SIEMENS

SIMATIC HMI

WinCC V7.5 SP2
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

11/2020
A5E50503070-AA

法律上の注意

警告事項

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

⚠ 危険
回避しなければ、直接的な死または重傷に至る危険状態を示します。

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

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

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

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

有資格者

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

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

⚠ 警告
シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限ります。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て更新いたします。

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
ドイツ

A5E50503070-AA
Ⓟ 10/2020 変更する権利を留保

Copyright © Siemens AG 2020.
無断複写・転載を禁じます

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 23

5.1 通信の設定 23

5.2 WinCC チャンネルとの接続確認 25

5.3 WinCC のタグ 26

5.4 チャンネルの追加 27

5.5 接続の作成 32

5.6 タググループのプロセスタグの作成 37

5.7 WinCC でのプロセスタグのスケーリング 43

5.8 内部タグの作成 45

6 プロセス画面の設定 49

6.1 プロセス画面の設定 49

6.2 グラフィックシステム 50

6.3 プロセス画面の作成 52

6.4 プロセス画面の編集 55

6.4.1 プロセス画面の編集 55

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 56

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 61

6.4.4 プロセス画面"SAMPLE.pdl"の編集 64

6.5 画面ダイナミックスの処理 66

6.5.1 画面ダイナミックスの処理 66

6.5.2 フィルレベルインジケータのダイナミック化 67

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) 71

6.6 ユーザー定義メニューおよびツールバーの作成 77

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 77

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 77

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 82
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 87
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) 92
6.8 プロジェクトの有効化(グラフィックシステム) 96
6.9 プロジェクトのテスト(グラフィックシステム) 99
6.10 ランタイムシステムダイアログの使用 104
7 値のアーカイブと表示 111
7.1 値のアーカイブと表示 111
7.2 アーカイブシステム 112
7.3 タグロギングの開始 114
7.4 タイマの設定 116
7.5 プロセス値アーカイブの作成 118
7.6 プロセス値アーカイブの編集 121
7.7 プロセス画面の設定 124
7.7.1 プロセス画像の設定(アーカイブシステム) 124
7.7.2 [トレンド]ウィンドウの設定 124
7.7.3 [テーブル]ウィンドウの設定 131
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) 136
7.9 ランタイムプロパティの定義(アーカイブシステム) 139
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) 142
8 プロセスアーカイブからの値の出力 145
8.1 プロセスアーカイブからの値の出力 145
8.2 レポートシステム 146
8.3 ページレイアウトの作成 149
8.4 ページレイアウトの編集 152
8.4.1 ページレイアウトの編集 152
8.4.2 ページレイアウトのプロパティの確立 154
8.4.3 ログの内容の決定 157
8.4.4 ヘッダーの編集 161
8.4.5 フッターの編集 164
8.5 印刷ジョブの編集 167
8.5.1 印刷ジョブの編集 167
8.5.2 印刷ジョブの定義と編集 167
8.6 ランタイムプロパティの定義(レポートシステム) 171
8.7 プロジェクトの有効化とテスト(レポートシステム) 174

8.8 ログの印刷 177

9 メッセージの設定 181

9.1 メッセージの設定 181

9.2 メッセージシステム 182

9.3 アラームロギングの開始 184

9.4 メッセージブロックの設定 186

9.5 ビットメッセージの設定 189
9.5.1 ビットメッセージの設定 189
9.5.2 ビットメッセージの作成 190

9.6 アナログメッセージの設定 194
9.6.1 アナログメッセージの設定 194
9.6.2 限界値の設定 194
9.6.3 トレンドウィンドウでのメッセージの表示 197

9.7 メッセージステータスの色の定義 203

9.8 プロセス画面の設定 205
9.8.1 プロセス画像の設定(アラームロギング) 205
9.8.2 アラームメッセージウィンドウの設定 205
9.8.3 スライダオブジェクトの挿入とダイナミック化 210
9.8.4 スケールの挿入 213
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) 216

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) 222

9.10 ランタイムプロパティの定義(アラームロギング) 225

9.11 プロジェクトの有効化(アラームロギング) 228

9.12 プロジェクトのテスト(アラームロギング) 232

用語集 237

索引 247

ようこそ 1

WinCC「スタートガイド」へようこそ

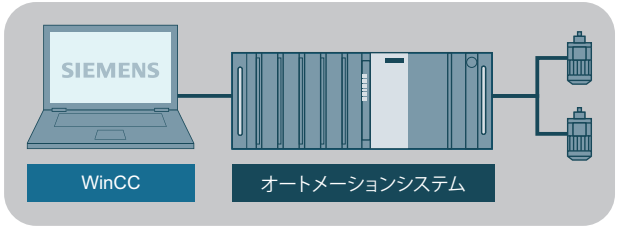
「スタートガイド」は、WinCC バージョン 7 に関する概要ドキュメントです。

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 181)

プロセスアーカイブからの値の出力 (ページ 145)

値のアーカイブと表示 (ページ 111)

プロセス画面の設定 (ページ 49)

通信の設定 (ページ 23)

プロジェクトの作成 (ページ 17)

アイコン

2

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続けてドラッグします(ドラッグアンドドロップ)
	ドラッグした後、マウスの左ボタンを放します(ドラッグアンドドロップ)
	個々の操作ステップのナンバリング

WinCC の操作

3

はじめに

WinCC はモジュラシステムです:

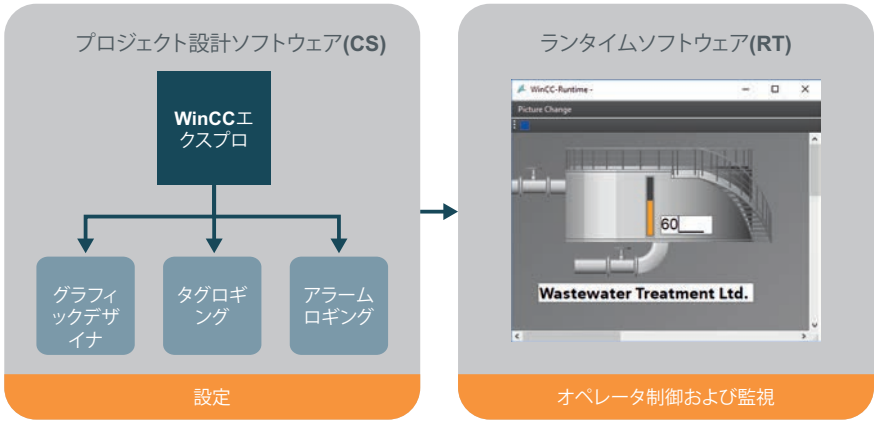
- WinCC を使用して、プロセスを視覚化し、グラフィカルユーザーインターフェースを設定します。
- プロセスのオペレータ制御および監視用ユーザーインターフェースを使用します。

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	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	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

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。

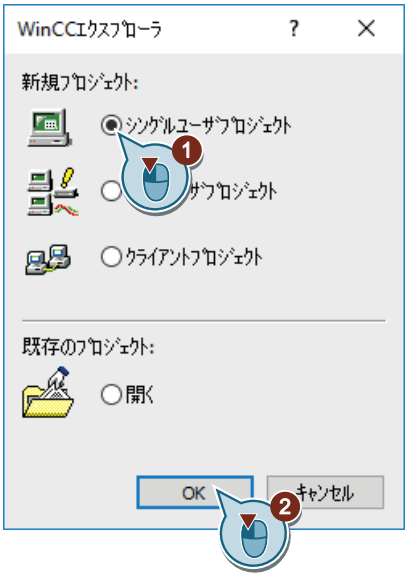
必要条件

- WinCC V7 が最低限、「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

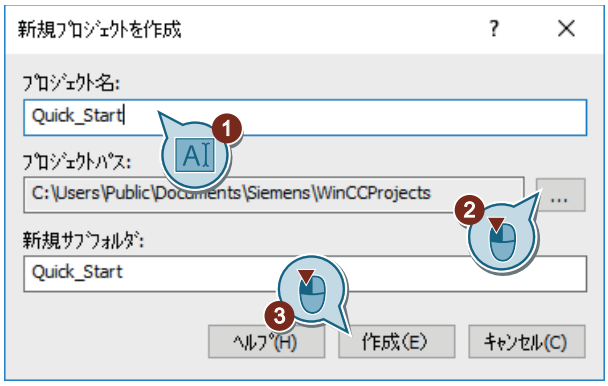
1. WinCC を起動します。
   必要な場合、<Windows+R>を使用して入力画面を開きます。

2. [WinCC エクスプローラ]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、WinCC エクスプローラのメニューバーで[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。

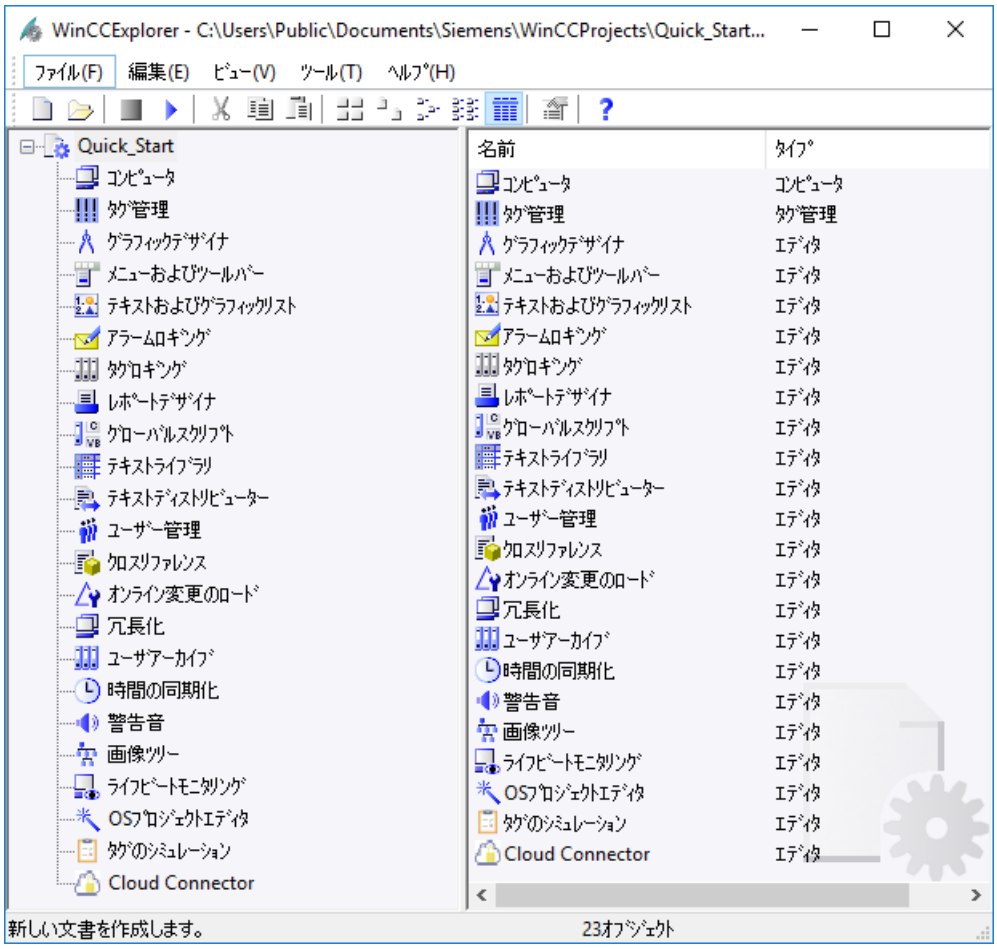
結果

"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。


WinCCExplorer - C:\Users\Public\Documents\Siemens\WinCCProjects\Quick_Start...
ﾌｧｲﾙ(F) 編集(E) ﾋﾞｭｰ(V) ﾂｰﾙ(T) ﾍﾙﾌﾟ(H)
Quick_Start
ｺﾝﾋﾟｭｰﾀ
ﾀｸﾞ管理
ｸﾞﾗﾌｨｯｸﾃﾞｻﾞｲﾅ
ﾒﾆｭｰおよびﾂｰﾙﾊﾞｰ
ﾃｷｽﾄおよびｸﾞﾗﾌｨｯｸﾘｽﾄ
ｱﾗｰﾑﾛｷﾞﾝｸﾞ
ﾀｸﾞﾛｷﾞﾝｸﾞ
ﾚﾎﾟｰﾄﾃﾞｻﾞｲﾅ
ｸﾞﾛｰﾊﾞﾙｽｸﾘﾌﾟﾄ
ﾃｷｽﾄﾗｲﾌﾞﾗﾘ
ﾃｷｽﾄﾃﾞｨｽﾄﾘﾋﾞｭｰﾀｰ
ﾕｰｻﾞｰ管理
ｸﾛｽﾘﾌｧﾚﾝｽ
ｵﾝﾗｲﾝ変更のﾛｰﾄﾞ
冗長化
ﾕｰｻﾞｱｰｶｲﾌﾞ
時間の同期化
警告音
画像ﾂﾘｰ
ﾗｲﾌﾋﾞｰﾄﾓﾆﾀﾘﾝｸﾞ
OSﾌﾟﾛｼﾞｪｸﾄｴﾃﾞｨﾀ
ﾀｸﾞのｼﾐｭﾚｰｼｮﾝ
Cloud Connector
名前
ﾀｲﾌﾟ
ｺﾝﾋﾟｭｰﾀ ｺﾝﾋﾟｭｰﾀ
ﾀｸﾞ管理 ﾀｸﾞ管理
ｸﾞﾗﾌｨｯｸﾃﾞｻﾞｲﾅ ｴﾃﾞｨﾀ
ﾒﾆｭｰおよびﾂｰﾙﾊﾞｰ ｴﾃﾞｨﾀ
ﾃｷｽﾄおよびｸﾞﾗﾌｨｯｸﾘｽﾄ ｴﾃﾞｨﾀ
ｱﾗｰﾑﾛｷﾞﾝｸﾞ ｴﾃﾞｨﾀ
ﾀｸﾞﾛｷﾞﾝｸﾞ ｴﾃﾞｨﾀ
ﾚﾎﾟｰﾄﾃﾞｻﾞｲﾅ ｴﾃﾞｨﾀ
ｸﾞﾛｰﾊﾞﾙｽｸﾘﾌﾟﾄ ｴﾃﾞｨﾀ
ﾃｷｽﾄﾗｲﾌﾞﾗﾘ ｴﾃﾞｨﾀ
ﾃｷｽﾄﾃﾞｨｽﾄﾘﾋﾞｭｰﾀｰ ｴﾃﾞｨﾀ
ﾕｰｻﾞｰ管理 ｴﾃﾞｨﾀ
ｸﾛｽﾘﾌｧﾚﾝｽ ｴﾃﾞｨﾀ
ｵﾝﾗｲﾝ変更のﾛｰﾄﾞ ｴﾃﾞｨﾀ
冗長化 ｴﾃﾞｨﾀ
ﾕｰｻﾞｱｰｶｲﾌﾞ ｴﾃﾞｨﾀ
時間の同期化 ｴﾃﾞｨﾀ
警告音 ｴﾃﾞｨﾀ
画像ﾂﾘｰ ｴﾃﾞｨﾀ
ﾗｲﾌﾋﾞｰﾄﾓﾆﾀﾘﾝｸﾞ ｴﾃﾞｨﾀ
OSﾌﾟﾛｼﾞｪｸﾄｴﾃﾞｨﾀ ｴﾃﾞｨﾀ
ﾀｸﾞのｼﾐｭﾚｰｼｮﾝ ｴﾃﾞｨﾀ
Cloud Connector ｴﾃﾞｨﾀ
新しい文書を作成します。
23ｵﾌﾞｼﾞｪｸﾄ

通信の設定 5

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

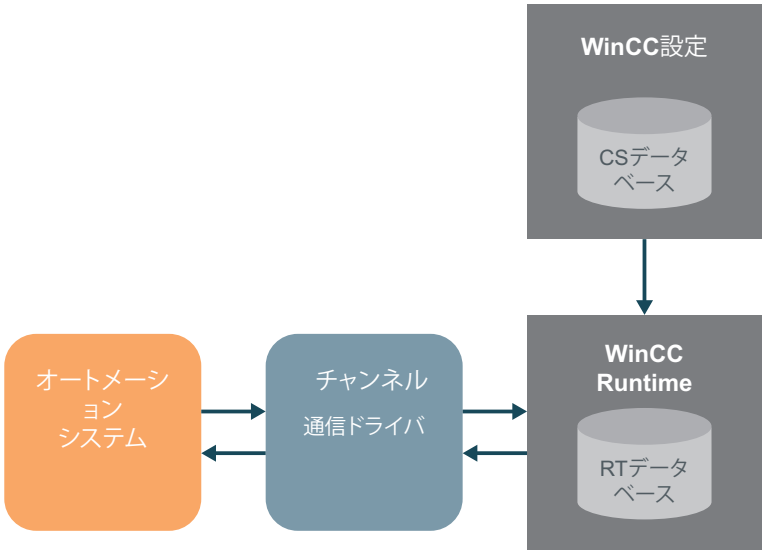
[WinCC Configuration Studio]エディタを使用して、通信を設定します。

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。

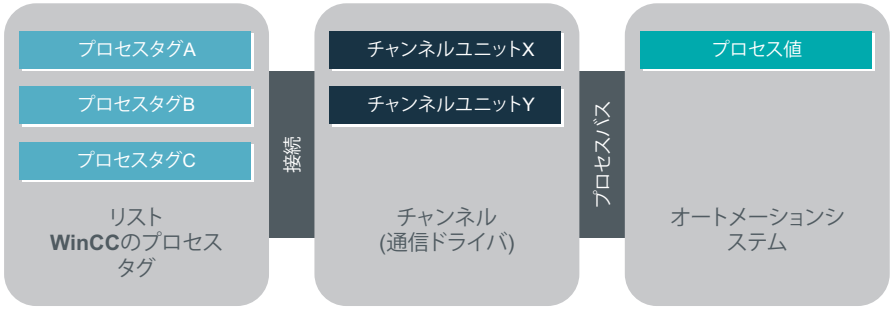
さらに、このチャンネルユニットは、特定のタイプの AS にアクセスするために使用されます。

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。

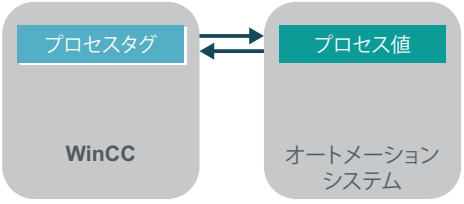
プロセスタグ

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。

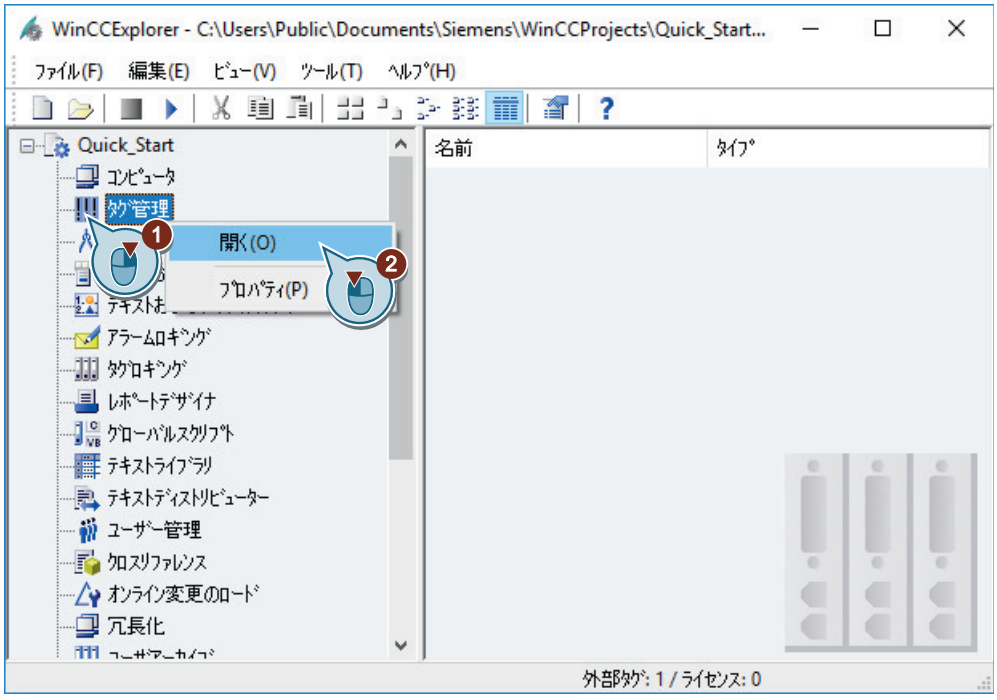
WinCC は、このチャンネルによってオートメーションシステムと通信します。これは、プロセス値を WinCC のプロセスタグにも提供します。

必要条件

- "Quick_Start"プロジェクトが開いている。

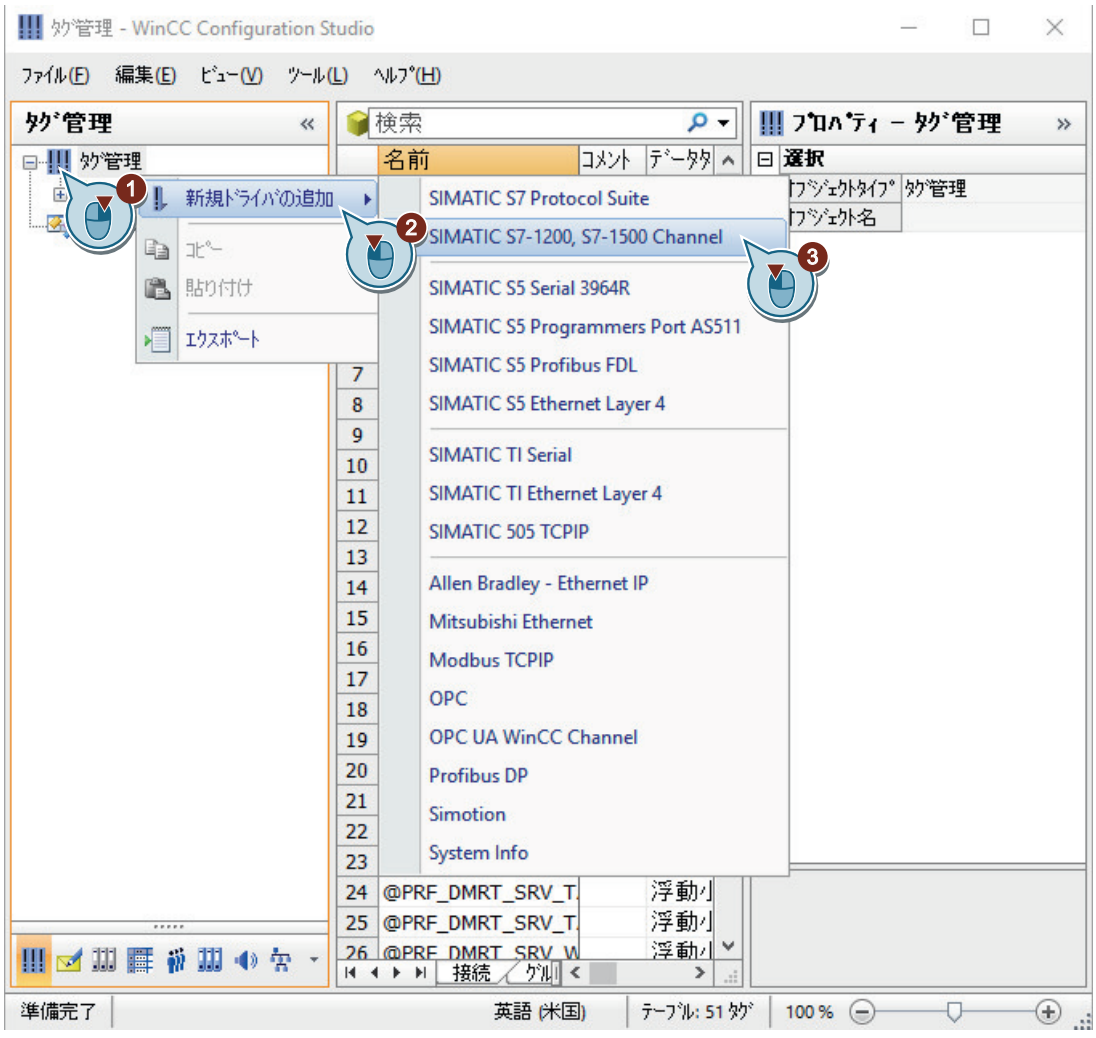
手順

1. [タグ管理]エディタを開きます。

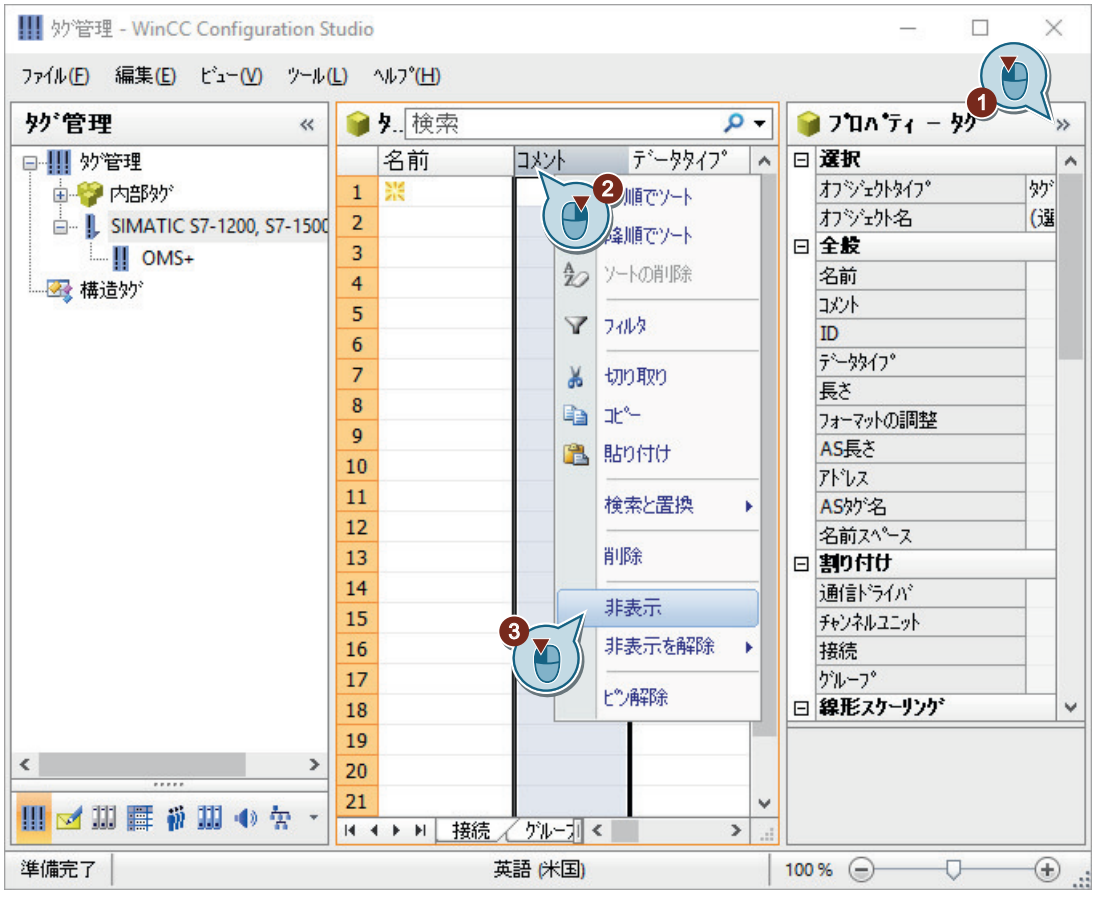


[タグ管理]エディタは、WinCC Configuration Studio で開きます。

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


ﾀｸﾞ管理 - WinCC Configuration Studio
ﾌｧｲﾙ(F) 編集(E) ﾋﾞｭｰ(V) ﾂｰﾙ(L) ﾍﾙﾌﾟ(H)
ﾀｸﾞ管理
ﾀｸﾞ管理
新規ﾄﾞﾗｲﾊﾞの追加
ｺﾋﾟｰ
貼り付け
ｴｸｽﾎﾟｰﾄ
検索
名前
ｺﾒﾝﾄ
ﾃﾞｰﾀﾀ
SIMATIC S7 Protocol Suite
SIMATIC S7-1200, S7-1500 Channel
SIMATIC S5 Serial 3964R
SIMATIC S5 Programmers Port AS511
SIMATIC S5 Profibus FDL
SIMATIC S5 Ethernet Layer 4
SIMATIC TI Serial
SIMATIC TI Ethernet Layer 4
SIMATIC 505 TCPIP
Allen Bradley - Ethernet IP
Mitsubishi Ethernet
Modbus TCPIP
OPC
OPC UA WinCC Channel
Profibus DP
Simotion
System Info
ﾌﾟﾛﾊﾟﾃｨ - ﾀｸﾞ管理
選択
ｵﾌﾞｼﾞｪｸﾄﾀｲﾌﾟ
ﾀｸﾞ管理
ｵﾌﾞｼﾞｪｸﾄ名
@PRF_DMRT_SRV_T
浮動小
接続
準備完了
英語 (米国)
ﾃｰﾌﾞﾙ: 51 ﾀｸﾞ
100 %

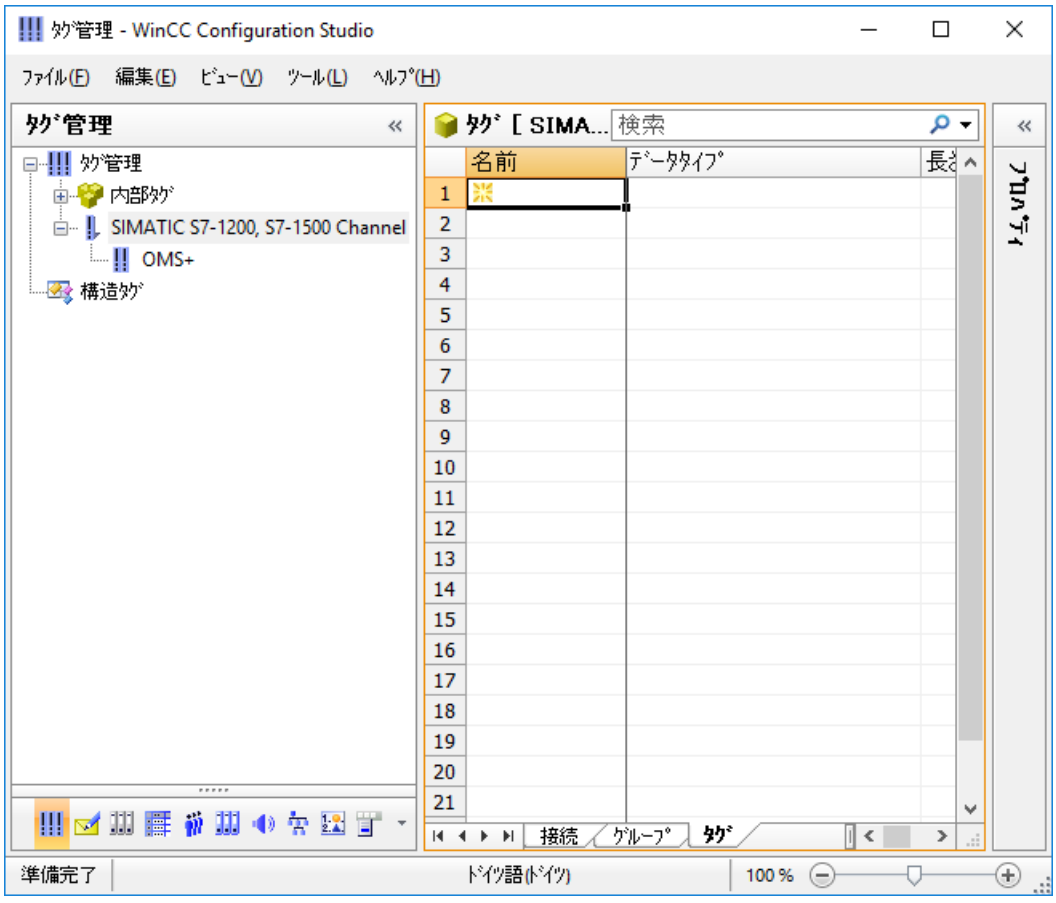
3. データ領域を拡大するには、»ボタンを使用して、[プロパティ]ウィンドウのサイズを減らします。
   [プロパティ]ウィンドウを再度表示するには、«ボタンをクリックします。
4. WinCC Configuration Studio のエディタで、データ領域にある列を非表示および表示することができます。



結果

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

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。


ﾀｸﾞ管理 - WinCC Configuration Studio
ﾌｧｲﾙ(F)
編集(E)
ﾋﾞｭｰ(V)
ﾂｰﾙ(L)
ﾍﾙﾌﾟ(H)
ﾀｸﾞ管理
ﾀｸﾞ管理
内部ﾀｸﾞ
SIMATIC S7-1200, S7-1500 Channel
OMS+
構造ﾀｸﾞ
ﾀｸﾞ [SIMA...
検索
名前
ﾃﾞｰﾀﾀｲﾌﾟ
接続
ｸﾞﾙｰﾌﾟ
ﾀｸﾞ
ﾌﾟﾛﾊﾟﾃｨ
準備完了
ﾄﾞｲﾂ語(ﾄﾞｲﾂ)
100 %

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。

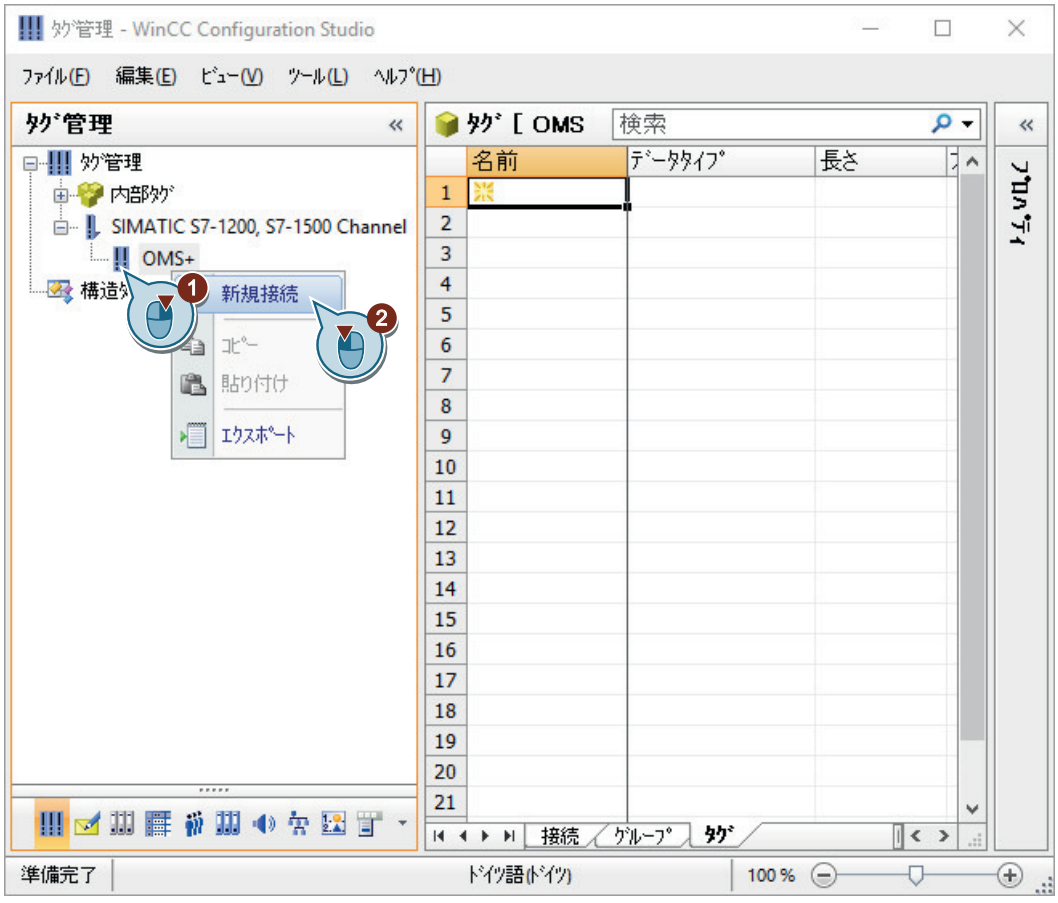
ランタイムには、この接続を通してデータ交換が行なわれます。

必要条件

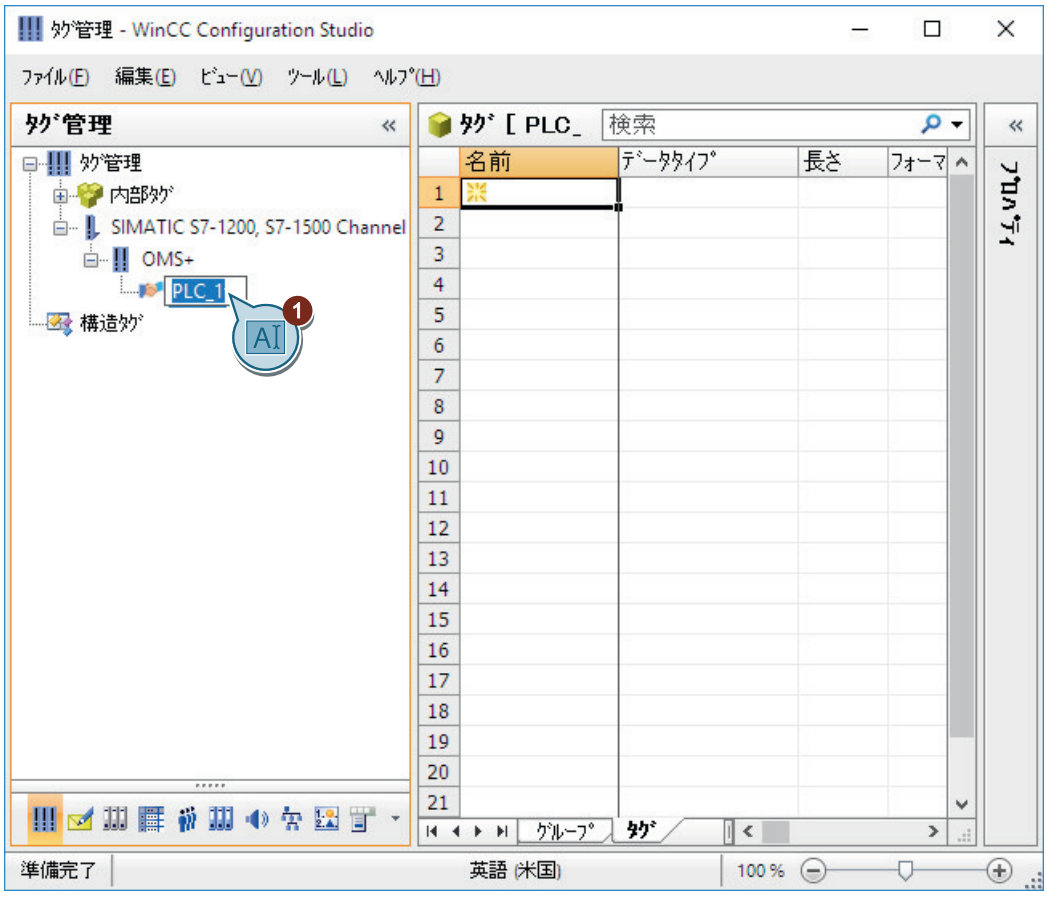
- "SIMATIC S7-1200、S7-1500 チャンネル"がタグ管理に追加されていること。

