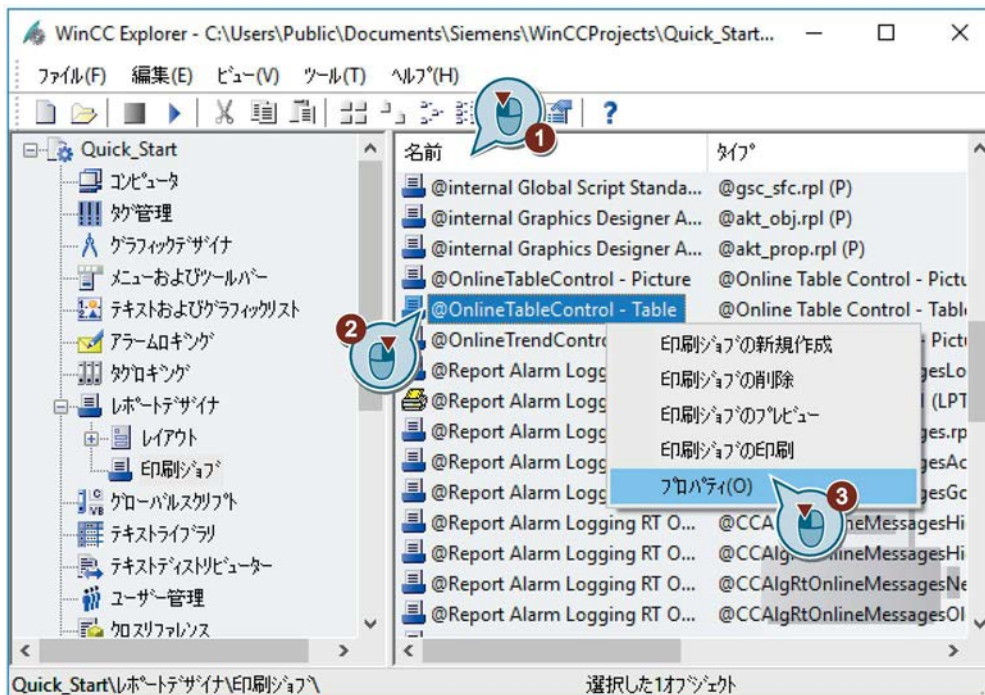
1. WinCC Explorerで[レポートデザイナー]エディタの[印刷ジョブ]コンポーネントを選択します。



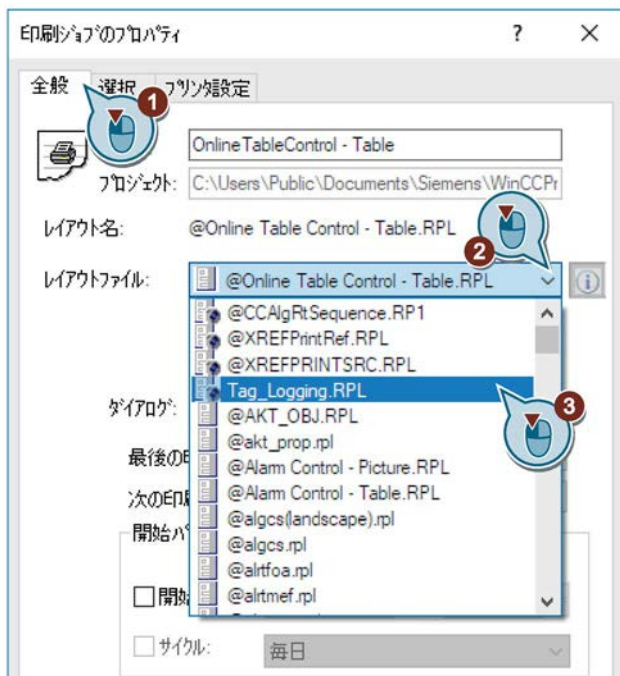
事前に定義された印刷ジョブが表示されます。

8.5 印刷ジョブの編集

- 印刷ジョブ"@OnlineTableControl - Table"を選び、[印刷ジョブのプロパティ]ダイアログを開きます。



- 印刷ジョブを"Tag_Logging.rpl"レイアウトファイルとリンクします。



4. [プリンタの選択]タブをクリックして、[プリンタ]への出力を有効にします。

[プリンタプロパティ]領域で、接続されたプリンタを選択するか、初期設定のプリンタを受け入れます。



5. [OK]をクリックします。

結果

ログの出力のための印刷ジョブが、定義、編集されています。

ログは、"Tag_Logging"ページレイアウトをベースにしています。

ログが選択したプリンタに出力されます。

8.6 ランタイムプロパティの定義(レポートシステム)

はじめに

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レポート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。レポートランタイムは、[テーブル]ウィンドウ "Tank_Level_Tables"から値を読み取り、プリンタ出力を制御します。

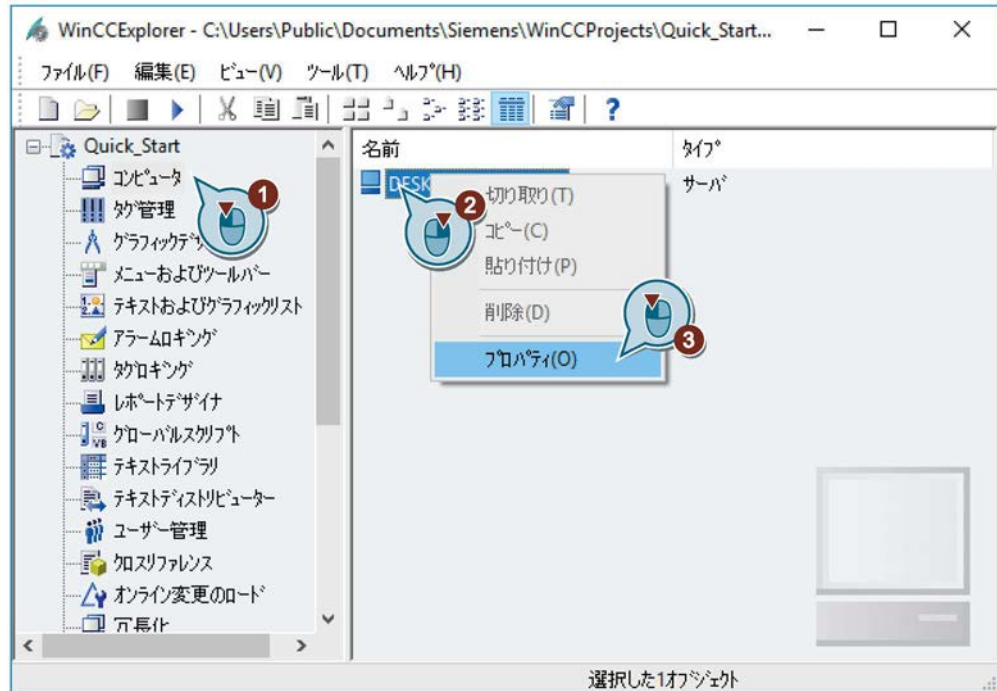
"Tag_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

必要条件

- "Quick_Start"プロジェクトが開いている。
- プロセス画像"Tag_Logging.pdl"が作成されている。

手順

1. [コンピュータのプロパティ]ダイアログを開きます。



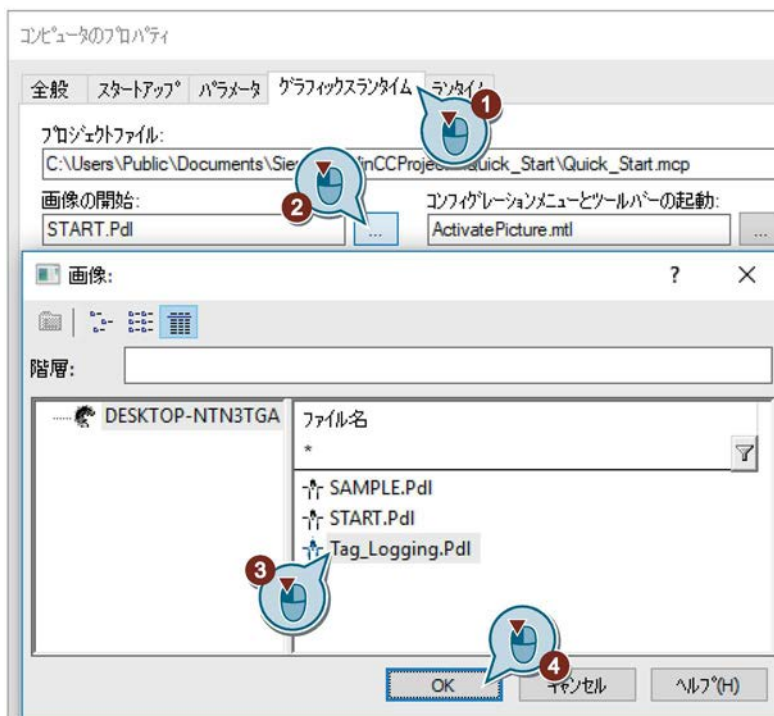
[コンピュータのプロパティ]ダイアログが開きます。

2. [スタートアップ]タブをクリックして、"レポートランタイム"アプリケーションを有効にします。



8.6 ランタイムプロパティの定義(レポートシステム)

3. [グラフィックランタイム]タブで、"Tag_Logging.pdl"プロセス画像を、ランタイムウインドウの初期画像として設定します。



4. [OK]をクリックして[コンピュータのプロパティ]ダイアログを終了します。

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、レポートランタイムが実行されます。開始画像として、"Tag_Logging.pdl"プロセス画像が表示されます。

次のステップで、"Quick_Start"プロジェクトを有効にし、[タグシミュレーション]エディタを使用します。

8.7 プロジェクトの有効化とテスト(レポートシステム)

はじめに

次のステップは、ロギング用に"Quick_Start"プロジェクトを有効化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"に値を割り当てます。

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。

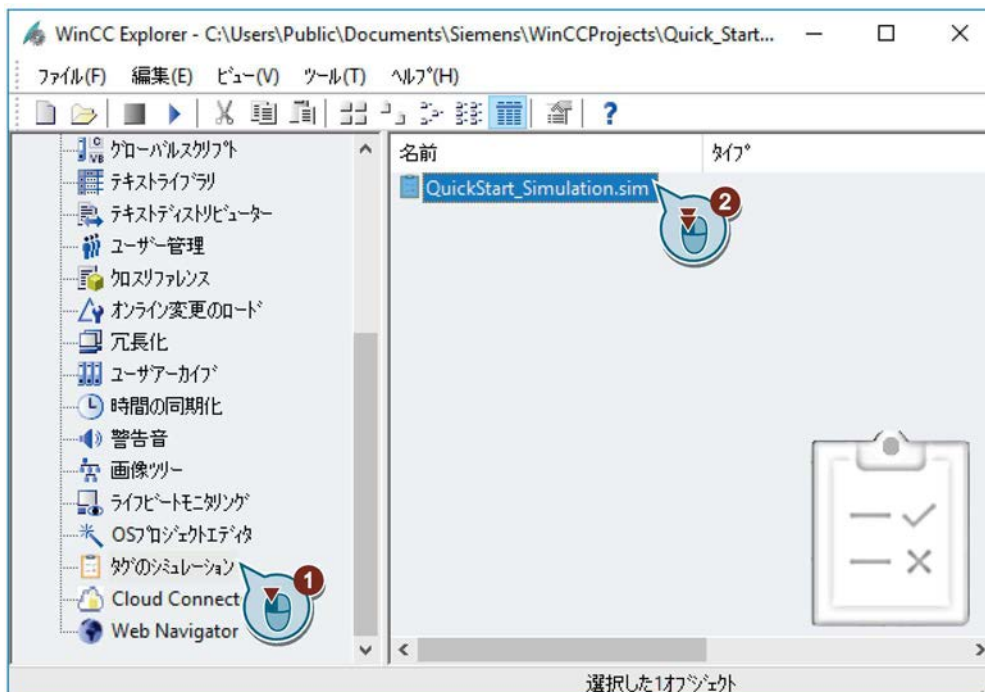
必要条件

- ランタイムプロパティが定義されている。

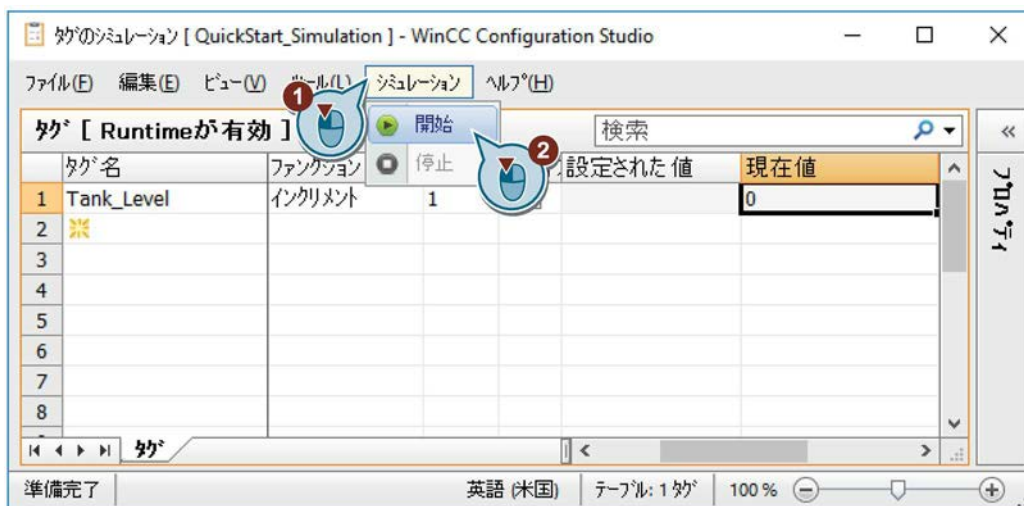
8.7 プロジェクトの有効化とテスト(レポートシステム)

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. [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。

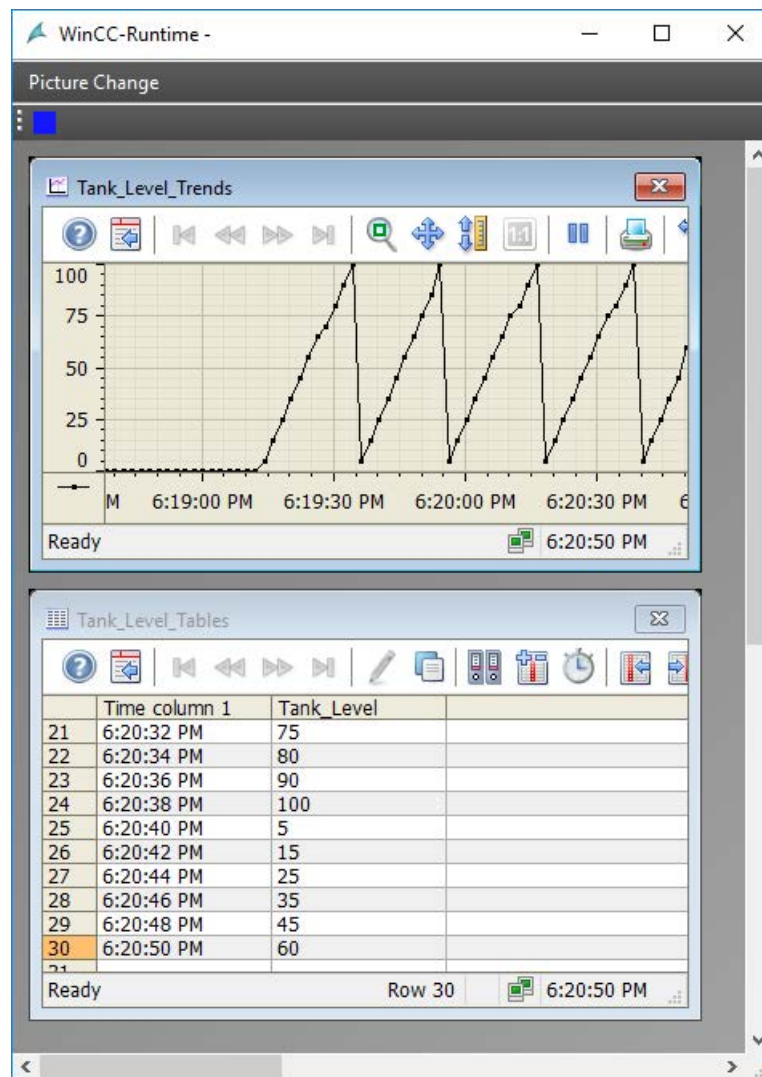


結果

"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。

内部タグ"Tank_Level"の推移が、[トレンド]ウィンドウおよび[テーブル]ウィンドウに表示されます。

次のステップで、[テーブル]ウィンドウの現在のビューの値を文書化するために、プロトコルを印刷します。



8.8 ログの印刷

はじめに

次のステップは、ログの印刷方法を示しています。

このセクションでは、[テーブル]ウィンドウの現在の表示の値が文書化されるログを印刷します。

これには、[テーブル]ウィンドウの[印刷]  ボタンを使用する必要があります。

ボタンを使用するために、 ボタンによってデータの更新を停止します。

データはクリップボードに保存され、ボタンを再びクリックすると追加されます。

ランタイムで  ボタンを押すと、印刷ジョブ"@OnlineTableControl - 表"が実行され、ログが印刷されます。

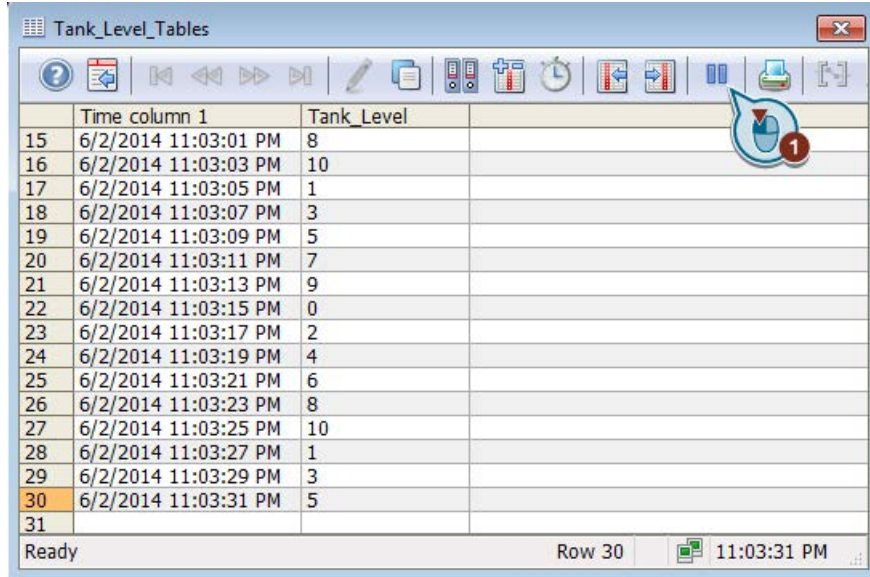
ログは、ページレイアウト"Tag_Logging.rpl"をベースにしています。