手順

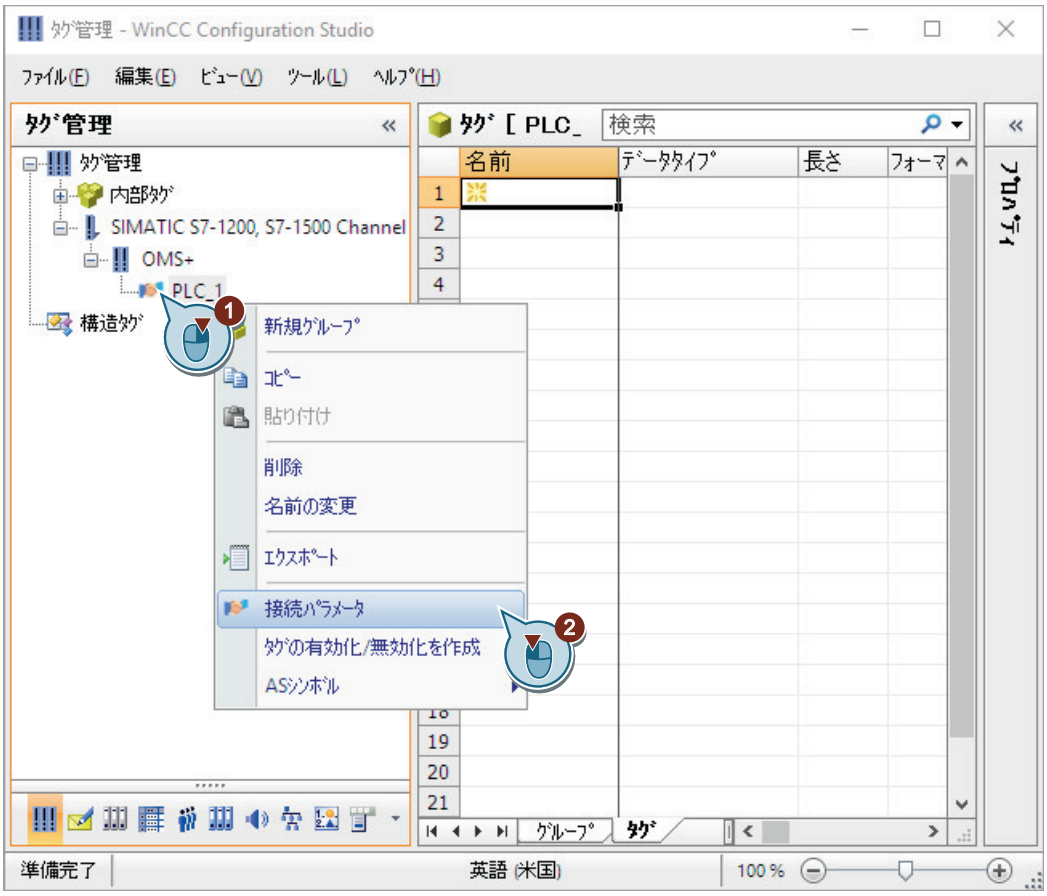
1. チャンネルユニット"OMS+"の下で、新しい接続を作成します。



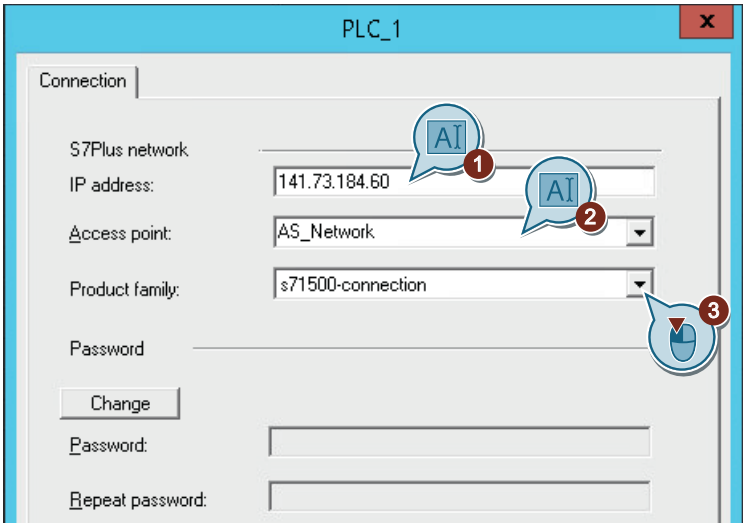
2. 接続名として"PLC_1"と入力します。


ﾀｸﾞ管理 - WinCC Configuration Studio
ﾌｧｲﾙ(F) 編集(E) ﾋﾞｭｰ(V) ﾂｰﾙ(L) ﾍﾙﾌﾟ(H)
ﾀｸﾞ管理
ﾀｸﾞ管理
内部ﾀｸﾞ
SIMATIC S7-1200, S7-1500 Channel
OMS+
PLC_1
構造ﾀｸﾞ
1
ﾀｸﾞ [PLC_
検索
名前
ﾃﾞｰﾀﾀｲﾌﾟ
長さ
ﾌｫｰﾏ
ﾌﾟﾛﾊﾟﾃｨ
ｸﾞﾙｰﾌﾟ
ﾀｸﾞ
準備完了
英語 (米国)
100 %

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。

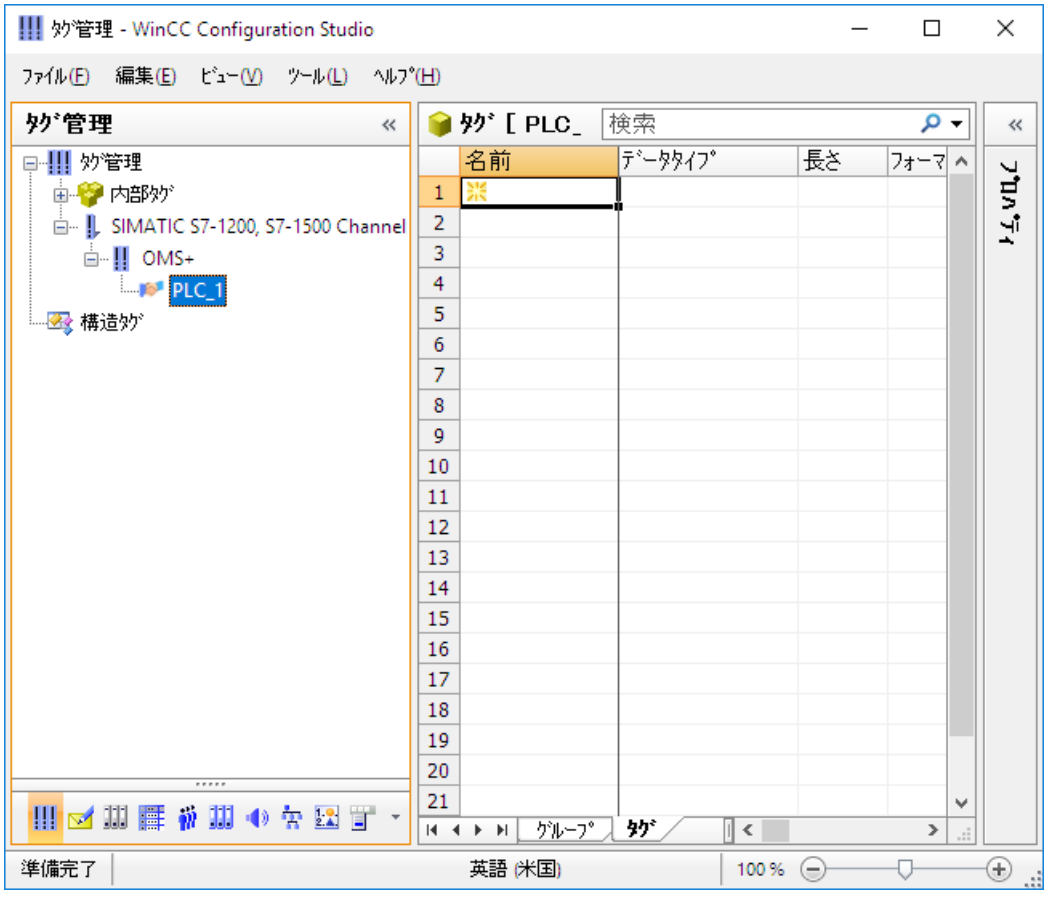


5. [OK]を押してダイアログを閉じます。

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。

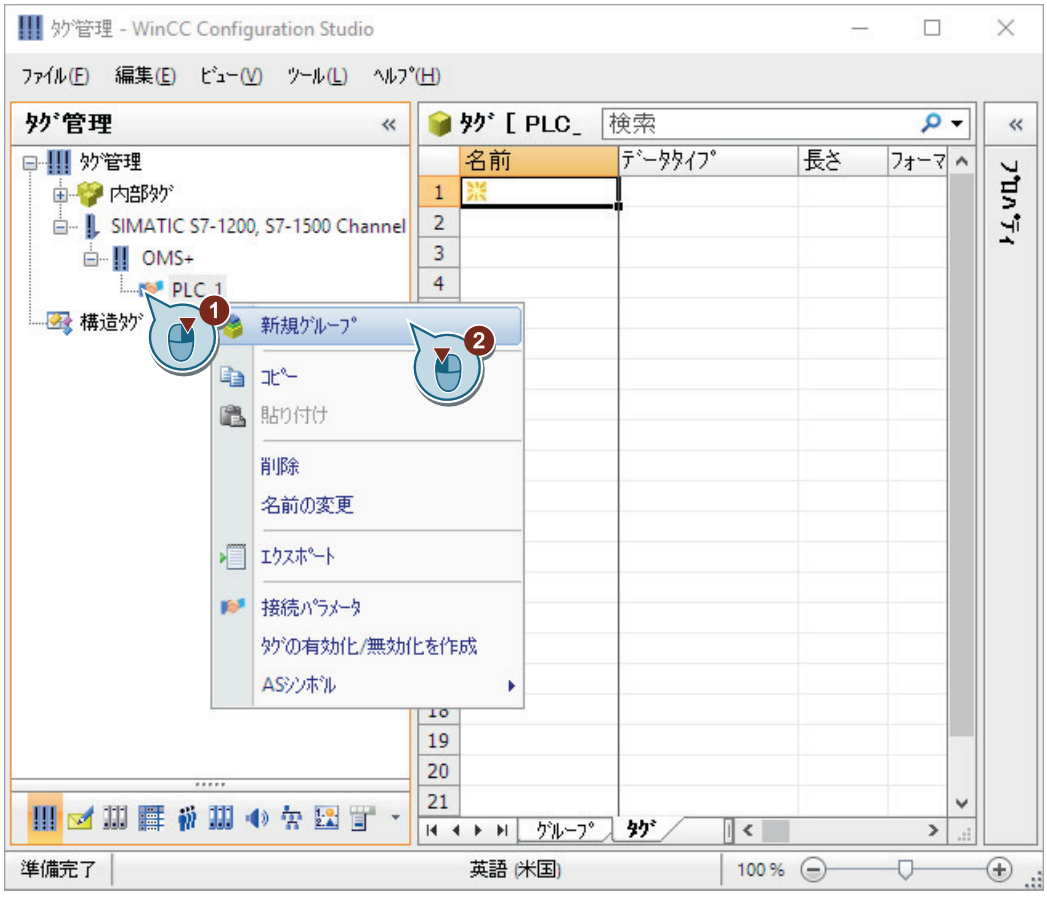
- 名前
- データタイプ
  データタイプは、WinCC におけるデータフォーマットを決定します。
  WinCC のタグのデータタイプは、AS で使用されるデータタイプとは異なります。
- アドレス
  アドレス指定によって、AS の特定のデータ範囲をプロセスタグに割り当てます。
  アドレス指定のタイプは、通信パートナのタイプに依存します。

必要条件

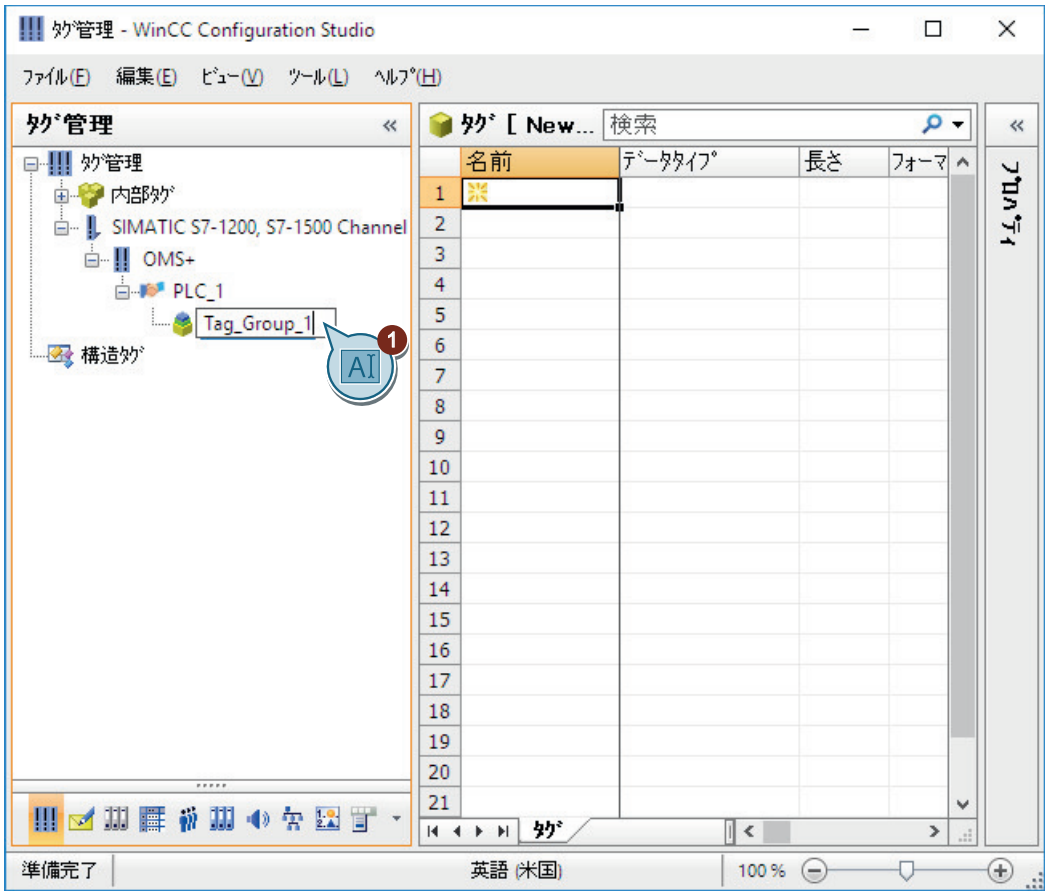
- "PLC_1"接続がタグ管理で作成されていること。

手順

1. タググループを作成します。

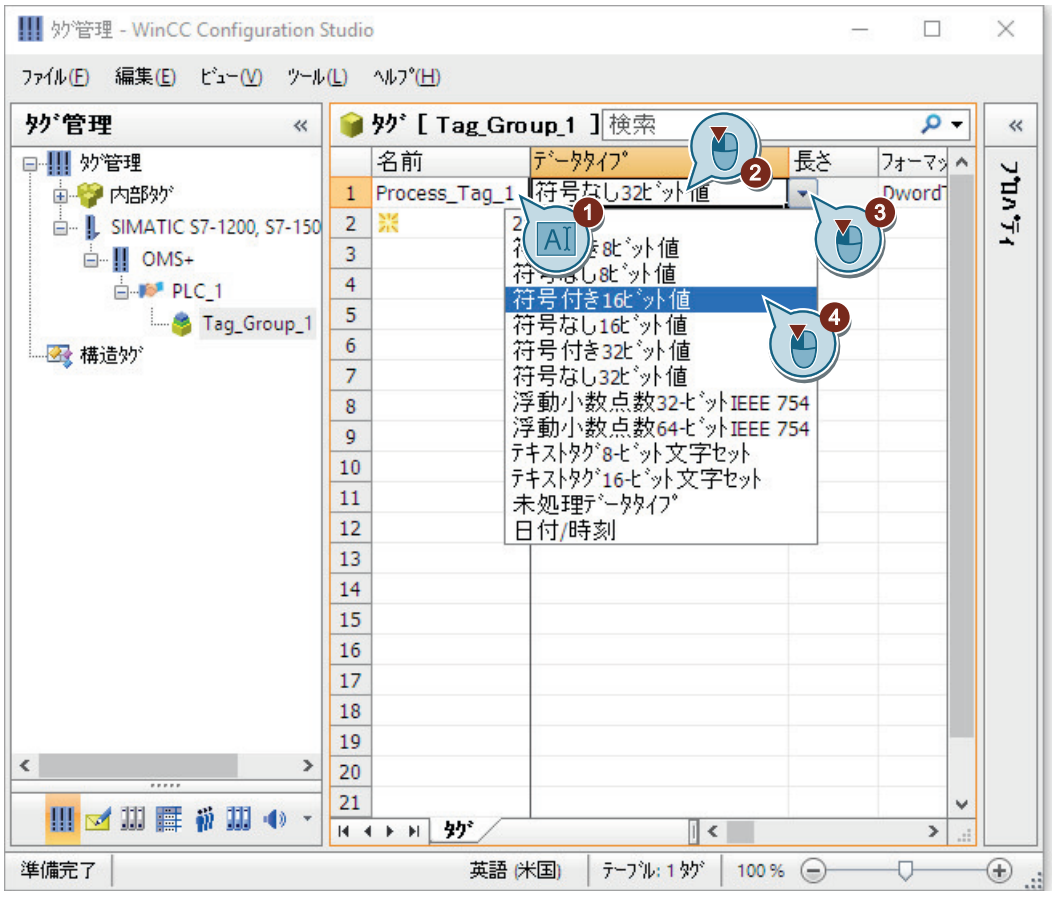


2. タググループ名として"Tag_Group_1"と入力します。

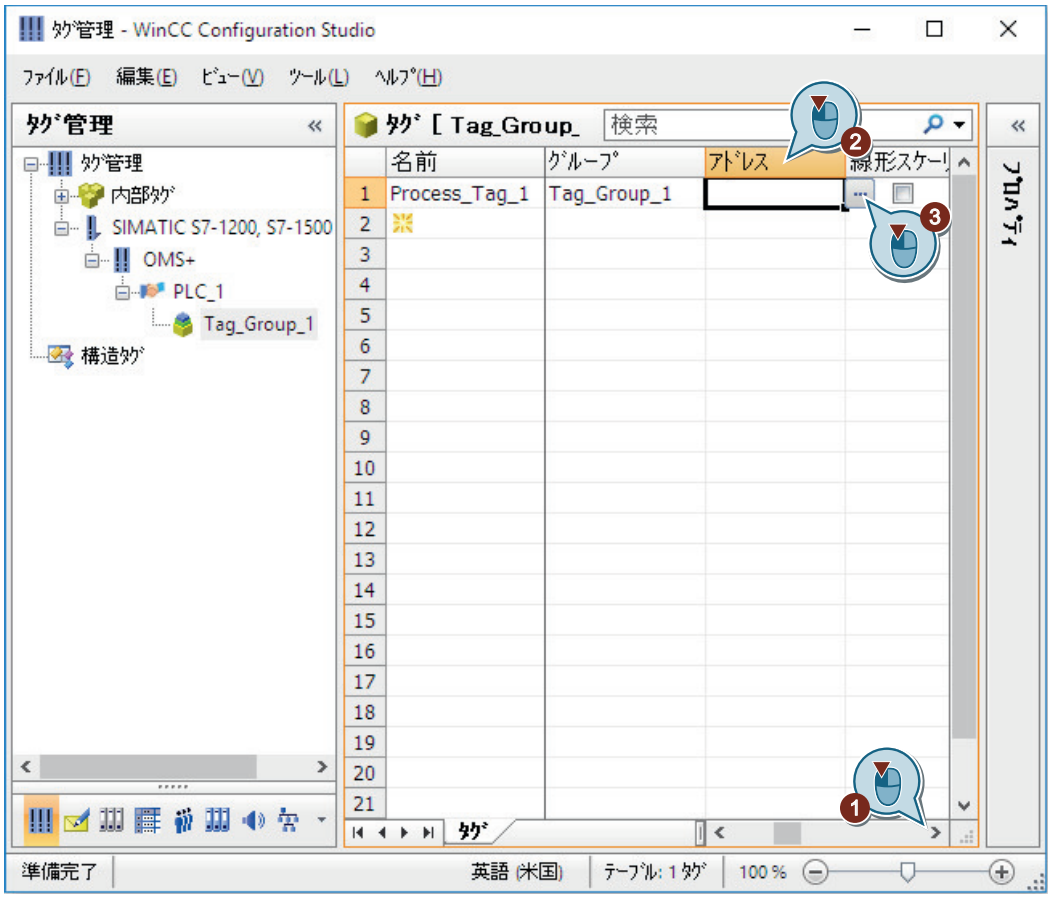


3. プロセスタグ名として、"Process_Tag_1"を入力して、タググループでプロセスタグを作成します。

4. "符号付き 16 ビット値"データタイプを選択します。

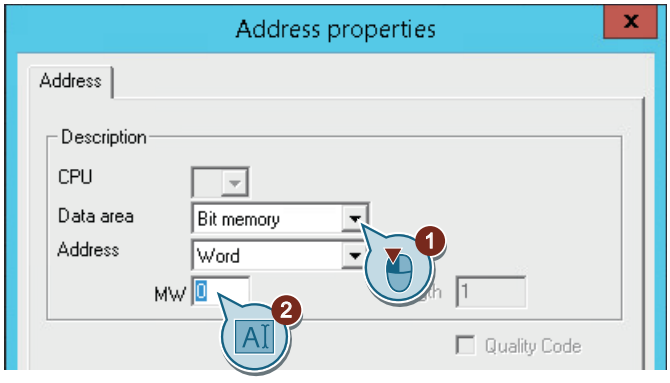


5. [アドレス]欄の[...]をクリックして、タグのアドレス指定のプロパティを指定します。



[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

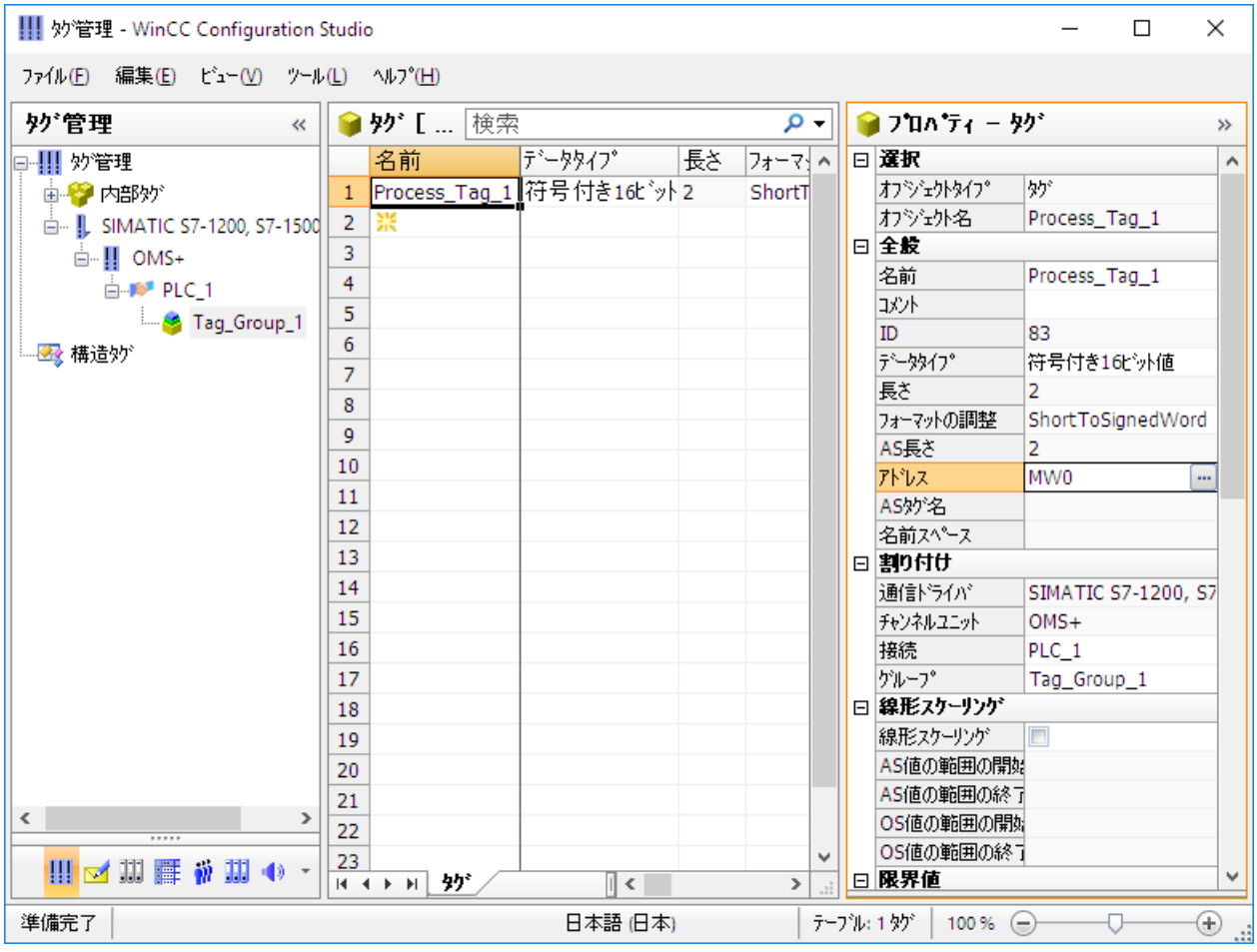
プロセスタグ"Process_Tag_1"が作成されます。[タグ管理]エディタのテーブルエリアに、プロセスタグが表示されます。

[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リ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。

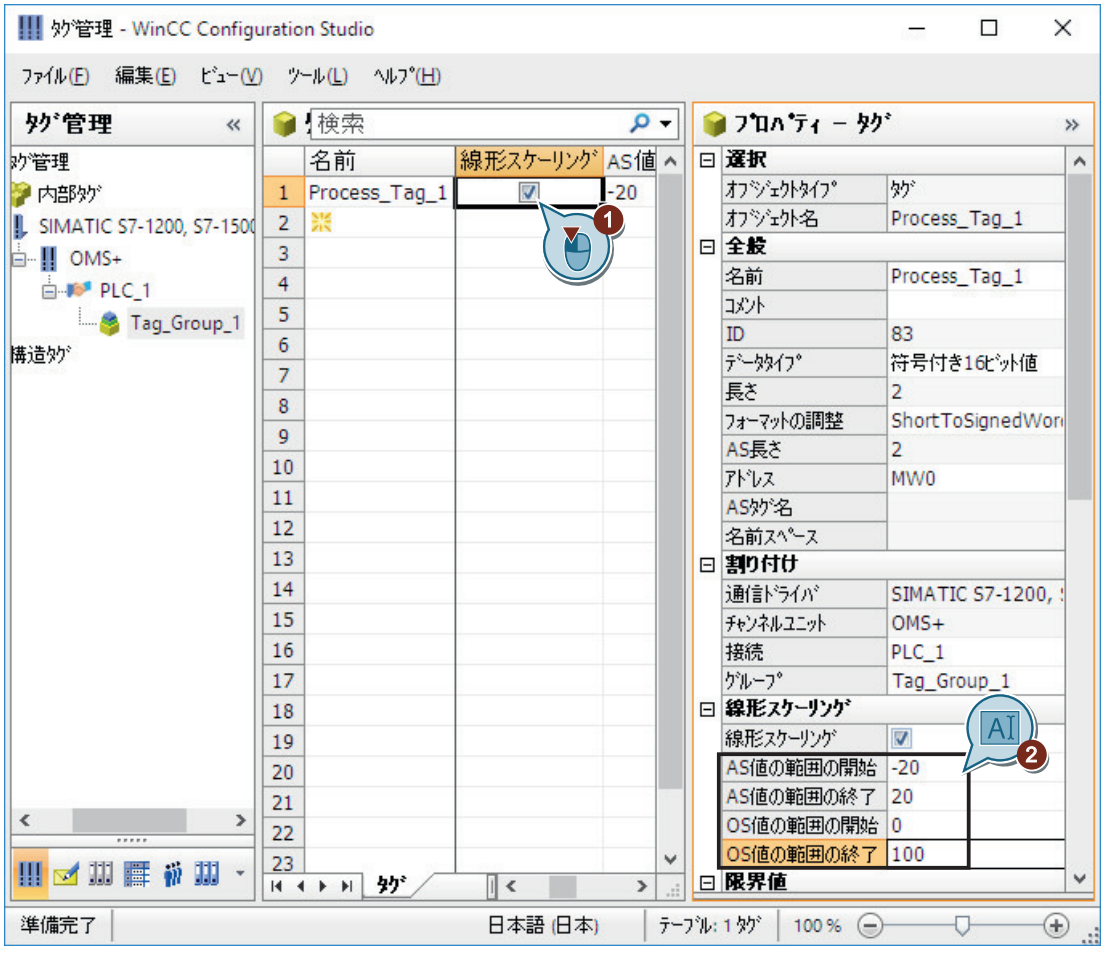
タグ管理では、抵抗値は[AS 値の範囲]列の設定に対応しています。温度は"OS 値の範囲"列の設定に対応しています。

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。

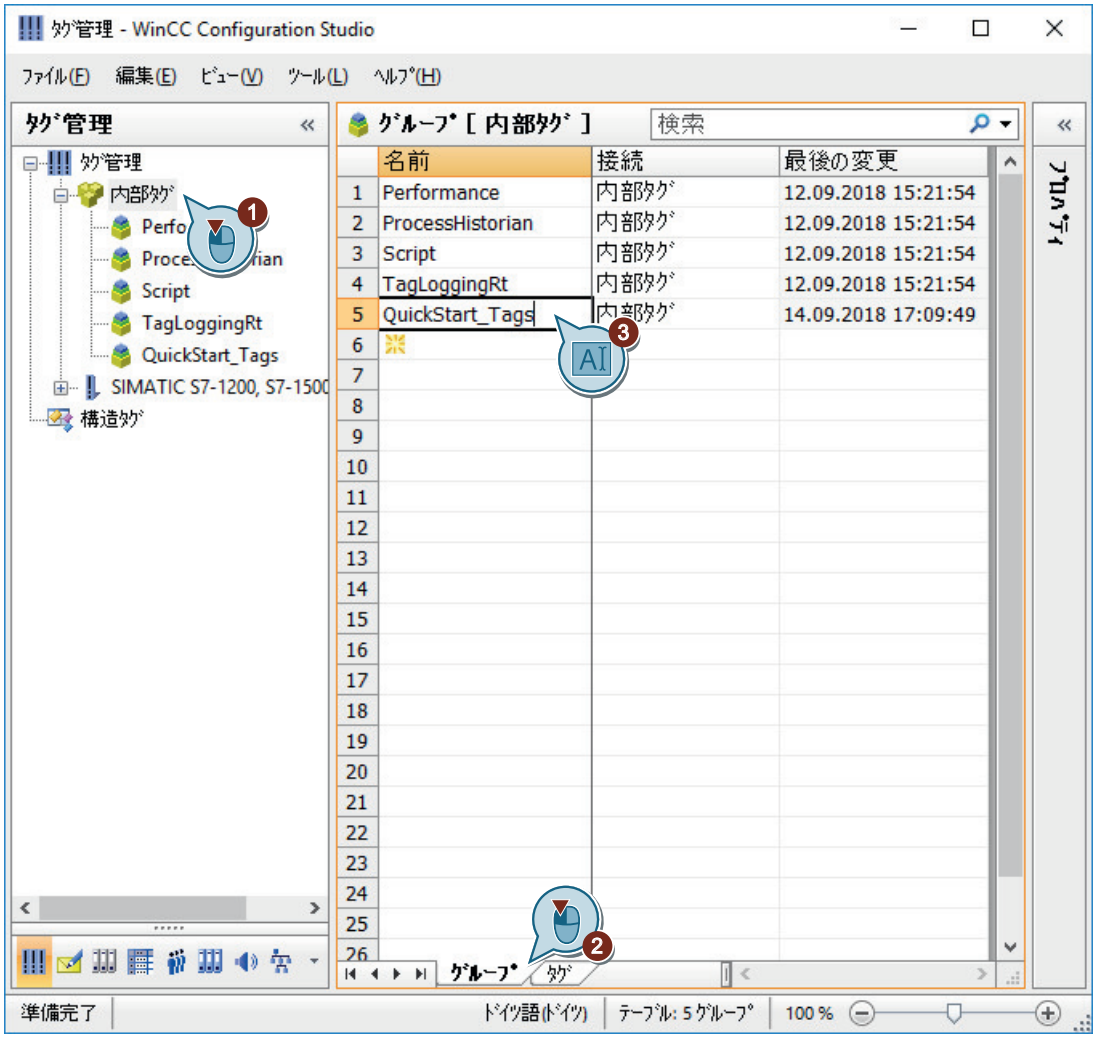
内部タグは、WinCC 内で値を転送するために使用します。

必要条件

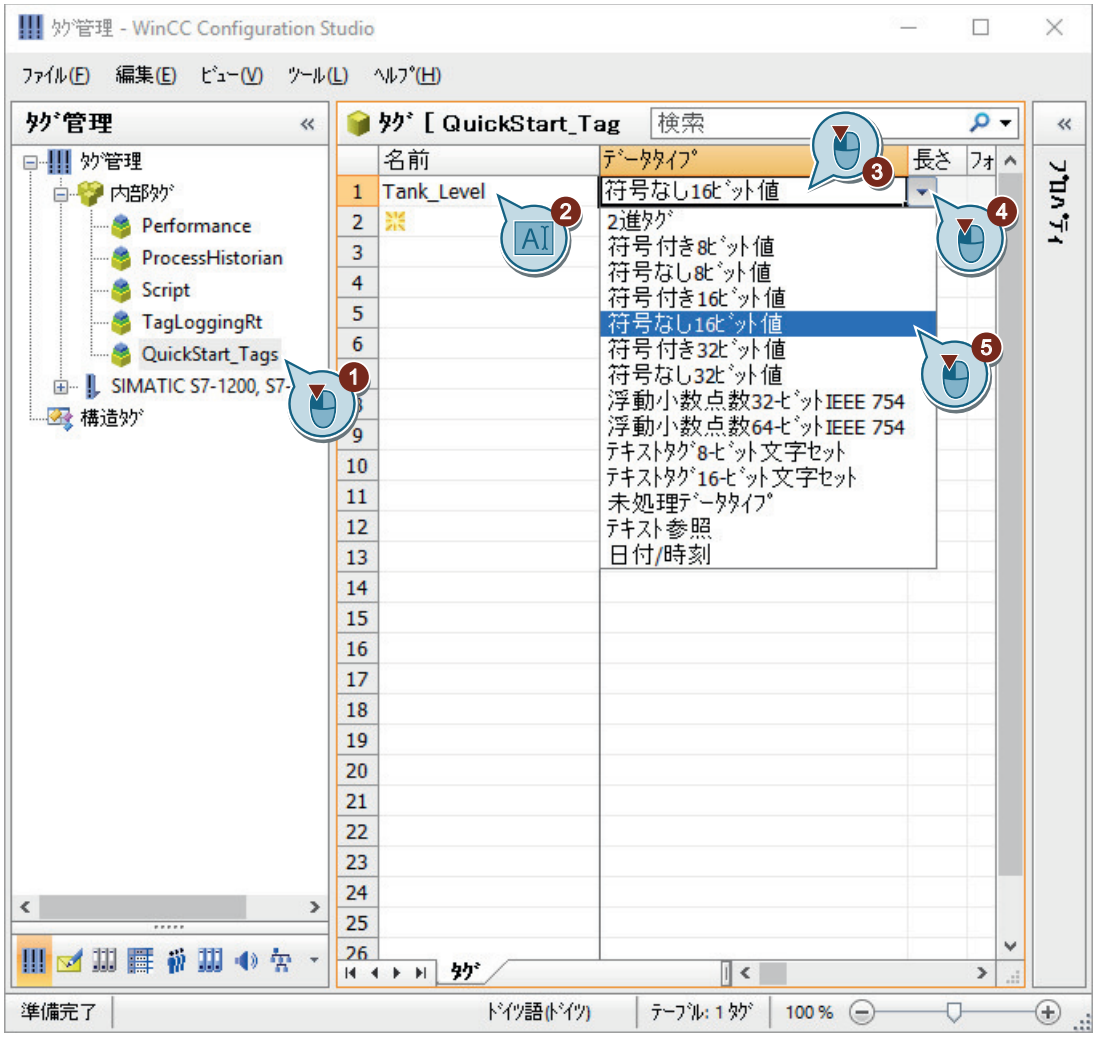
- "Quick_Start"プロジェクトが開いている。

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。



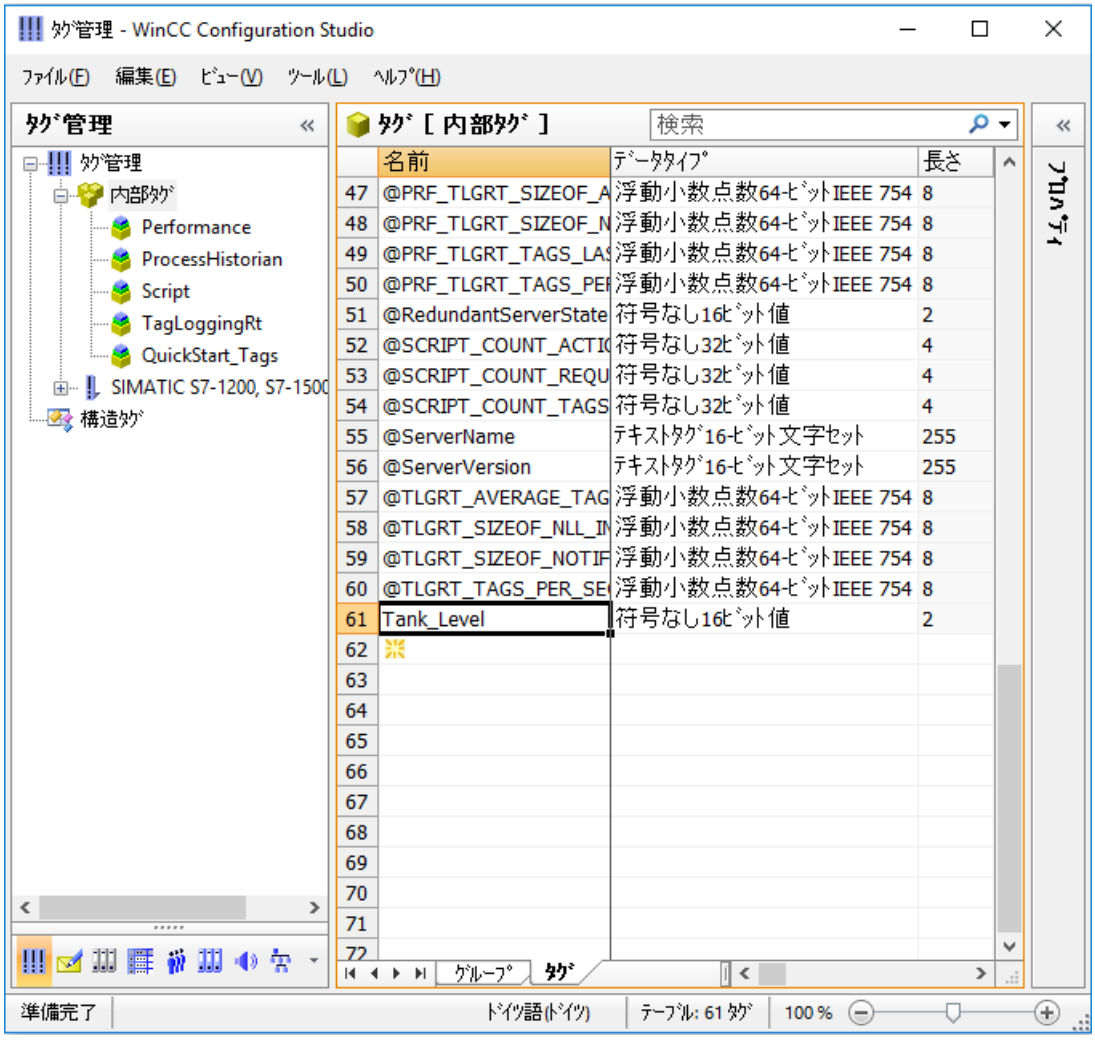
3. 内部タグの名前として"Tank_Level"と入力して、内部タグをタググループで作成します。
4. データタイプ[符号なし 16 ビット値]を選択します。



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。


ﾀｸﾞ管理 - WinCC Configuration Studio
ﾌｧｲﾙ(F) 編集(E) ﾋﾞｭｰ(V) ﾂｰﾙ(L) ﾍﾙﾌﾟ(H)
ﾀｸﾞ管理
ﾀｸﾞ管理
内部ﾀｸﾞ
Performance
ProcessHistorian
Script
TagLoggingRt
QuickStart_Tags
SIMATIC S7-1200, S7-1500
構造ﾀｸﾞ
ﾀｸﾞ [内部ﾀｸﾞ]
検索
名前
ﾃﾞｰﾀﾀｲﾌﾟ
長さ
47 @PRF_TLGRT_SIZEOF_A 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
48 @PRF_TLGRT_SIZEOF_N 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
49 @PRF_TLGRT_TAGS_LAS 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
50 @PRF_TLGRT_TAGS_PER 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
51 @RedundantServerState 符号なし16ﾋﾞｯﾄ値 2
52 @SCRIPT_COUNT_ACTIC 符号なし32ﾋﾞｯﾄ値 4
53 @SCRIPT_COUNT_REQU 符号なし32ﾋﾞｯﾄ値 4
54 @SCRIPT_COUNT_TAGS 符号なし32ﾋﾞｯﾄ値 4
55 @ServerName ﾃｷｽﾄﾀｸﾞ16-ﾋﾞｯﾄ文字ｾｯﾄ 255
56 @ServerVersion ﾃｷｽﾄﾀｸﾞ16-ﾋﾞｯﾄ文字ｾｯﾄ 255
57 @TLGRT_AVERAGE_TAG 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
58 @TLGRT_SIZEOF_NLL_IN 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
59 @TLGRT_SIZEOF_NOTIF 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
60 @TLGRT_TAGS_PER_SE 浮動小数点数64-ﾋﾞｯﾄIEEE 754 8
61 Tank_Level 符号なし16ﾋﾞｯﾄ値 2
62
63
64
65
66
67
68
69
70
71
72
ｸﾞﾙｰﾌﾟ
ﾀｸﾞ
ﾌﾟﾛﾊﾟﾃｨ
準備完了
ﾄﾞｲﾂ語(ﾄﾞｲﾂ)
ﾃｰﾌﾞﾙ: 61 ﾀｸﾞ
100 %

プロセス画面の設定 6

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]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。

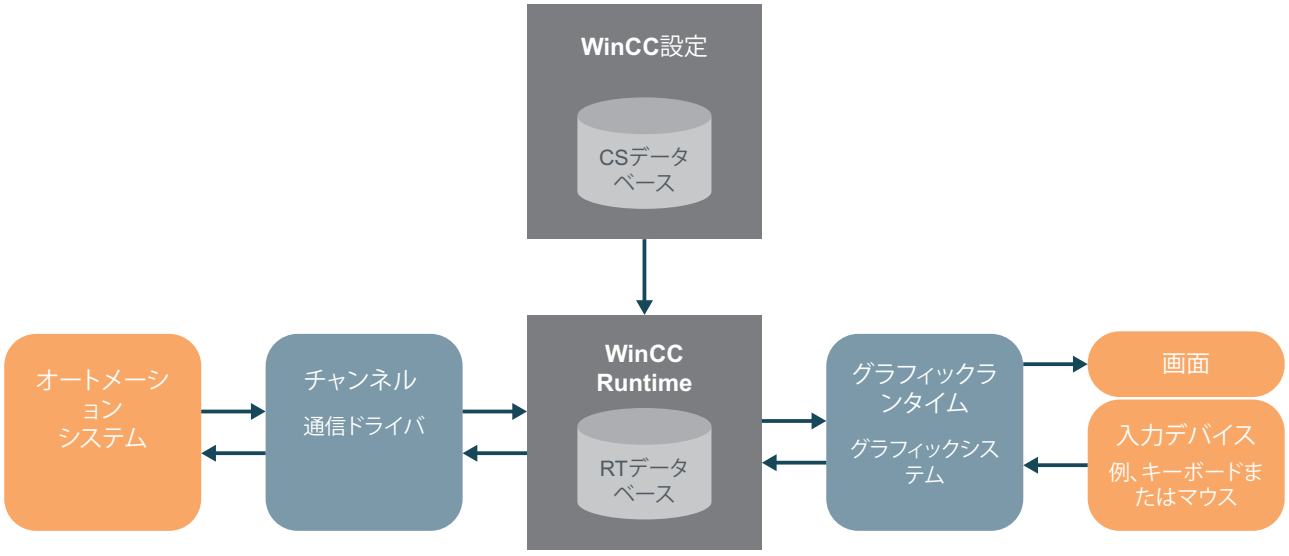
- テキスト、グラフィックまたはボタンなどのスタティックでオペレータ操作可能なオブジェクトを表示します。
- プロセス値に応じて棒グラフの長さを変更するなど、ダイナミックオブジェクトを更新します。
- ボタンをクリックしたり、テキストボックスにテキストを入力するなどのオペレータ入力に対応します。

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]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]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。

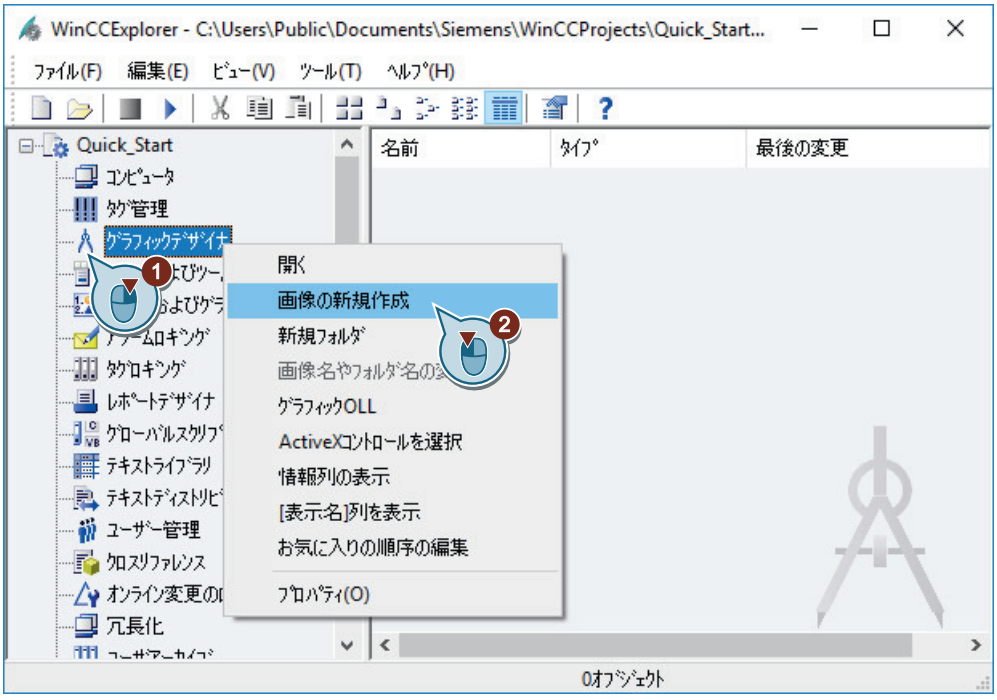
- すぐにプロセス画像を編集する場合は、[グラフィックデザイナ]エディタを使用して作成します。
- 処理する前に必要なプロセス画像をすべて作成しておく場合は、WinCC エクスプローラを使用することを推奨します。

必要条件

- "Quick_Start"プロジェクトが開いている。

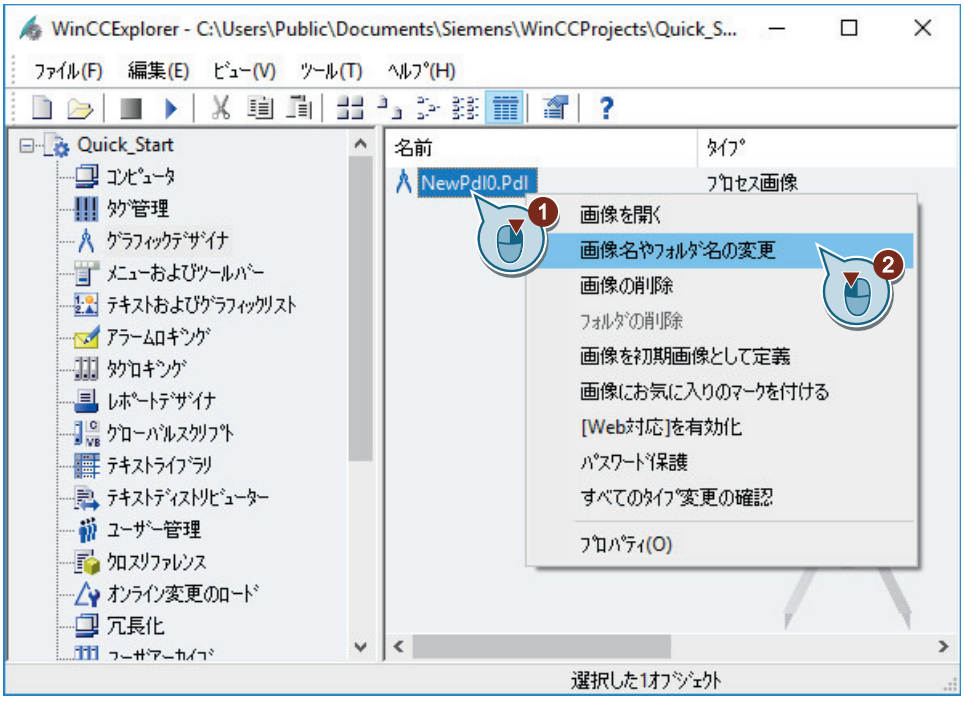
手順

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。

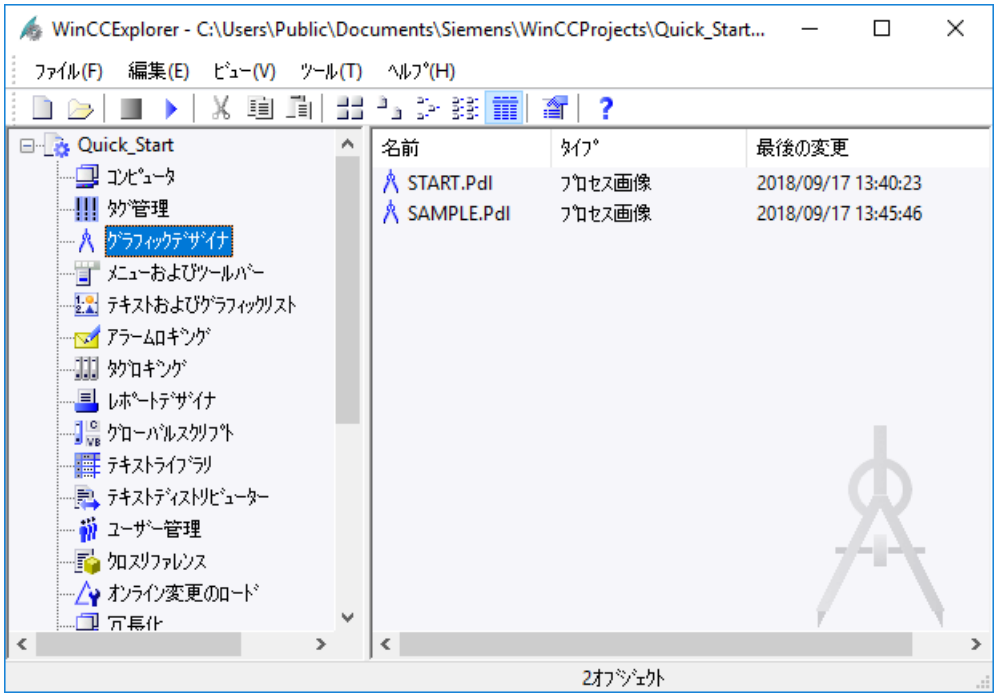
3. ステップ 1 およびステップ 2 と同様にして 2 番目のプロセス画像を作成し、それに "SAMPLE.pdl"という名前を付けます。

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]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]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]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]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、ドブル 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。

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"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]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。

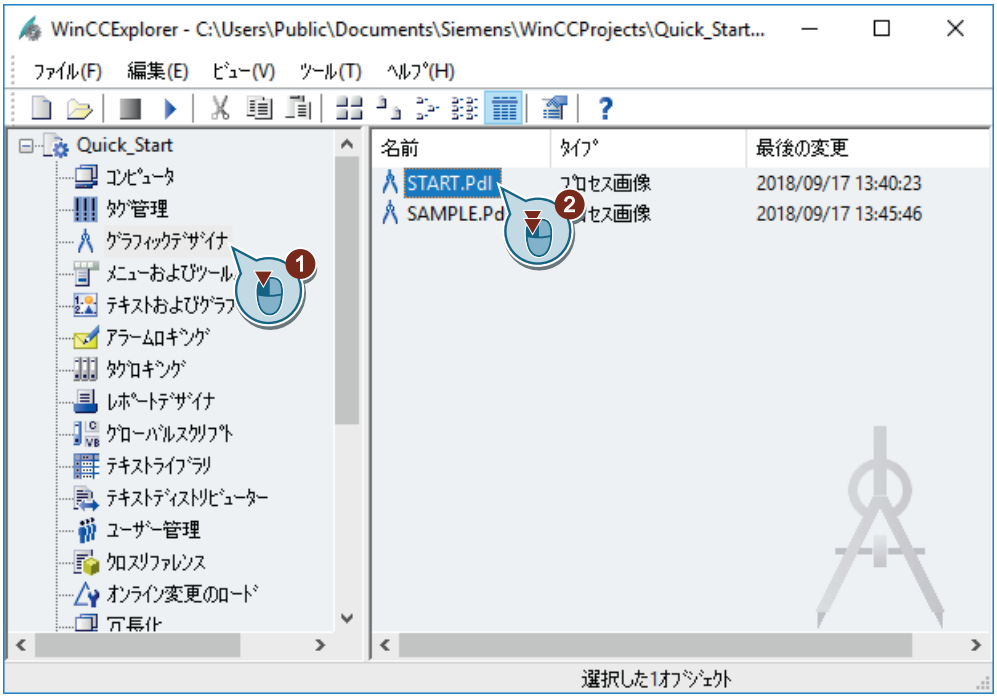
- グローバルライブラリ
  [グローバル SVG ライブラリ]エリアは、ディレクトリツリーの構造です。
  マシンやシステムのパーツ、測定デバイス、操作コントロールやビルなどの様々な事前に作成されたグラフィックオブジェクトを提供します。
- プロジェクトライブラリ
  [プロジェクト SVG ライブラリ]エリアには、カスタムプロジェクトを保存できます。

"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。

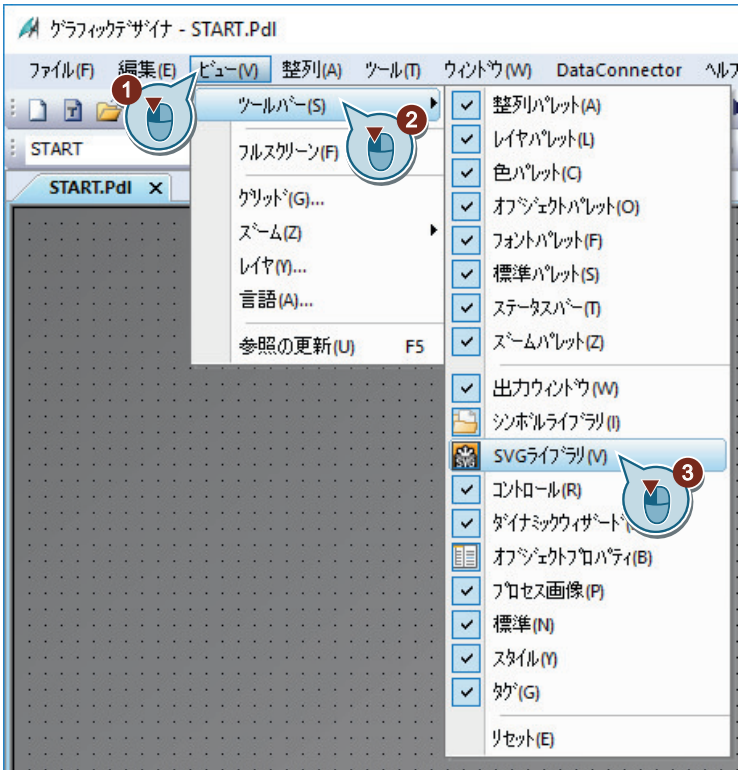
これらのオブジェクトを使用して、"START.pdl"プロセス画像で水処理システムを描写します。

手順

1. "グラフィックデザイナ"エディタで、"START.pdl"プロセス画像を開きます。

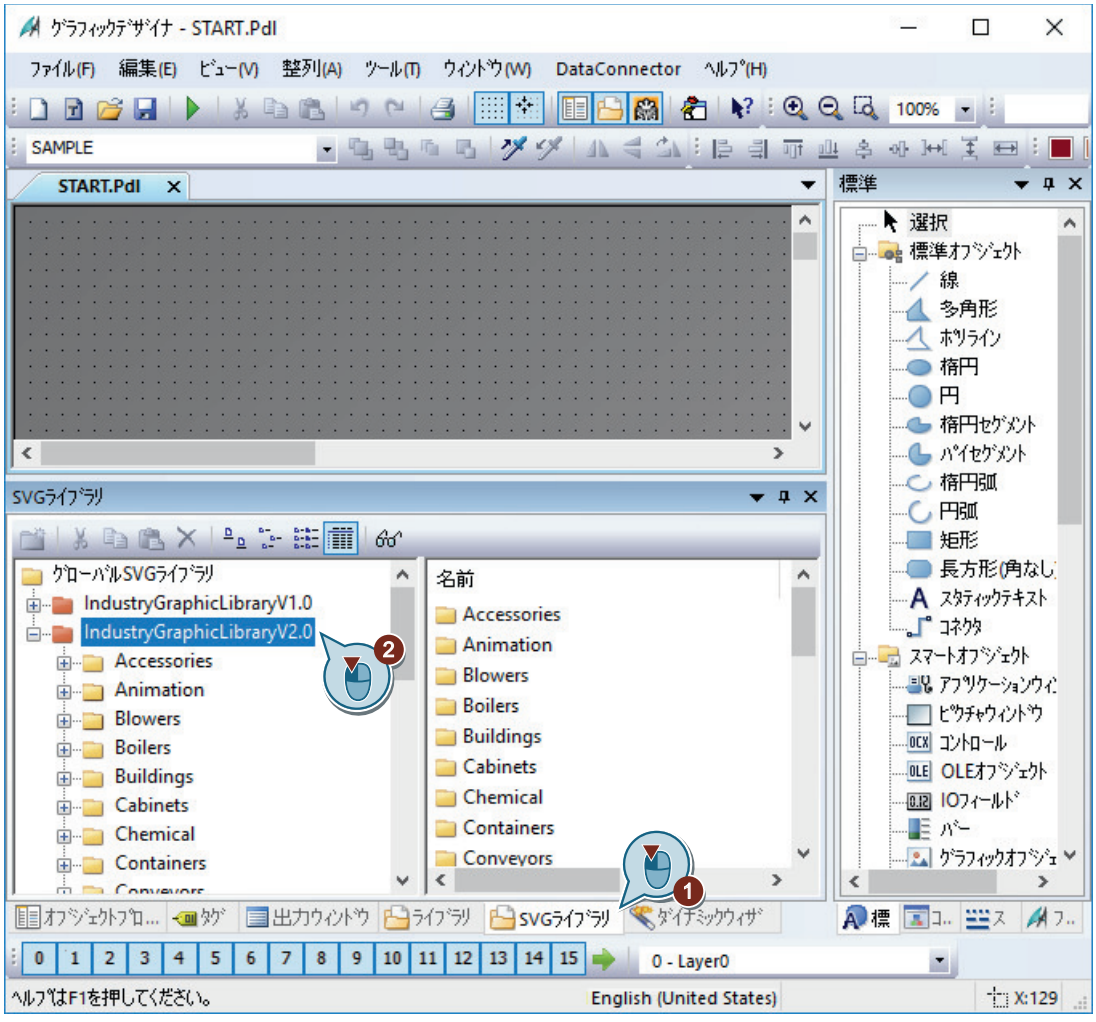


2. [SVG ライブラリ]ウィンドウが表示されない場合、ビューを有効にします。



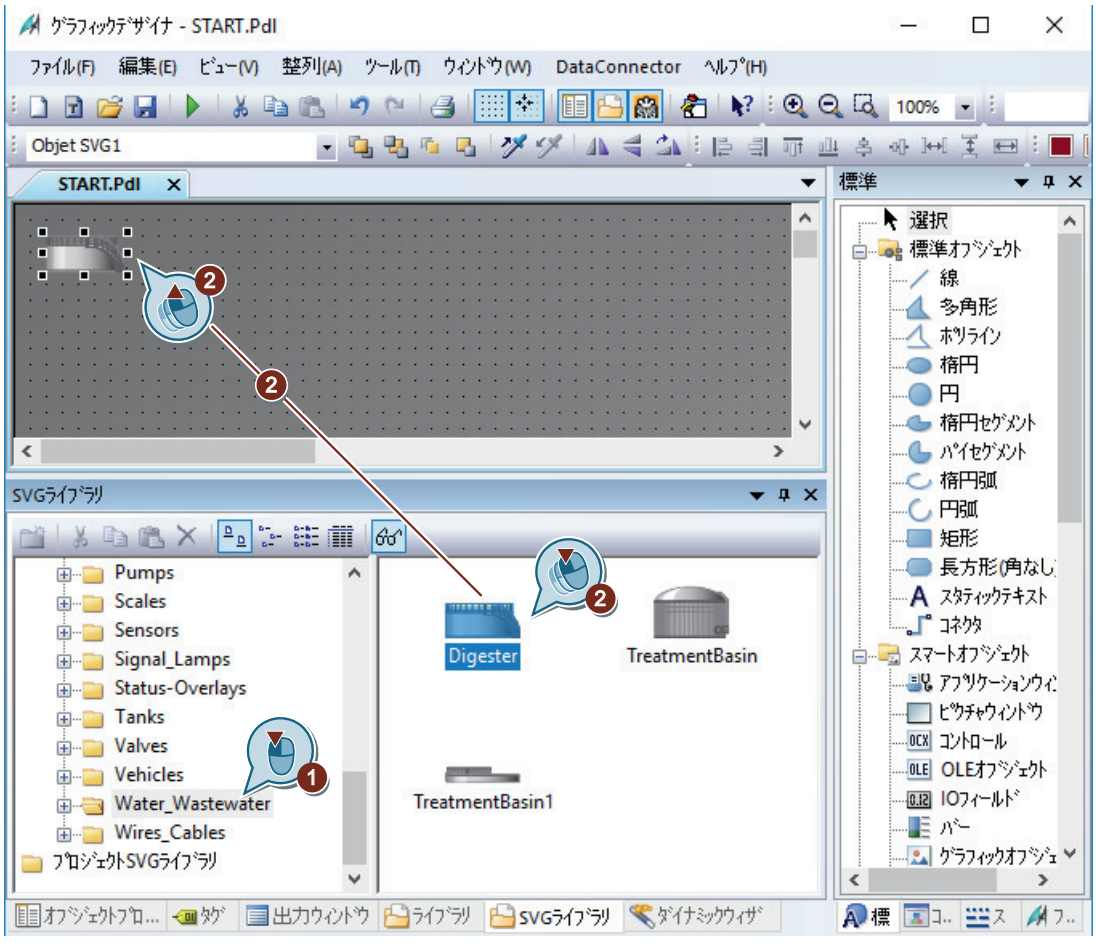
[SVG ライブラリ]ウィンドウが開きます。

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



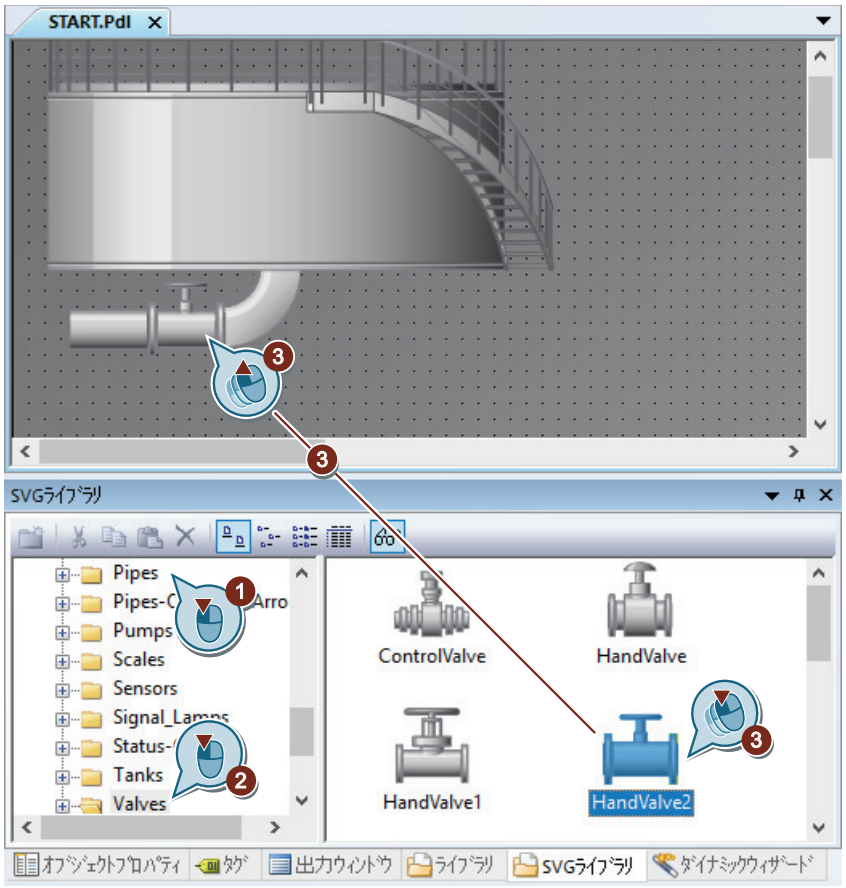
ライブラリのツールバーの ボタンを使用して、使用可能なオブジェクトのプレビューを表示します。
と ボタンを使用して表示されたシンボルのサイズを変更することができます。

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



バイオコンバータの図が、作業エリアに表示されます。
マウスで表示されたポイントのいずれかをクリックアンドドラッグして画像を拡大します。

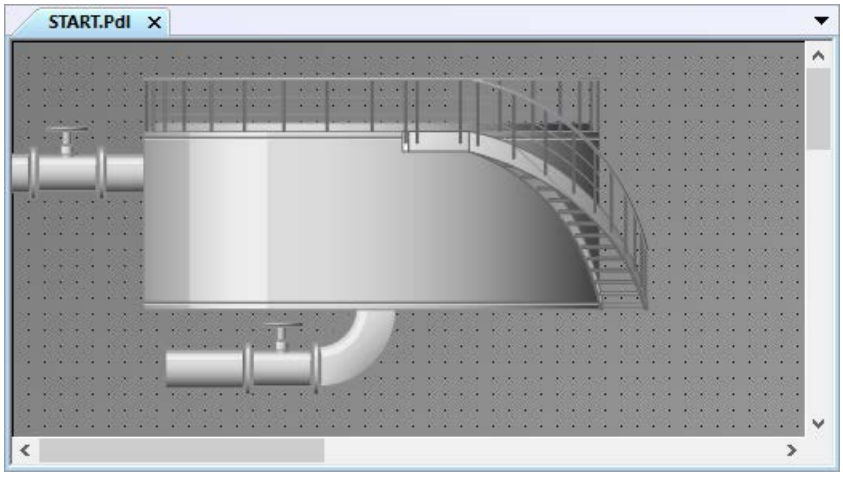
5. 他のオブジェクトの表示を追加します(例、パイプラインやバルブ)。
マウスまたはカーソルキーを使用して、プロセス画像上でオブジェクトの画像を移動します。



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。



6.4.3 [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。

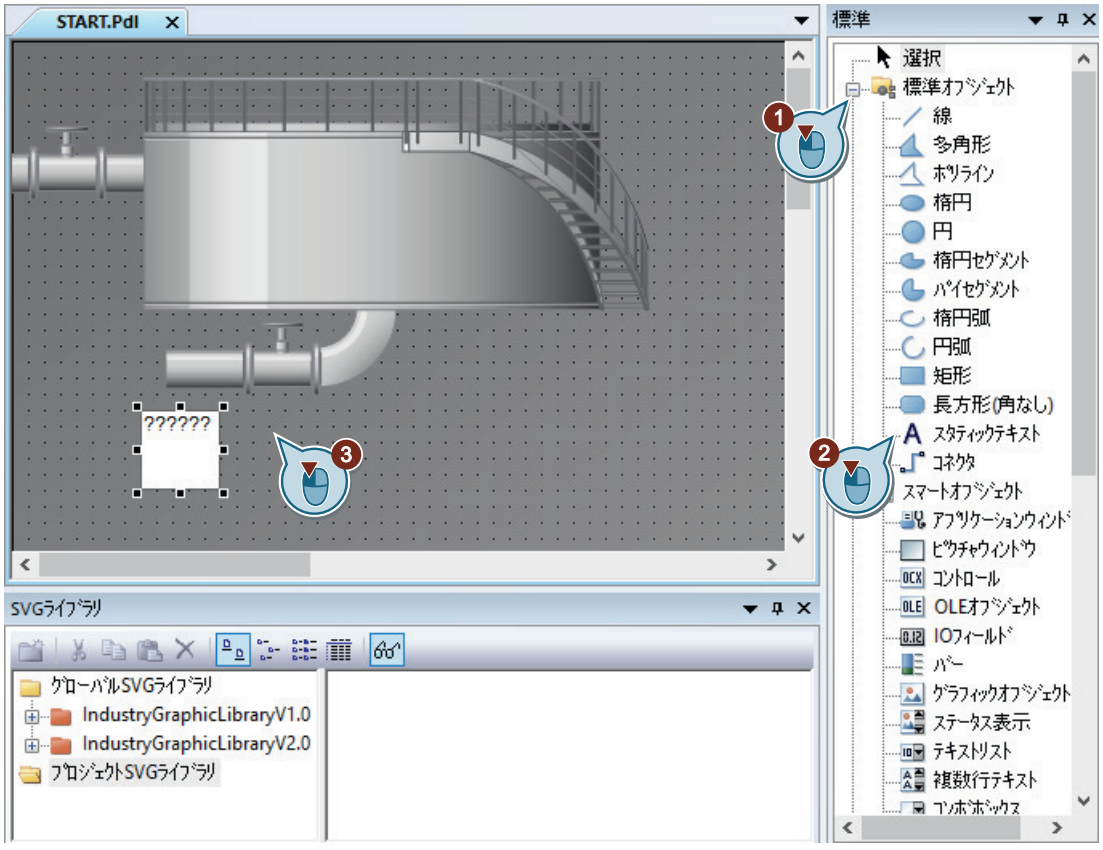
"Quick_Start"プロジェクトでは、表示されたプロセスにラベル付けするスタティックテキストが必要です。複数の図を作成する場合は、表示されたプロセスに一意のラベルを付けることが非常に重要です。

必要条件

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

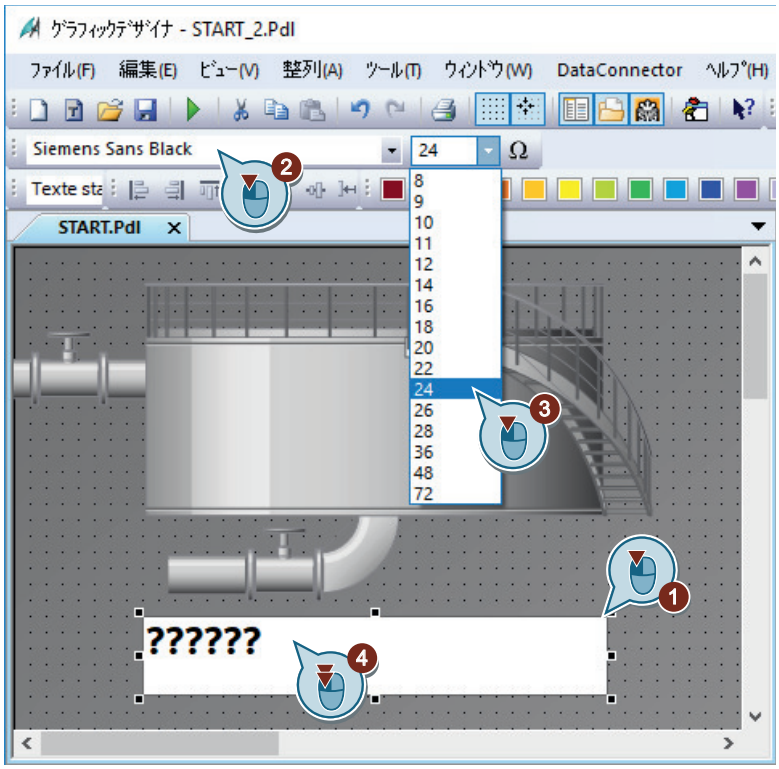
手順

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。

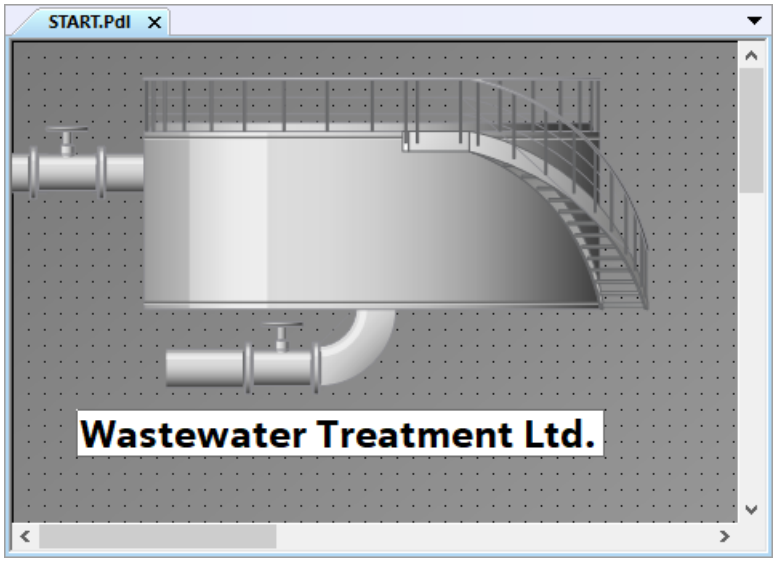
2. マウスを使用してテキストボックスをドラッグし、フォントタイプとサイズを設定します。



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。



6.4.4 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"エディタが開いている。

手順

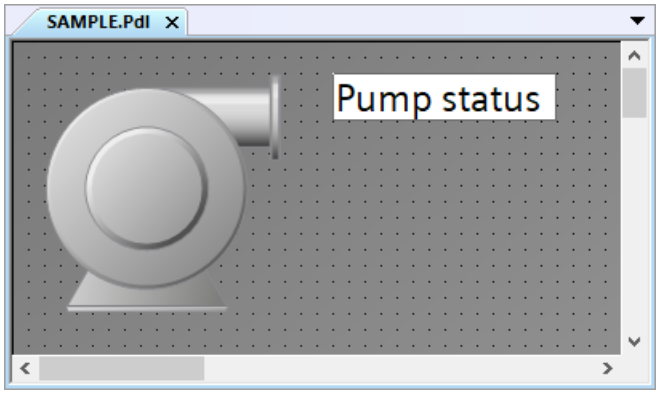
1. [グラフィックデザイナ]エディタのツールバーのボタンを使用して、"SAMPLE.pdl"プロセス画像を開きます。
2. ライブラリのグラフィックオブジェクトを使用して、任意のプロセスを表示します。

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 56)

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]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。