必要条件

- "Quick_Start"プロジェクトが有効である。
- [タグシミュレーション]エディタでシミュレーションが起動されている。
- "@OnlineTableControl - Table"印刷ジョブが[Tag_Logging.rpl]ページレイアウトとリンク付けされている。

手順

1.  ボタンによって、データ更新を停止します。

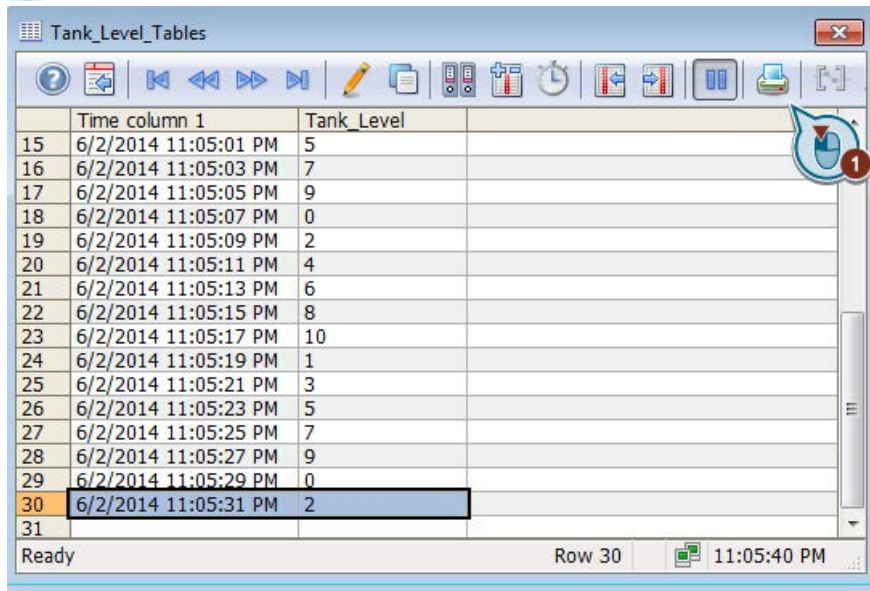


	Time column 1	Tank_Level
15	6/2/2014 11:03:01 PM	8
16	6/2/2014 11:03:03 PM	10
17	6/2/2014 11:03:05 PM	1
18	6/2/2014 11:03:07 PM	3
19	6/2/2014 11:03:09 PM	5
20	6/2/2014 11:03:11 PM	7
21	6/2/2014 11:03:13 PM	9
22	6/2/2014 11:03:15 PM	0
23	6/2/2014 11:03:17 PM	2
24	6/2/2014 11:03:19 PM	4
25	6/2/2014 11:03:21 PM	6
26	6/2/2014 11:03:23 PM	8
27	6/2/2014 11:03:25 PM	10
28	6/2/2014 11:03:27 PM	1
29	6/2/2014 11:03:29 PM	3
30	6/2/2014 11:03:31 PM	5
31		

Ready Row 30 11:03:31 PM

データ更新の表示が停止します。

2.  ボタンをクリックして、ログを印刷します。



	Time column 1	Tank_Level
15	6/2/2014 11:05:01 PM	5
16	6/2/2014 11:05:03 PM	7
17	6/2/2014 11:05:05 PM	9
18	6/2/2014 11:05:07 PM	0
19	6/2/2014 11:05:09 PM	2
20	6/2/2014 11:05:11 PM	4
21	6/2/2014 11:05:13 PM	6
22	6/2/2014 11:05:15 PM	8
23	6/2/2014 11:05:17 PM	10
24	6/2/2014 11:05:19 PM	1
25	6/2/2014 11:05:21 PM	3
26	6/2/2014 11:05:23 PM	5
27	6/2/2014 11:05:25 PM	7
28	6/2/2014 11:05:27 PM	9
29	6/2/2014 11:05:29 PM	0
30	6/2/2014 11:05:31 PM	2
31		

Ready Row 30 11:05:40 PM

ログが印刷されるか、印刷可能ファイルとして保存されます。

3. シミュレーションおよびWinCC Runtimeを終了します。

ログが印刷されるだけです。

SELECT * FROM DEMO.DEMO_PROJECTS ORDER BY PROJECT_CODE, START_DATE		
1	Project Name	Project Code
2	Project Name	Project Code
3	Project Name	Project Code
4	Project Name	Project Code
5	Project Name	Project Code
6	Project Name	Project Code
7	Project Name	Project Code
8	Project Name	Project Code
9	Project Name	Project Code
10	Project Name	Project Code
11	Project Name	Project Code
12	Project Name	Project Code
13	Project Name	Project Code
14	Project Name	Project Code
15	Project Name	Project Code
16	Project Name	Project Code
17	Project Name	Project Code
18	Project Name	Project Code
19	Project Name	Project Code
20	Project Name	Project Code
21	Project Name	Project Code
22	Project Name	Project Code
23	Project Name	Project Code
24	Project Name	Project Code
25	Project Name	Project Code
26	Project Name	Project Code
27	Project Name	Project Code
28	Project Name	Project Code
29	Project Name	Project Code
30	Project Name	Project Code
31	Project Name	Project Code
32	Project Name	Project Code
33	Project Name	Project Code
34	Project Name	Project Code
35	Project Name	Project Code
36	Project Name	Project Code
37	Project Name	Project Code
38	Project Name	Project Code
39	Project Name	Project Code
40	Project Name	Project Code
41	Project Name	Project Code
42	Project Name	Project Code
43	Project Name	Project Code
44	Project Name	Project Code
45	Project Name	Project Code
46	Project Name	Project Code
47	Project Name	Project Code
48	Project Name	Project Code
49	Project Name	Project Code
50	Project Name	Project Code
51	Project Name	Project Code
52	Project Name	Project Code
53	Project Name	Project Code
54	Project Name	Project Code
55	Project Name	Project Code
56	Project Name	Project Code
57	Project Name	Project Code
58	Project Name	Project Code
59	Project Name	Project Code
60	Project Name	Project Code
61	Project Name	Project Code
62	Project Name	Project Code
63	Project Name	Project Code
64	Project Name	Project Code
65	Project Name	Project Code
66	Project Name	Project Code
67	Project Name	Project Code
68	Project Name	Project Code
69	Project Name	Project Code
70	Project Name	Project Code
71	Project Name	Project Code
72	Project Name	Project Code
73	Project Name	Project Code
74	Project Name	Project Code
75	Project Name	Project Code
76	Project Name	Project Code
77	Project Name	Project Code
78	Project Name	Project Code
79	Project Name	Project Code
80	Project Name	Project Code
81	Project Name	Project Code
82	Project Name	Project Code
83	Project Name	Project Code
84	Project Name	Project Code
85	Project Name	Project Code
86	Project Name	Project Code
87	Project Name	Project Code
88	Project Name	Project Code
89	Project Name	Project Code
90	Project Name	Project Code
91	Project Name	Project Code
92	Project Name	Project Code
93	Project Name	Project Code
94	Project Name	Project Code
95	Project Name	Project Code
96	Project Name	Project Code
97	Project Name	Project Code
98	Project Name	Project Code
99	Project Name	Project Code
100	Project Name	Project Code

メッセージの設定

9.1 メッセージの設定

はじめに

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。

[アラームロギング]エディタのアラームロギング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值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。

9.2 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。

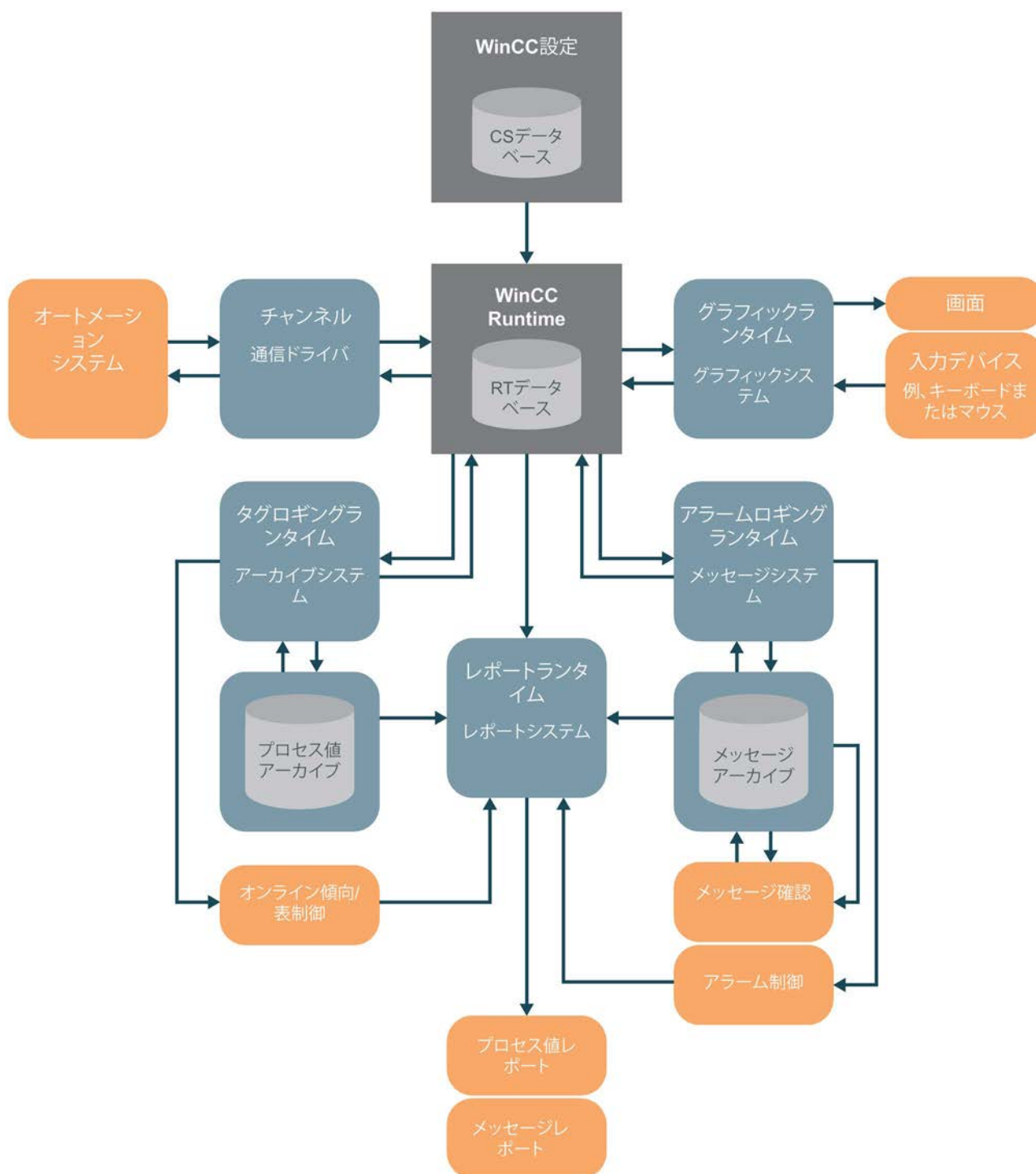
特定のステータスやプロセスにおける変化によって、メッセージシステムはメッセージを生成し、それらをランタイムのテーブルに出力します。メッセージは、危機的な状況を初期段階で認識し、ダウンタイムを回避するのに役立ちます。

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



9.3 アラームロギングの開始

はじめに

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。

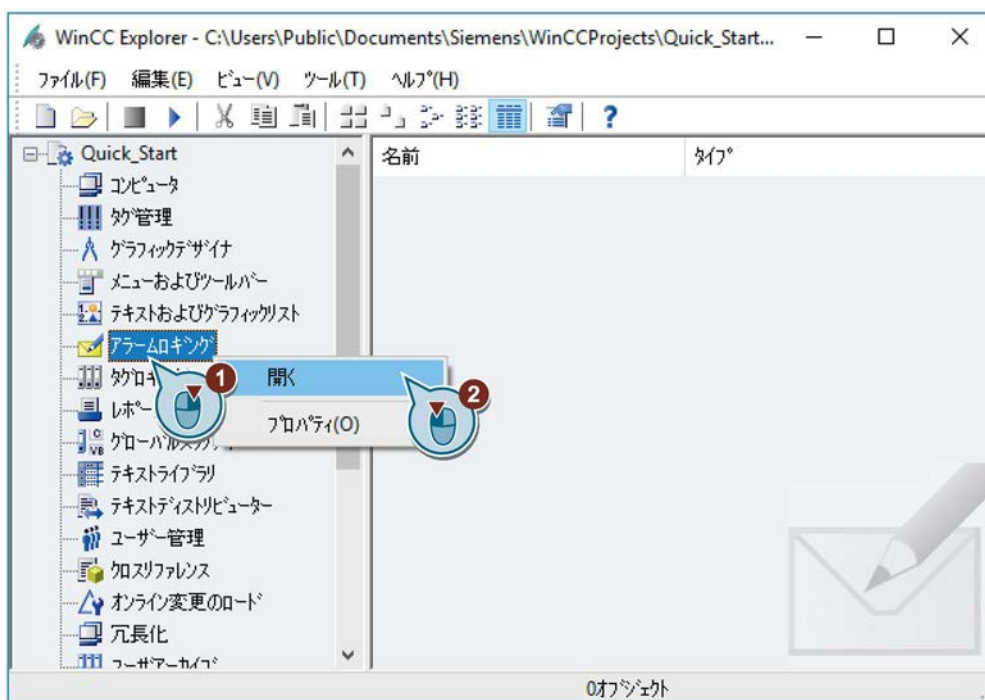
"アラームロギング"エディタで、"Quick_Start"プロジェクトに必要な全てのビットメッセージおよびアナログメッセージを設定します。

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。

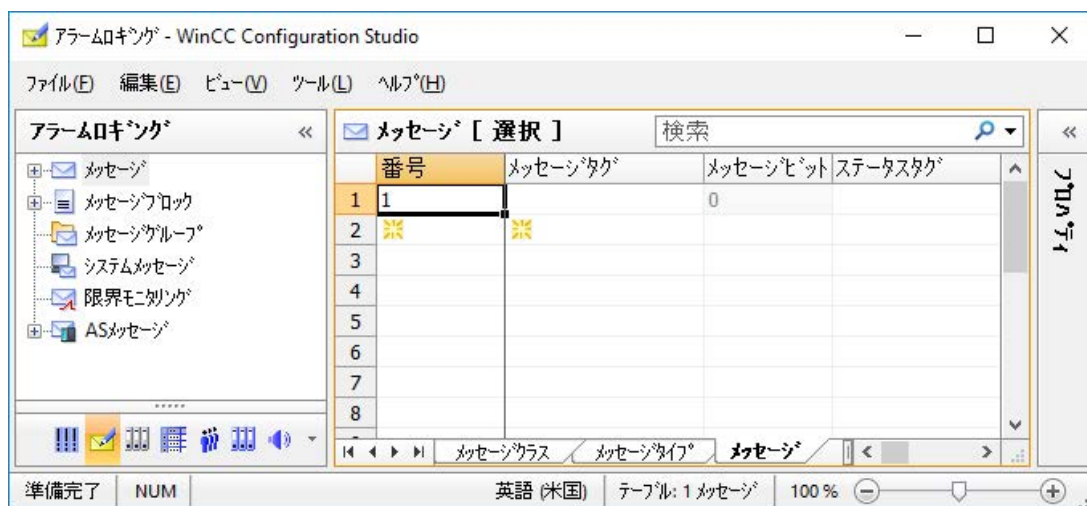


[アラームロギング]エディタが開きます。

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。



9.4 メッセージブロックの設定

はじめに

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。

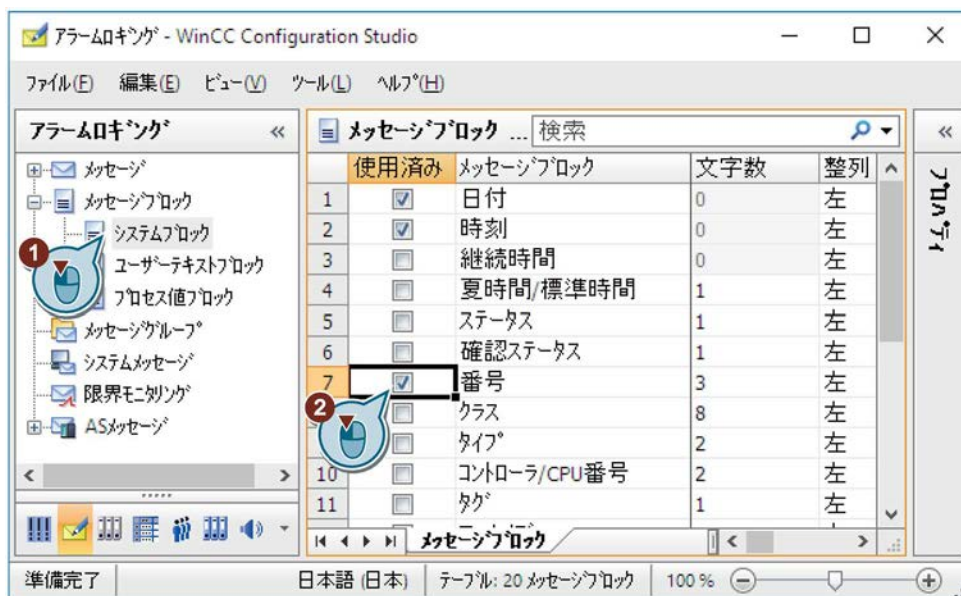
- システムデータ(例:
日付、時刻、メッセージ番号、ステータス)を有するシステムブロック。
システムブロックは、事前定義されています。
- ユーザーテキストブロックには、説明テキスト(例:
ロケーションおよび障害の原因に関する情報のあるテキスト)があります。
テキストは自由にカスタマイズできます。
- メッセージをプロセス値(例:
現在のフィルレベル、温度、速度)に接続するために使用するプロセス値ブロック。
メッセージブロックをランタイムに表示するためにプロパティを変更できます。
"Quick_Start"プロジェクトに使用されるユーザーテキストブロックの長さを変更します。

必要条件

- [アラームロギング]エディタが開いている。

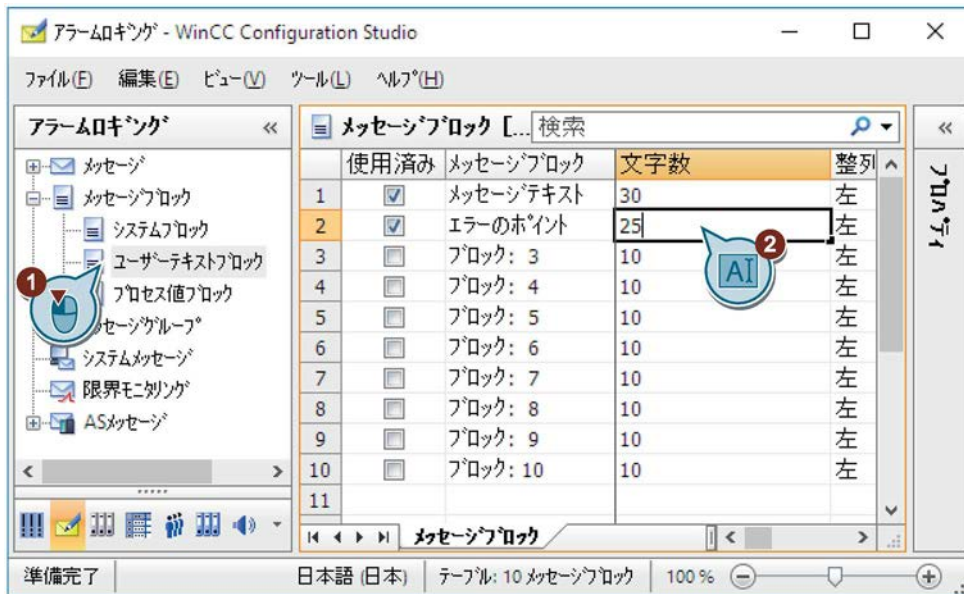
手順

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"
 - "時刻"
 - "番号"



9.4 メッセージブロックの設定

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。

次のステップで、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。

9.5 ビットメッセージの設定

9.5.1 ビットメッセージの設定

はじめに

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。

9.5.2 ビットメッセージの作成

はじめに

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 42)」で列挙されているステップに従います。

手順

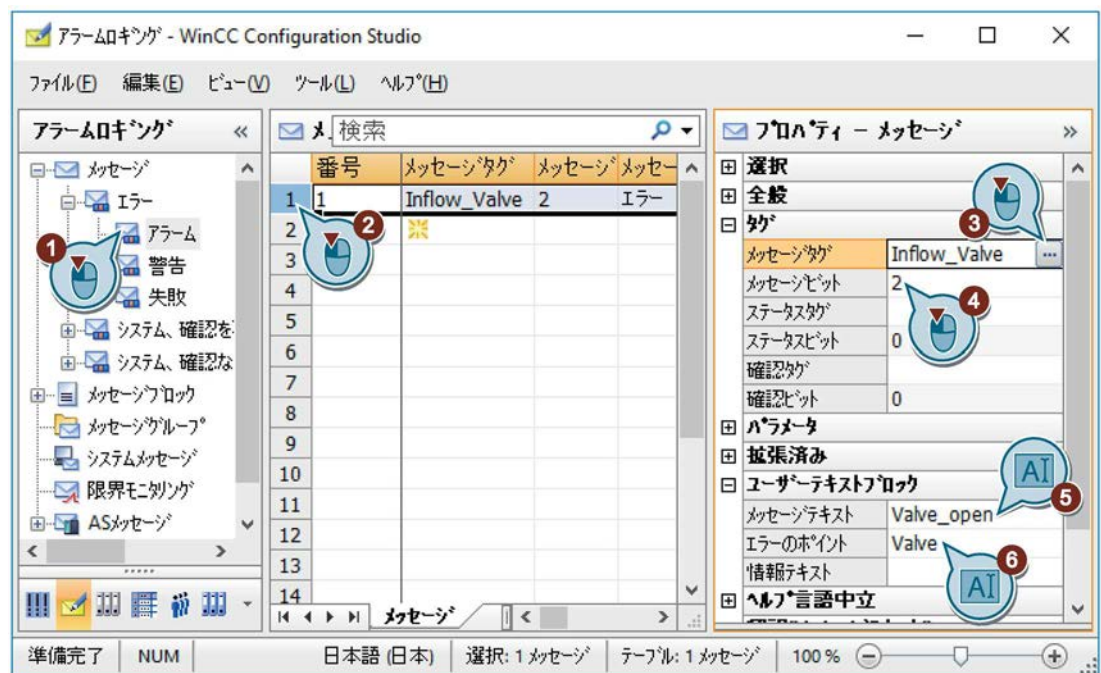
1. ナビゲーションエリアの[メッセージ]フォルダで、[エラー]の下にある[アラーム]メッセージタイプを選択します。

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同様に作成します。

テーブルエリアで、それぞれの状況について、次の空の行の[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。

9.5 ビットメッセージの設定

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"

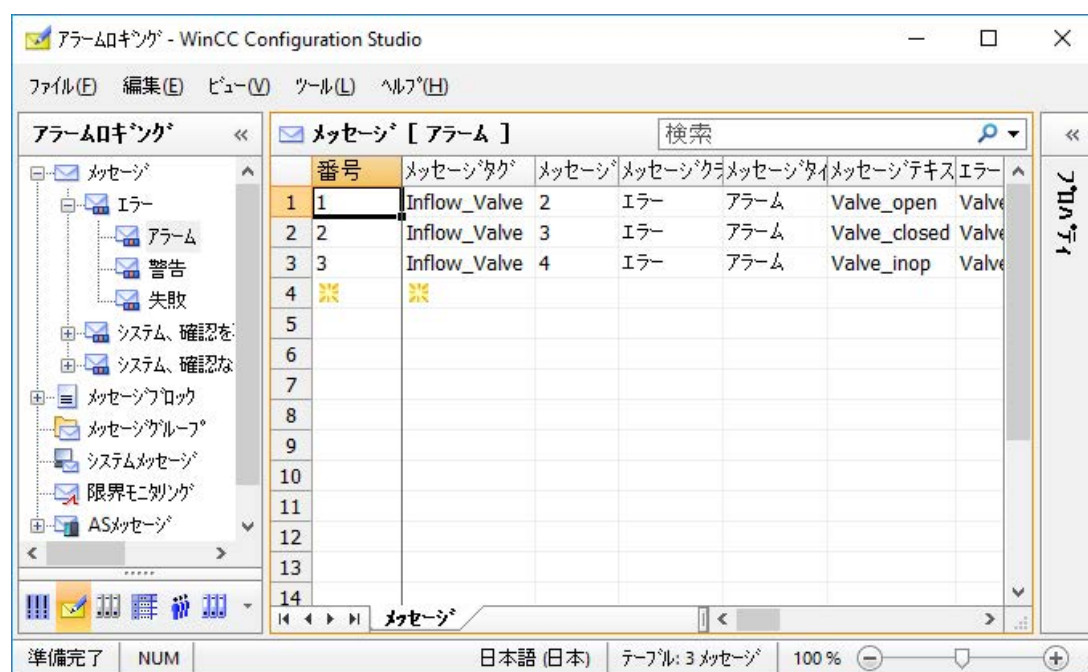
結果

"Quick_Start"プロジェクトのプロパティを持つディスクリットアラームを定義しました。

作成されたディスクリットアラームは、[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ットアラームの出力はメッセージの設定で制御されています。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、ディスクリットアラーム"Valve_open"がトリガされます。

次のステップで、アナログメッセージを2つ設定して、内部タグ"Tank_Level"の動作をモニタします。



9.6 アナログメッセージの設定

9.6.1 アナログメッセージの設定

はじめに

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。

9.6.2 限界値の設定

はじめに

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"の上限值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。



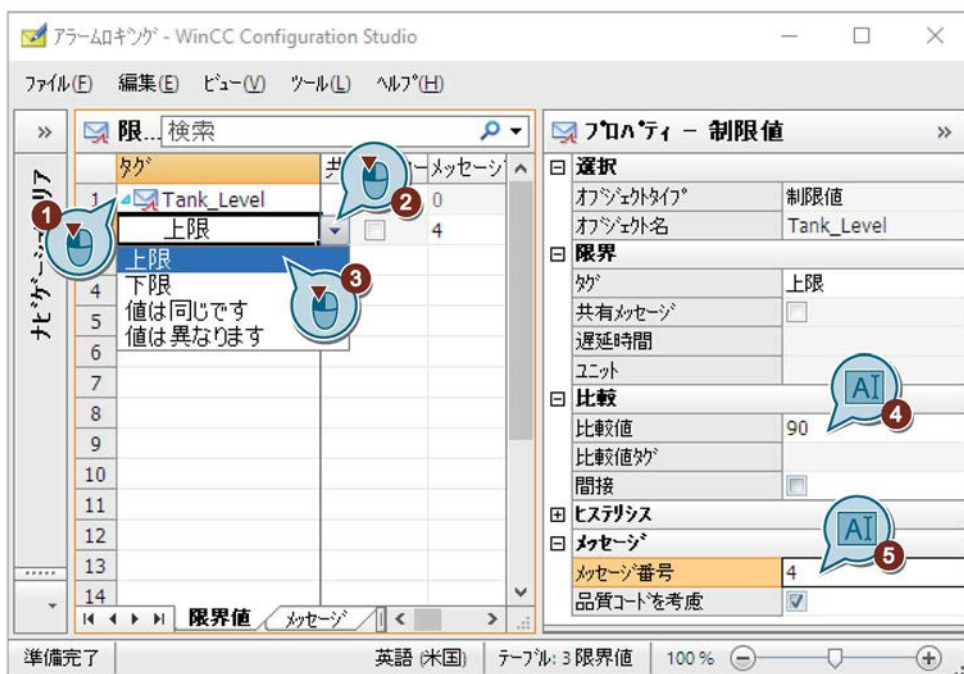
9.6 アナログメッセージの設定

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。

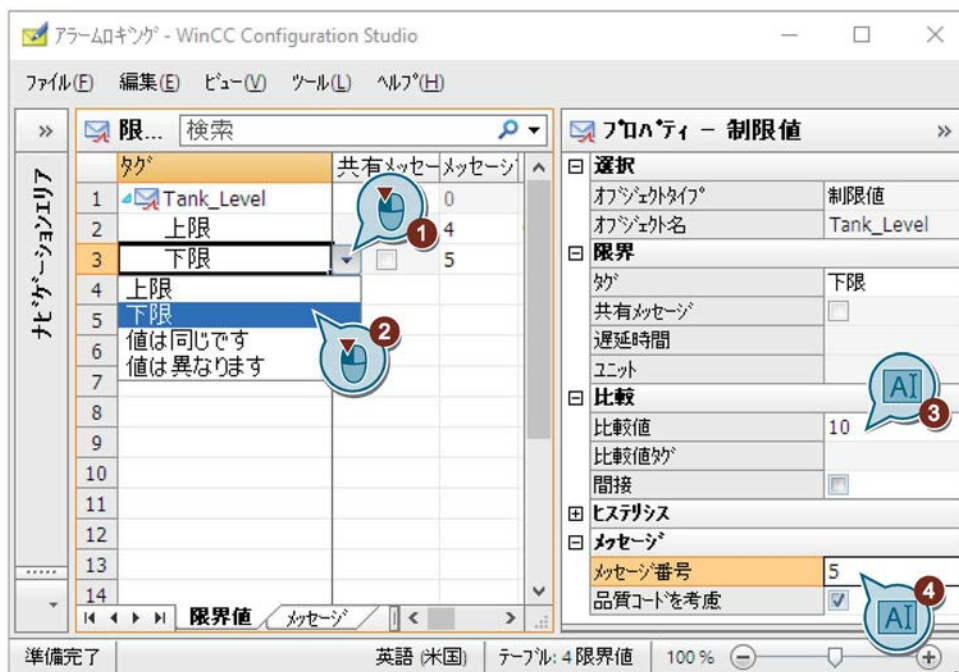
4. 上限値のプロパティを[プロパティ - 制限値]エリアまたはテーブルエリアに入力します。

新しい、未使用のメッセージ番号を使用します。例、"4"。



- 2番目の制限値のデータを入力するには、タグの下次の行で[下限]エントリを選択します。
- 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"。



9.6 アナログメッセージの設定

結果

内部タグ"Tank_Level"の制限値が定義されています。

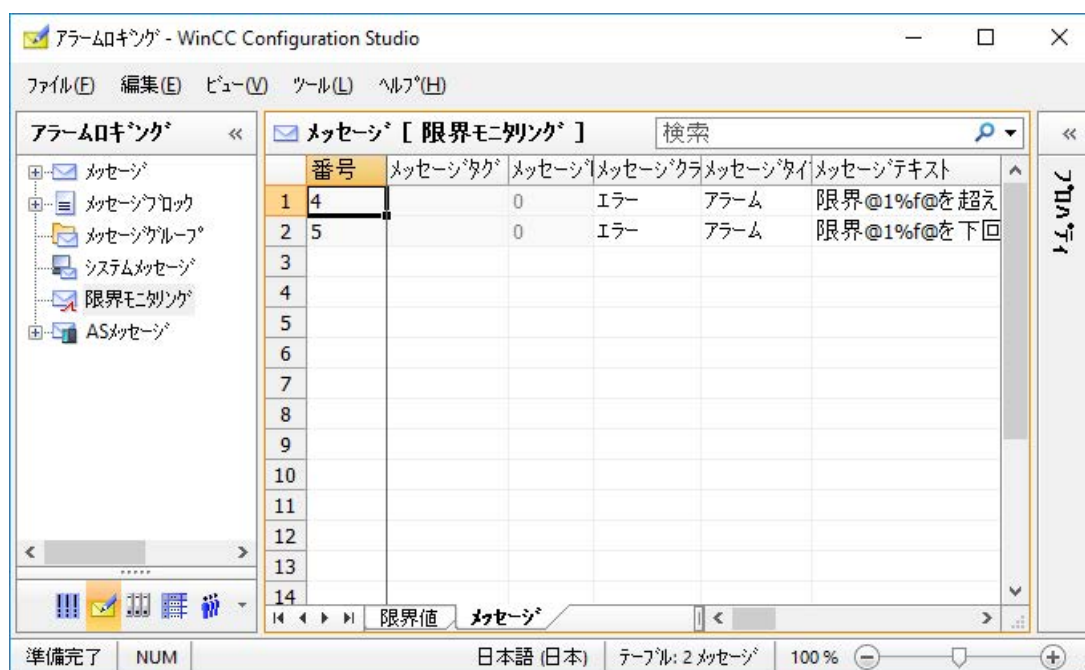
- タグ値が90より大きい場合、アナログメッセージ[上限値]がトリガされ、ランタイムで表示されます。