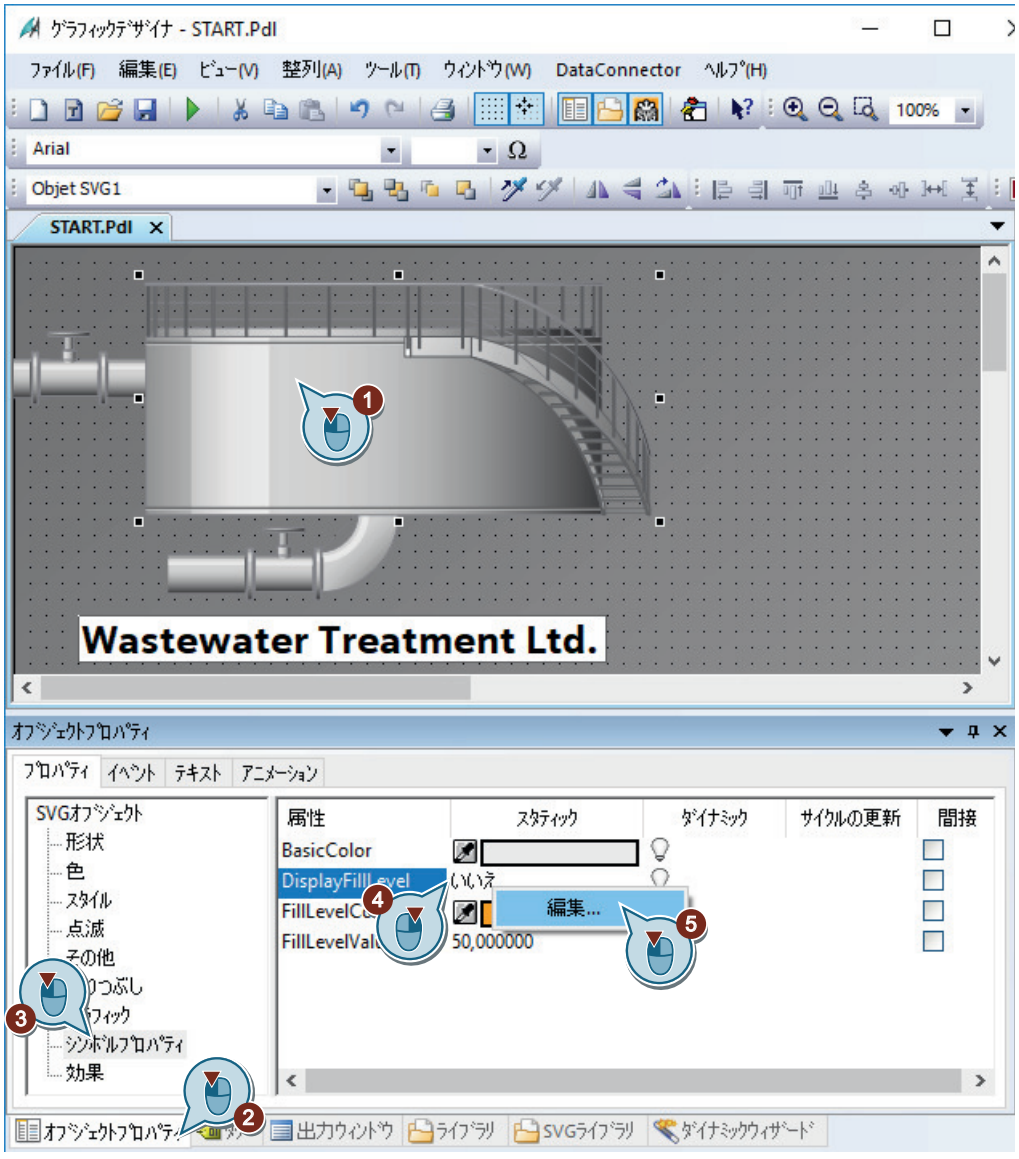
- バイオコンバータのグラフィック画像と内部タグ[Tank_Level]の接続
  内部タグ[Tank_Level]への接続によって、バイオコンバータを表すグラフィックオブジェクトにタグ値を転送できます。充填レベルインジケータが、ランタイムでタグ値に従って変化します。
  タグとオブジェクト間のリンクが、アイコンと太字で"オブジェクトプロパティ"ウィンドウに表示されます。
- 更新サイクルの指定
  更新サイクルは、フィルレベルインジケータを更新する時間間隔を決定します。

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。

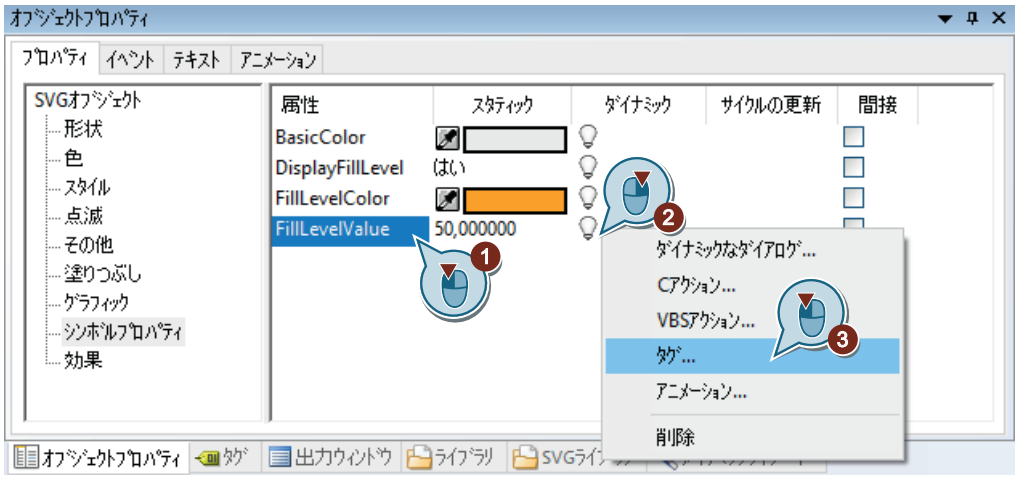
手順

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。

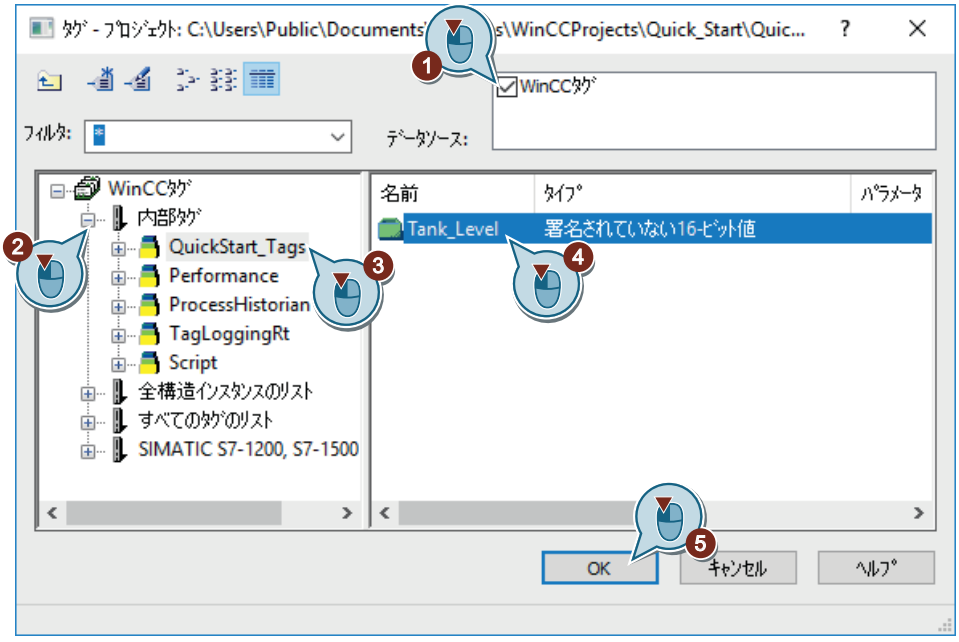


バーグラフが[Digester] SVG オブジェクトで、充填レベルインジケータとして表示されます。

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。

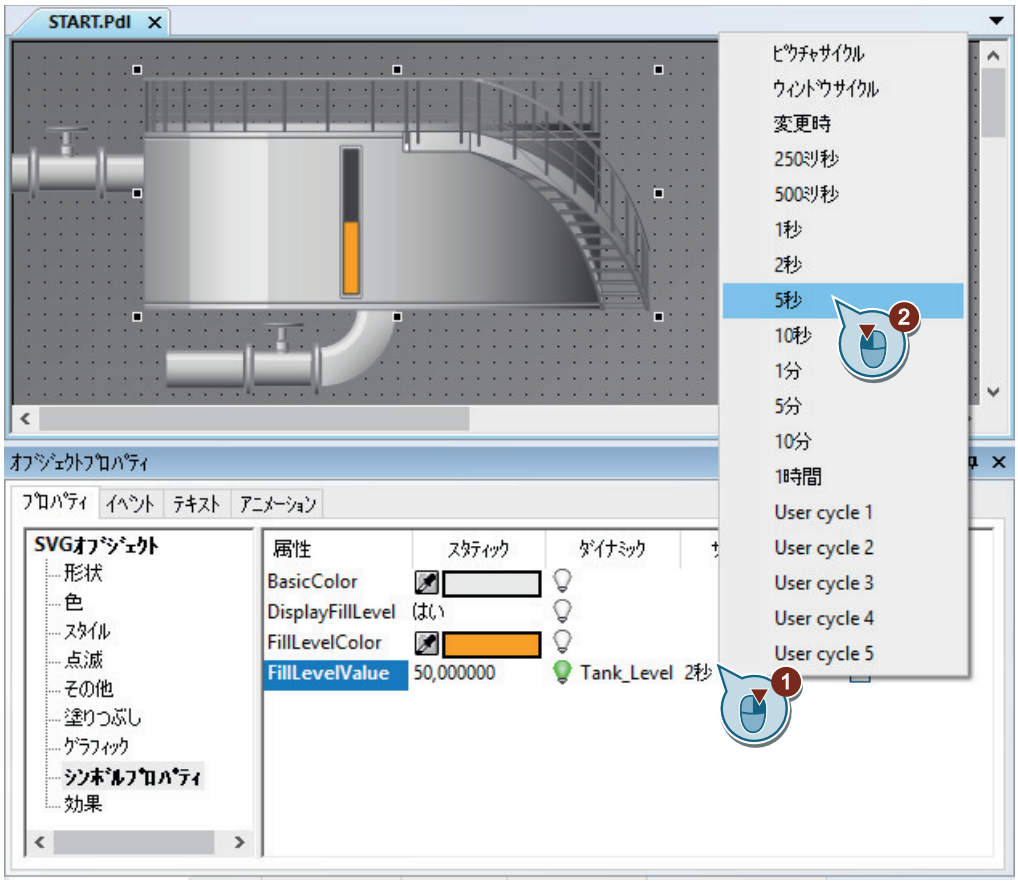


4. 内部タグ"Tank_Level"を選択します。
   必要な場合、"データソース"エリアで"WinCC タグ"オプションを選択します。



   "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。

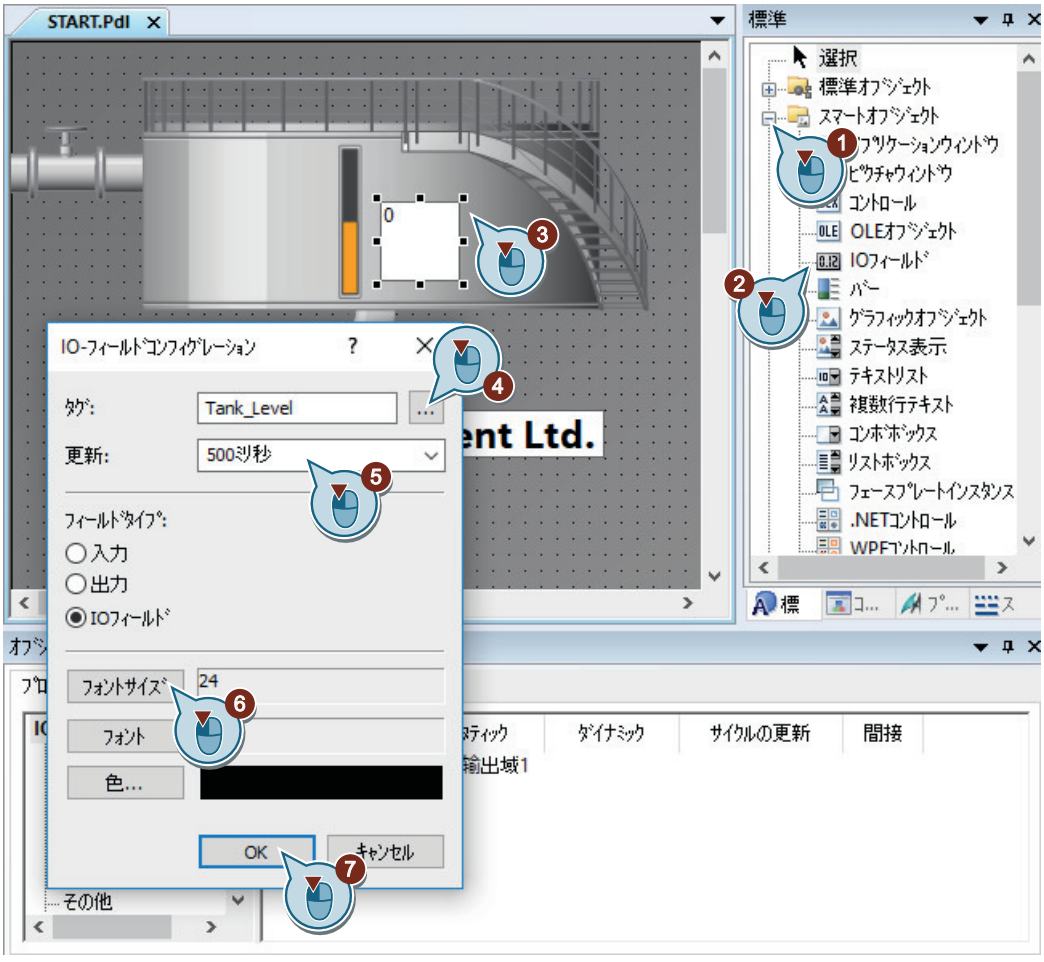
- I/O フィールドと内部タグ"Tank_Level"の接続
  I/O フィールドを、"Quick_Start"プロジェクトの内部タグ"Tank_Level"と接続します。これにより、I/O フィールドとバイオコンバータのグラフィック画像の間に間接的接続が作成されます。
  ランタイムに I/O フィールドの値を入力すると、この値は内部タグ"Tank_Level"によって取得されます。タグは、値をバイオコンバータを表すグラフィックオブジェクトに転送します。バイオコンバータの充填レベルインジケータが、タグ値に従ってランタイムに変化します。
- 更新の定義
  更新によって、I/O フィールド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]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。

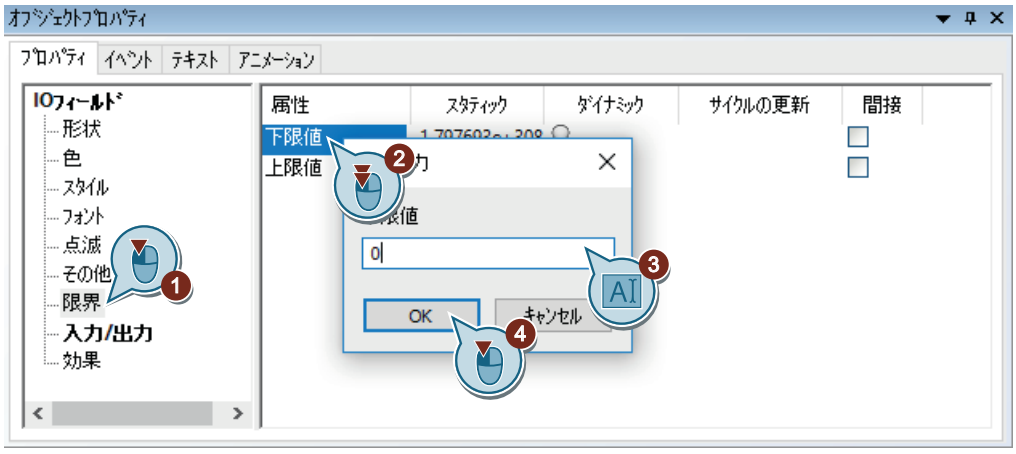
手順

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。

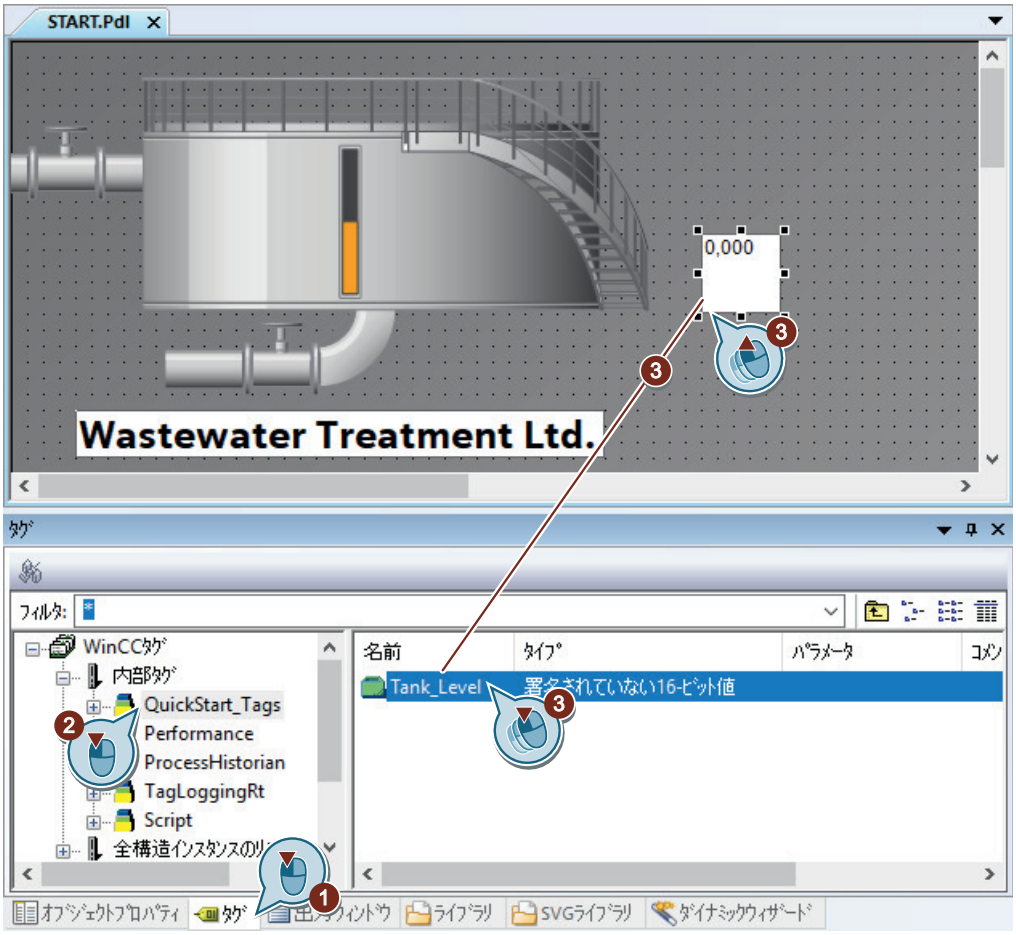
3. [制限]グループを[オブジェクトプロパティ]ウィンドウで開き、[下限値]を 0 に設定します。



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]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。

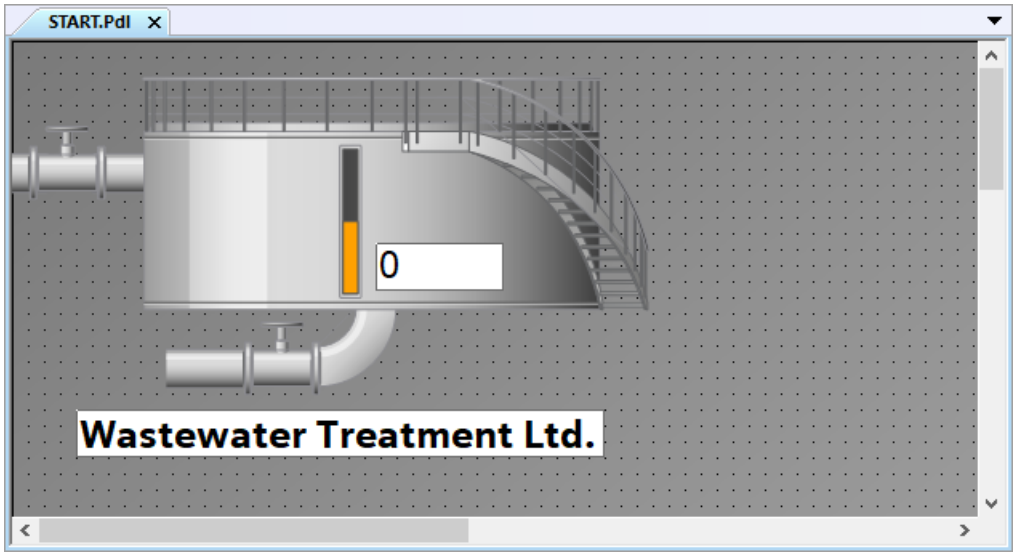
ランタイムに I/O フィールドの値を入力すると、この値は内部タグによってバイオコンバータのグラフィック画像に転送されます。充填レベルインジケータが、入力された値に従って変化します。

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ができます。

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

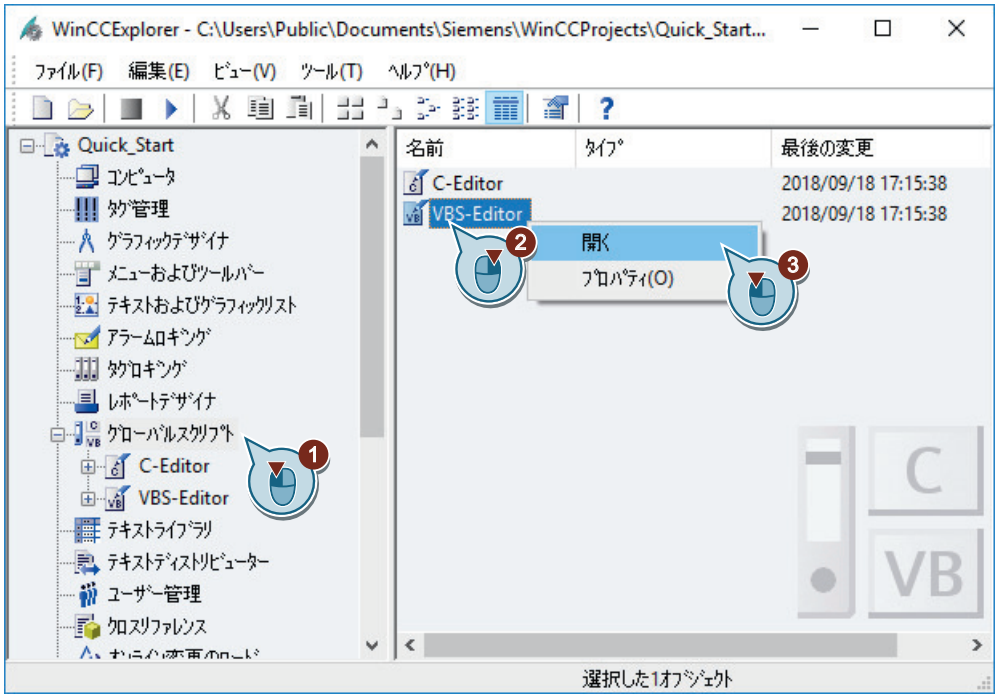
次のステップは、グローバルスクリプトのモジュールでのプロシージャの作成方法を示しています。

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。

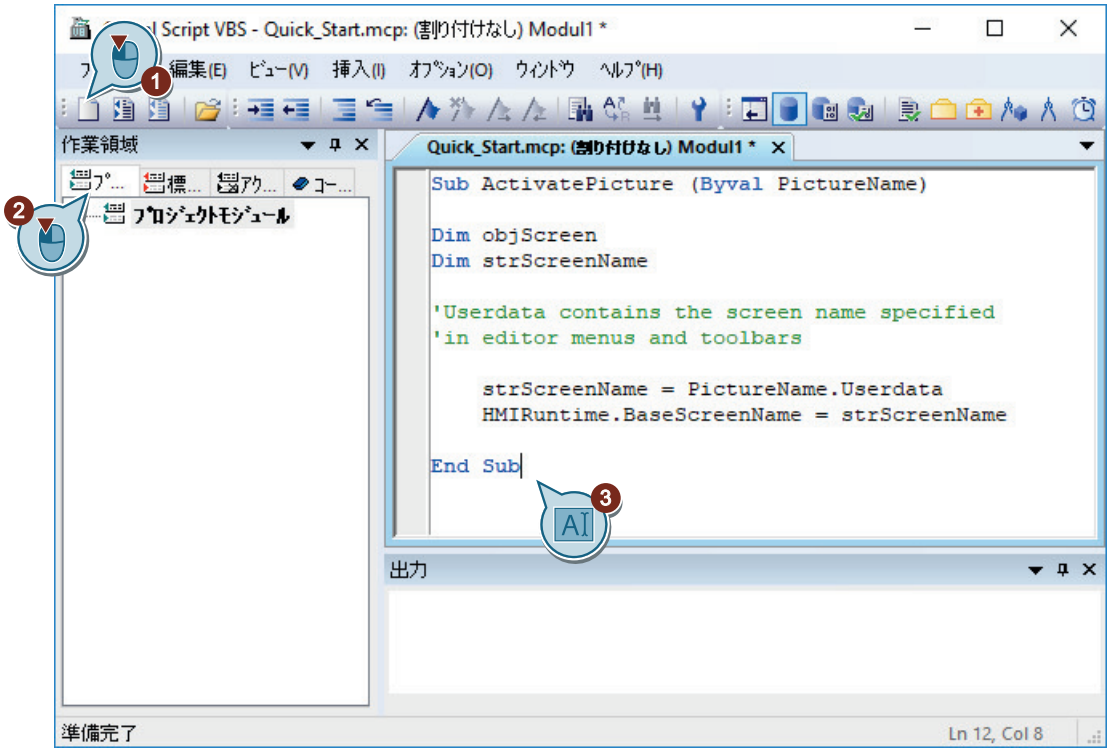
- ActivatePicture(ByVal PictureName):
  "PictureName"パラメータと一緒に転送される画像に対して、画像の変更を実行します。
- StopRuntime(ByVal Item):
  ランタイムを終了します。

手順

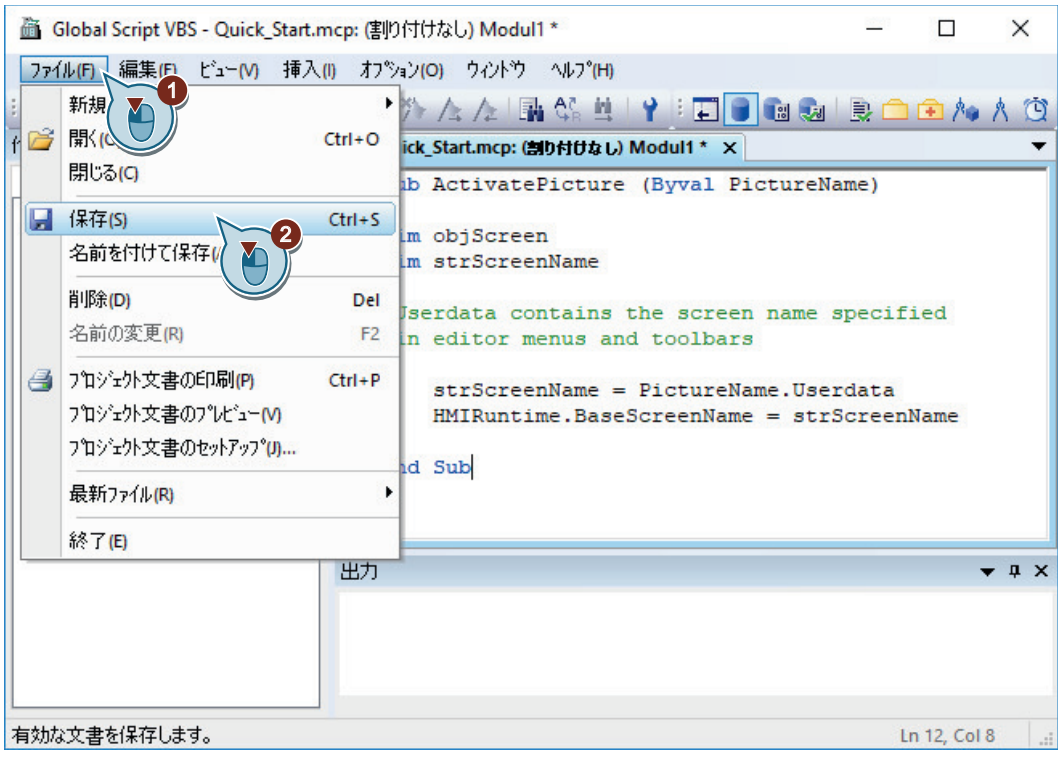
1. VBS エディタを開きます。



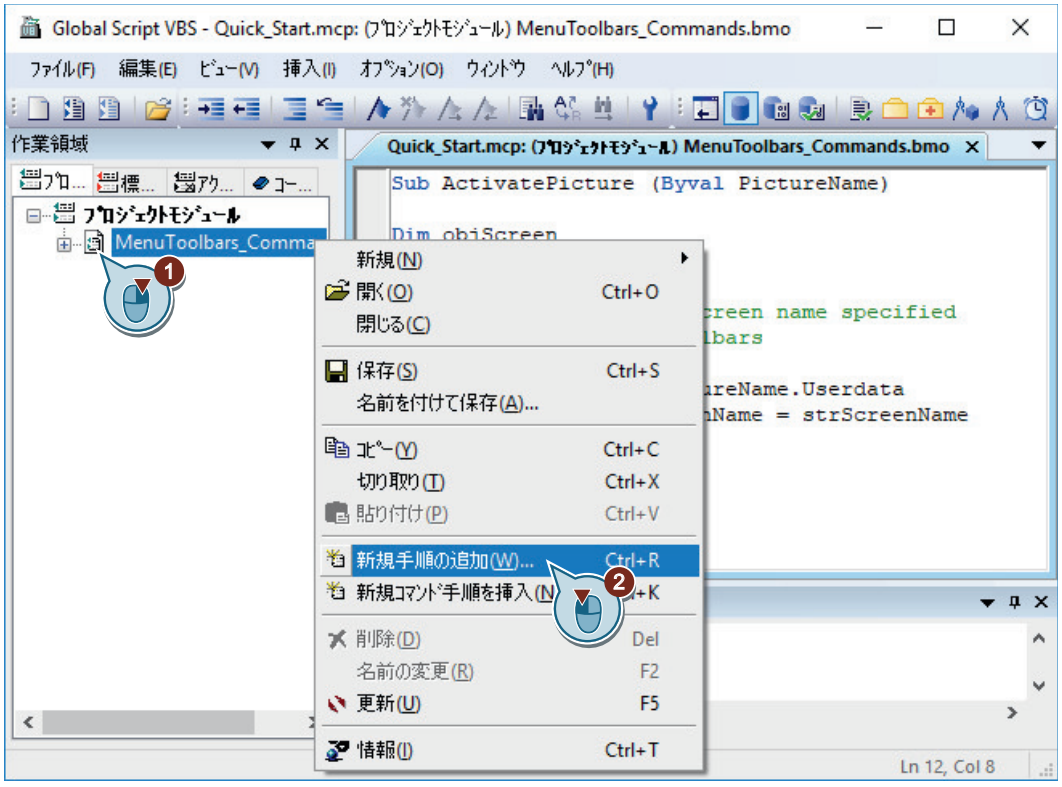
2. [プロジェクトモジュール]タブを選択し、以下のプロシージャコードを書き込みます。



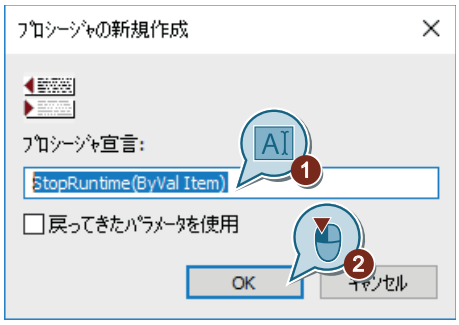
3. モジュールを"MenuToolbars_Commands.bmo"ファイルとして保存します。



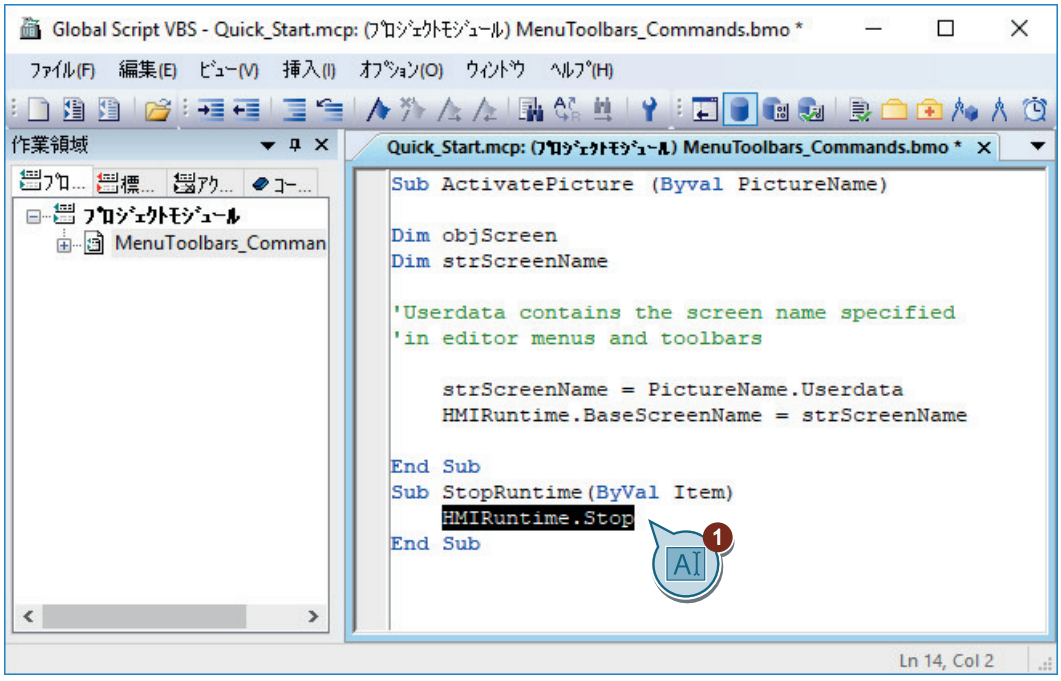
4. 新しいプロシージャを挿入します。



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。

6.6.3 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。

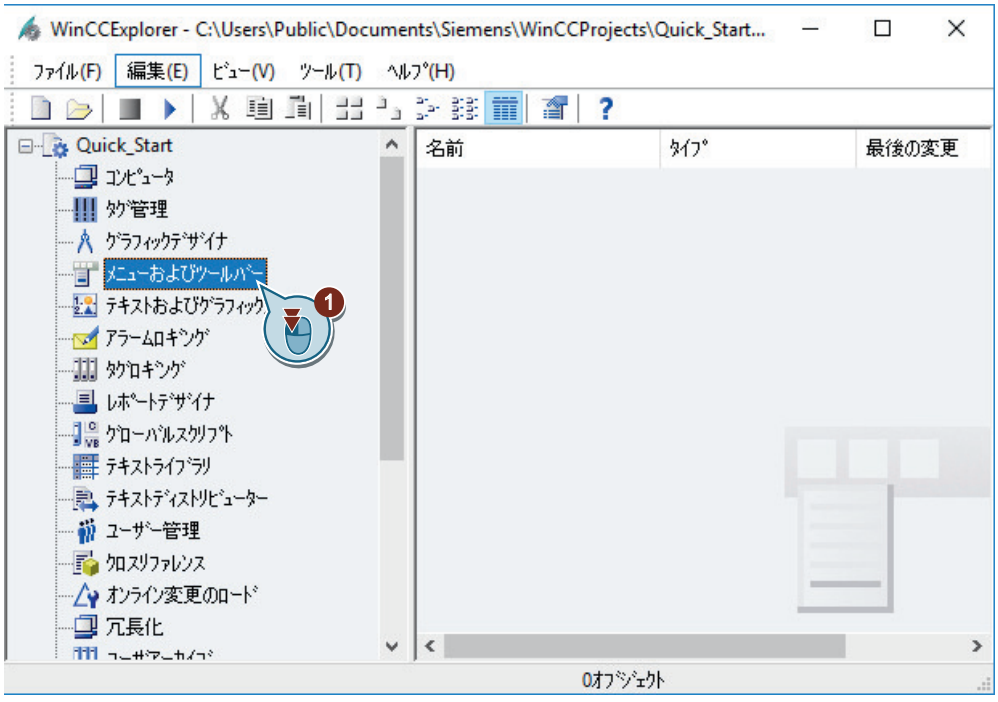
変更するプロセス画像の名前を、"ユーザーデータ"ボックスに入力します。

必要条件

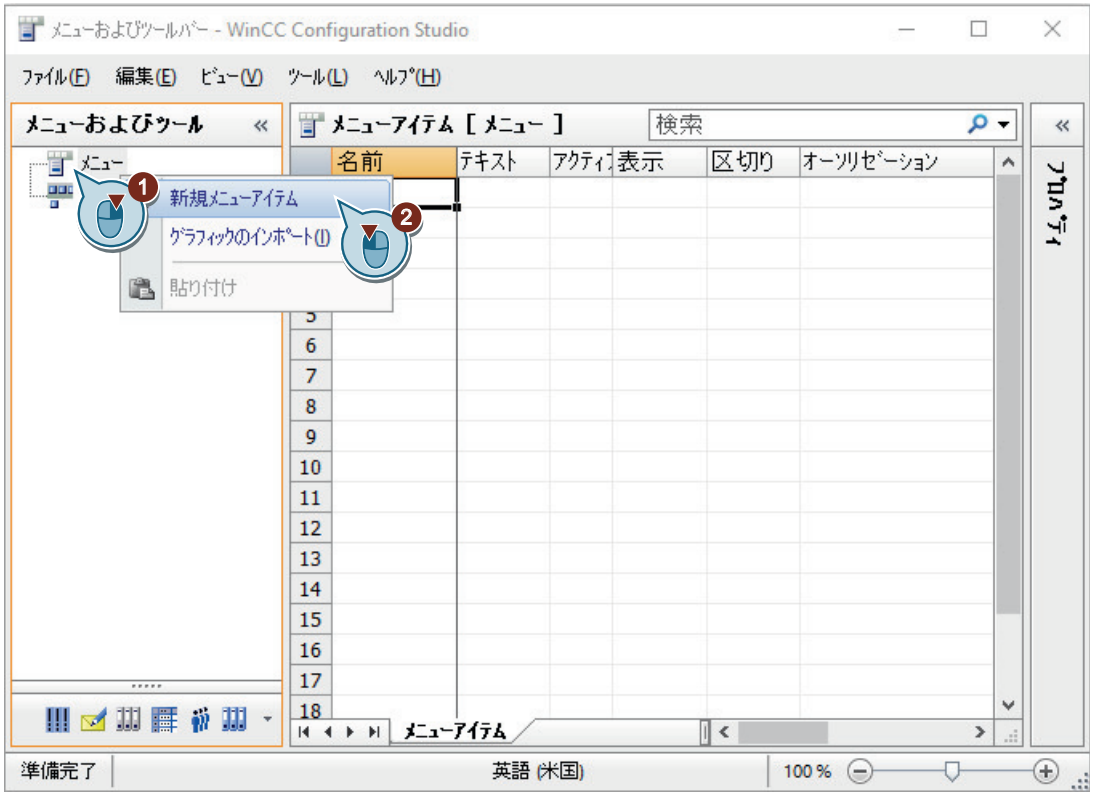
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。