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。



次のステップで、WinCC OnlineTrendControlでの制限値違反の表示を設定します。

9.6.3 トレンドウィンドウでのメッセージの表示

はじめに

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。

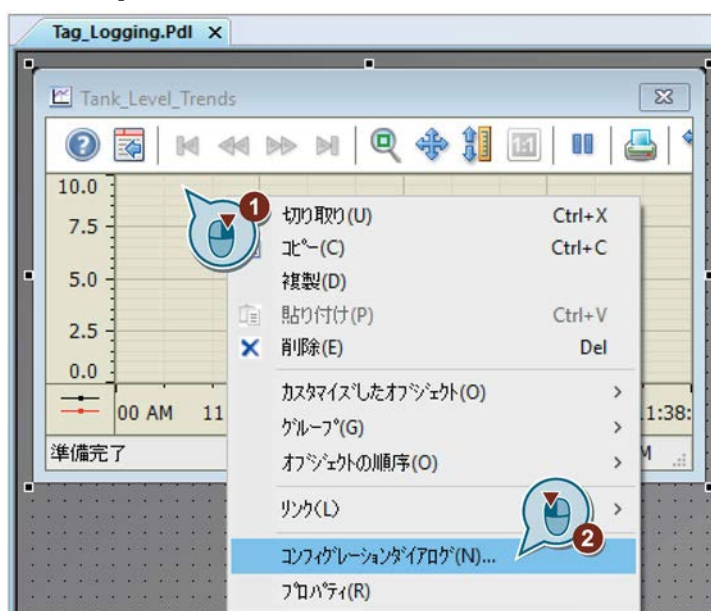
必要条件

- 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。

"限界値の設定 (ページ 205)"

手順

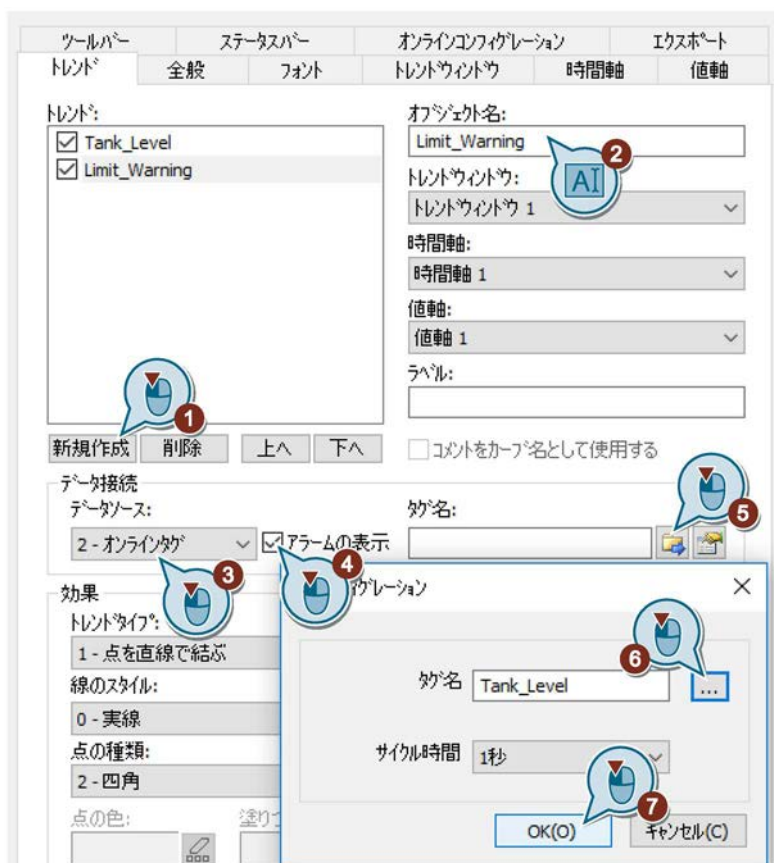
- WinCC OnlineTrendControl [Tank_Level_Trends]を選択します。
- ショートカットメニューを使用して、[WinCCオンライントレンドコントロールのプロパティ]設定ダイアログを開きます。



9.6 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。

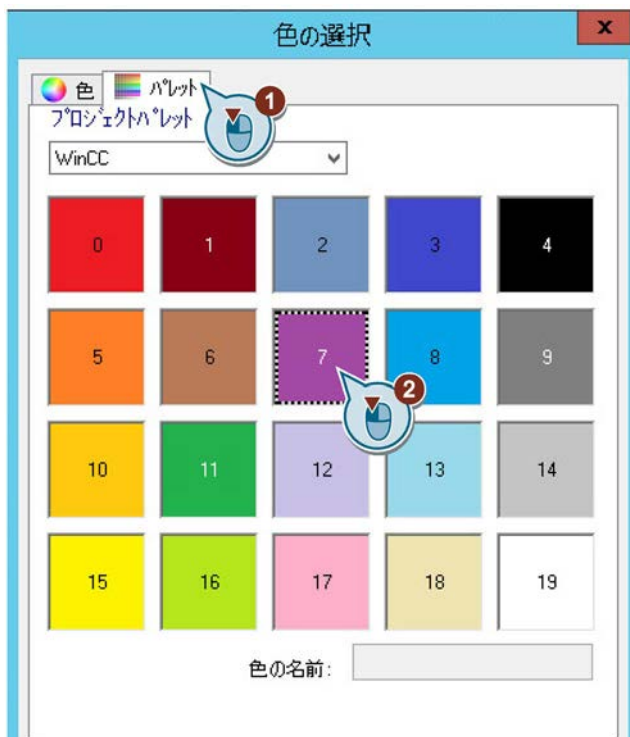


4. 下限のトレンドの色変更を設定します。



9.6 アナログメッセージの設定

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。



5. 上限および不特定なステータスの値のトレンドの色変更を設定します。

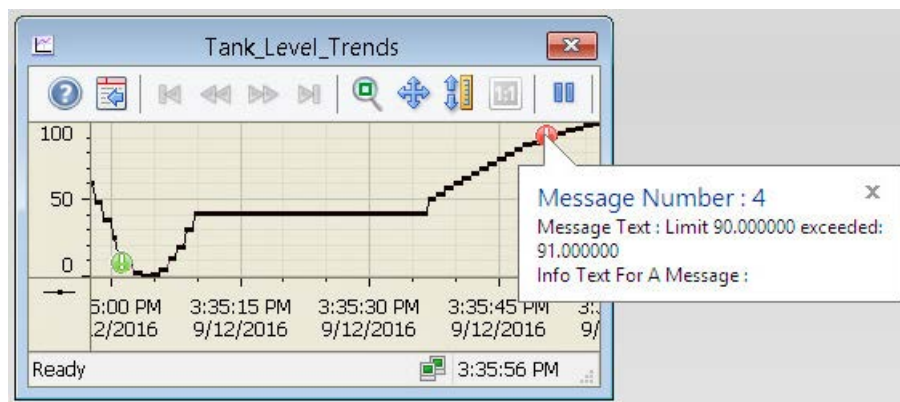
結果

[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。



次のステップで、ランタイムでのメッセージ状態の表示色を定義します。

9.7 メッセージステータスの色の定義

はじめに

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。

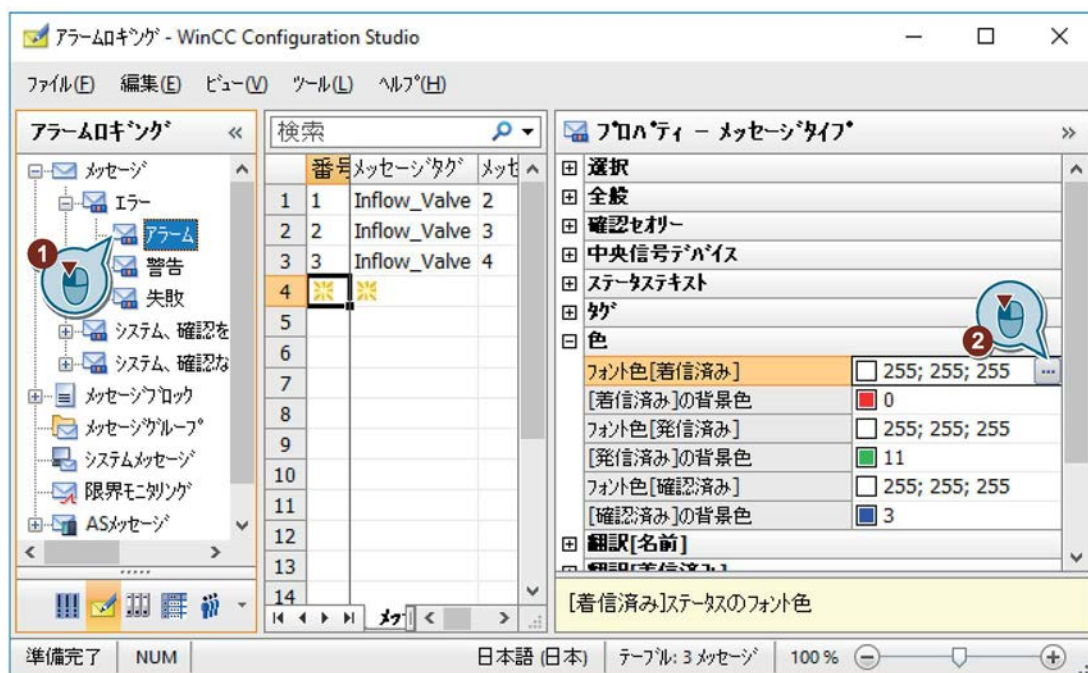
"Quick_Start"プロジェクトで、3つの基本ステータスに対して異なるテキスト色と背景色を定義します。この定義は、メッセージクラス[エラー]のメッセージタイプ[アラーム]全体に適用されます。これは、設定を"Quick_Start"プロジェクトの全てのメッセージに適用します。

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。

9.7 メッセージステータスの色の定義

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、メッセージはそれぞれのステータスに応じてそれぞれの色で表示されます。

次のステップで、"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、ランタイム中にメッセージをテーブル形式で表示させます。

9.8 プロセス画面の設定

9.8.1 プロセス画像の設定(アラームロギング)

はじめに

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"と接続します。

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。

9.8.2 アラームメッセージウィンドウの設定

はじめに

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。

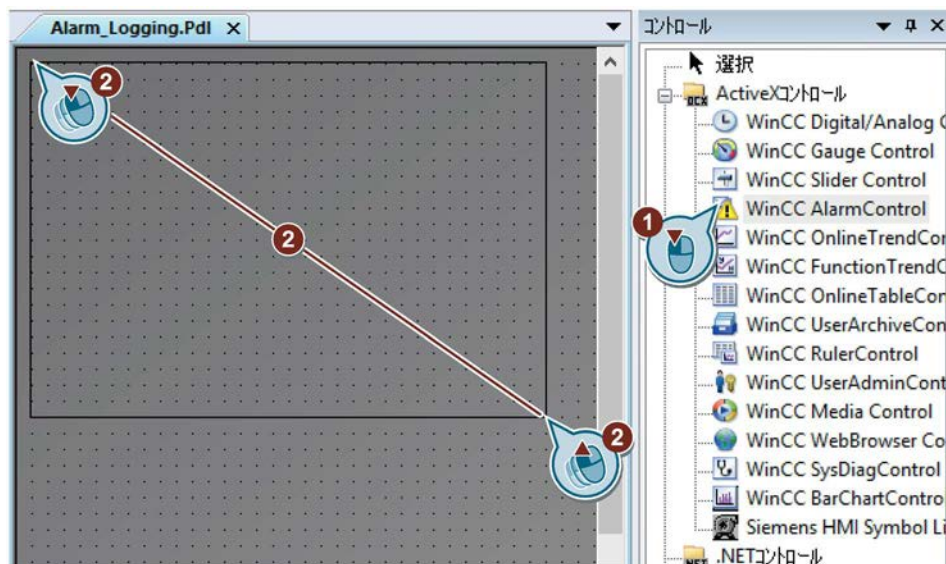
ランタイムで、メッセージはこれらのメッセージブロックで構成されます。

必要条件

- "Quick_Start"プロジェクトが開いている。

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。

3. メッセージウィンドウの名前を指定します。

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		

ウィンドウ

ウィンドウタイトル:

1 - 通常

テキスト

Wastewater

☒ 移動可能

☒ 加え可能

☒ サイズ変更可能

スタイル:

プロジェクト設定

行スクロールバー:

1 - 必要時

列スクロールバー:

1 - 必要時

現在の印刷ジョブの表示

AlarmControl - Table

タイムベース

2 - プロジェクト設定

プロパティ

開いたセグメントについてのアクティビリスト

0 - メッセージリスト

サーバーの選択

localhost;

☒ すべてのサーバー

☒ メッセージ色の表示

☒ 自動スクローリング

デフォルトのソート:

0 - 昇順

表示オプション

メッセージの表示:

1 - 表示メッセージのみ

長期アーカイブリストのページ

☐ ページングの有効化

ページ当たりのメッセージ:

50

ダブルクリック時にアクション

4 - 列依存

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ギュレーション 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

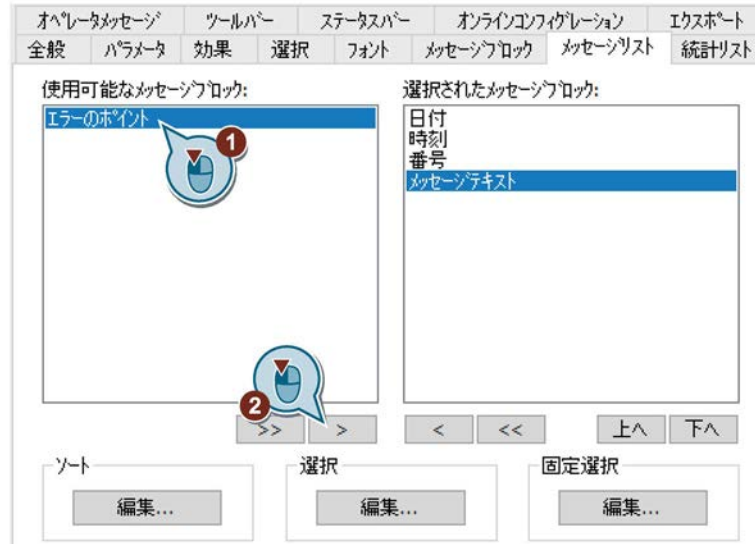
時間の形式:

日付の形式: dd/MM/yy

☒ 日付の表示

☒ プロジェクト設定の適用

5. [メッセージリスト]タブをクリックして、[メッセージテキスト]および[エラーのポイント]ユーザーテキストブロックを[選択した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。

6. [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"プロジェクトで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

次のステップで、スライダオブジェクトを設定して、内部タグ"Tank_Level"にアナログ値を提供します。

	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	TEXT
92	04/10/18	04:43:49 午 92	TEXT	TEXT	TEXT
93	04/10/18	04:43:50 午 93	TEXT	TEXT	TEXT
94	04/10/18	04:43:51 午 94	TEXT	TEXT	TEXT
95	04/10/18	04:43:52 午 95	TEXT	TEXT	TEXT
96	04/10/18	04:43:53 午 96	TEXT	TEXT	TEXT
97	04/10/18	04:43:54 午 97	TEXT	TEXT	TEXT
98	04/10/18	04:43:55 午 98	TEXT	TEXT	TEXT
99	04/10/18	04:43:56 午 99	TEXT	TEXT	TEXT
100	04/10/18	04:43:57 午 100	TEXT	TEXT	TEXT

準備完了 保留中: 0 未確認: 0 非表示: 0 リスト: 100 16:42:24

9.8.3 スライドオブジェクトの挿入とダイナミック化

はじめに

次のステップは、スライド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。

スライドオブジェクトは、タグ値の表示や変更のために使用します。スライド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、プロセス画像"Alarm_Logging.pdl"にスライド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、スライド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ド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。

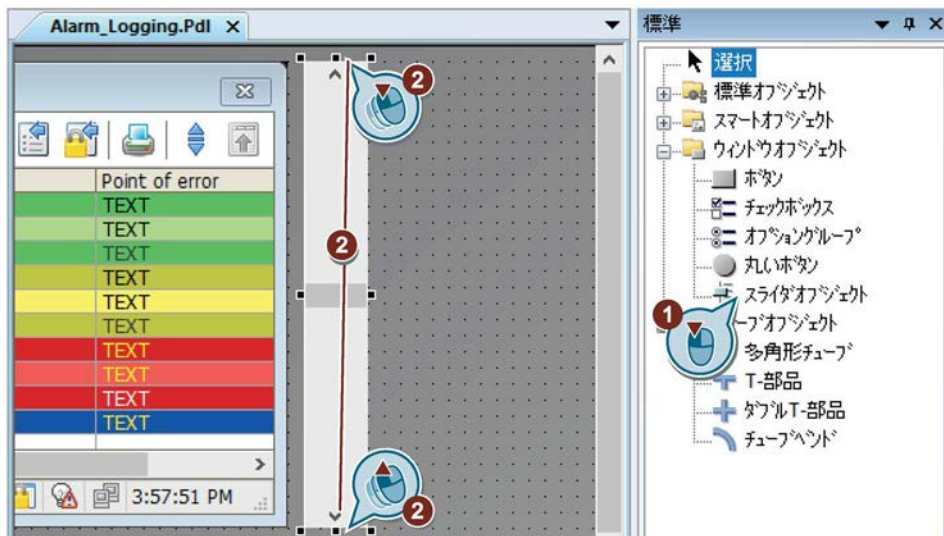
"Quick_Start"プロジェクトに対して、スライドオブジェクトのプリセットプロパティを変更します。[オブジェクトプロパティ]ウィンドウで、スライドオブジェクトの新しい名前を入力し、その高さを定義します。

必要条件

- "グラフィックデザイナー"エディタが開いている。
- プロセス画像"Alarm_Logging.pdl"が作成されている。
- 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。