手順

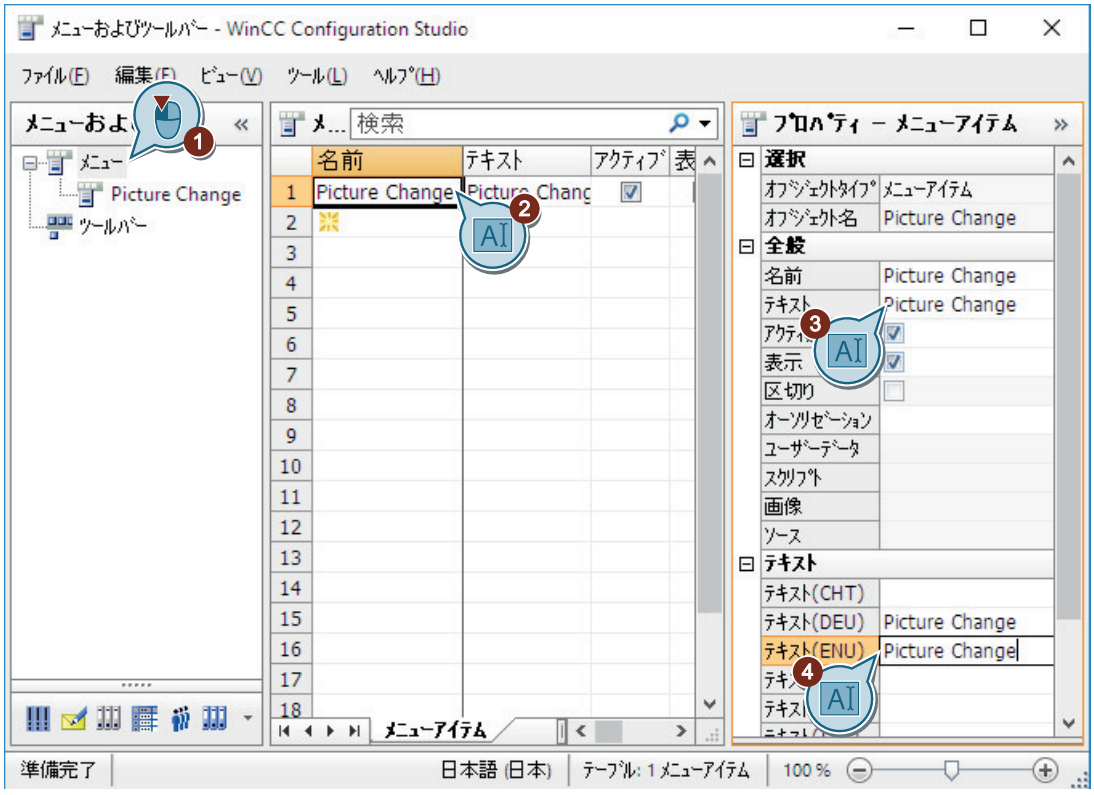
1. [メニューとツールバー]エディタを開きます。



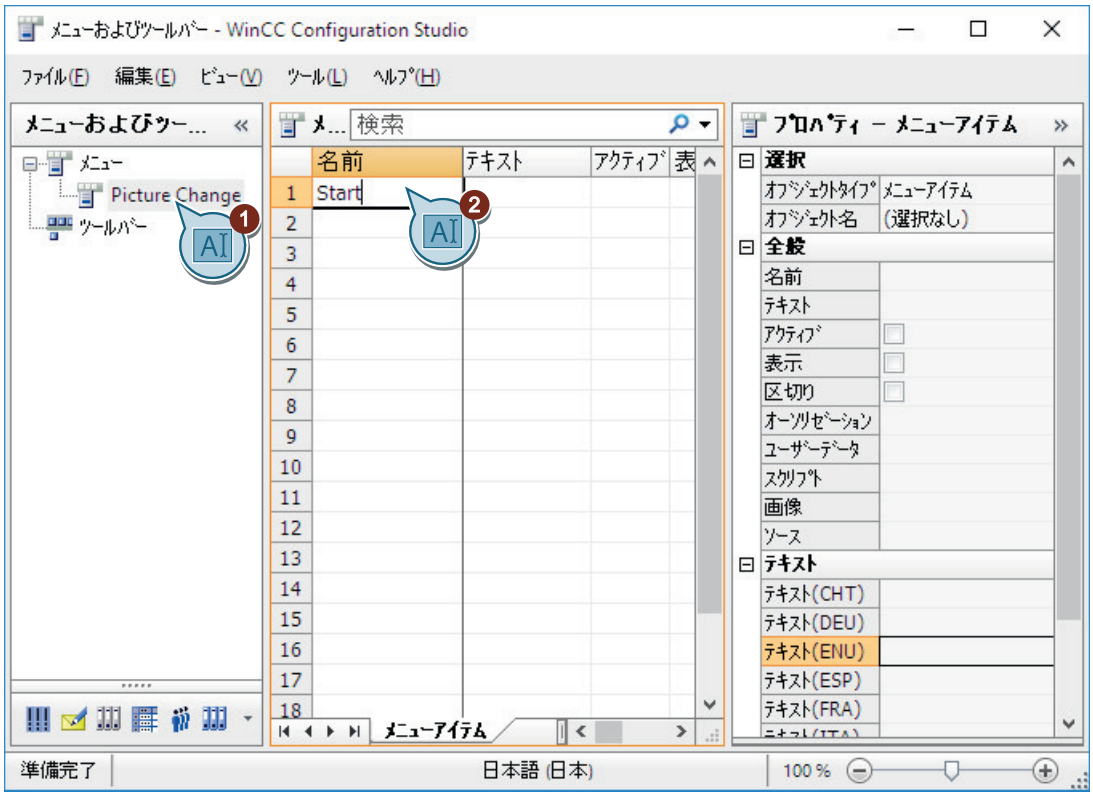
2. 新しいメニュー項目を作成します。



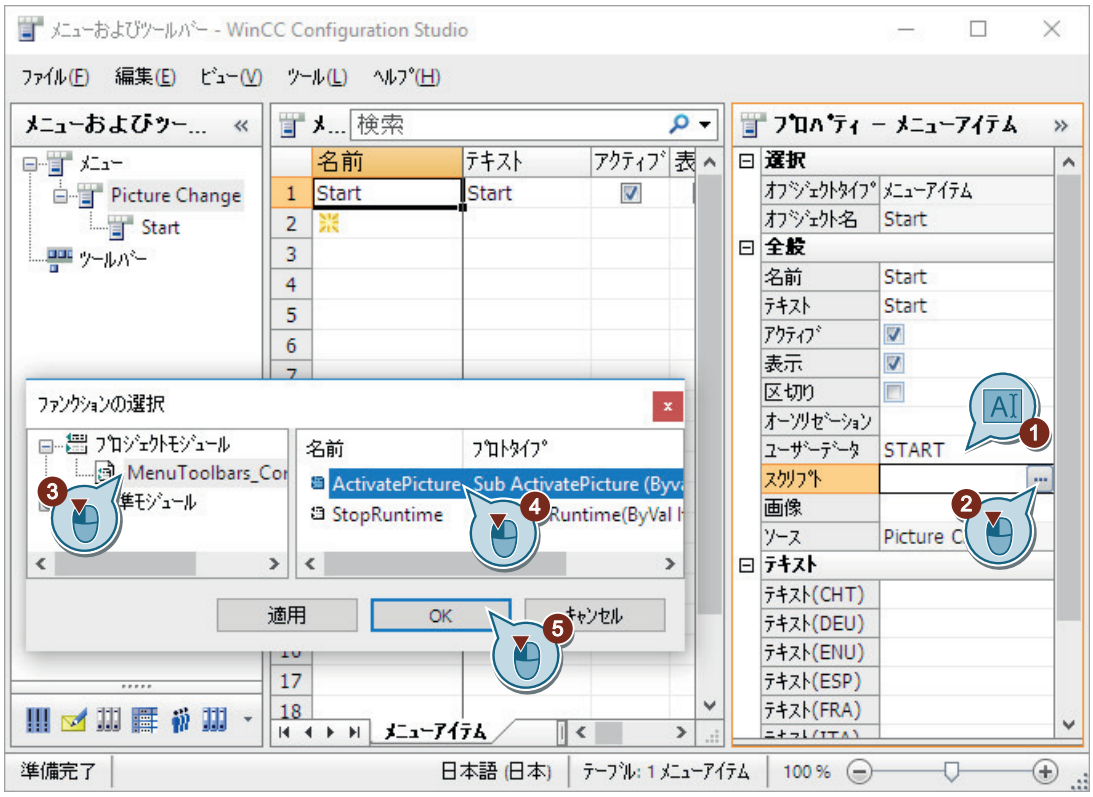
3. メニュー項目に"Picture Change"という名前を付けます。
   "プロパティ"ウィンドウでは、"テキスト"エリアにその他のランタイム言語の名前を入力します。



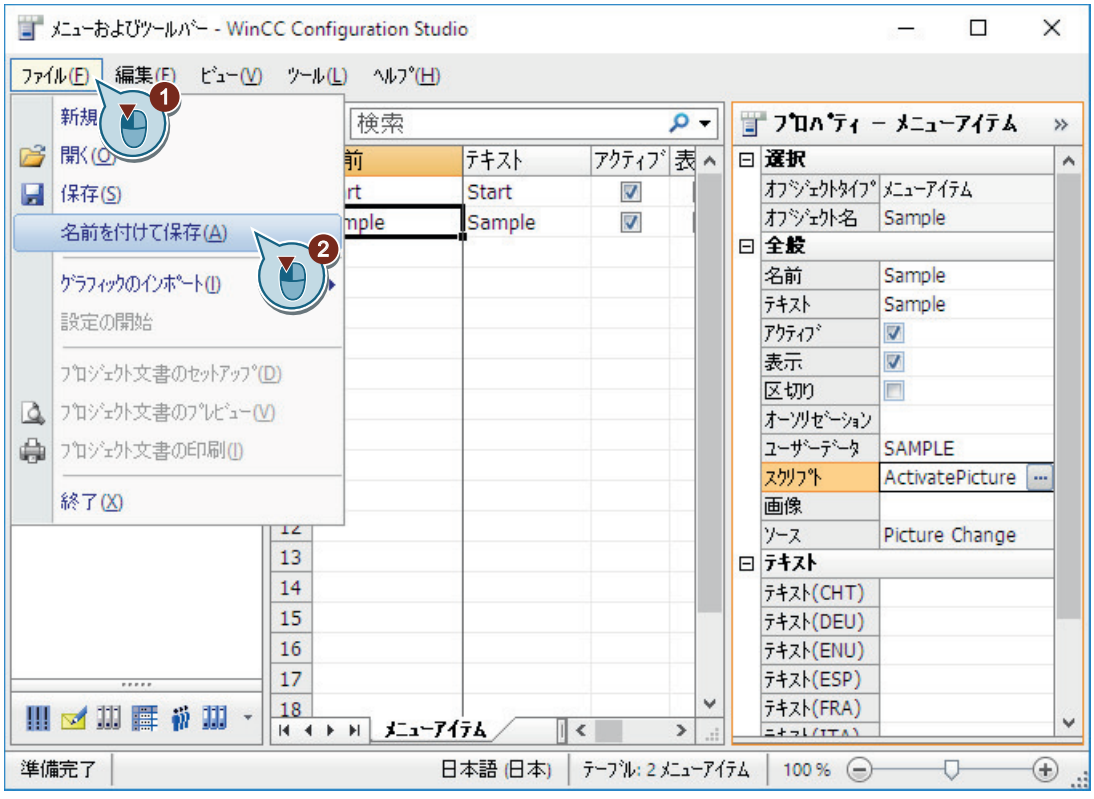
4. [スタート]メニューコマンドを作成します。



5. "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。



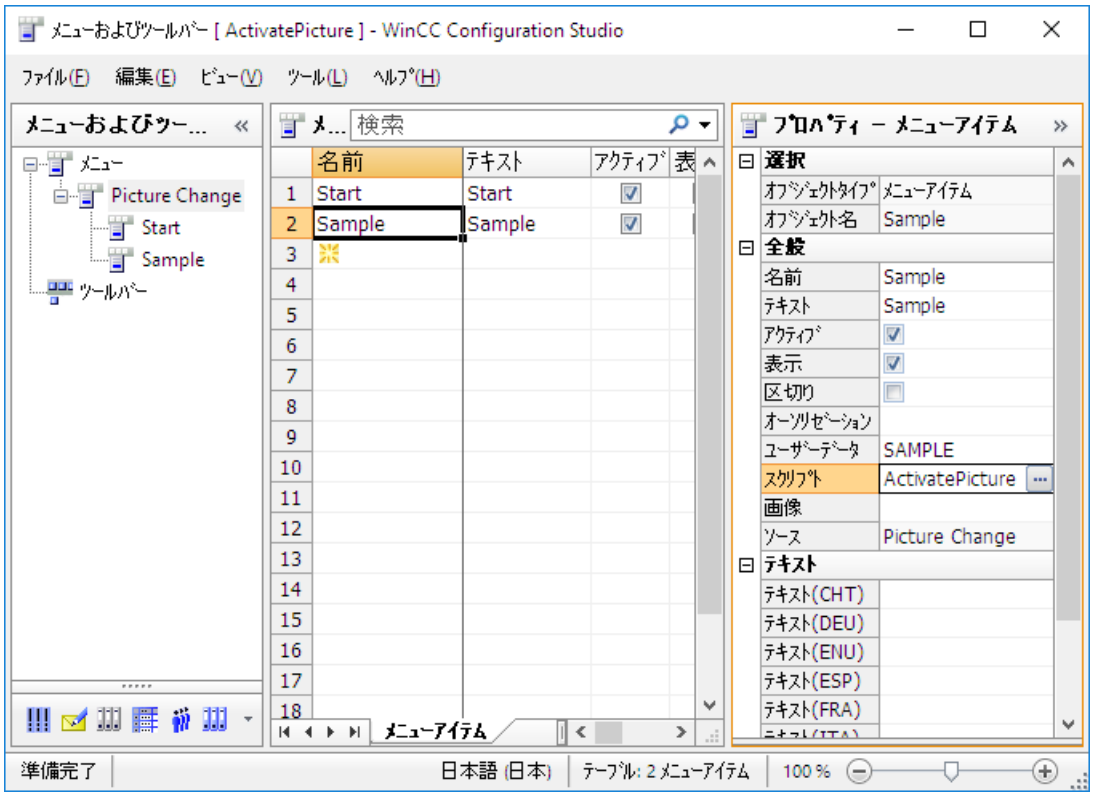
6. 同様に、"SAMPLE.pdl"画像に変更するために、[サンプル]メニューコマンドを作成します。
7. 設定ファイルに"ActivatePicture"という名前を付けて保存します。



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

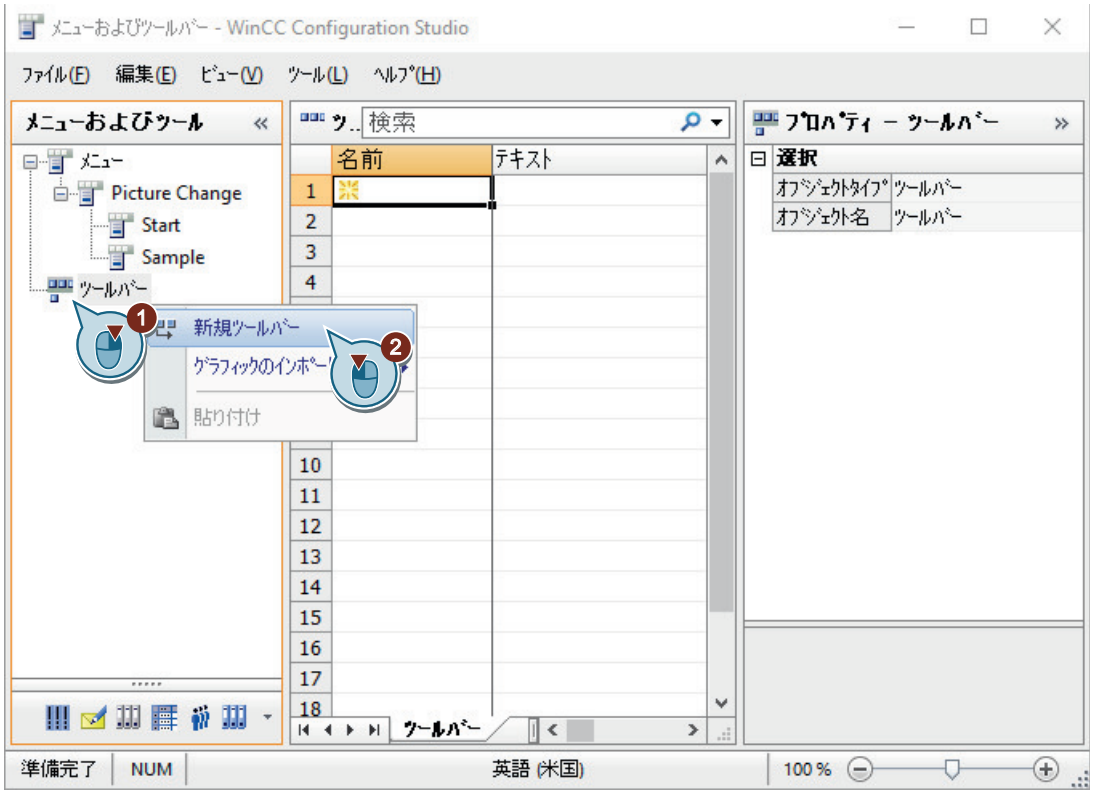
シンボルとして、BMP、PNG、JPG または GIF フォーマットのグラフィックを使用します。

必要条件

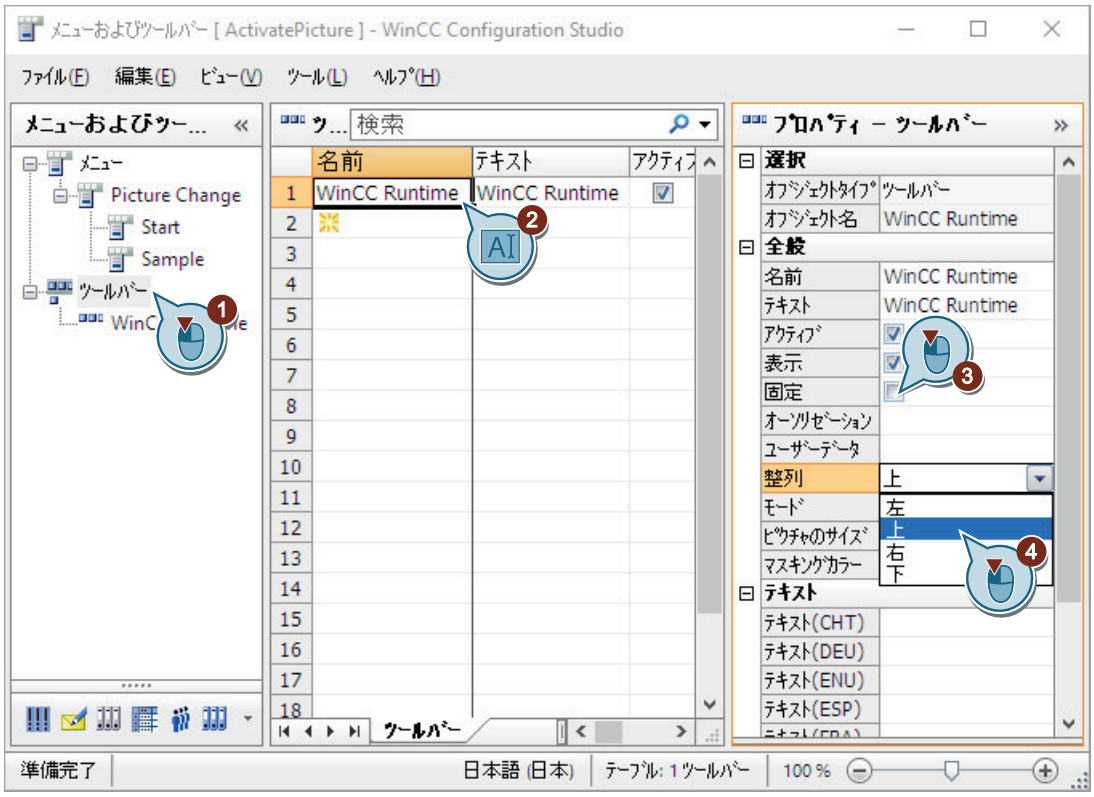
- [メニューとツールバー]エディタが開いていること。
- "StopRuntime(ByVal 項目)"プロシージャが作成されていること。
- グラフィックファイルが"ランタイムの終了"のシンボルとして WinCC プロジェクトフォルダにコピーされていること。例:
  - C:\Users\Public\Documents\Siemens\WinCCProjects\Quick_Start\GraMT\mtimages

手順

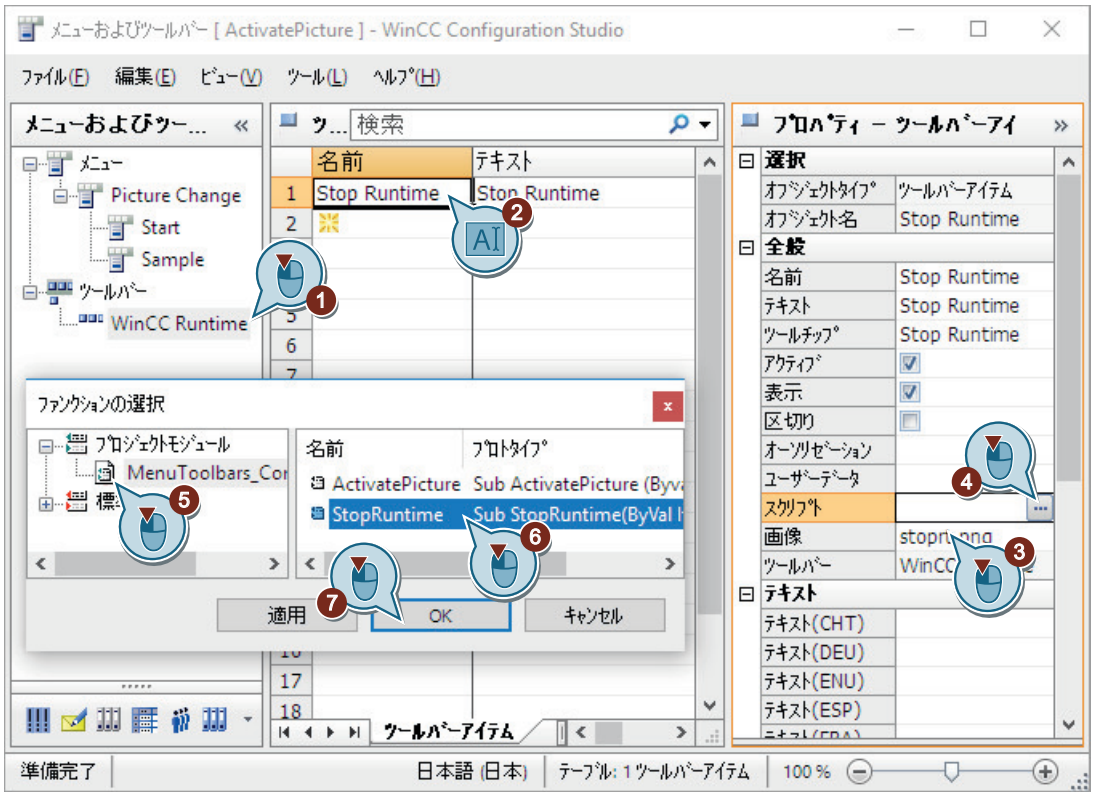
1. 新しい"WinCC Runtime"ツールバーを作成します。



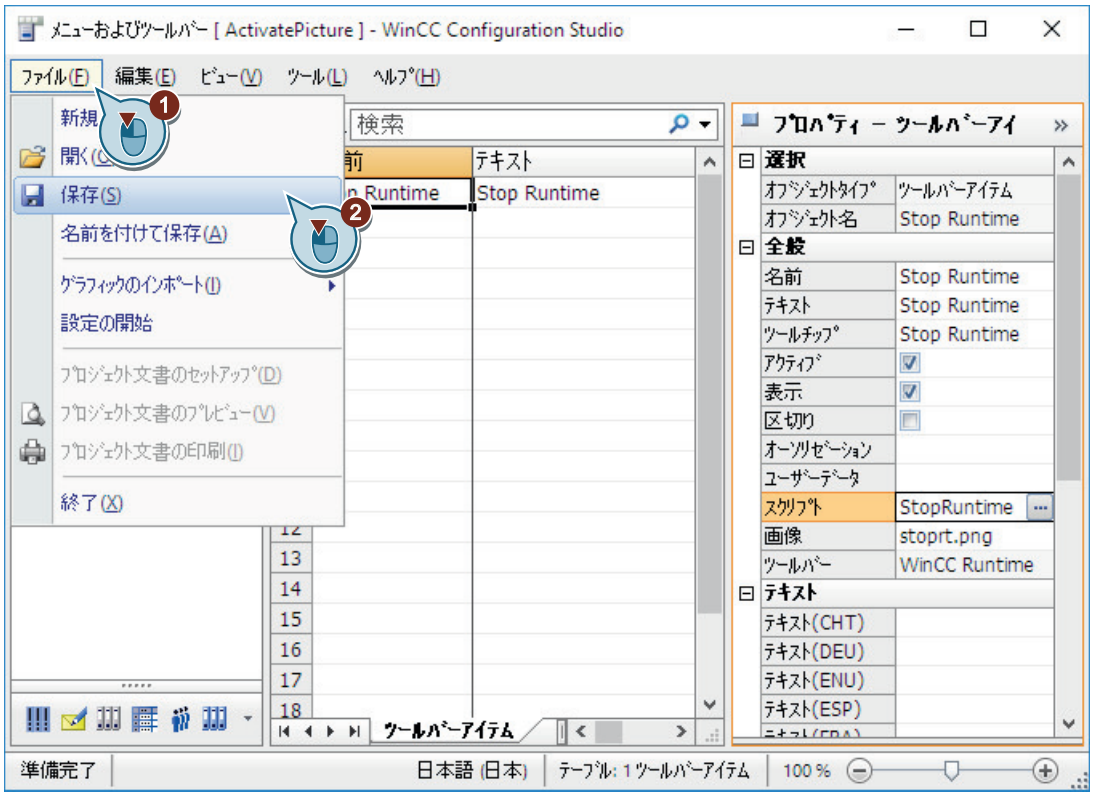
2. デフォルトで画像の上部端に表示されるようにツールバーを設定しますが、ユーザーはこの位置を変えることができます。



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。

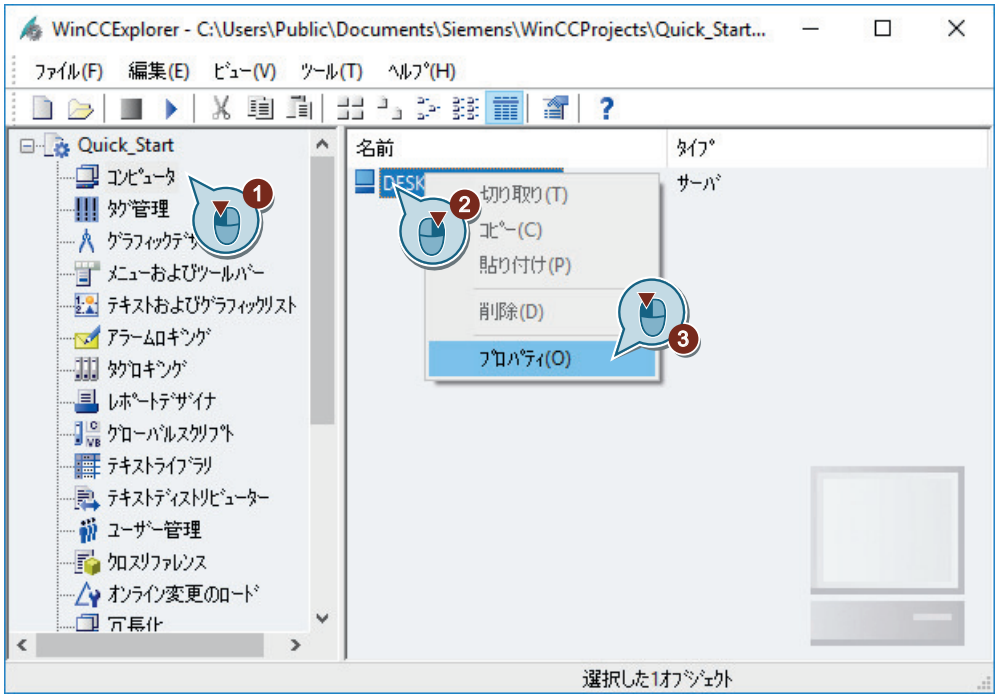
作成した設定ファイルをプロセス画像でユーザー定義メニューおよびツールバーを表示するために読み込みます。

必要条件

- "Quick_Start"プロジェクトが開いている。

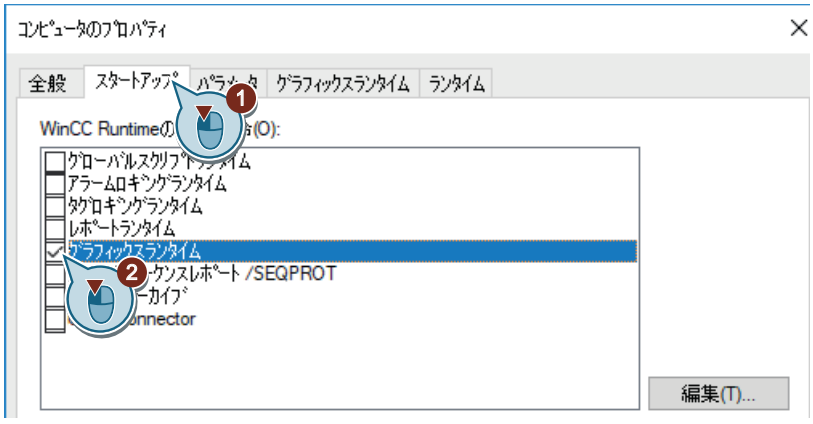
手順

1. [コンピュータのプロパティ]ダイアログを開きます。

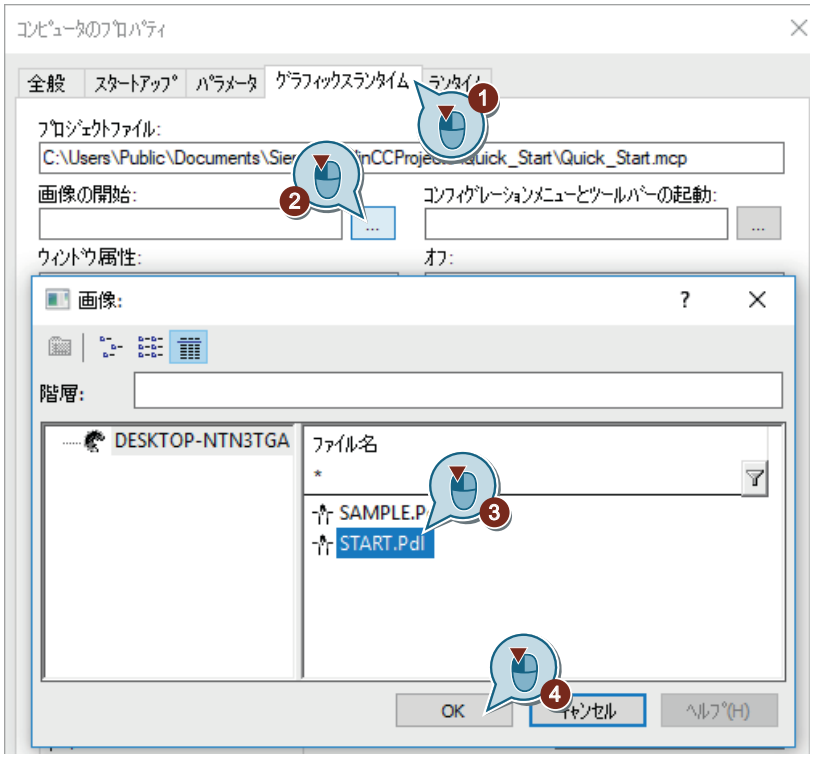


[コンピュータのプロパティ]ダイアログが開きます。

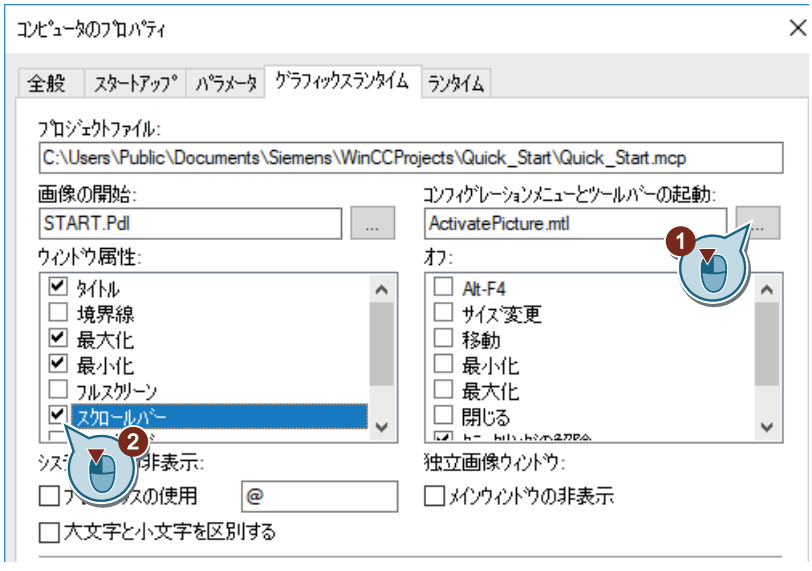
2. "スタートアップ"タブをクリックして、"グラフィックスランタイム"アプリケーションを有効にします。



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。.
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

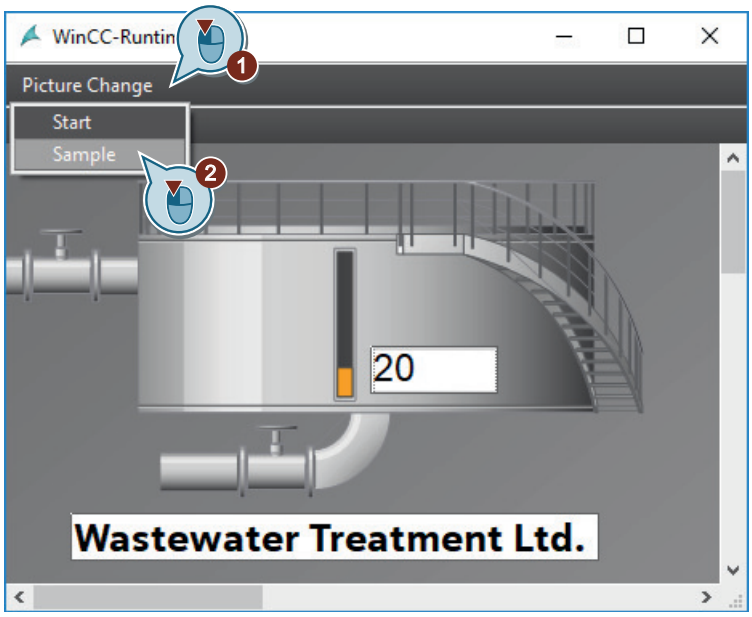
- "Quick_Start"プロジェクトが開いている。
- ランタイムプロパティが定義されている。

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。

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 87)」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 87)

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。

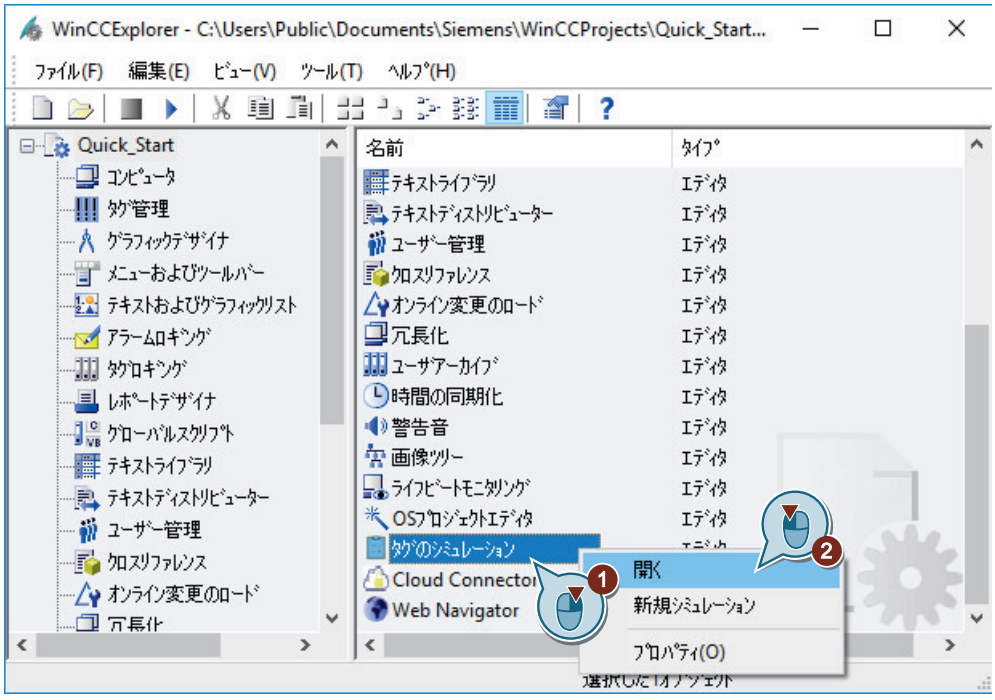
WinCC タグシミュレータは、内部タグ"Tank_Level"に異なる値を割り付けます。バイオコンバータのグラフィック描写に内部タグを接続すると、バイオコンバータの充填レベルインジケータはタグ値に従って変化します。