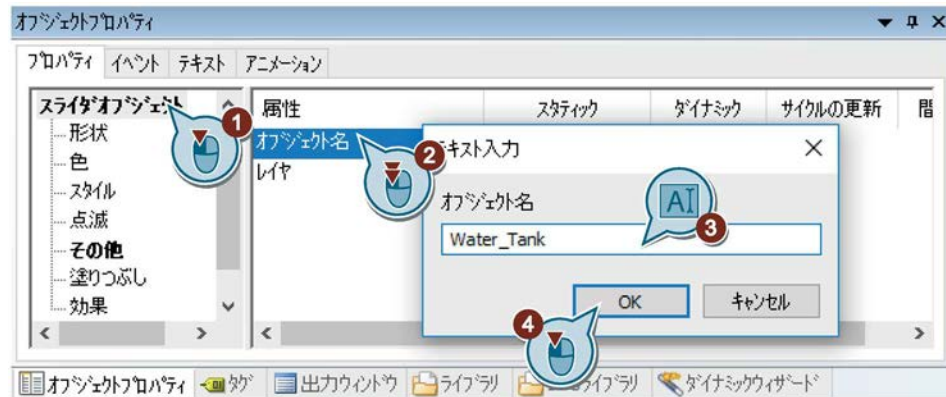
[スライダ設定]ダイアログが開きます。

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。

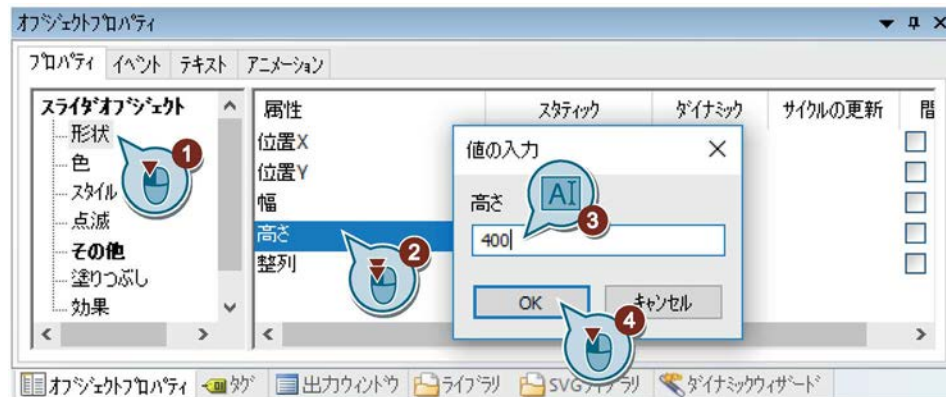


3. [オブジェクトプロパティ]ウィンドウに切り替えます。

スライダオブジェクトの名前として、"Water_Tank"を入力します。



4. 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。

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、次のステップでスケールを挿入します。

9.8.4 スケールの挿入

はじめに

次のステップは、"グラフィックデザイナー"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、プロセス画像"Alarm_Logging.pdl"にスケールを挿入します。

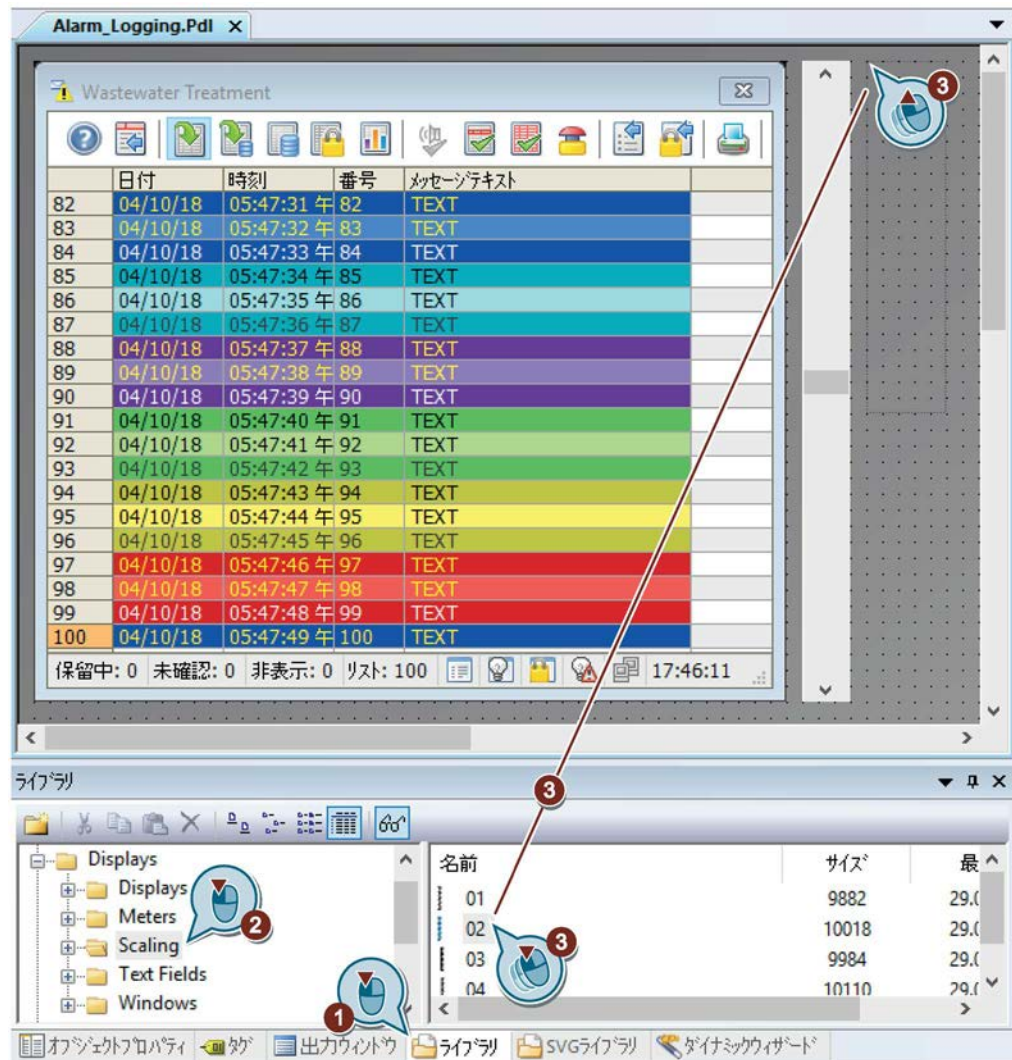
このスケールによって、スライダオブジェクト"Water_Tank"が想定できる値を表示します。スケール上のラインは、スライダオブジェクトの操作ステップに対応します。

必要条件

- プロセス画像"Alarm_Logging.pdl"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。

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。

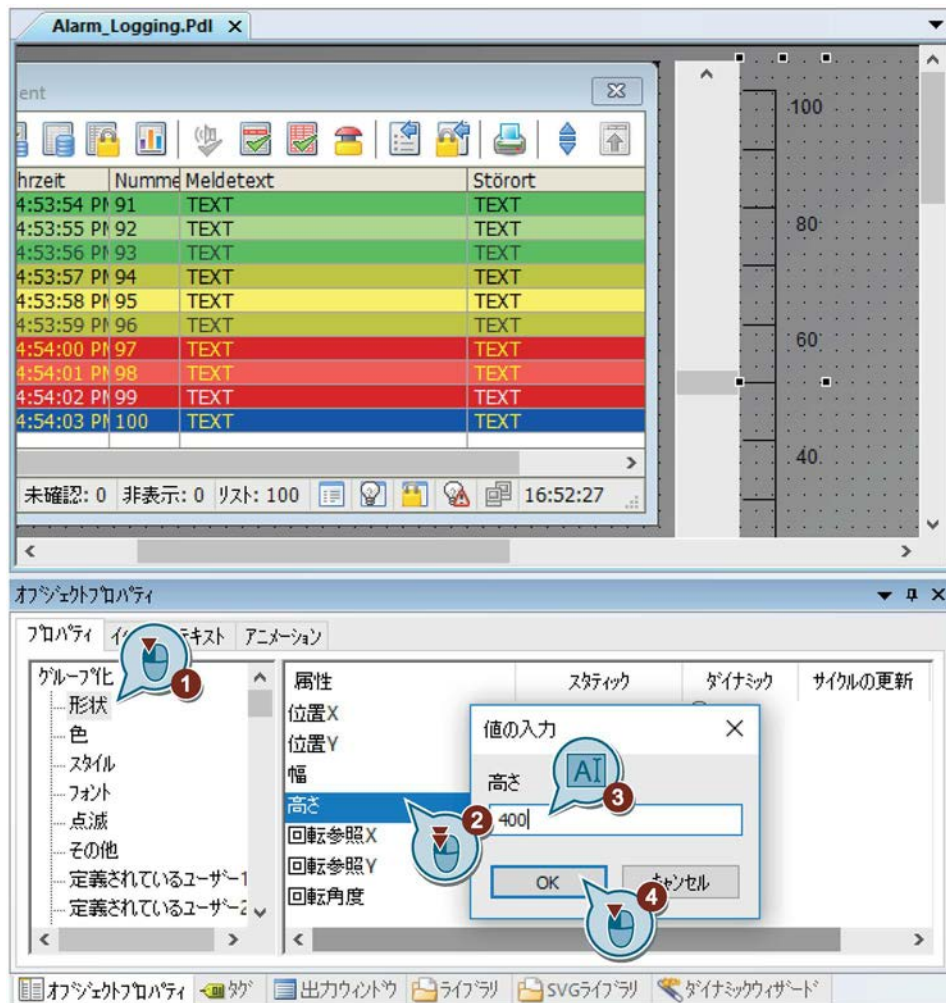


スケールは、プロセス画像"Alarm_Logging.pdl"に表示されます。

9.8 プロセス画面の設定

3. [オブジェクトプロパティ]に切り替えます。

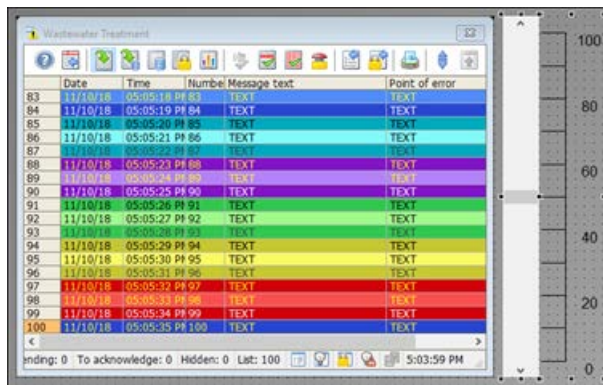
スケールの高さとして、[400]と定義します。



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、プロセス画像"Alarm_Logging.pdl"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。

次のステップで、I/Oフィールドを挿入して、内部タグ"Inflow_Valve"を2進数値と共に提供します。

9.8.5 I/Oフィールドの挿入とダイナミック化(アラームロギング)

はじめに

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、プロセス画像"Alarm_Logging.pdl"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。

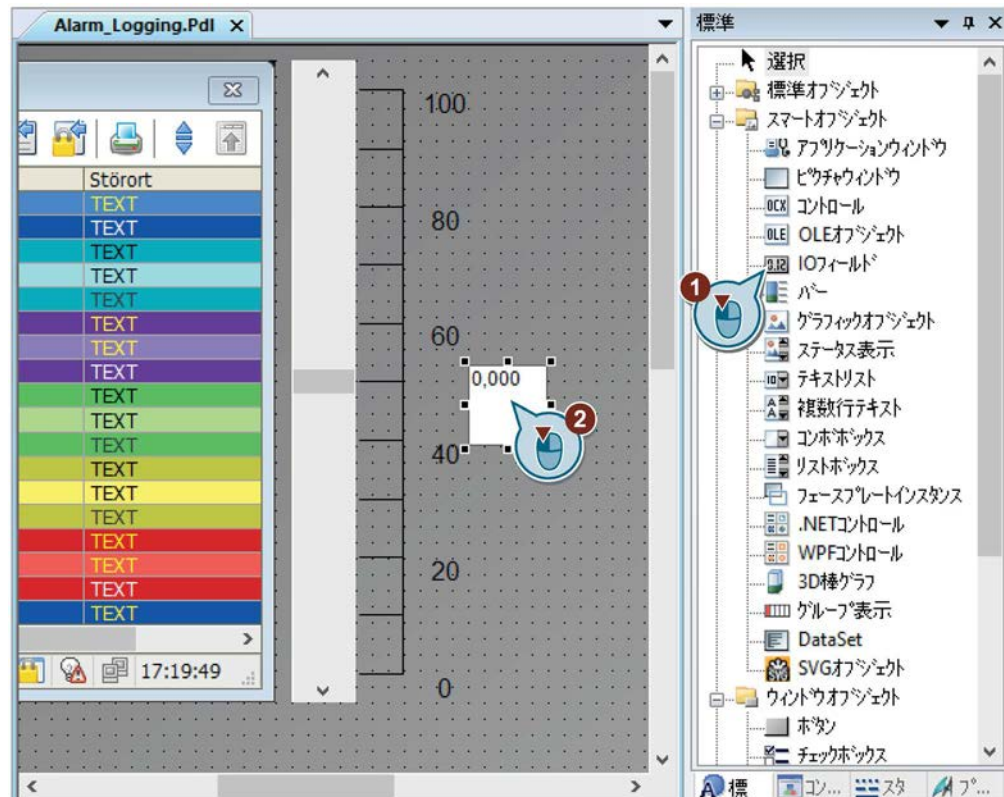
I/Oフィールドは、ランタイム中にバイナリ値を内部[Inflow_Valve]タグに転送するために使用されます。

必要条件

- プロセス画像"Alarm_Logging.pdl"が開いている。
- 内部タグ"Inflow_Valve"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。



9.8 プロセス画面の設定

3. [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。



4. 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位置調整:右揃え
- Y位置調整:中央揃え



5. 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。



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、プロセス画像"Alarm_Logging.pdl"を保存します。
8. [グラフィックデザイナー]を閉じます。

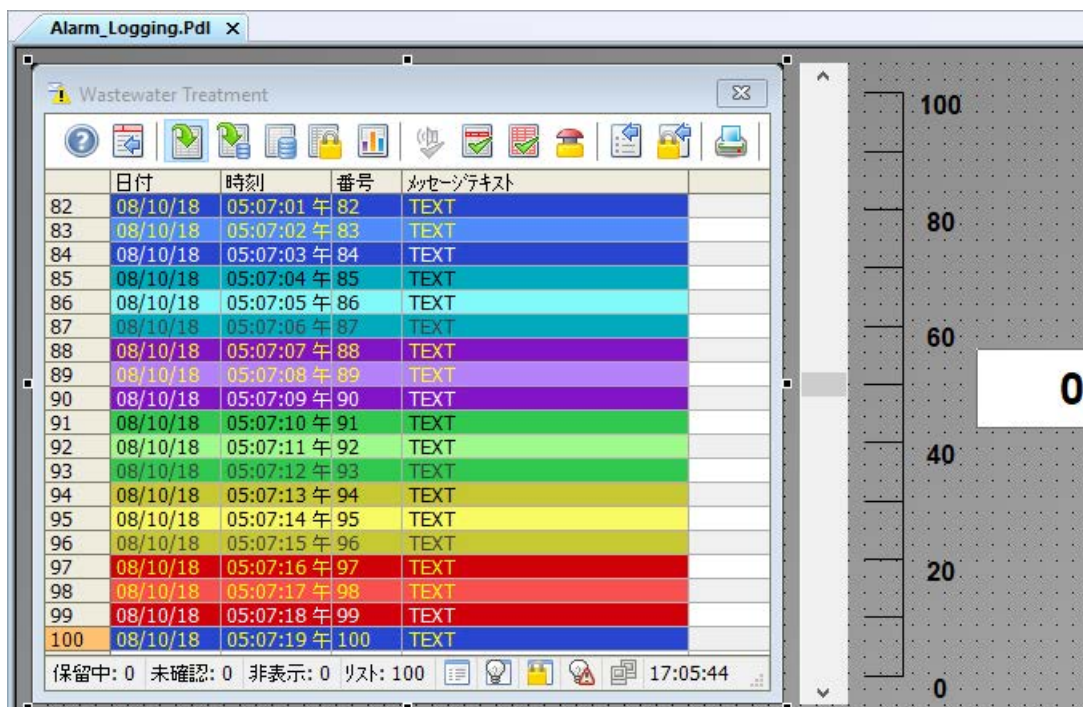
結果

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。

次のステップで、WinCC Runtimeのプロパティを定義し、ランタイムでのメッセージの出力を確認します。



9.9 ユーザー定義メニューのカスタマイズ(アラームロギング)