必要条件

- ランタイムプロパティが定義されている。

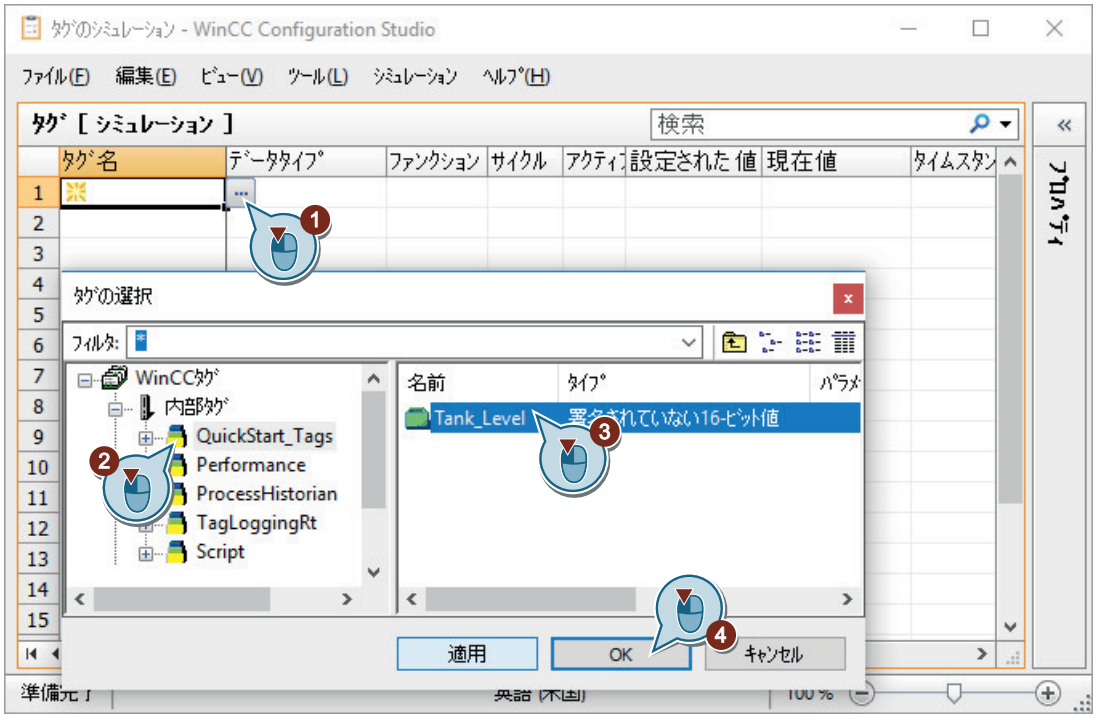
手順

1. [タグシミュレーション]エディタを起動します。

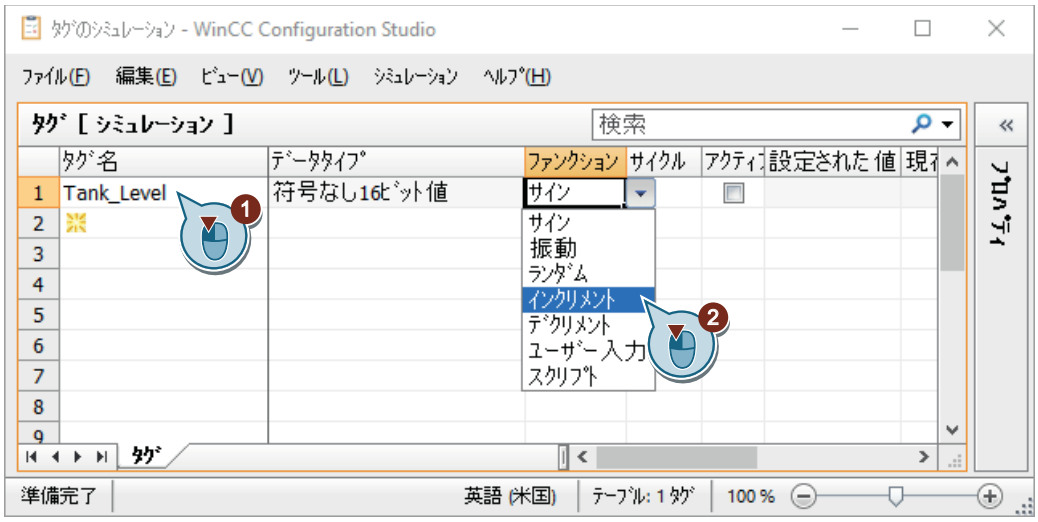


[タグシミュレーション]エディタが開きます。

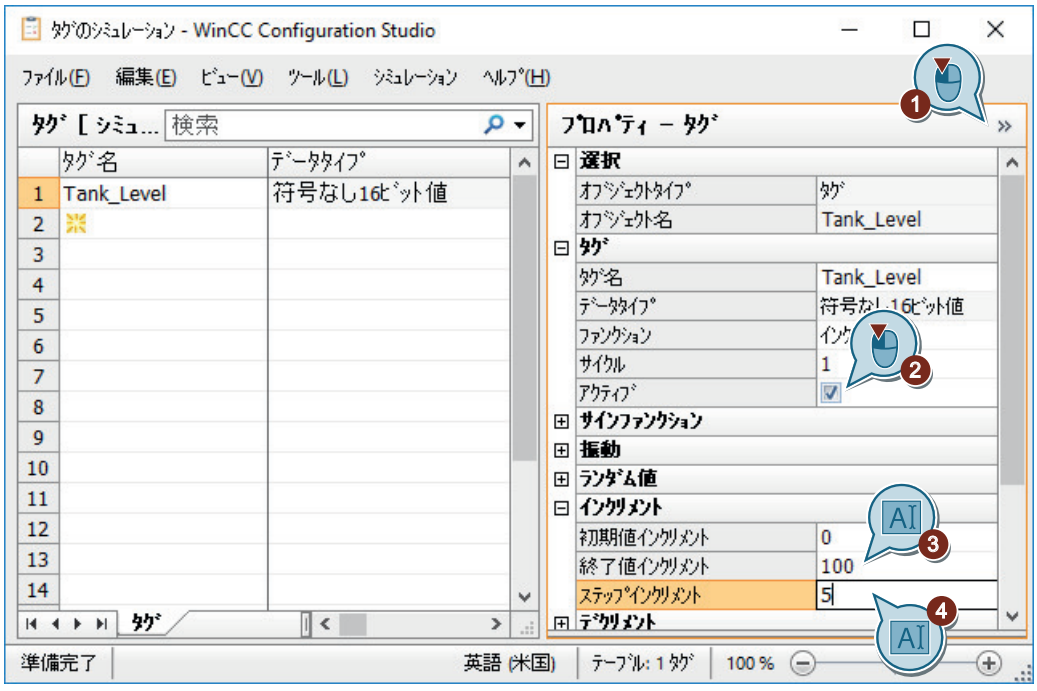
2. タグ選択ダイアログを開き、内部タグ[Tank_Level]をタググループ[QuickStart_Tags]から選択します。



3. シミュレーションタイプを選択します。

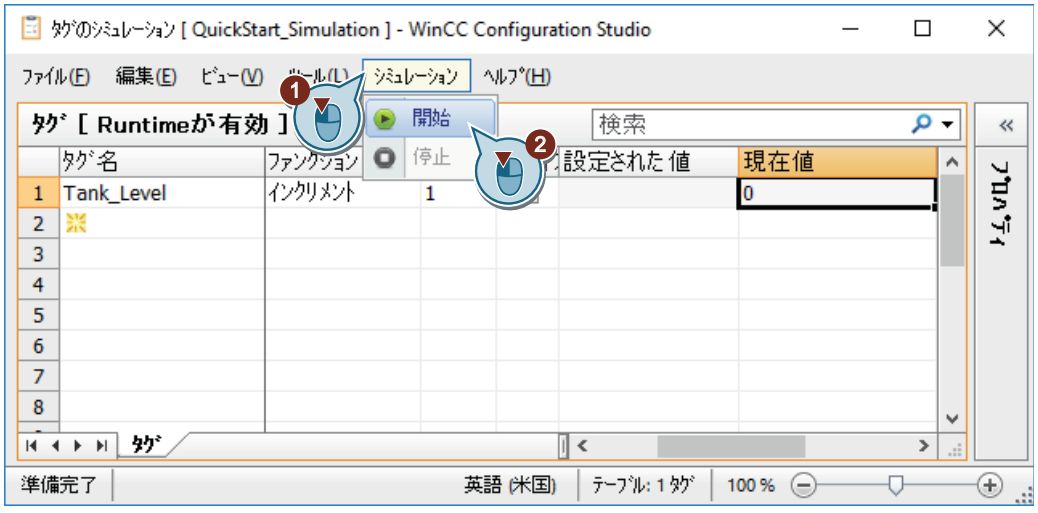


4. [プロパティ]ウィンドウを開き、シミュレーションタイプのプロパティを定義します。



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

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



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、頻繁に必要になる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。

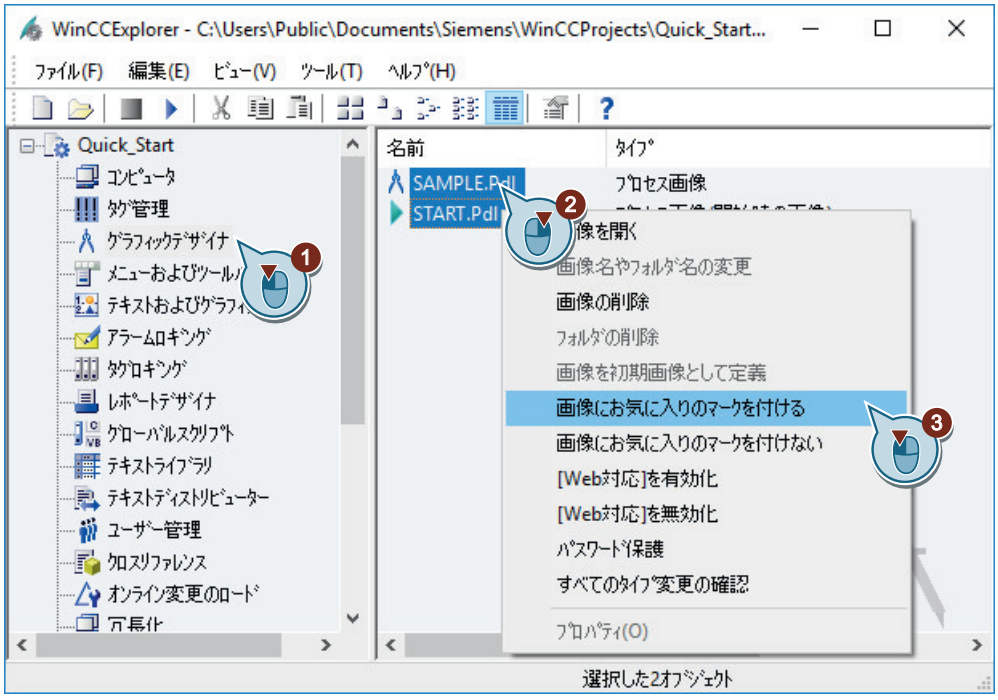
例では、2 つの画像を取り替える手順を示しています。

必要条件

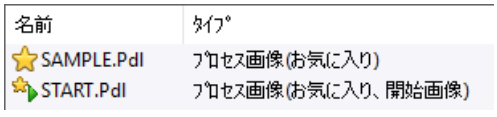
- "START.pdl"と"SAMPLE.pdl"プロセス画像が作成されている。
- "START.pdl"プロセス画像が開始画像として定義されている。

手順

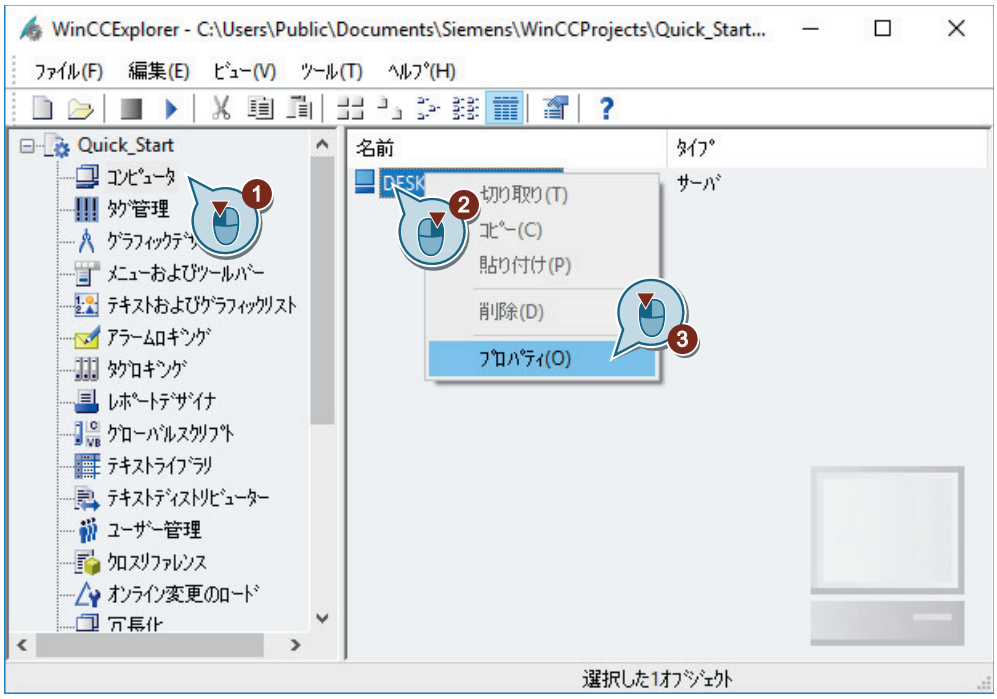
1. WinCC エクスプローラで、"START.pdl"と"SAMPLE.pdl"プロセス画像を選択します。
   両方の画像を選択するには、クリックするときに<Ctrl>ボタンを押します。
2. プロセス画像をお気に入りとして選択します。



"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。

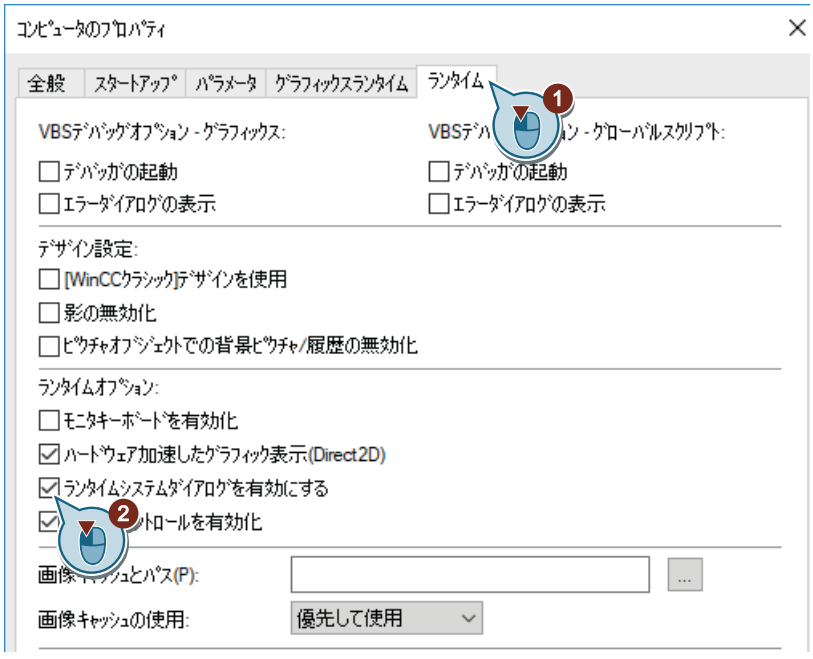


3. [コンピュータのプロパティ]ダイアログを開きます。



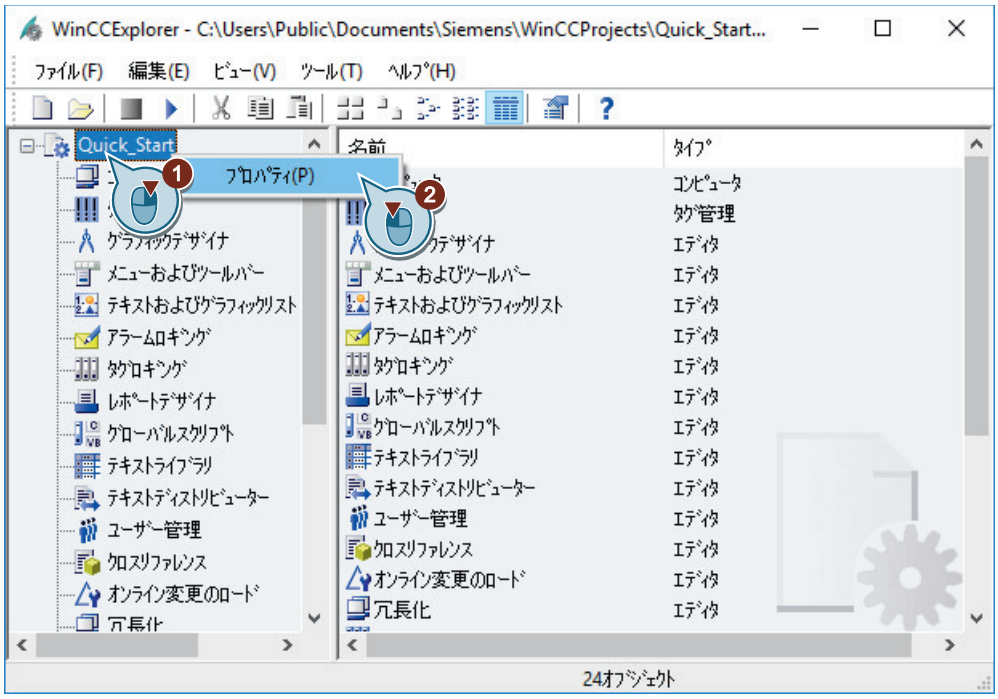
[コンピュータのプロパティ]ダイアログが開きます。

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。



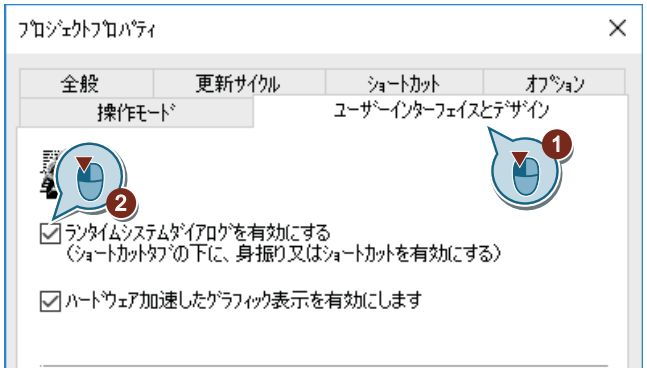
5. [OK]をクリックしてダイアログボックスを閉じます。

6. [プロジェクトプロパティ]ダイアログを開きます。

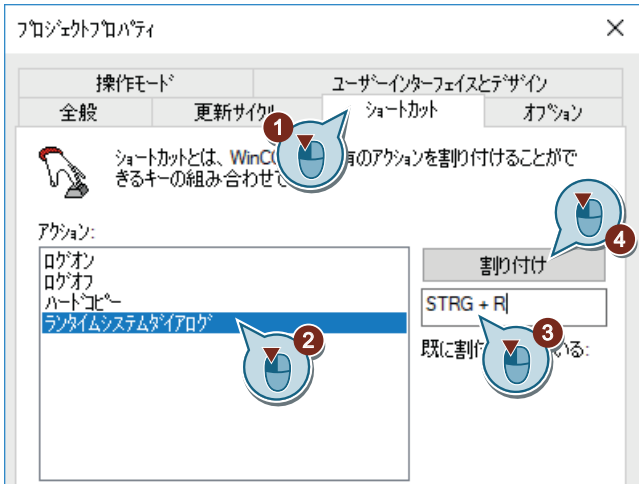


[プロジェクトプロパティ]ダイアログが開きます。

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。



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。

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

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ので、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]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

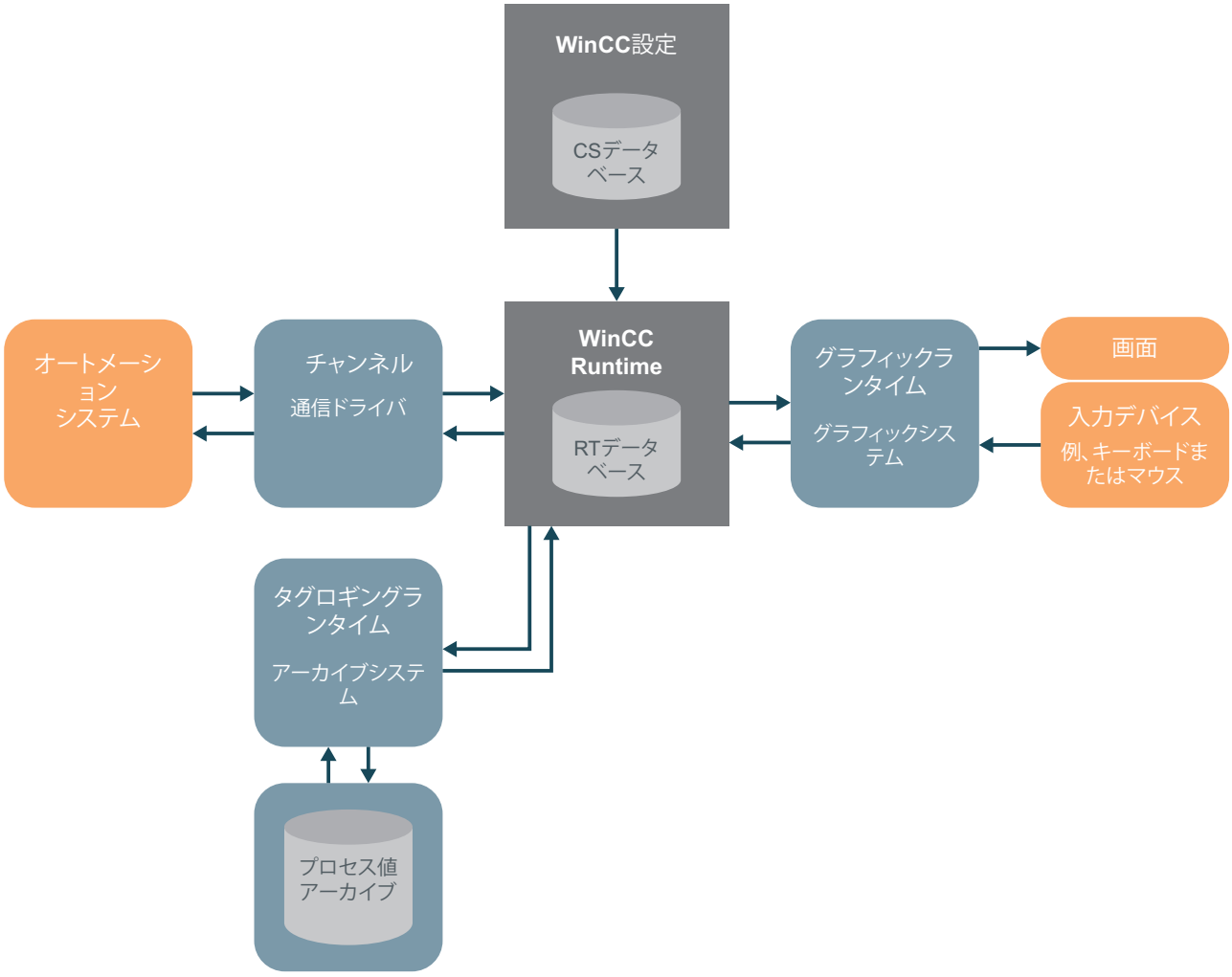
アーカイブサイクルとイベントを使用すると、アーカイブ時間をコントロールできます。

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。

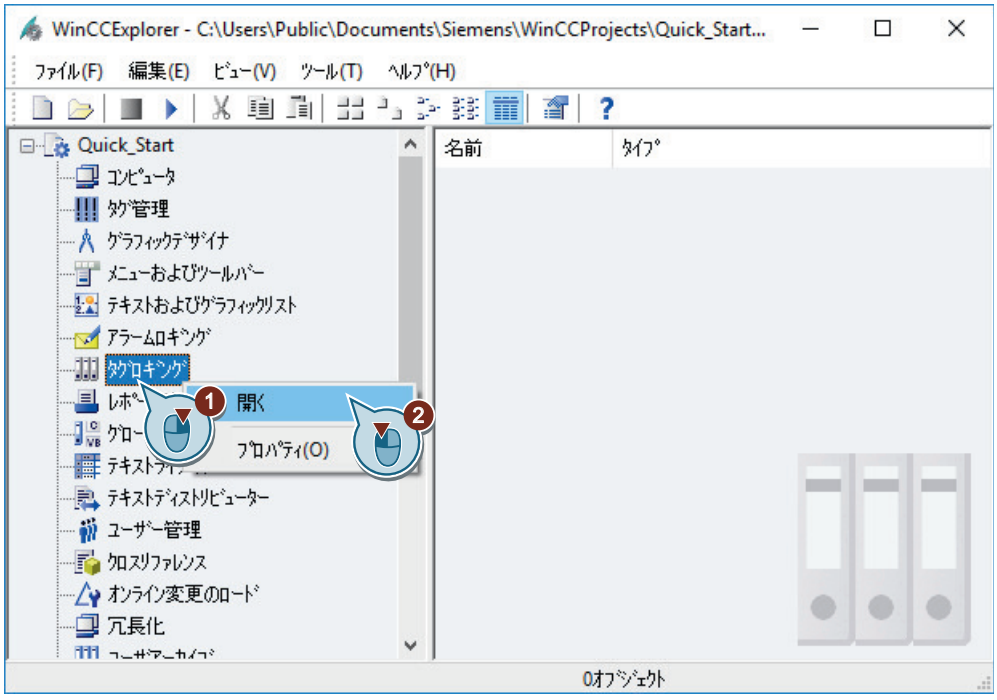
"タグロギング"エディタで、取得サイクルやアーカイブサイクルの時間と共に、プロセス値アーカイブを設定します。

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。



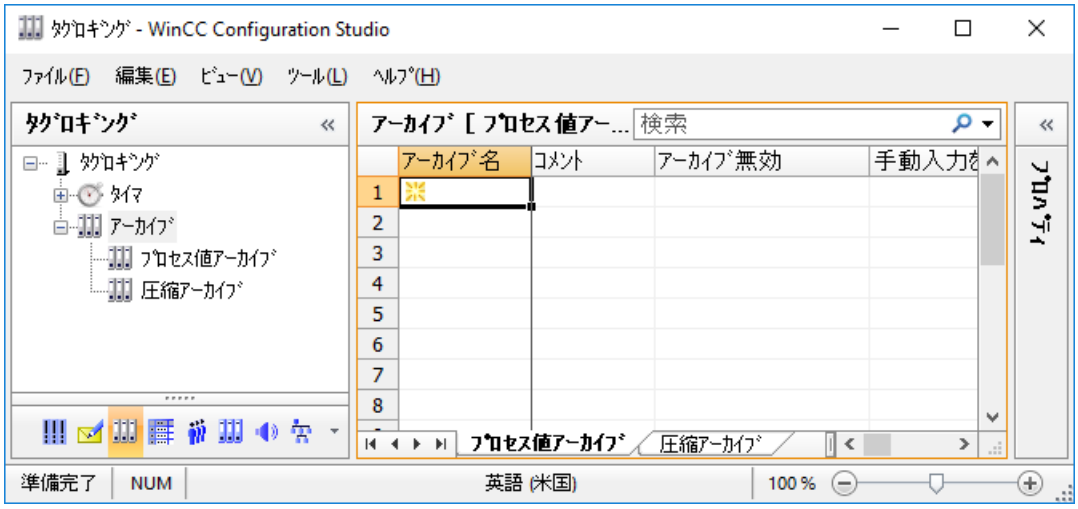
"タグロギング"エディタが開きます。

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、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。

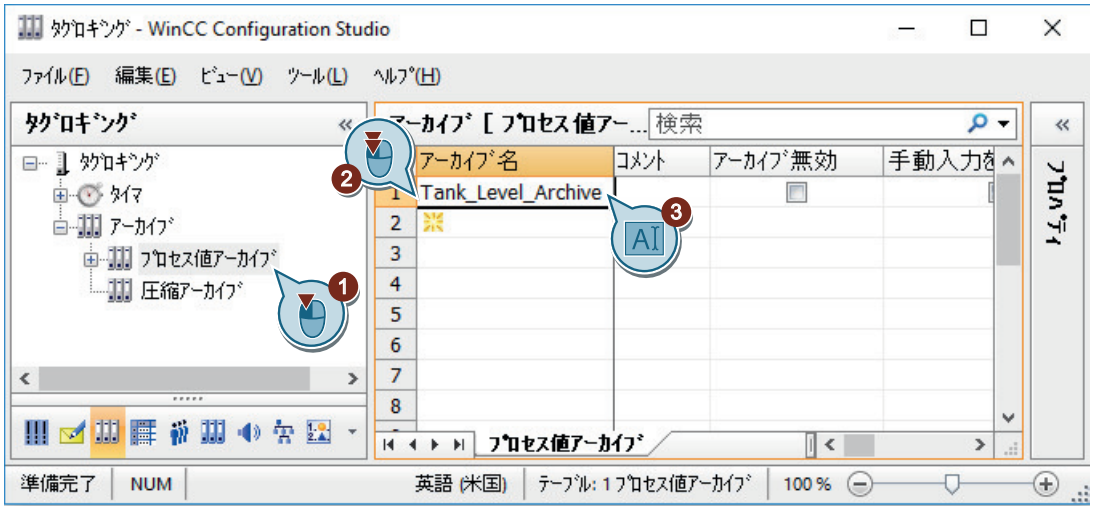
タグを定義すると、[タグロギング]はプロセス値アーカイブにアーカイブタグを作成します。アーカイブする値は、アーカイブタグに保存されます。

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。

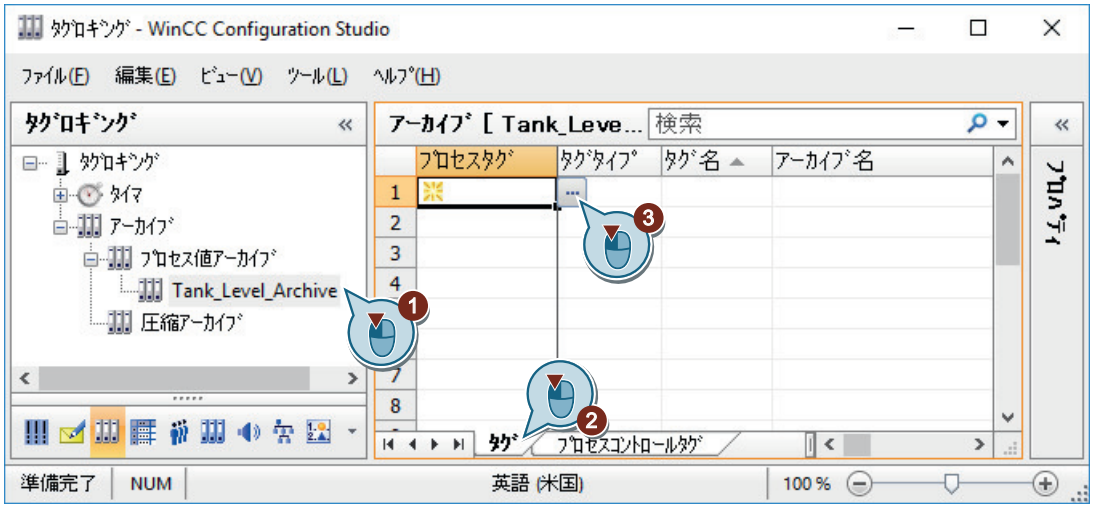
手順

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。

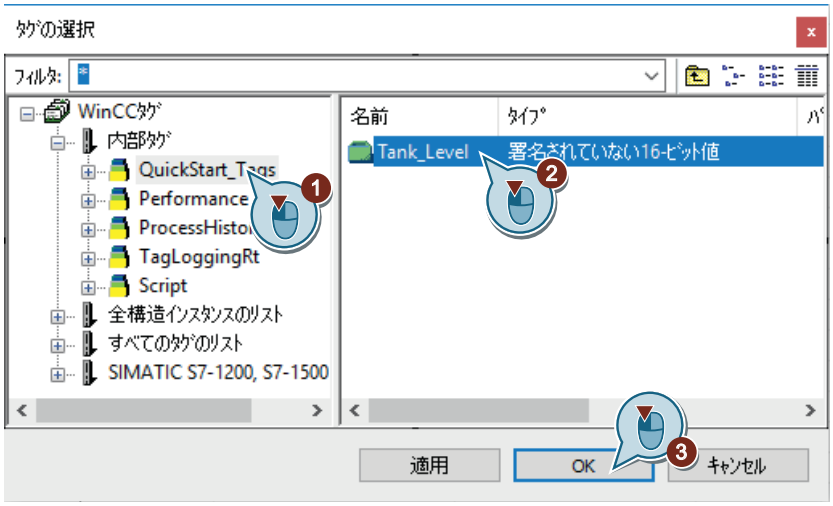


3. ナビゲーションエリアで、アーカイブのフォルダを選択します。
テーブルエリアで[タグ]タブを選択します。

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。

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。

ﾀｸﾞﾛｷﾞﾝｸﾞ - WinCC Configuration Studio
ﾌｧｲﾙ(F) 編集(E) ﾋﾞｭｰ(V) ﾂｰﾙ(L) ﾍﾙﾌﾟ(H)
ﾀｸﾞﾛｷﾞﾝｸﾞ
ﾀｸﾞﾛｷﾞﾝｸﾞ
ﾀｲﾏ
ｱｰｶｲﾌﾞ
ﾌﾟﾛｾｽ値ｱｰｶｲﾌﾞ
Tank_Level_Archive
圧縮ｱｰｶｲﾌﾞ
ｱｰｶｲﾌﾞ [Tank_Leve...
検索
ﾌﾟﾛｾｽﾀｸﾞ
ﾀｸﾞﾀｲﾌﾟ
ﾀｸﾞ名
ｱｰｶｲﾌﾞ名
1 Tank_Level アナログ Tank_Level Tank_Level_Archive
ﾀｸﾞ
ﾌﾟﾛｾｽｺﾝﾄﾛｰﾙﾀｸﾞ
ﾌﾟﾛﾊﾟﾃｨ
準備完了
NUM
英語 (米国)
ﾃｰﾌﾞﾙ: 1 ﾀｸﾞ
100 %

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。

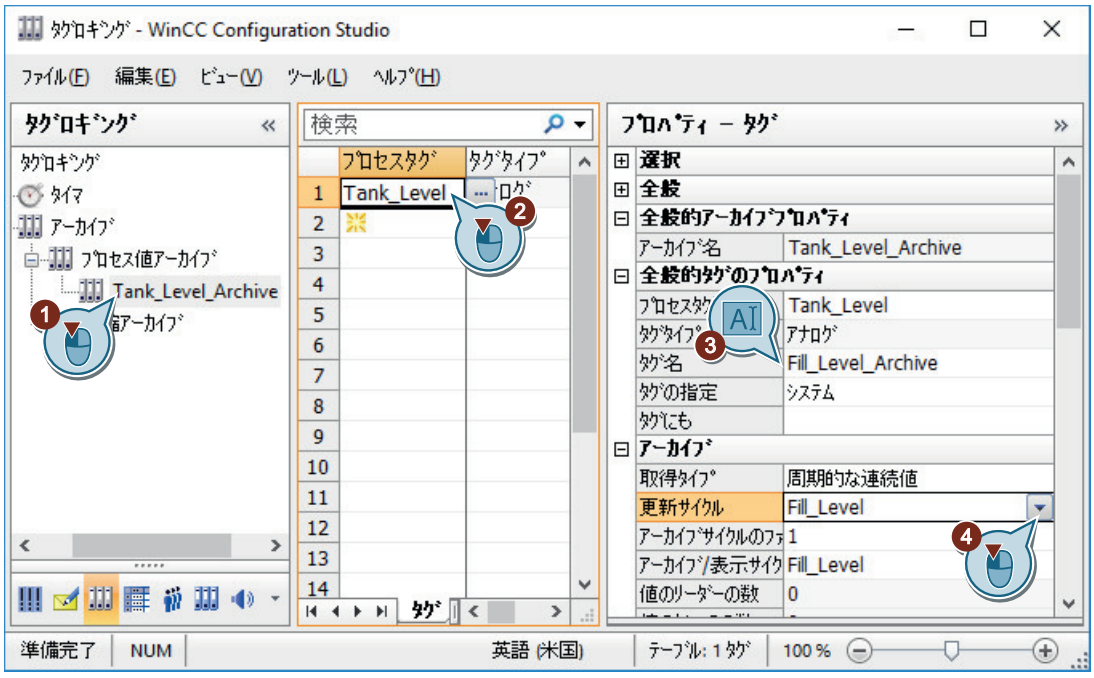
必要条件

- [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。

手順

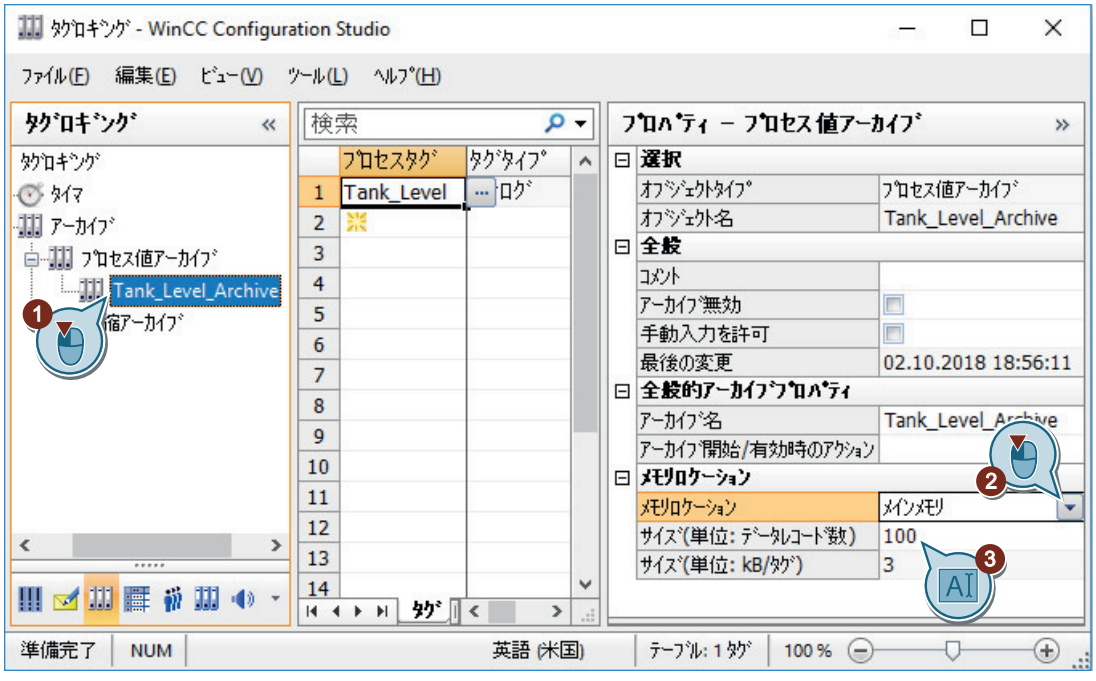
1. 設定済みのアーカイブタグでラインにあるテーブルエリアをクリックし、«ボタンを使用して[プロパティ]ウィンドウを開きます。
   必要な場合は、プロパティエリアを拡大します。
2. アーカイブタグ名として"Fill_Level_Archive"を入力します。

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"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]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"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。

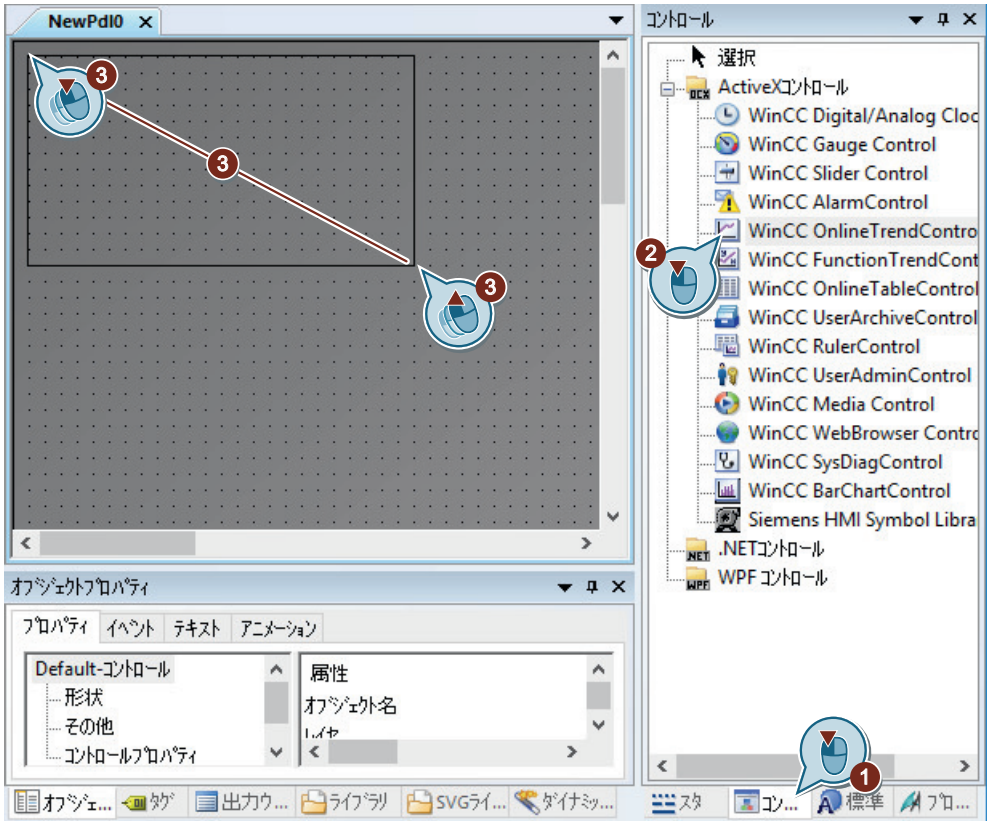
これで、ランタイムでアーカイブタグに保存された値がトレンドとして出力されます。

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。

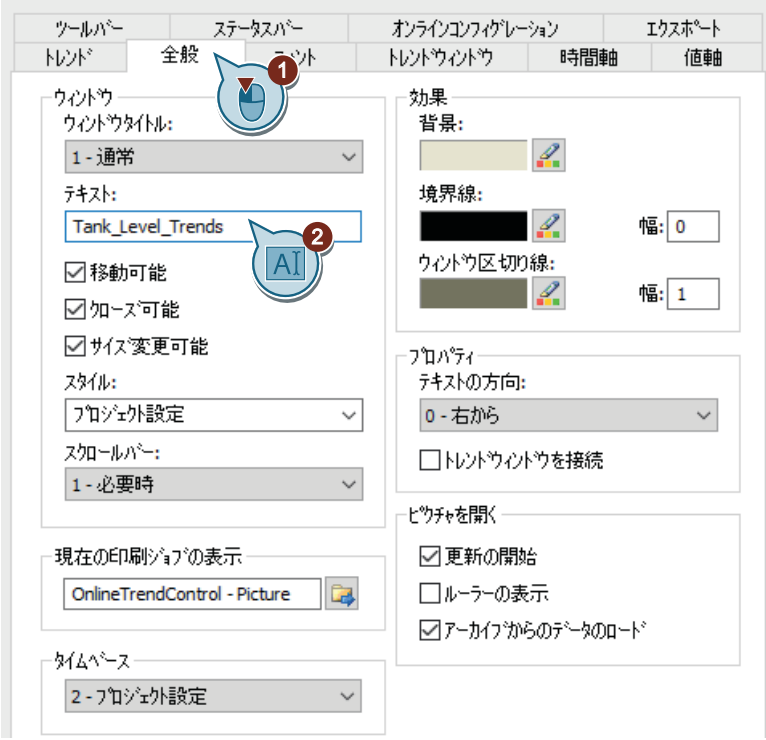
手順

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]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。



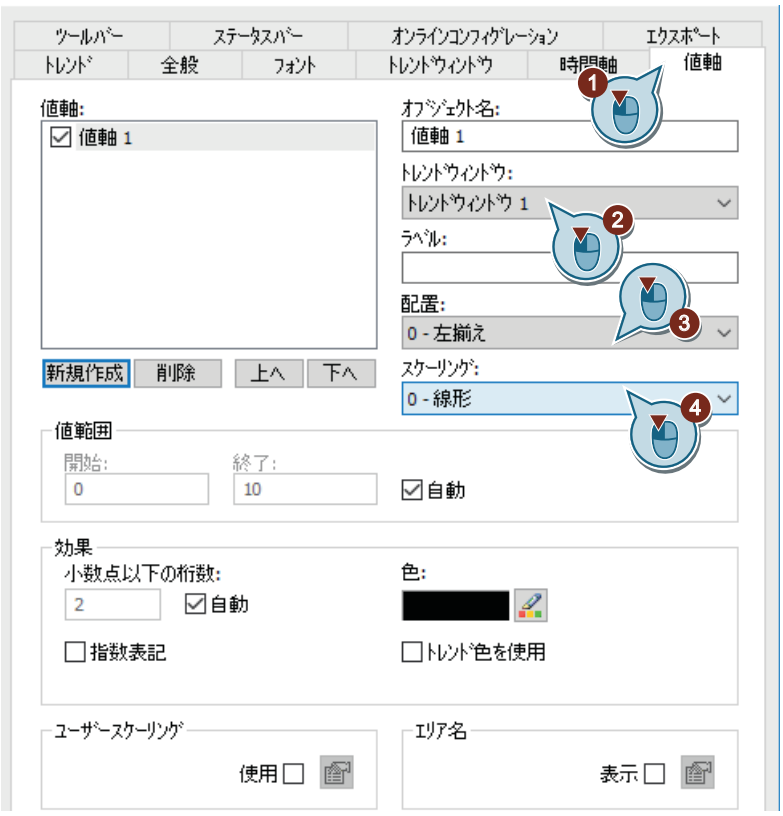
"グラフィックデザイナ"エディタの作業エリアに、[トレンド]ウィンドウが表示されます。[WinCC OnlineTrendControl のプロパティ]ダイアログが開きます。

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。

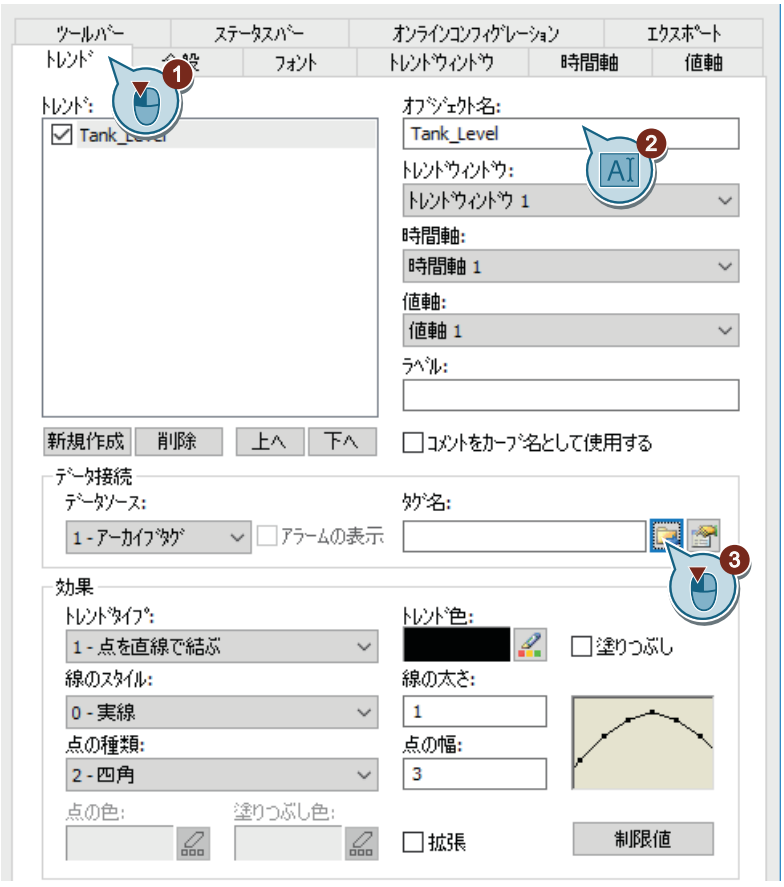


4. 時間軸の設定を定義します。

5. 値軸の設定を定義します。

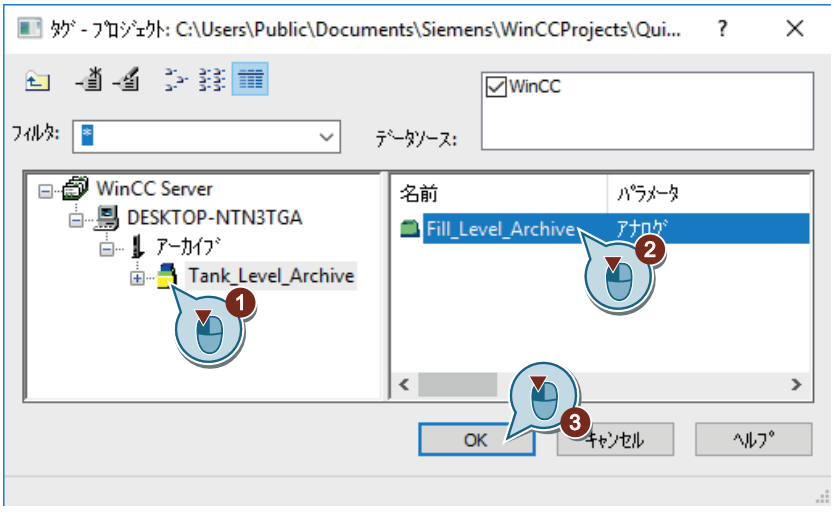


6. トレンド名前"Tank_Level"を入力し、をクリックします。

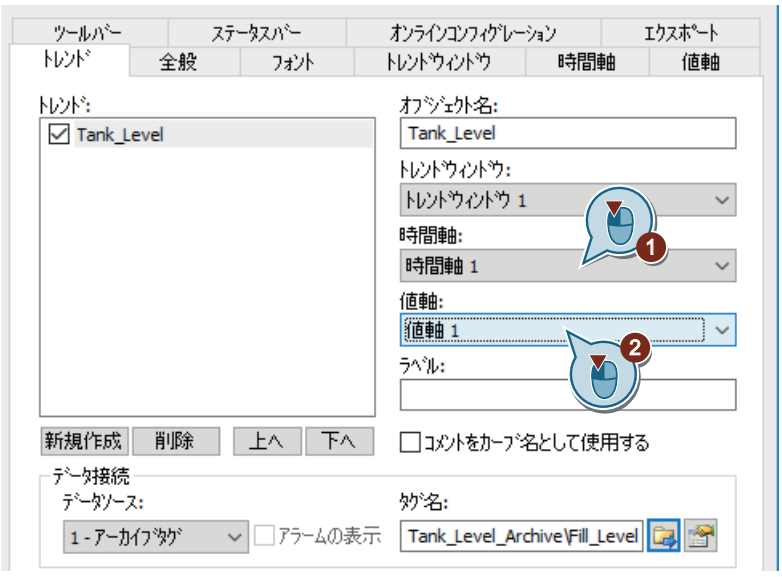


タグの選択用ダイアログが開きます。

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

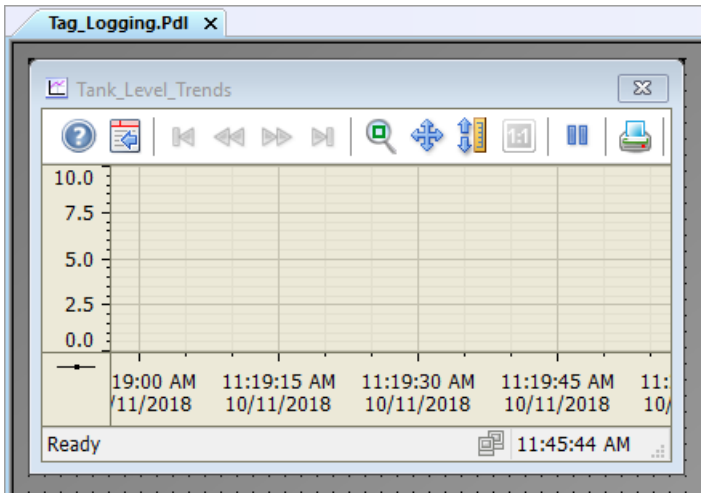
1. "Tag_Logging.pdl"という名前の新しいプロセス画像を作成し、[グラフィックデザイナ]エディタで開きます。
2. [タグロギング]エディタのテーブルペインにあるアーカイブタグ"Fill_Level_Archive"の付いた行を選択します。
   マウスを選択長方形の端に動かします。
   カーソルが十字型ポインタ[+]に変わります。
3. タグの選択内容をグラフィックデザイナの画像の空白エリアにドラッグしている間、マウスの左ボタンを押したままにします。
4. 画像でマウスボタンから指を放します。
   WinCC OnlineTrendControl が作成されます。
   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"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"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。

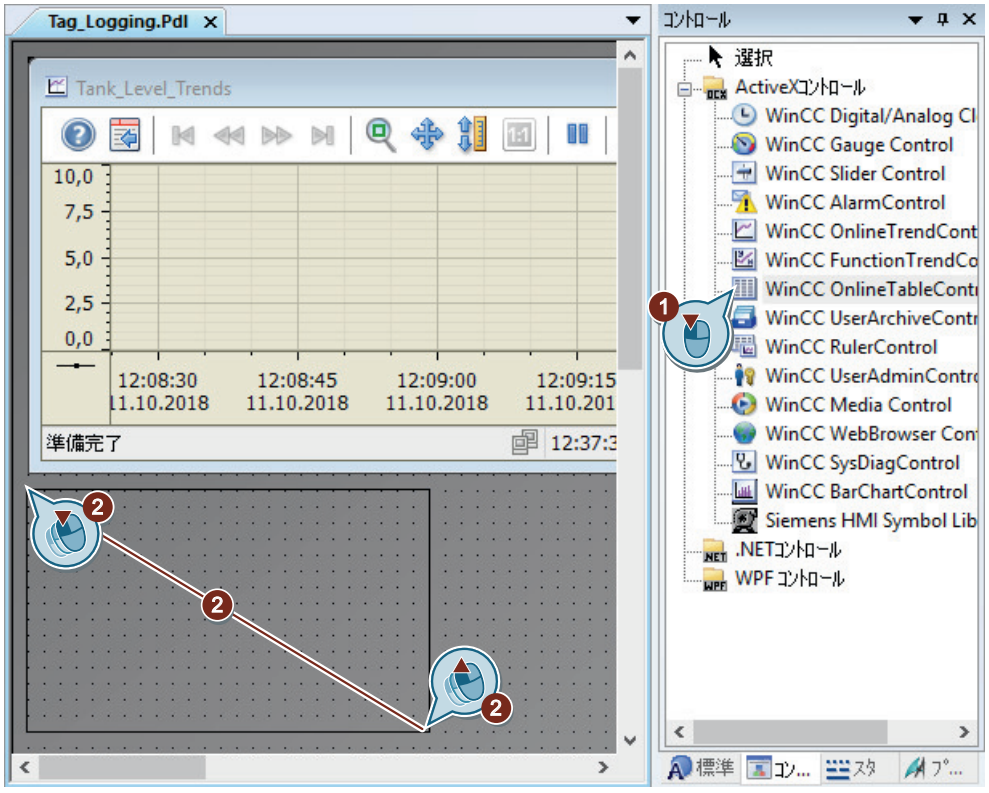
これで、ランタイムでアーカイブタグに保存された値がテーブルとして出力されます。

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。

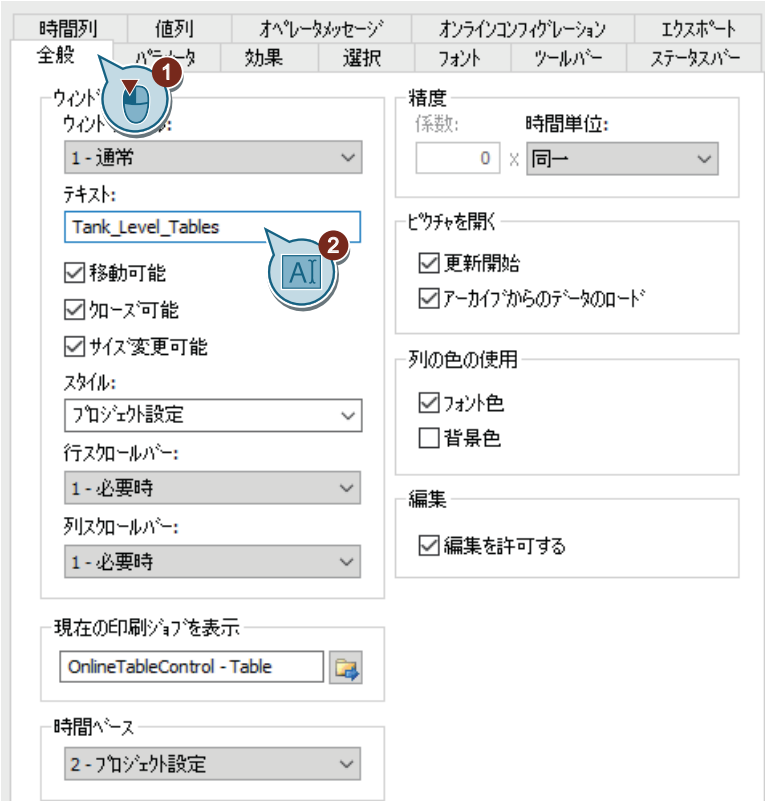
手順

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。

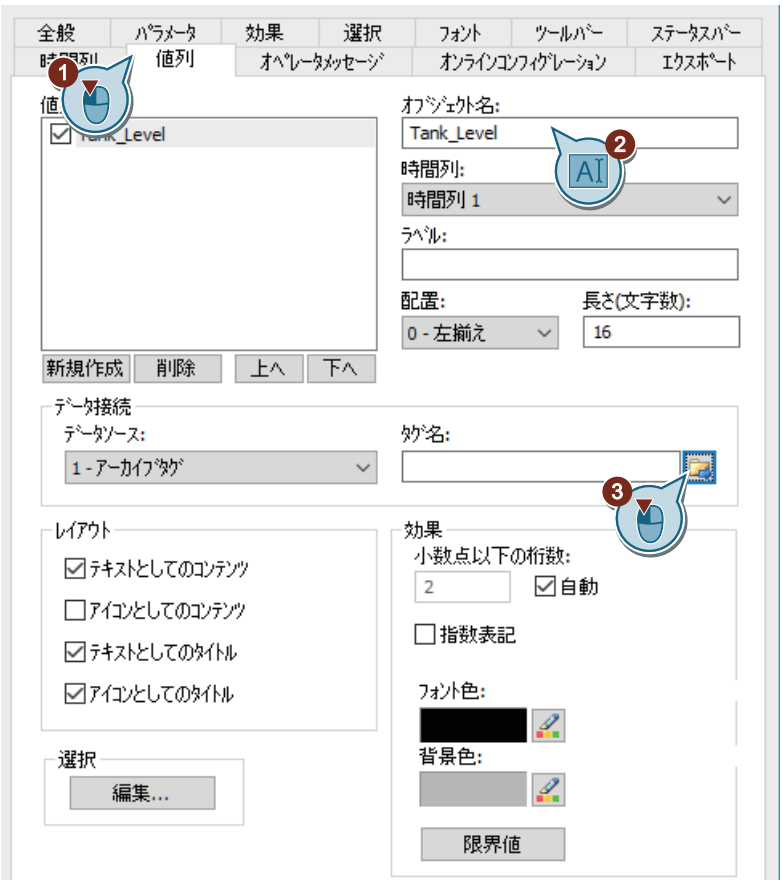


"グラフィックデザイナ"エディタの作業エリアに、[テーブル]ウィンドウが表示されます。[WinCC OnlineTableControl のプロパティ]ダイアログが開きます。

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。

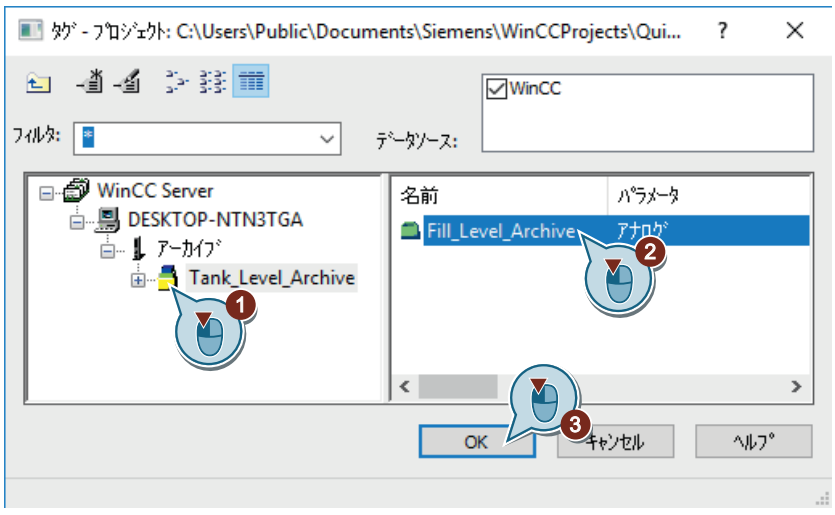

時間列
値列
オペレータメッセージ
オンラインコンフィグレーション
エクスポート
全般
パラメータ
効果
選択
フォント
ツールバー
ステータスバー
1
ウィンドウ
ウィンドウ:
1 - 通常
テキスト:
Tank_Level_Tables
2
AI
移動可能
クローズ可能
サイズ変更可能
スタイル:
プロジェクト設定
行スクロールバー:
1 - 必要時
列スクロールバー:
1 - 必要時
現在の印刷ジョブを表示
OnlineTableControl - Table
時間ベース
2 - プロジェクト設定
精度
係数:
時間単位:
0
x
同一
ピクチャを開く
更新開始
アーカイブからのデータのロード
列の色の使用
フォント色
背景色
編集
編集を許可する

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。

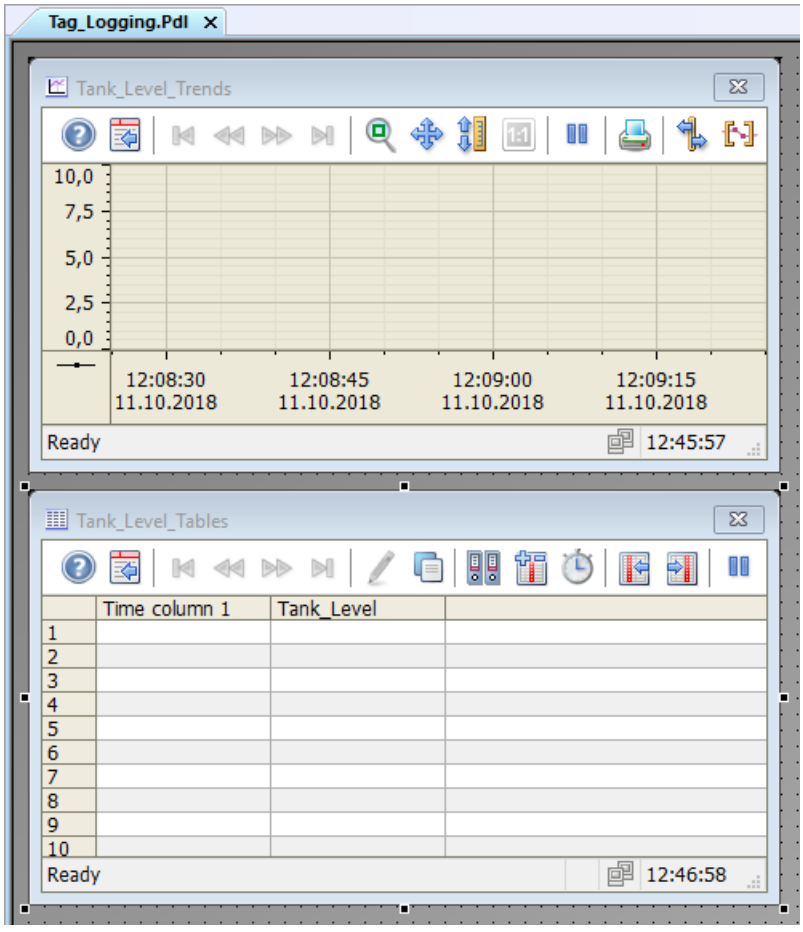
6. プロセス画像"Tag_Logging.pdl"を保存します。
7. グラフィックデザイナを閉じます。

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。

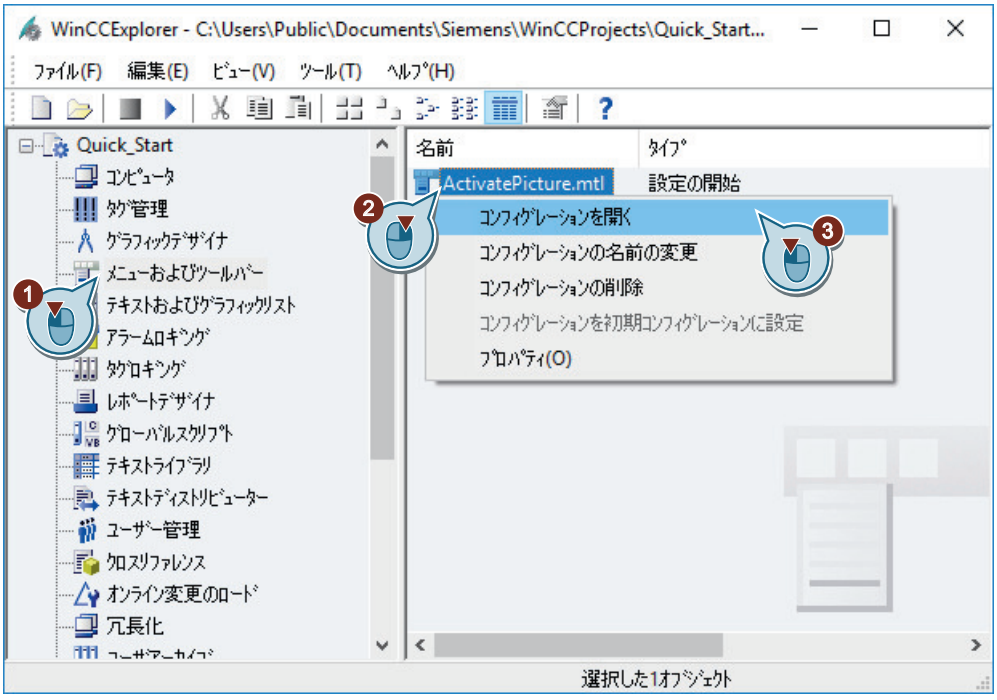
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

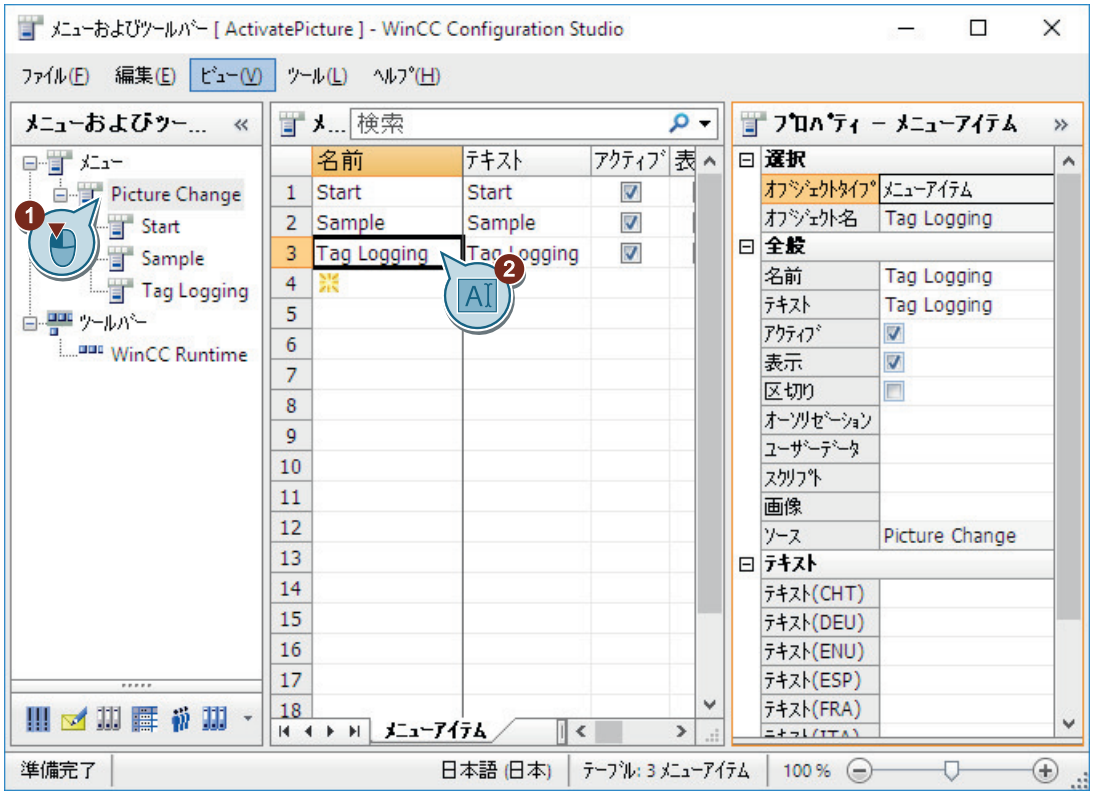
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- カスタマイズされた[画像変更]メニューがプロセス画像"START.pdl"および"SAMPLE.pdl"用に作成されていること。

手順

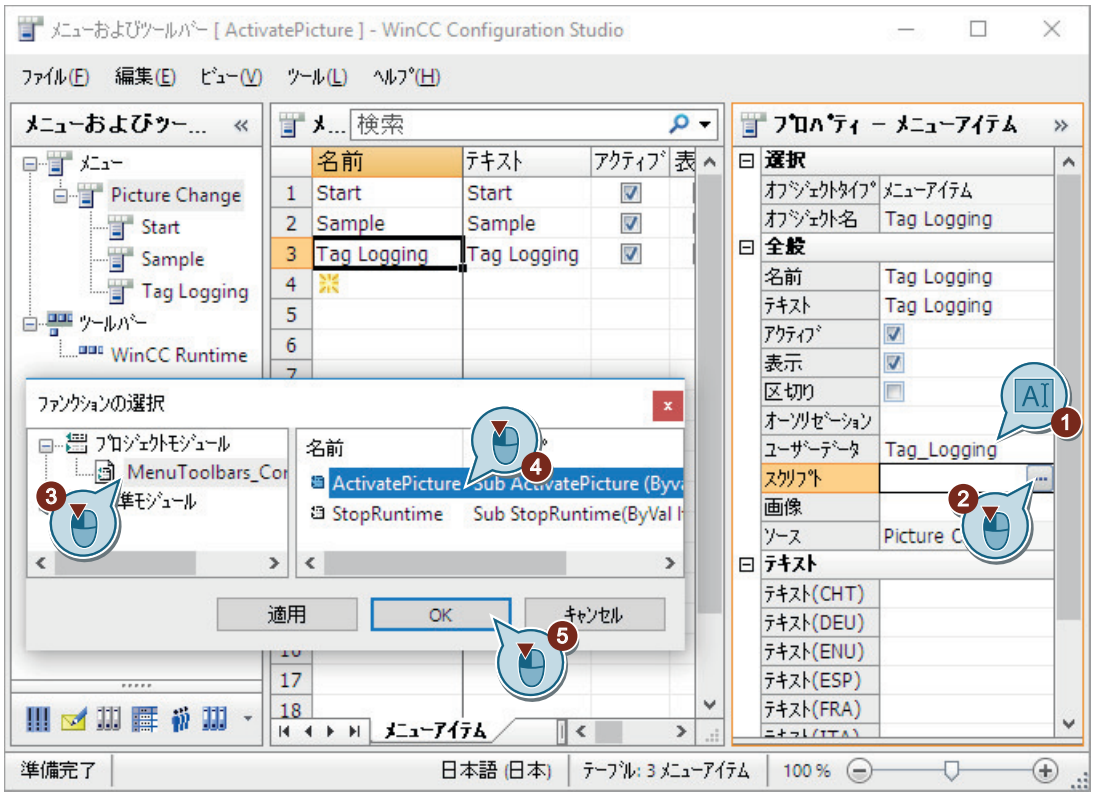
1. 保存された設定[ActivatePicture.mtl]を[メニューとツールバー]エディタで開きます。



2. [タグロギング]メニューコマンドを作成します。



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。

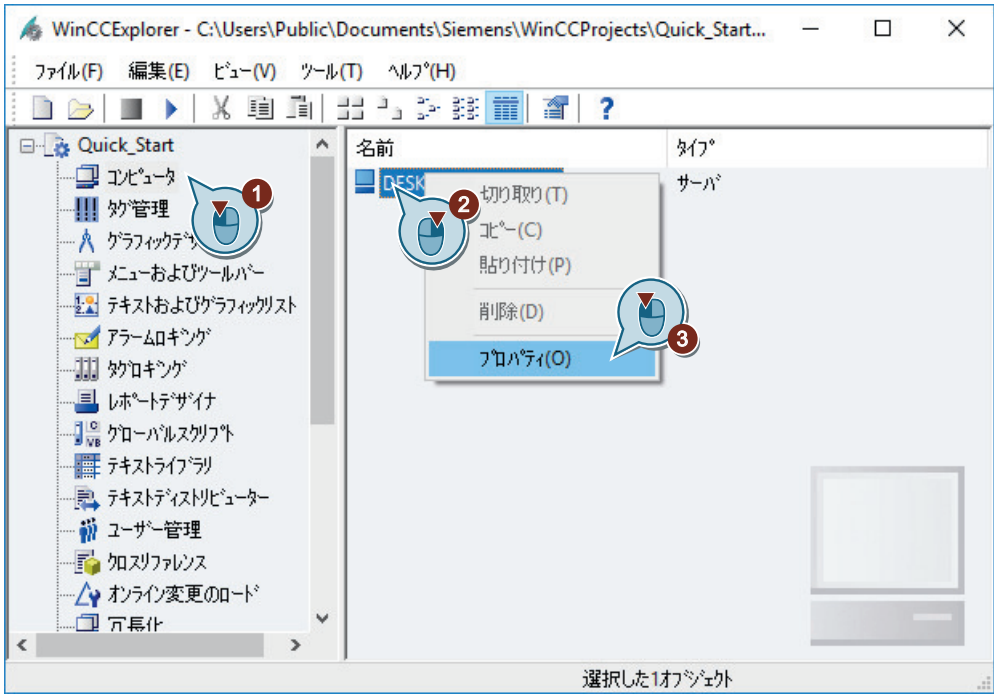
このセクションでは、プロジェクトの起動時にタグロギングランタイムが実行されるように WinCC Runtime をセットアップします。"Tag_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

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。

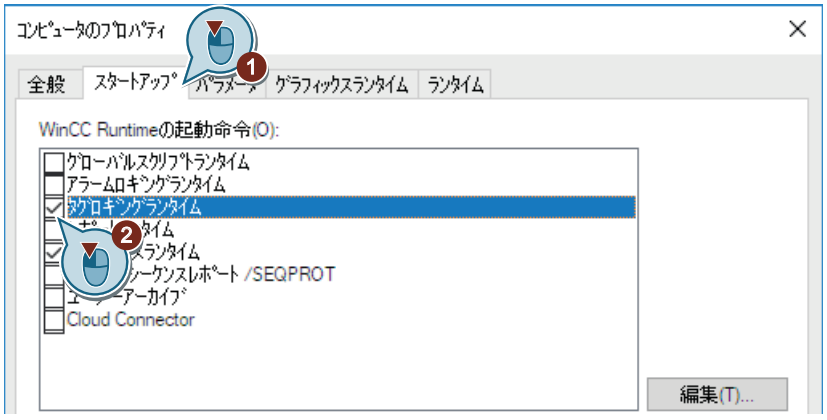
手順

1. [コンピュータのプロパティ]ダイアログを開きます。

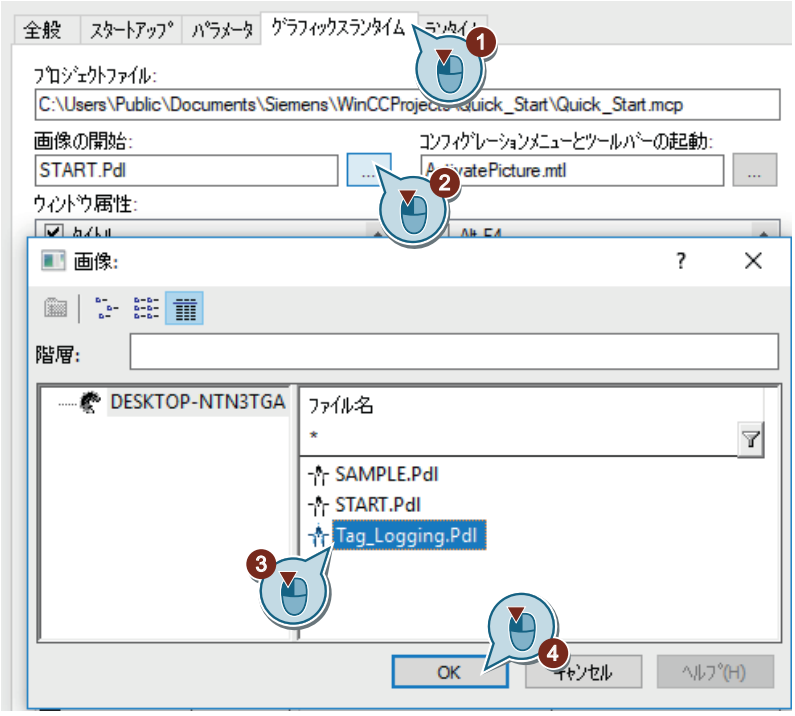


[コンピュータのプロパティ]ダイアログが開きます。

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。

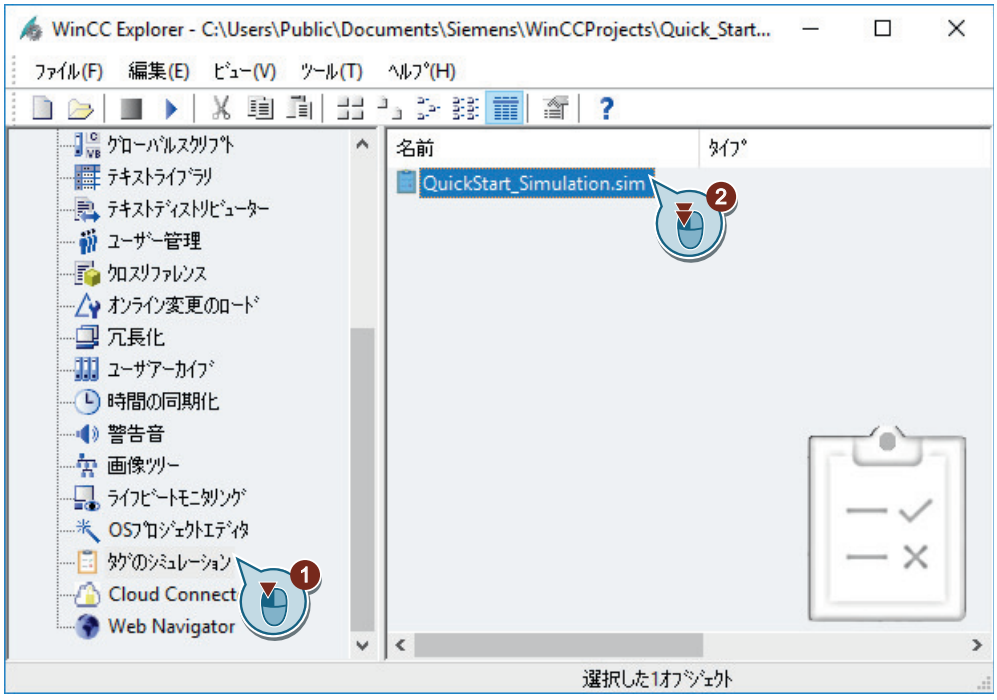
タグロギングランタイムがアーカイブ値を読み取り、[トレンド]ウィンドウおよび[テーブル]ウィンドウに転送します。値は、トレンドおよびテーブルとして出力されます。