はじめに

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。

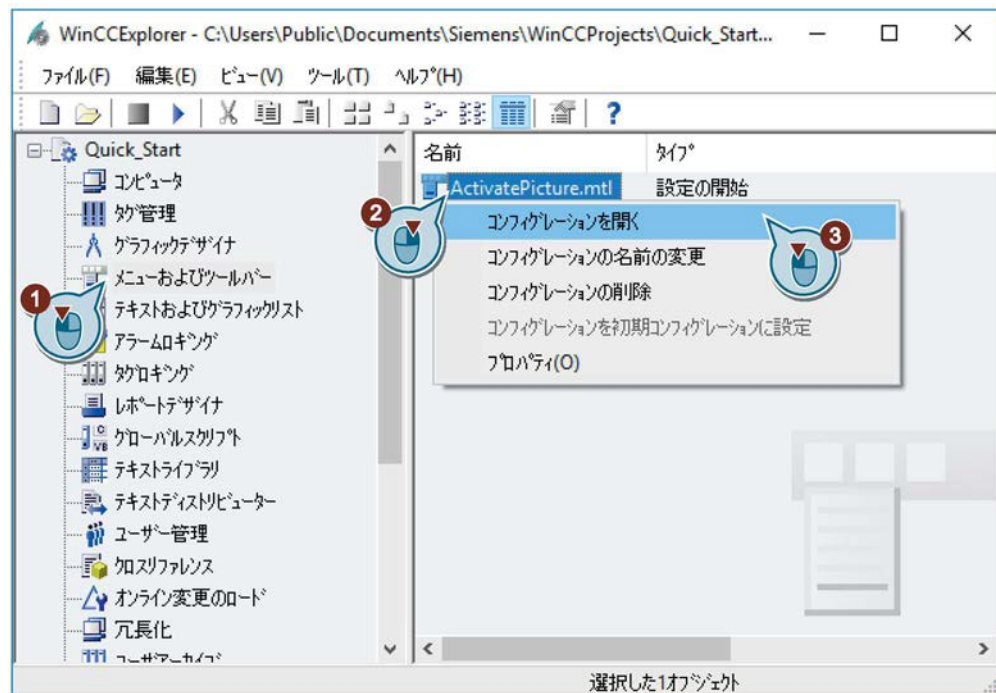
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- カスタマイズされた[画像変更]メニューがプロセス画像"START.pdl"、"SAMPLE.pdl"および"Tag_Logging.pdl"用に作成されていること。

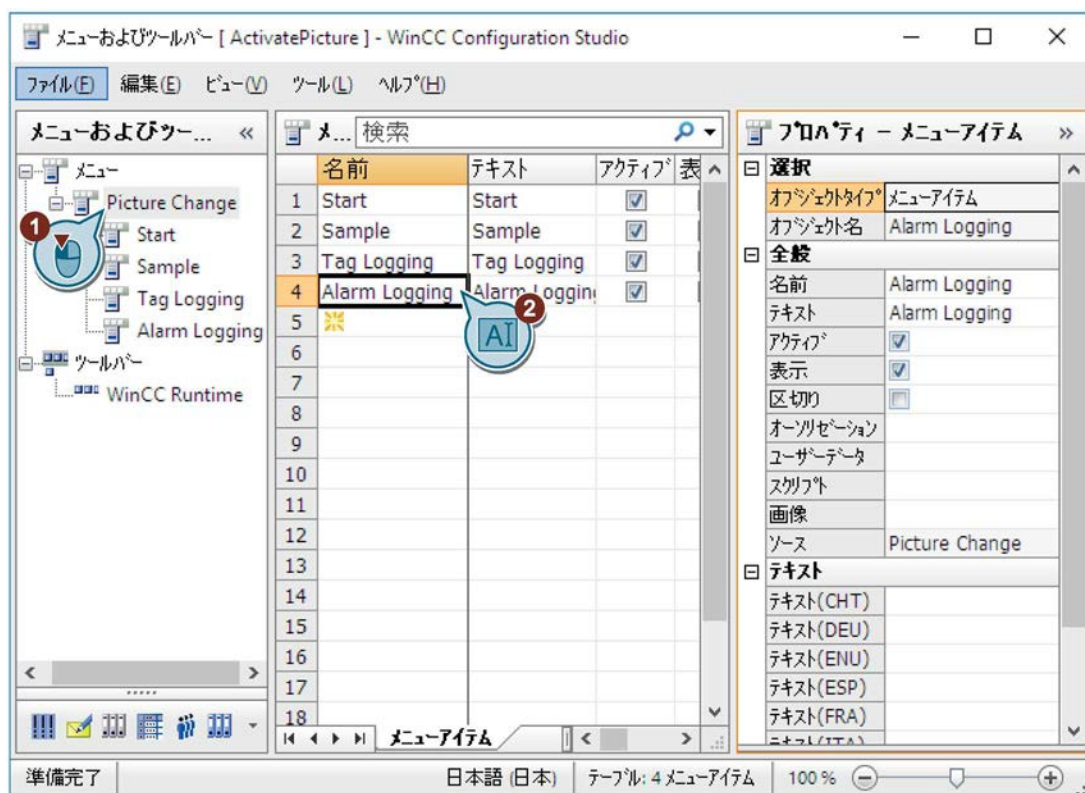
手順

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。



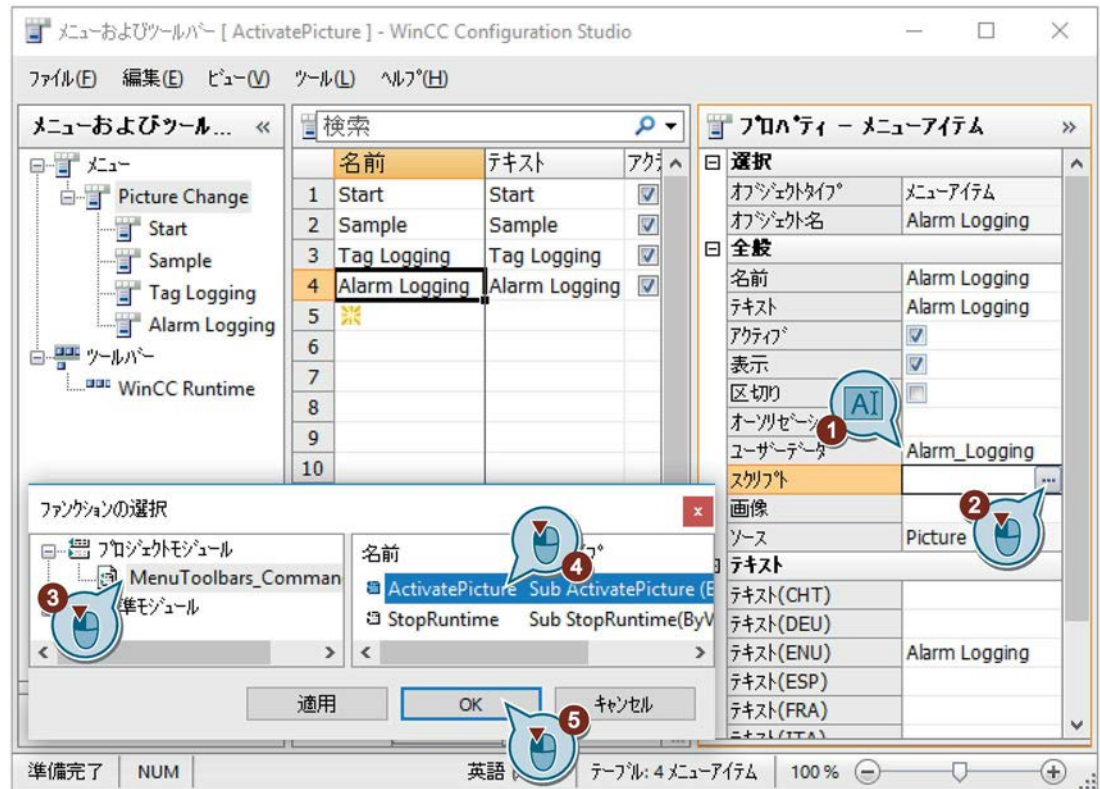
9.9 ユーザー定義メニューのカスタマイズ(アラームロギング)

2. [アラームロギング]メニューコマンドを作成します。



9.9 ユーザー定義メニューのカスタマイズ(アラームロギング)

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、プロセス画像"START.pdl"、
"SAMPLE.pdl"、"Tag_Logging.pdl"および"Alarm_Logging.pdl"に切り替えます。

9.10 ランタイムプロパティの定義(アラームロギング)

はじめに

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。

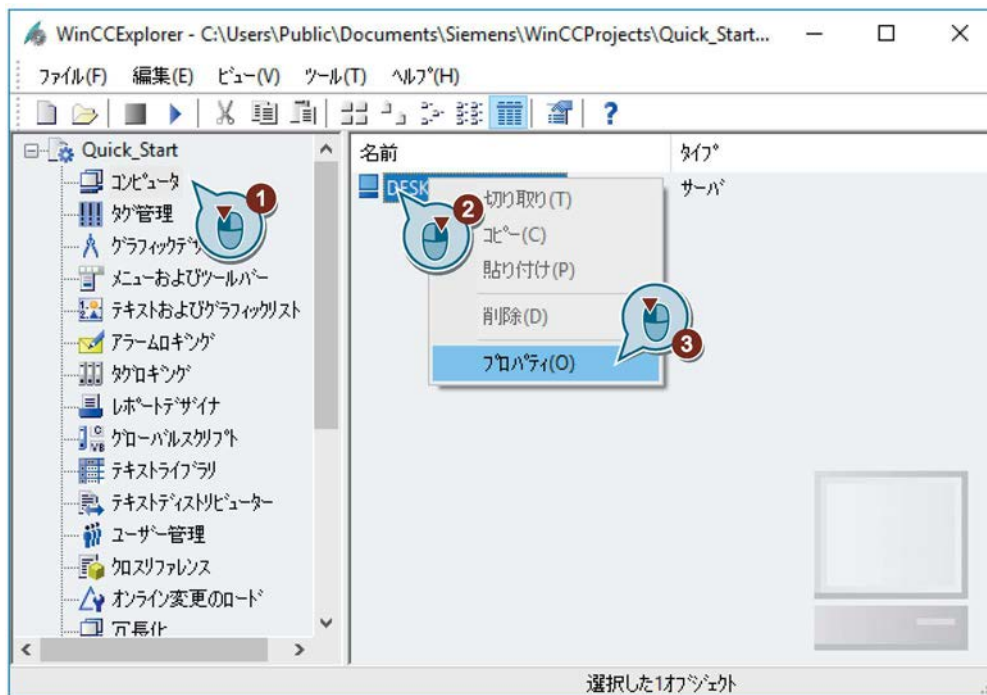
"Alarm_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

必要条件

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

手順

1. [コンピュータのプロパティ]ダイアログを開きます。



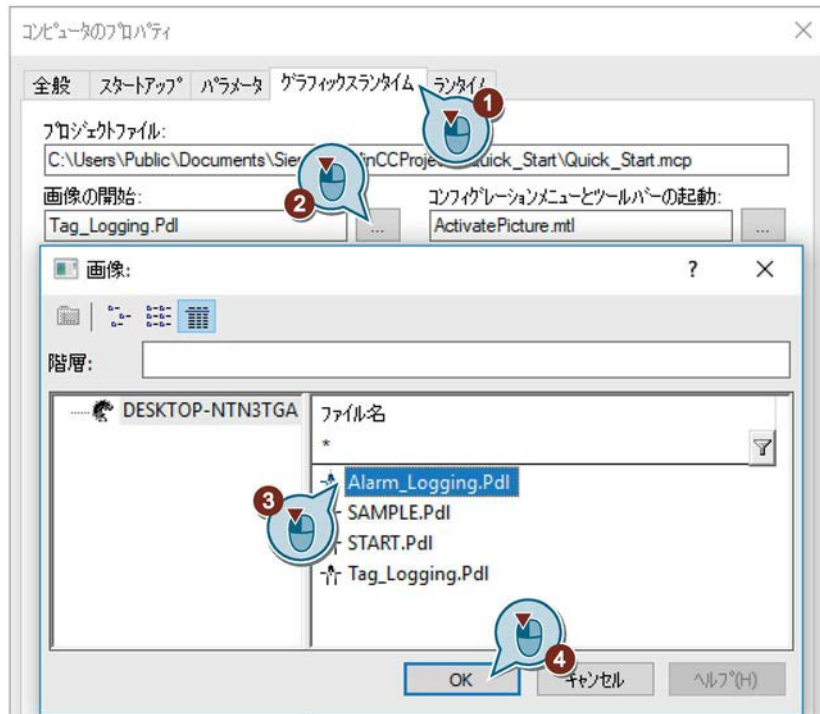
[コンピュータのプロパティ]ダイアログが開きます。

2. [スタートアップ]タブをクリックして、"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。



9.10 ランタイムプロパティの定義(アラームロギング)

3. [グラフィックランタイム]タブで、"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。



4. [OK]をクリックして[コンピュータのプロパティ]ダイアログを終了します。

結果

WinCC Runtimeのプロパティが定義されています。

"Quick_Start"プロジェクトの起動時に、タグロギングランタイムが実行され、プロセス画像"Alarm_Logging.pdl"が表示されます。

次のステップで、"Quick_Start"プロジェクトを有効にして、ランタイムでのメッセージの出力を表示させます。

9.11 プロジェクトの有効化(アラームロギング)

はじめに

次のステップは、"Quick_Start"プロジェクトを有効化する方法や、ランタイムでのプロセス画像"Alarm_Logging.pdl"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。

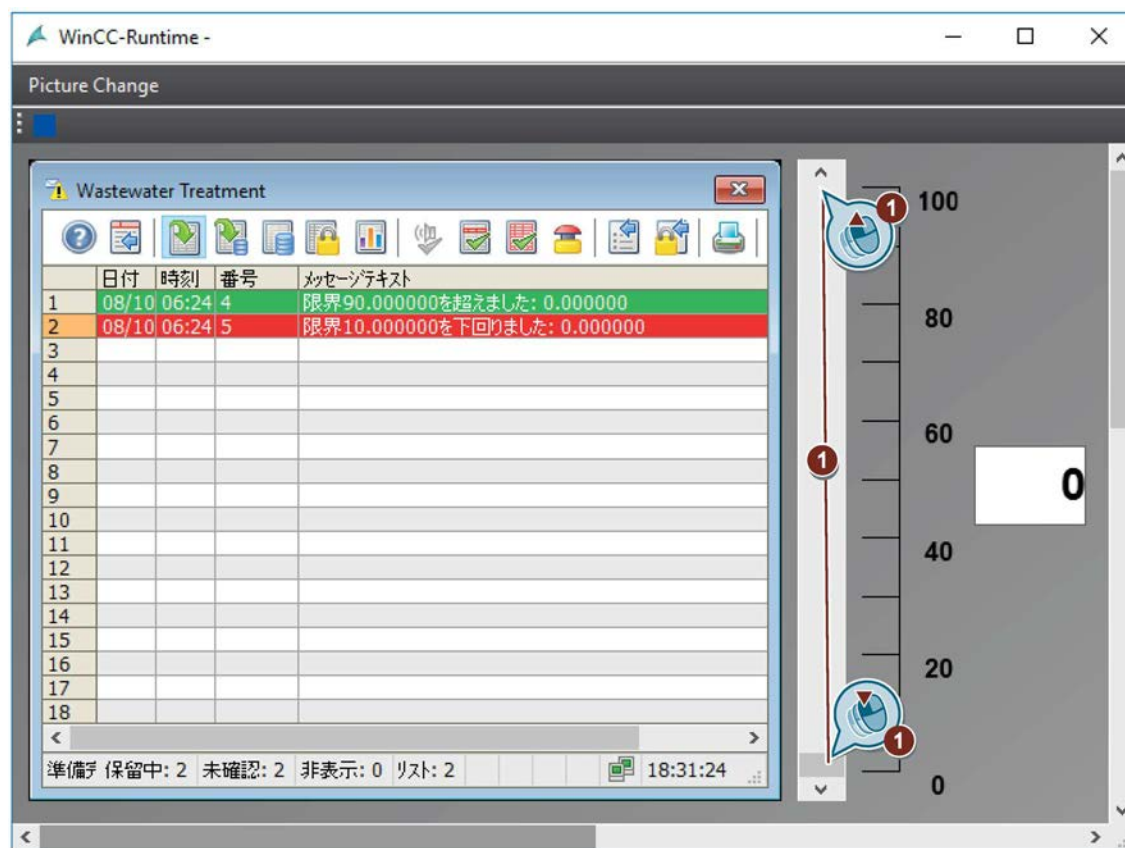
9.11 プロジェクトの有効化(アラームロギング)

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。

プロセス画像"Alarm_Locking.pdl"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。

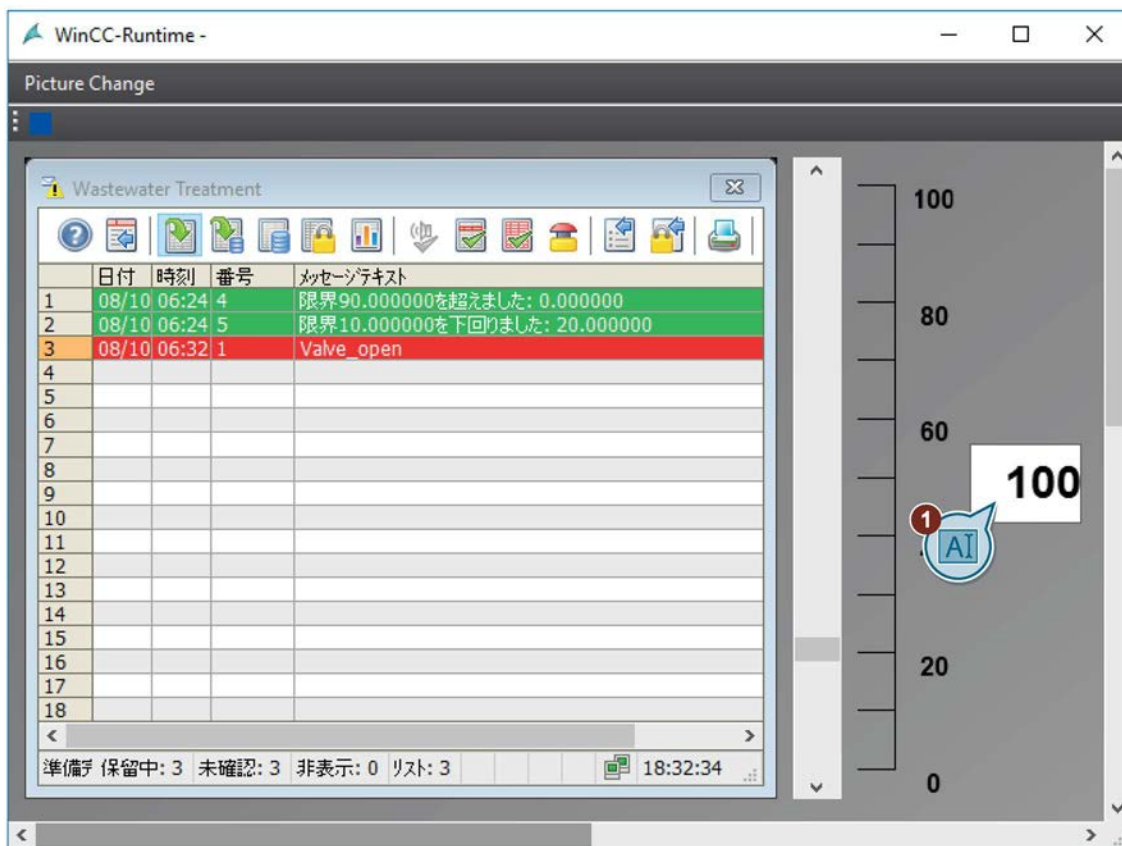


コントローラの位置によって、内部タグ[Tank_Level]に値が割り当てられます。

この値が設定済みの下限値(10)より低下した場合、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"がトリガされました。

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。

ディスクリートアラーム"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。

ディスクリートアラーム"Valve_closed"が表示されます。

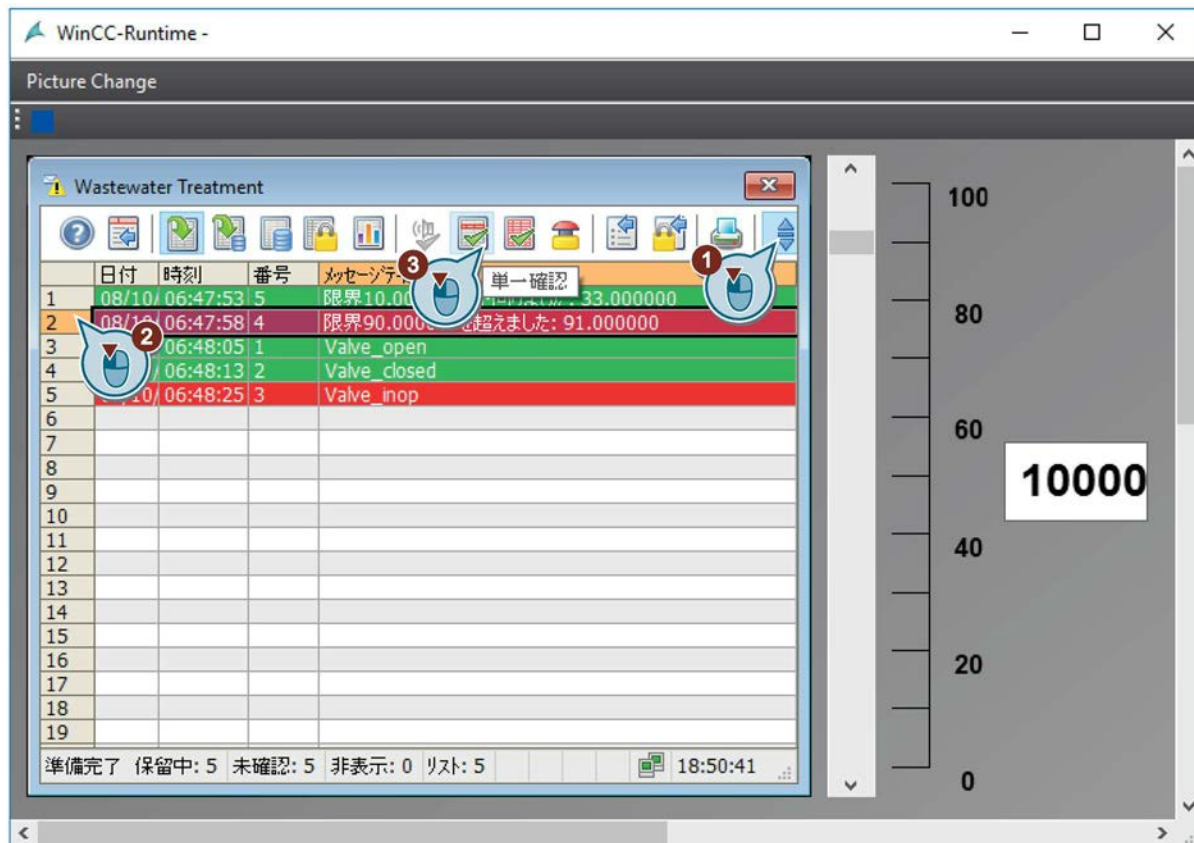
5. I/Oフィールドに値[10000]を入力し、<Return>で入力内容を確定します。

タグ値の4番目のビットが設定されます。

ディスクリートアラーム"Valve_inop"が表示されます。

9.11 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、メッセージウィンドウに表示されます。

内部タグ[Tank_Level]および[Inflow_Valve]を値と共に自動的に提供するには、次のステップで[タグシミュレーション]エディタを使用します。

9.12 プロジェクトのテスト(アラームロギング)

はじめに

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"に値を割り当てます。

これらのタグに対してモニタリングが設定され、タグ値に対応してメッセージがトリガされます。

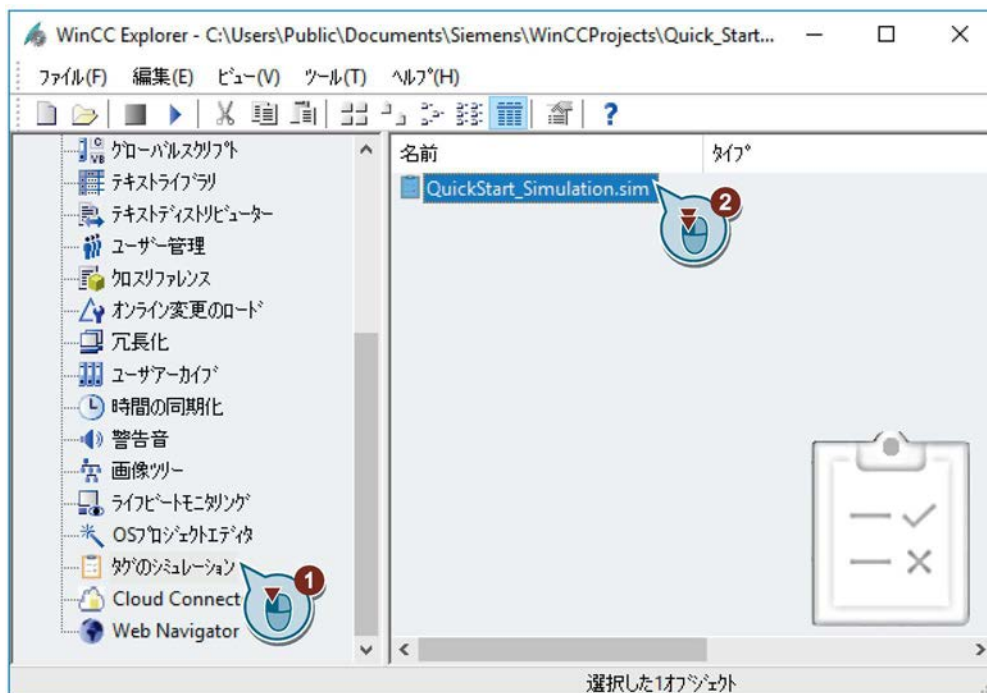
メッセージが、メッセージウィンドウに表示されます。メッセージステータスは、異なる表示色によってマークされます。

必要条件

- ランタイムプロパティが定義されている。

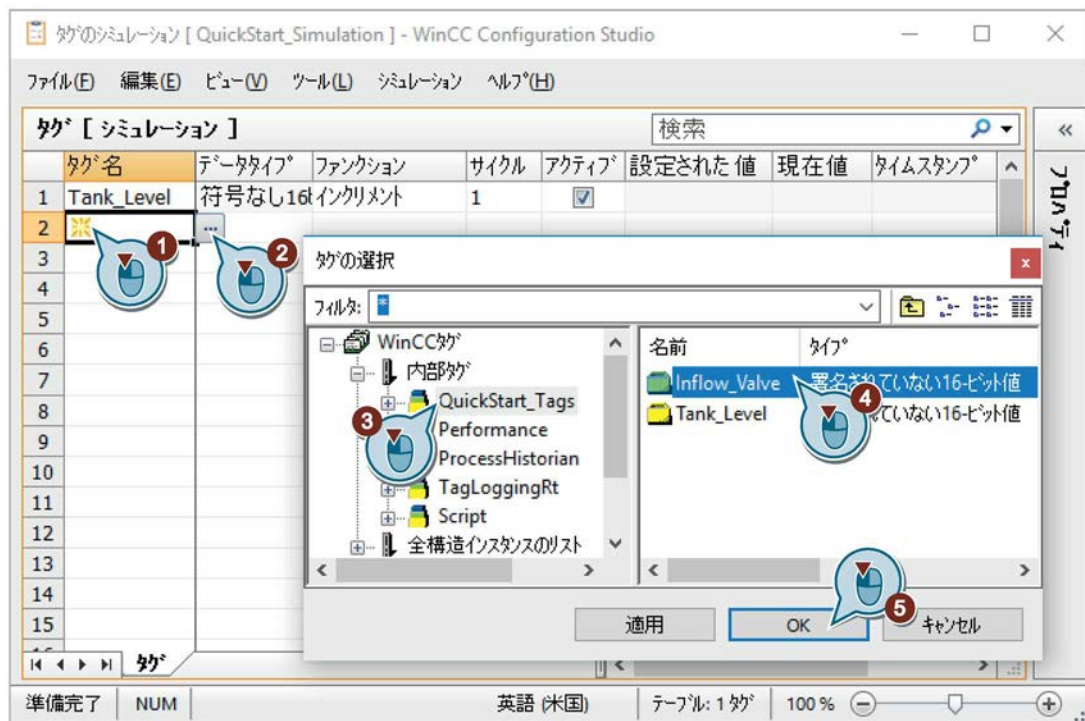
手順

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。



9.12 プロジェクトのテスト(アラームロギング)

2. 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。



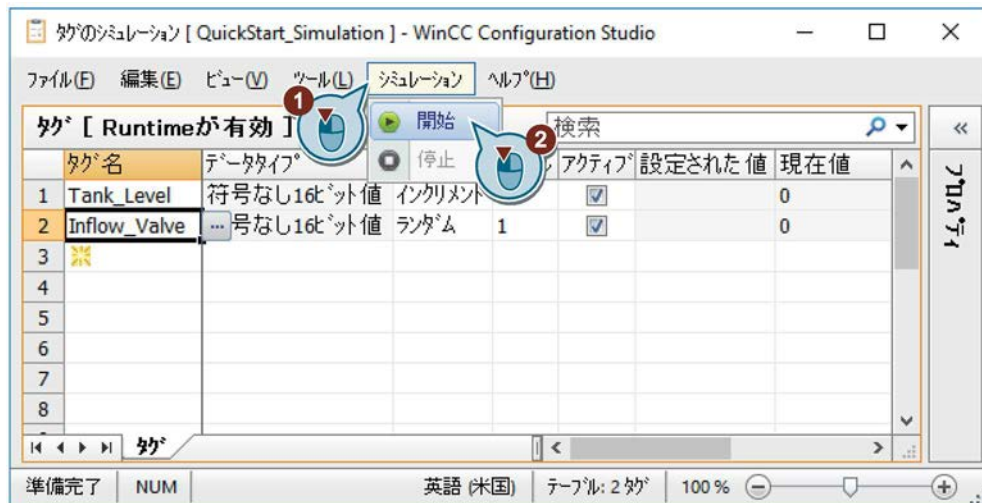
9.12 プロジェクトのテスト(アラームロギング)

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。



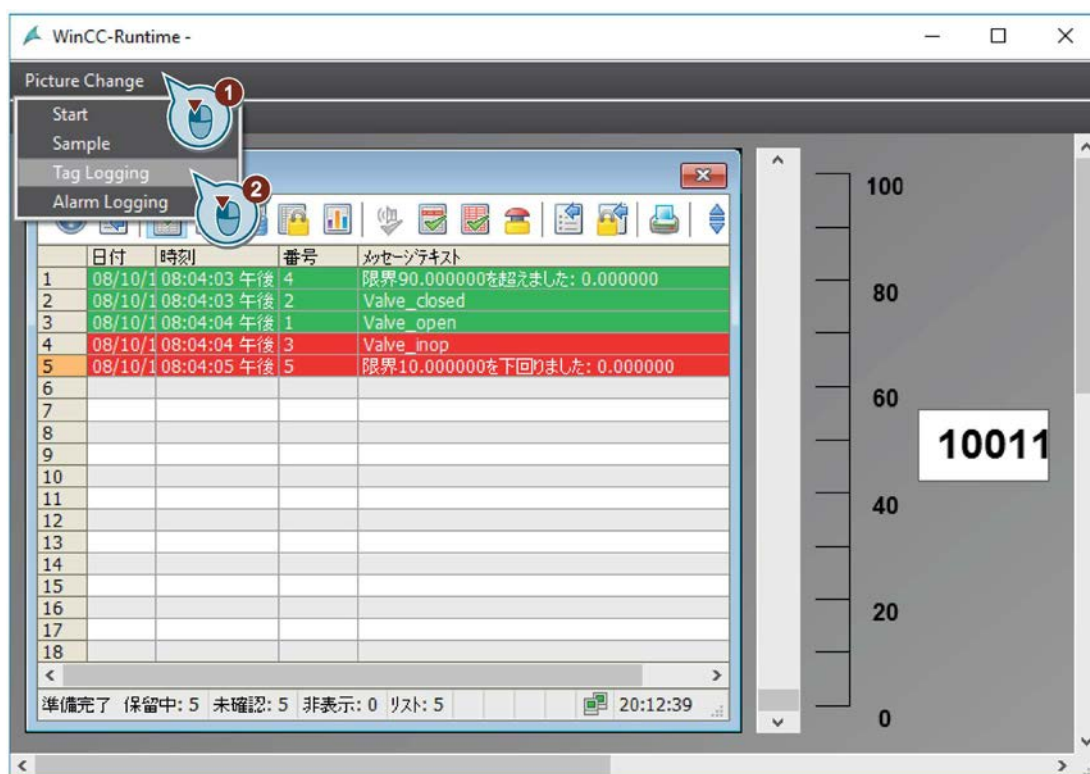
9.12 プロジェクトのテスト(アラームロギング)

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エクスペローラでWinCC Runtimeを有効にします。
6. [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。

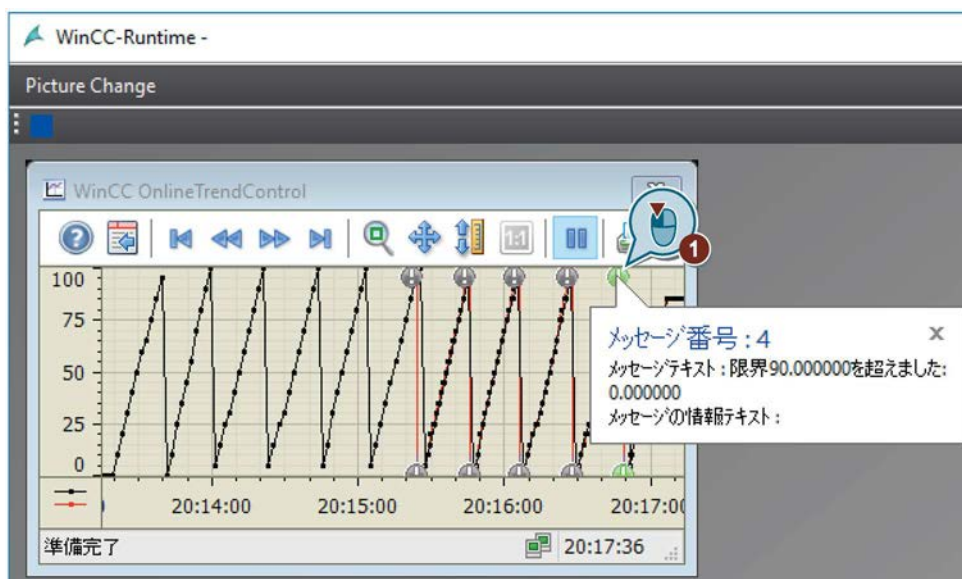


7. "Alarm_Logging.pdl"プロセス画像で、シミュレーション値の出力を確認します。
シミュレーション値の変化が、メッセージウィンドウにどのように影響するかを監視します。

8. "Tag_Logging.pdl"プロセス画像に切り替えます。

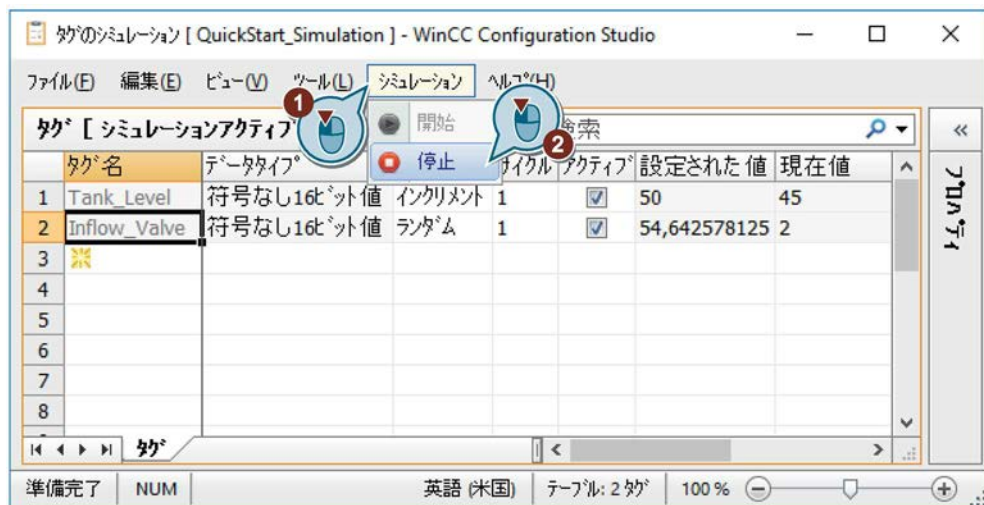


9. WinCC OnlineTrendControlでのトレンドの拡大表示を確認します。



9.12 プロジェクトのテスト(アラームロギング)

10. シミュレーションを終了します。



11. ツールバーの  ボタンを使用して、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、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。

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の設定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

設定ソフトウェアは、WinCCの一部です。

WinCCエクスプローラは設定ソフトウェアの核となっています。

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。

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。

システムオブジェクト

システムオブジェクトは、システム時間、ページ番号、プロジェクト名およびレイアウト名のプレースホルダとして使用されます。

システムオブジェクトを挿入できるのは、スタティックレイアウトのスタティックレベルだけです。

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。

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。

ページレイアウトエディタ

ページレイアウトエディタは、ページレイアウトを作成、編集するために使用します。

ページレイアウトエディタを使用できるのは、現在WinCCエクスプローラで開いているプロジェクトだけです。レイアウトは、特定のオブジェクトに対して保存されます。

ページレイアウトのスタティックレベル

レイアウトのヘッダーとフッターは、スタティックレベルで定義されます。

例えば、スタティックレベルは、会社名、会社のロゴ、時刻やページ番号を出力するために使用します。

ページレイアウトのダイナミックレベル

ページレイアウトのダイナミックレイヤは、設定やラインタイムデータを出力するオブジェクトを含みます。

ページレイアウトのページサイズ

ページサイズは、レイアウトにおける全体エリアを示します。

ページサイズは、レポートの出力フォーマットを決定します。

ページレイアウトの印刷余白

印刷余白は、ページレイアウトで印刷されない余白の領域を定義します。

この領域は、デフォルトではページレイアウトエディタでグレー表示され、編集できません。

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を含むテキスト)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。

リニアスケーリング

リニアスケー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。

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

@

@OnlineTableControl - 表, 178

[

[テーブル]ウィンドウ, 137

WinCC OnlineTableControl, 137
設定, 137

[トレンド]ウィンドウ, 129

WinCC OnlineTrendControl, 129
設定, 129

A

Alarm_Logging.pdl, 218

I

I/Oフィールド, 68, 74, 217

ダイナミック化, 74, 230
フォントプロパティの定義, 230
更新の定義, 74
出力フォーマットの定義, 74, 230
挿入, 74, 230
入力制限, 74

M

MPI, 26, 30

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, 50, 66

作成, 50

編集, 66

START.pdl, 50

作成, 50

編集, 66

SVGオブジェクト

ダイナミック化, 70

SVGライブラリ, 55

T

Tag_Logging.pdl, 129

Tag_Logging.rpl, 155

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, 217, 218

WinCC OnlineTableControl, 128, 137

挿入, 137

WinCC OnlineTrendControl, 128, 129

メッセージの表示, 209

挿入, 129

WinCC Runtime, 13

プロパティの指定, 96, 145, 182, 238

開始画像, 145

WinCCエクスプローラ, 13

WinCCタグシミュレータ, 102, 147, 185, 245

シミュレーションタイプの定義, 245

開始, 102, 147, 245

Windowsオブジェクト, 53

あ

アーカイブ, 115, 122

プロセス値アーカイブ, 115, 122

値, 115

アーカイブサイクル, 120

時間の割り付け, 125

時間の設定, 120

アーカイブシステム, 116

タグロギング, 116

タグロギングランタイム, 116

アーカイブタグ, 122

作成, 122

名前の変更, 125

アイコン, 11

アドレス指定, 34

プロセスタグ, 34

アナログアラーム, 191

ランタイムでの表示, 209

制限値の設定, 205

設定, 204

アラームウィンドウ, 218

WinCC AlarmControl, 218

設定, 218

アラームロギング, 191, 192

開始, 194

アラームロギングランタイム, 192

有効, 238

う

ウィンドウ属性, 96

有効, 96

え

エディタ, 13, 25, 48

アラームロギング, 13, 194

グラフィックデザイナー, 48, 53

タグロギング, 116, 118

タグ管理, 25

ページレイアウトエディタ, 158, 161

レポートデザイナー, 13, 152, 155

お

オブジェクト, 47, 48, 55, 158

I/Oフィールド, 217, 230

WinCC AlarmControl, 217

WinCC OnlineTableControl, 128, 137

WinCC OnlineTrendControl, 128, 129

Windowsオブジェクト, 53

スケール, 226

スタティックオブジェクト, 47, 158

スタティックテキスト, 62

スマートオブジェクト, 53

スライダオブジェクト, 217, 223

ダイナミックオブジェクト, 47

タグテーブル, 165

プロジェクト名, 170

挿入, 53

操作可能なオブジェクト, 47
標準オブジェクト, 53
オブジェクトパレット, 53
Windowsオブジェクト, 53
コントロール, 53
スマートオブジェクト, 53
標準オブジェクト, 53
オブジェクトプロパティ, 70, 74, 223, 226
印刷余白, 161
オブジェクトプロパティダイアログ, 165

か

カバーシート, 158
出力の無効化, 161

く

グラフィックシステム, 48
グラフィックデザイナ, 48
グラフィックランタイム, 48
グラフィックデザイナ, 48, 53
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, 53
グラフィックランタイム, 48
有効, 96

こ

コントロール, 53
WinCC AlarmControl, 217, 218
WinCC OnlineTableControl, 128, 137
WinCC OnlineTrendControl, 128, 129
コンポーネント, 13
WinCC, 13
アーカイブシステム, 116

グラフィックシステム, 48
メッセージシステム, 192
レポートシステム, 152

し

システムオブジェクト, 158
プロジェクト名, 170
ページ番号, 174
システムダイアログ
使用, 108
システムブロック, 196
シミュレーション, 102
WinCCタグシミュレータ, 102, 147, 185, 245
シミュレーションタイプ, 102, 147, 185, 245
開始, 102, 245
終了, 245
シミュレーションタイプ, 245

す

スケール, 40, 226
サイズの定義, 226
フォントサイズの定義, 226
プロセスタグ, 40
挿入, 226
スタティックテキスト, 62
フォントサイズを変更する, 62
挿入, 62
スタティックレベル, 158
開く, 170
ステータス, 200
バルブ, 191
メッセージ, 214
スマートオブジェクト, 53
スライダオブジェクト, 217
サイズの定義, 223

ダイナミック化, 223

色の定義, 223

挿入, 223

名前の定義, 223

た

ダイナミックオブジェクト, 47, 158

ダイナミック化, 68, 70

I/Oフィールド, 74, 230

スライダオブジェクト, 223

タイプ変換, 34

プロセスタグ, 34

タイマ, 120

作成, 120

設定, 120

タグ, 25

アーカイブタグ, 122

スケーリング, 40

プロセスタグ, 25, 34, 40

リンク, 70

作成, 34, 42

内部タグ, 25, 42

タググループ, 25

作成, 34

タグテーブル, 165

挿入, 165

タグロギング, 116

開始, 118

タグロギングランタイム, 116

有効, 145

タグ管理, 25

タググループ, 34

プロセスタグ, 25, 34

内部タグ, 25, 42

タグ接続, 68

直接タグ接続, 68

ち

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, 30

MPI, 26

挿入, 26

て

ディスクリートアラーム, 199

作成, 200

データタイプ, 34, 42

プロセスタグ, 34

符号なし16ビット値, 42

符号付き16ビット値, 34

データ交換, 25

プロセスタグ, 25

テキストフィールド, 62

テスト, 102

プロジェクト, 102, 147, 185, 245

ふ

フォントサイズ, 62

変更, 62, 74, 226

変更, 62, 74, 226

フッター, 158

ページ番号の挿入, 174

編集, 174

プリンタ, 178

印刷ジョブの定義, 178

- プロジェクト, 17, 99, 102
 - WinCCの起動, 18
 - シングルユーザープロジェクト, 18
 - テスト, 102, 147, 185, 245
 - 作成, 18
 - 無効化, 188
 - 有効, 99, 147, 185, 241
- プロジェクト設計ソフトウェア, 13
 - WinCC, 13
- プロジェクト名, 170
 - サイズの変更, 170
 - 再位置決め, 170
 - 挿入, 170
- プロセス, 13
 - モニタリング, 13, 191
 - 視覚化, 13, 47, 48
 - 操作, 13, 21, 23, 25
- プロセスタグ, 25
 - アドレス指定, 34
 - スケーリング, 40
 - タイプ変換, 34
 - データタイプ, 34
 - 作成, 34
- プロセス画像, 47
 - Alarm_Logging.pdl, 218
 - SAMPLE.pdl, 50
 - START.pdl, 50
 - Tag_Logging.pdl, 129
 - ダイナミック化, 68
 - 作成, 50
 - 設定, 47, 128, 217
 - 編集, 53, 66
 - 保存, 62, 66
 - 名前の変更, 50
- プロセス値, 21, 25, 115
 - アーカイブ, 115, 120
 - ウィンドウ, 115
 - シミュレーション, 102, 147, 185, 245
 - 入力, 120
- プロセス値アーカイブ, 115
 - サイズの定義, 125
 - メモリの場所の定義, 125
 - 作成, 122
 - 値の出力, 151
 - 編集, 125
 - 保存, 125
- プロセス値ブロック, 196
- へ
- ページレイアウト, 155, 158, 161
 - Tag_Logging.rpl, 155
 - カバーシート, 158
 - スタティックレベル, 158
 - ダイナミックレベル, 158, 165
 - フッター, 158
 - プロパティの指定, 161
 - ページサイズ, 161
 - ヘッダー, 158
 - レポート内容, 158, 165
 - 印刷余白, 161
 - 開く, 161
 - 最終ページ, 158
 - 作成, 155
 - 編集, 158
 - 保存, 174
 - 名前の変更, 155
- ページレイアウトエディタ, 152, 158
 - 開始, 161
- ページ番号, 174, 174
 - サイズの変更, 174

再位置決め, 174
挿入, 174
ページ番号, 174, 174
ヘッダー, 158, 170
 プロジェクト名の挿入, 170
 編集, 170
ヘッダー, 158, 170

ま

マニュアル類, 115
 値, 151

め

メッセージ, 191
 アナログアラーム, 191
 ディスクリートアラーム, 191
 メッセージステータス, 214
 メッセージステータスの色の定義, 214
 確認, 241
 制限値モニタリング, 204
 設定, 191
 保存, 205

メッセージクラス, 196
 定義, 196

メッセージシステム, 192
 アラームロギング, 192
 アラームロギングランタイム, 192

メッセージステータス, 200, 214

メッセージブロック, 196
 システムブロック, 196
 プロセス値ブロック, 196
 ユーザーテキストブロック, 196
 定義, 196
 表示の有効化, 218

メッセージリスト, 241

ゆ

ユーザーテキストブロック, 196

ら

ライブラリ, 53, 55

 グローバルライブラリ, 55

 プロジェクトライブラリ, 55

 開く, 55

ランタイム, 13

ランタイムウィンドウ, 96

 ウィンドウ属性, 96

 開始画像, 96, 145, 182, 238

ランタイムコンポーネント, 13

 アーカイブシステム, 116

 グラフィックシステム, 48

 メッセージシステム, 192

 レポートシステム, 152

ランタイムシステムダイアログ, 108

ランタイムソフトウェア, 13

 WinCC, 13

ランタイムデータ, 151, 152

り

リニアスケーリング, 40

 例, 40

れ

レイアウト, 152

レイヤ, 158

 スタティックレベル, 158, 174

 ダイナミックレベル, 158

レポートシステム, 152
 レポートデザイナー, 152
 レポートランタイム, 152
 レポートデザイナー, 152
 ページレイアウトエディタ, 152
 レイアウト, 152
 印刷ジョブ, 152
 行レイアウトエディタ, 152
 レポートランタイム, 152
 有効, 182
 レポート内容, 158, 165

ろ

ログ, 151, 152
 カバーシート, 161
 プロセスアーカイブからの値のロギング, 151
 ログの内容の決定, 165
 印刷, 188
 最終ページ, 158

漢字

印刷, 188
 プロセス値のログ, 188
 印刷ジョブ, 151, 152, 177
 @OnlineTableControl - 表, 178
 ページレイアウトのリンク, 178
 開く, 178
 出力媒体の定義, 178
 定義, 178
 開く, 18, 50
 [ページサイズ]ダイアログ, 161
 オブジェクトプロパティダイアログ, 165, 174
 スタティックレベル, 170
 ダイナミックレベル, 165

プロジェクト, 18
 プロセス画像, 50, 66
 ライブラリ, 55
 レポート内容ページ, 165
 印刷ジョブ, 178
 開始, 18
 WinCC, 18
 WinCCタグシミュレータ, 147
 アラームロギング, 194
 タグロギング, 118
 ページレイアウトエディタ, 161
 開始画像, 96, 145, 182
 定義, 96, 238
 確認, 241
 メッセージ, 241
 更新サイクル
 I/Oフィールド, 74
 行レイアウトエディタ, 152
 最終ページ, 158
 作成, 30, 34, 42, 129
 [テーブル]ウィンドウ, 137
 [トレンド]ウィンドウ, 129
 アナログアラーム, 205
 アラームウィンドウ, 218
 タイマ, 120
 タググループ, 34
 ディスクリートアラーム, 200
 フッター, 174
 プロジェクト, 17, 18
 プロセスタグ, 34
 プロセス画像, 50
 プロセス値アーカイブ, 122
 ページレイアウト, 155
 ヘッダー, 170
 接続, 30
 内部タグ, 42

- 実例, 55
 - スケール, 226
- 取得サイクル, 120
 - 時間の割り付け, 125
 - 時間の設定, 120
- 出力, 151
 - プロセス値のログ, 151
- 色, 214
 - メッセージステータスの色の定義, 214
- 制限値, 205
 - 下限値, 205
 - 上限値, 205
 - 定義, 205
- 制限値モニタリング, 204
 - ランタイムでのメッセージの表示, 209
 - 制限値の設定, 205
- 接続, 23, 30
 - 作成, 30
- 設定, 21, 53, 128
 - [テーブル]ウィンドウ, 137
 - [トレンド]ウィンドウ, 129
 - アラームウィンドウ, 218
 - プロセス画像, 53, 128, 217
 - プロセス値アーカイブ, 122, 125
 - メッセージ, 191, 199, 204
 - 取得サイクルおよびアーカイブサイクルの時間, 12
- 0
 - 通信, 21
- 設定コンポーネント, 48, 116, 152, 192
 - アーカイブシステム, 116
 - グラフィックシステム, 48
 - メッセージシステム, 192
 - レポートシステム, 152
- 設定データ, 151, 152
- 値, 21, 151
 - アーカイブ, 115
 - シミュレーション, 102
 - プロセス値, 21, 25, 115
 - 出力, 151
- 値の範囲, 40
 - プロセスタグのスケーリング, 40
- 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, 25, 34
 - 接続, 23, 30
 - 設定, 21
- 通信ドライバ, 23
 - SIMATIC S7-1200、S7-1500チャンネル, 26
- 貼り付け, 26
 - I/Oフィールド, 74, 230
 - SIMATIC S7-1200、S7-1500チャンネル, 26
 - WinCC AlarmControl, 218
 - WinCC OnlineTableControl, 137
 - WinCC OnlineTrendControl, 129
 - スケール, 226
 - スタティックテキスト, 62
 - スライダオブジェクト, 223
 - タグテーブル, 165
 - チャンネル, 26
 - チャンネルユニット, 26
 - プロジェクト名, 170
 - ページ番号, 174
 - ライブラリからのオブジェクト, 55
- 内部タグ, 25
 - データタイプ, 42
 - 作成, 42
- 標準オブジェクト, 53
 - スタティックテキスト, 62
- 表示色, 214
 - メッセージステータスの色の定義, 214

表示要素, 74, 115, 223

値, 74, 115, 223

符号なし16ビット値, 42

符号付き16ビット値, 34

編集, 53, 177

スタティックテキスト, 62

フッター, 174

プロセス画像, 53, 66

プロセス値アーカイブ, 125

ページレイアウト, 158

印刷ジョブ, 177, 178

保持, 147, 188

データの更新, 188

保存, 66, 174

プロセス画像, 62, 66

プロセス値アーカイブ, 125

ページレイアウト, 174

メッセージ, 205

無効化

カバーシートの出力, 161

プロジェクト, 188

名前の変更, 50, 125

アーカイブタグ, 125

プロセス画像, 50

有効, 99

アラームロギングランタイム, 238

ウィンドウ属性, 96

グラフィックランタイム, 96

タグロギングランタイム, 145

プロジェクト, 99, 147, 185, 241

リニアスケーリング, 40

レポートランタイム, 182