必要条件

- ランタイムプロパティが定義されている。

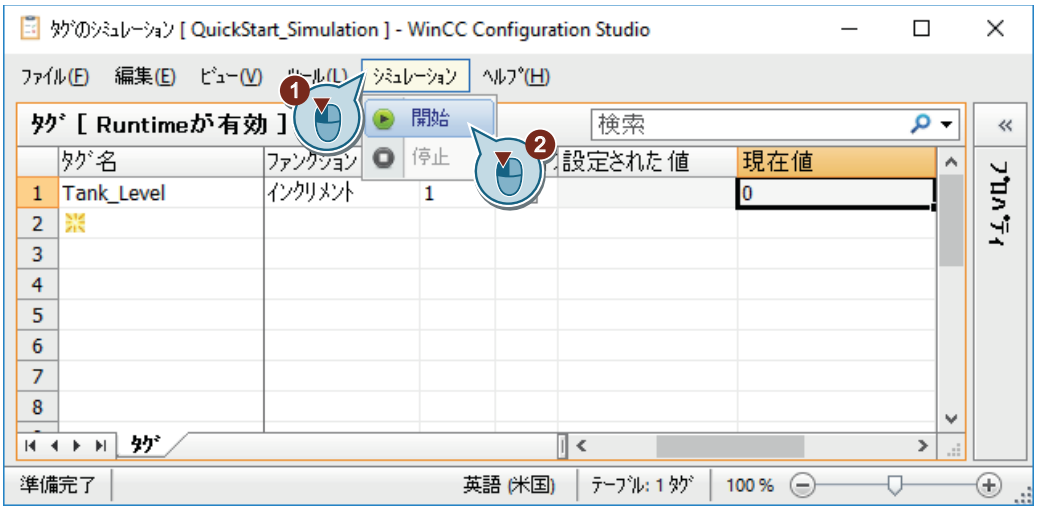
手順

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。

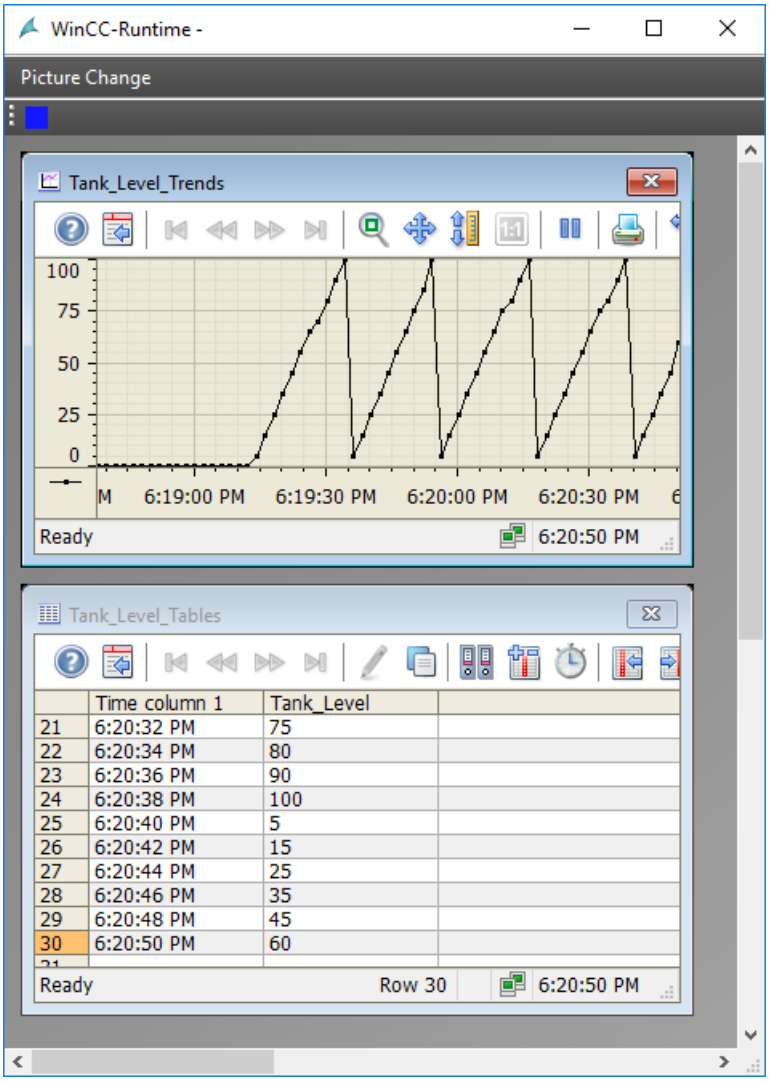


2. ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。

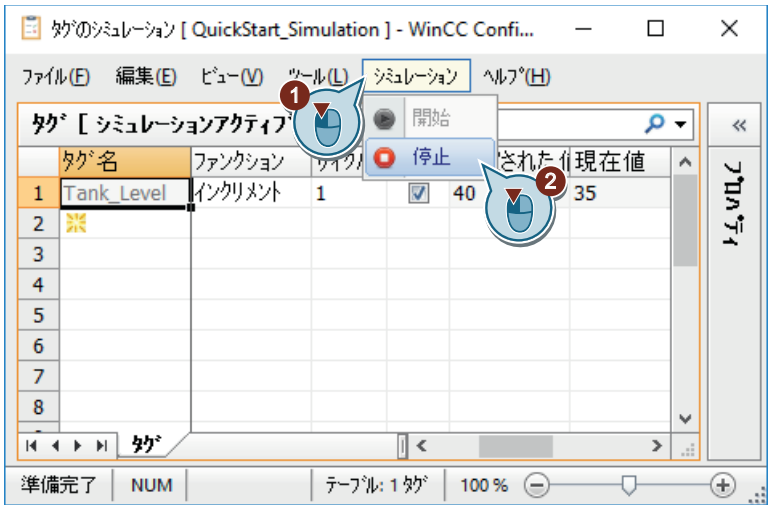
3. [タグシミュレーション]エディタでシミュレーションを起動します。



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。



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

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"エディタです。

[レポートデザイナ]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]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]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:

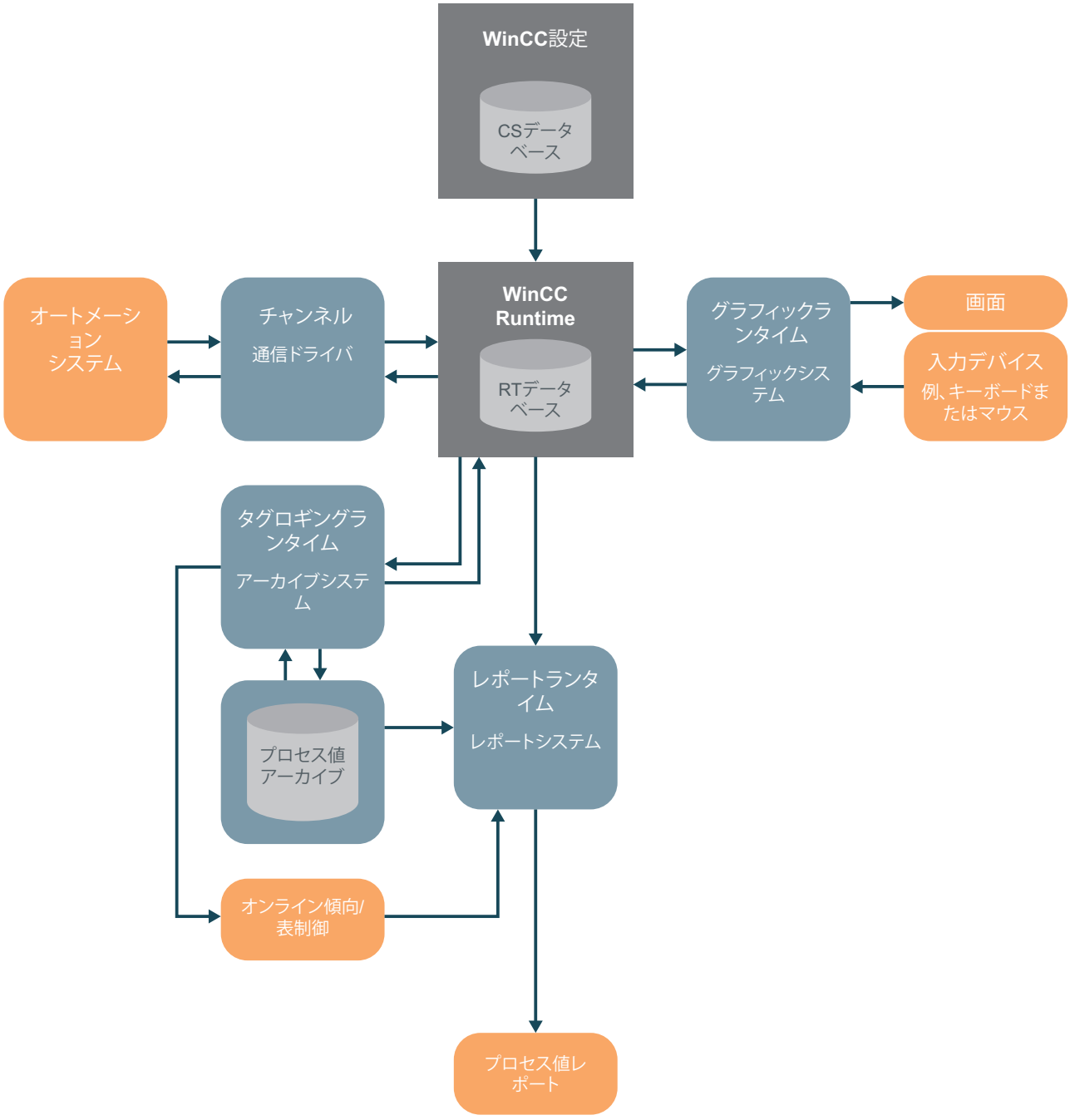
- アーカイブまたはコントロールからの文書化する値の読み取り
- プリンタ出力の制御

出力媒体

[レポートデザイナ]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

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"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。

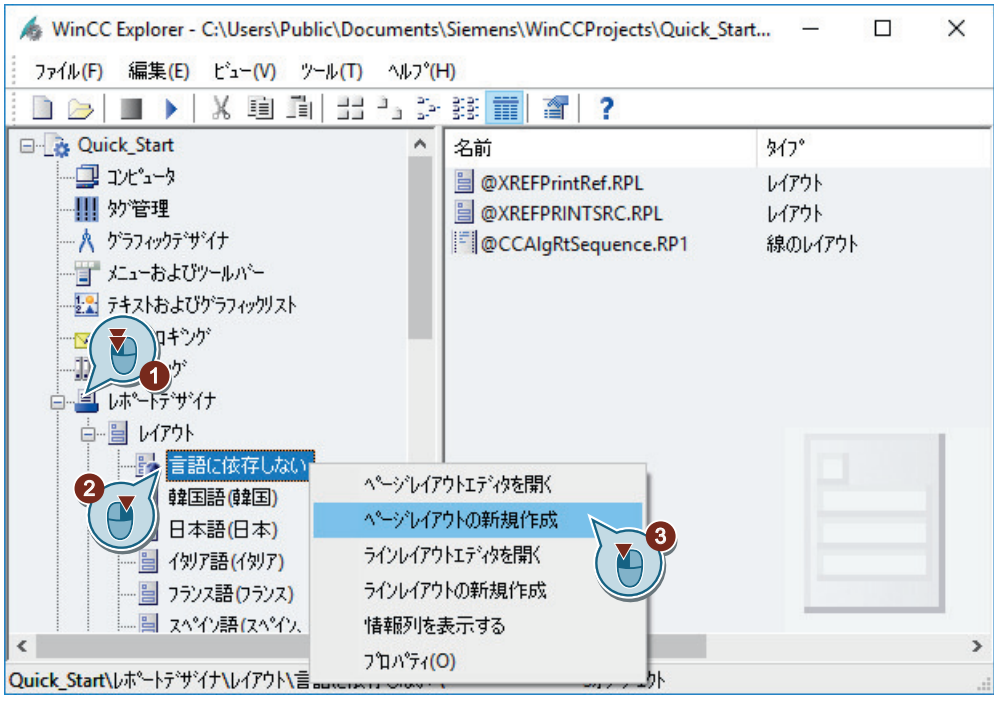
"Quick_Start"プロジェクトに対して、"レポートデザイナ"エディタで新しいページレイアウトを作成します。このページレイアウトを、プロセス値アーカイブからの値が文書化されるメッセージレポートのテンプレートとして使用します。

必要条件

- "Quick_Start"プロジェクトが開いている。

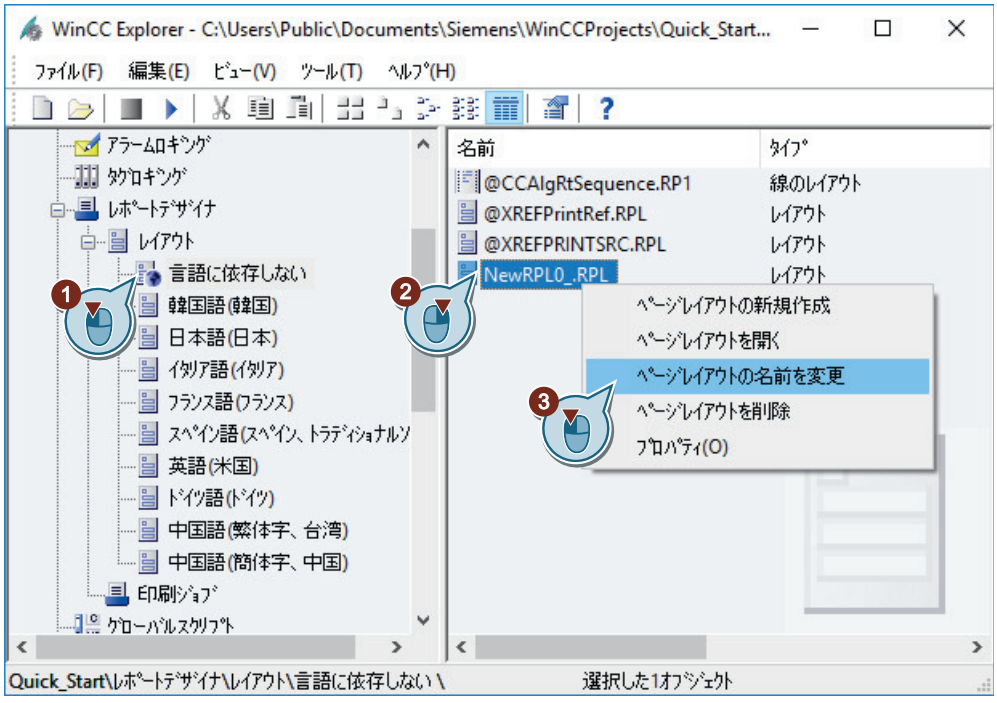
手順

1. WinCC エクスプローラで、新しい言語非依存ページレイアウトを作成します。

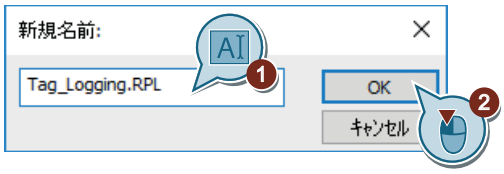


新しいレイアウトファイル"NewRPL0.RPL"は、[言語非依存]フォルダで作成されます。

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]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。

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"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。

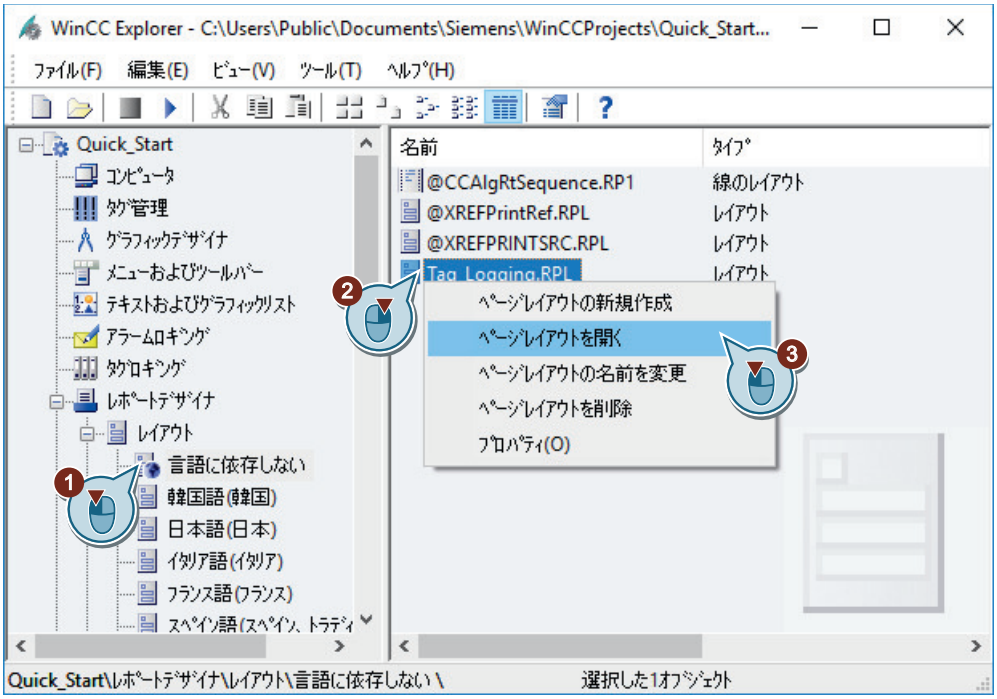
- ページサイズ
  ページサイズは、レイアウトにおける全体エリアを示します。
  ページサイズは、ログの出力フォーマットを決定します。
- 印刷余白
  印刷余白は、印刷不可の余白領域を定義します。
  この領域は、デフォルトではページレイアウトエディタでグレー表示され、編集できません。
- [カバーシート]ページの出力
  ページレイアウトの[オブジェクトプロパティ]ダイアログで、ログを最初のページ（カバー）と共に出力するかどうかを定義します。
  このセクションでは、[カバーシート]の出力を無効にします。

必要条件

- レイアウトファイル"Tag_Logging.rpl"が作成されている。

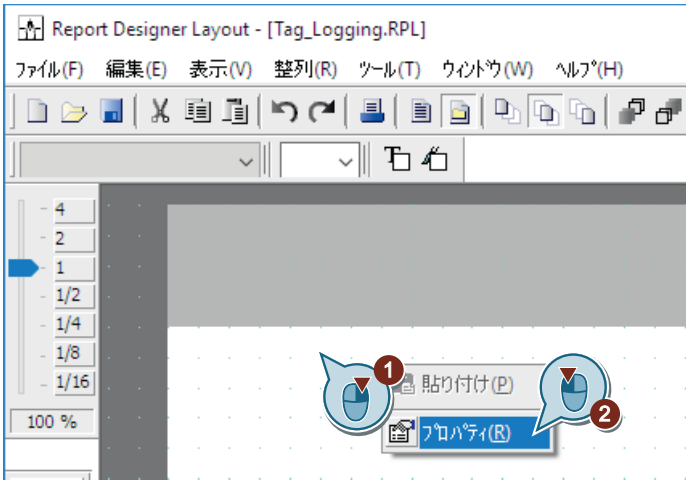
手順

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。



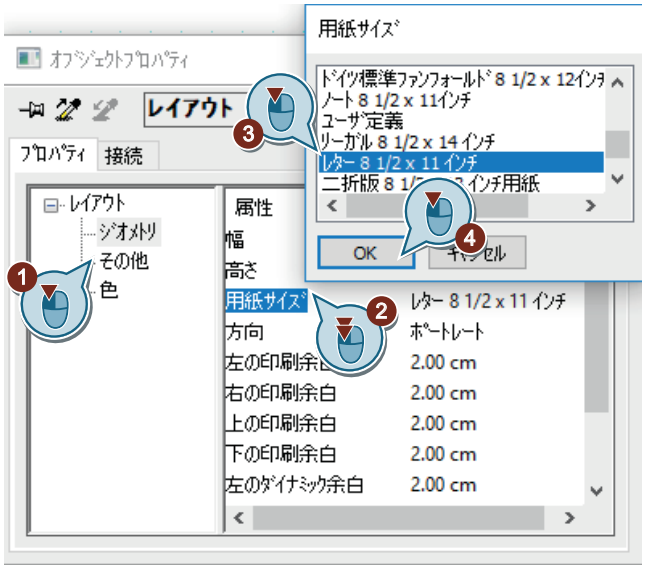
ページレイアウトエディタで、"Tag_Logging.rpl"レイアウトファイルが開きます。

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

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。

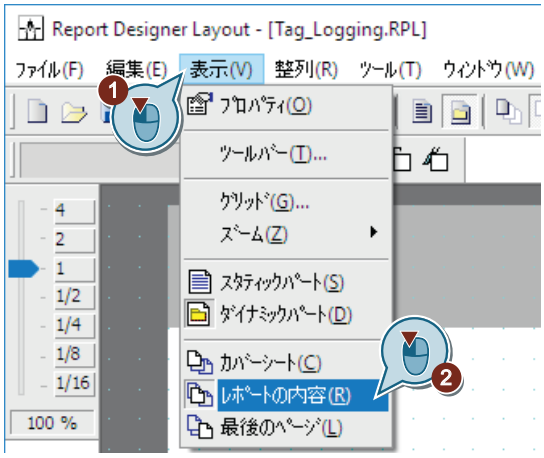
このセクションでは、プロセス値アーカイブからの値の表示に使用する[WinCC コントロールランタイム印刷プロバイダ - テーブル]オブジェクトを挿入します。

必要条件

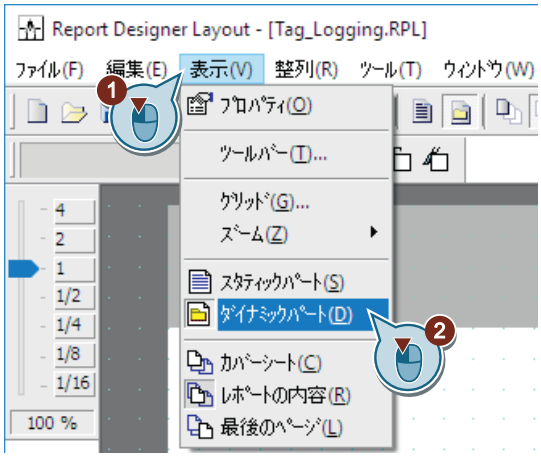
- ページレイアウトエディタで、[Tag_Logging.rpl]レイアウトファイルが開いている。

手順

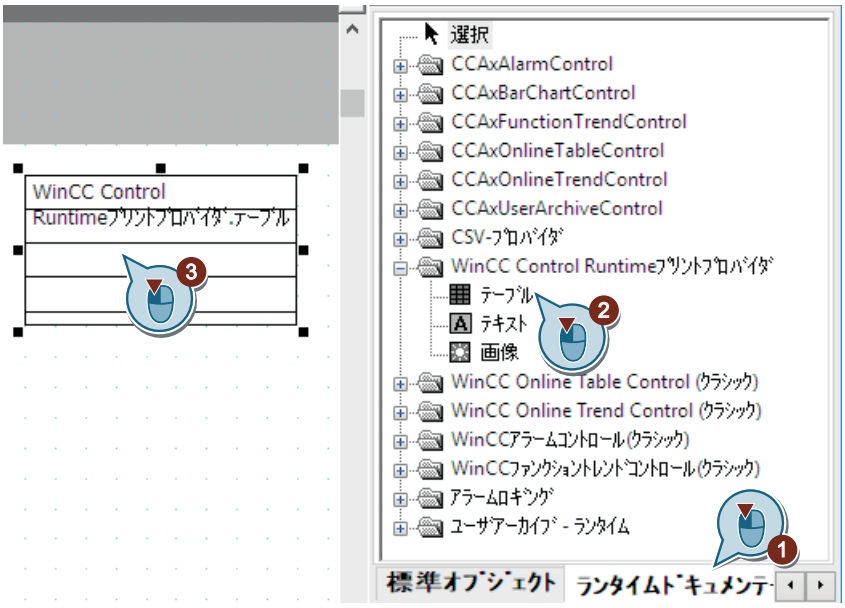
1. ページレイアウトの"レポートの内容"ページを開きます。



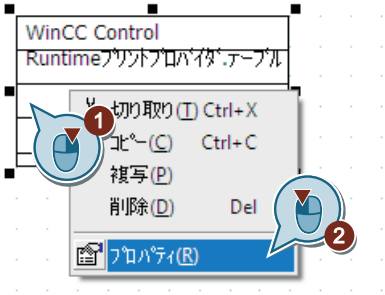
2. [レポートの内容]ページのダイナミックレベルへ変更します。



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。



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

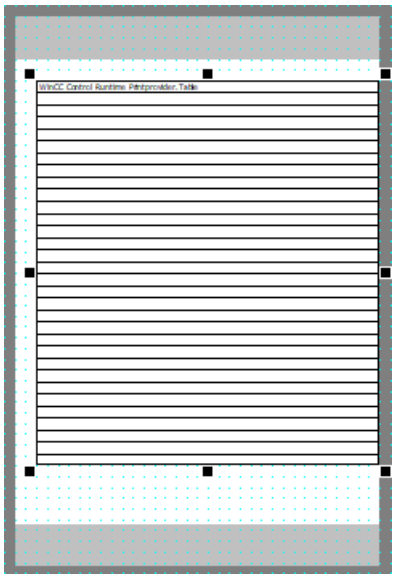
6. [オブジェクトプロパティ]ダイアログを閉じます。

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。



8.4.4 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。

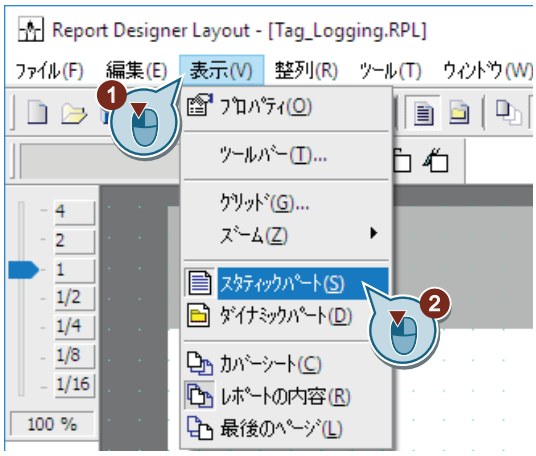
ヘッダーで、システムオブジェクト"プロジェクト名"を挿入します。このオブジェクトは、ログでのプロジェクト名の表示でプレースホルダになります。

必要条件

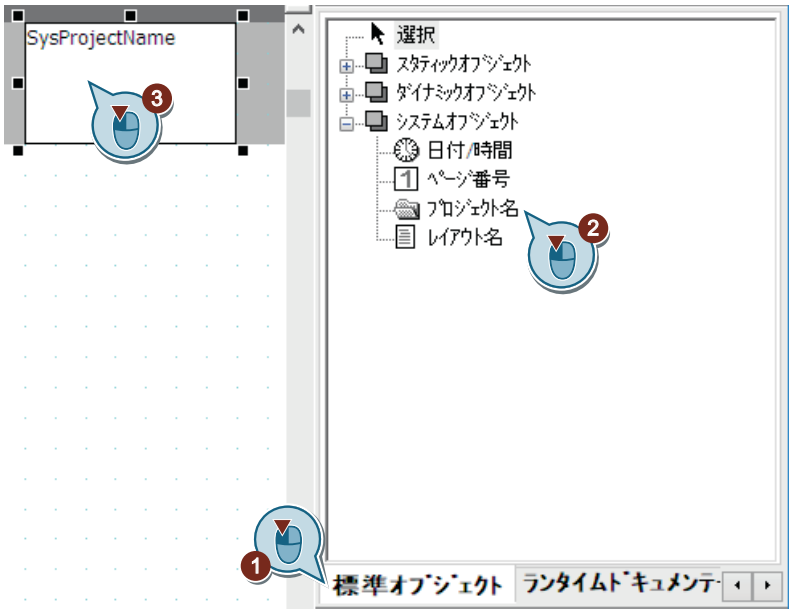
- ページレイアウトエディタで、[Tag_Logging.rpl]レイアウトファイルが開いている。

手順

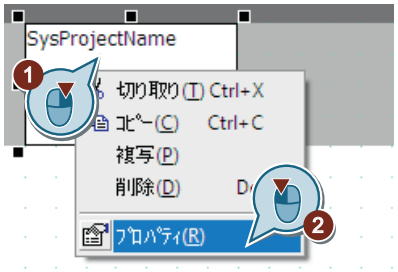
1. [表示]メニューのページレイアウトの"レポートの内容"ページを開きます。
2. [レポートの内容]ページのスタティックレベルへ変更します。



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。



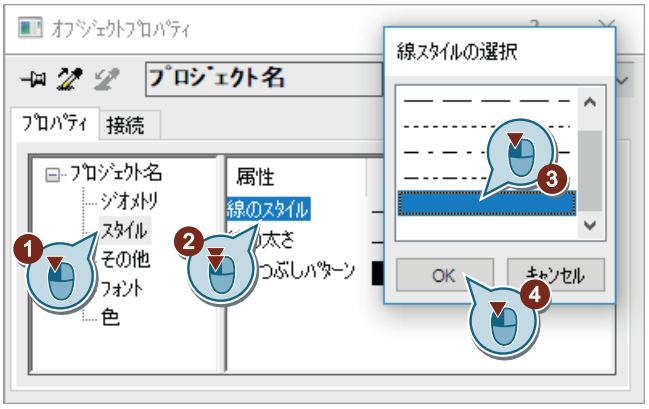
4. [オブジェクトプロパティ]ダイアログを開きます。



[オブジェクトプロパティ]ダイアログが開きます。

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。



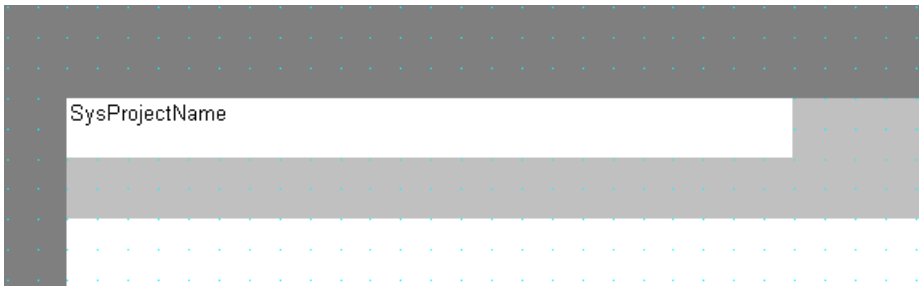
7. [オブジェクトプロパティ]ダイアログを閉じます。

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。



8.4.5 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。

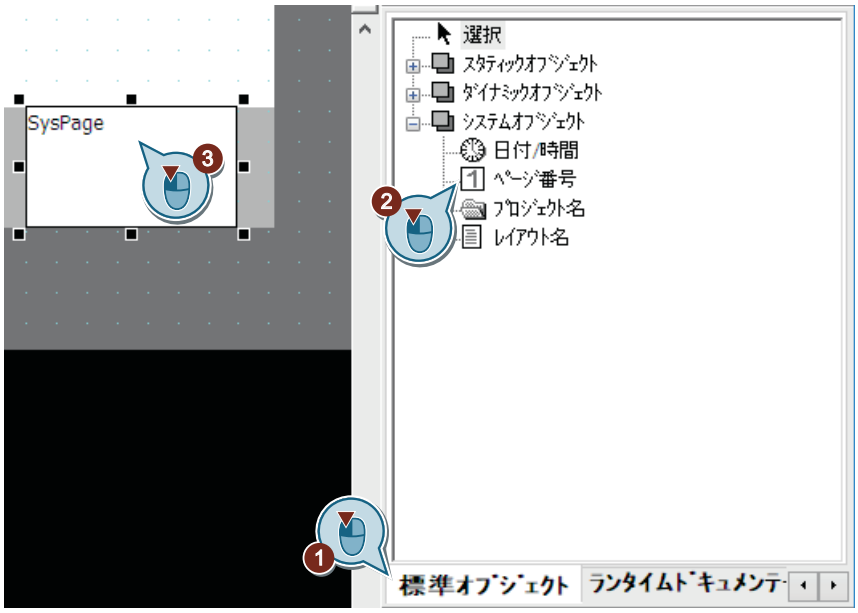
必要条件

- 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。

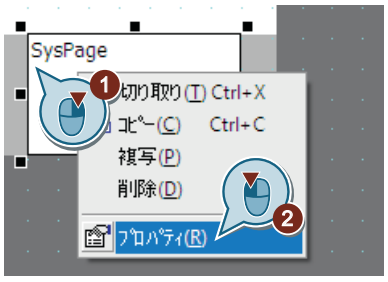
手順

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。



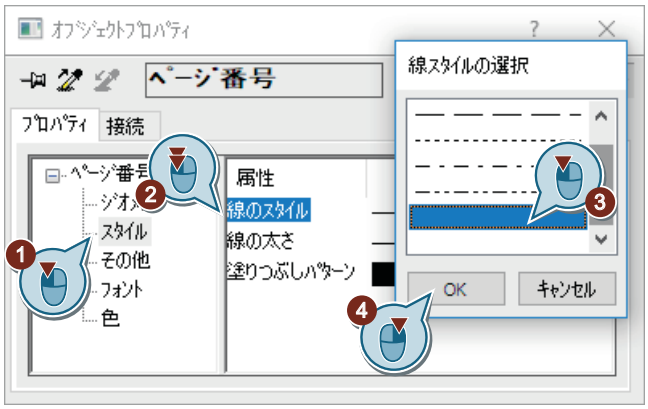
4. [オブジェクトプロパティ]ダイアログを開きます。



[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。

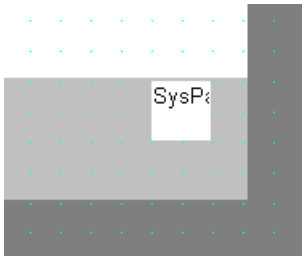


7. [オブジェクトプロパティ]ダイアログを閉じます。
8. ツールバーの ボタンで、[Tag_Logging.rpl]レイアウトファイルを保存します。
9. "レポートデザイナ"を閉じます。

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、一部の設定だけが変更できます。

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。

8.5.2 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。

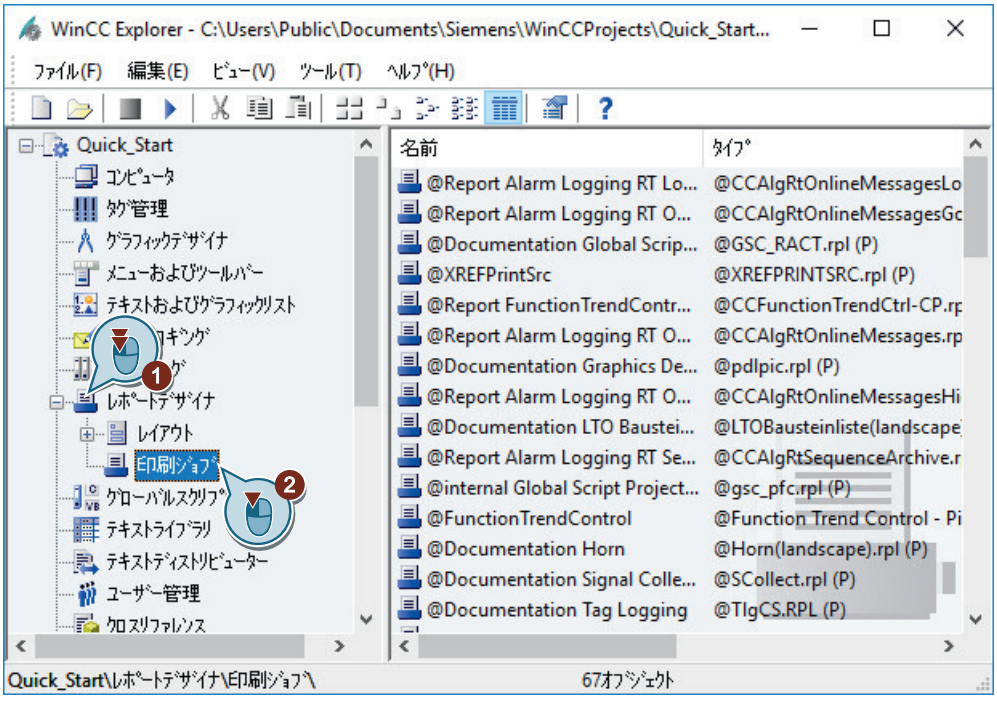
選択したプリンタを、ログの出力媒体として定義します。プリンタがない場合は、ログをファイルに出力できます。このトピックに関する詳細情報は、「プロジェクトのドキュメント」の「WinCC の操作」 > 「設定データとランタイムデータのドキュメント」 > 「プロジェクトのドキュメント」 > 「新しい印刷ジョブの作成方法」のセクションにあります。

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。

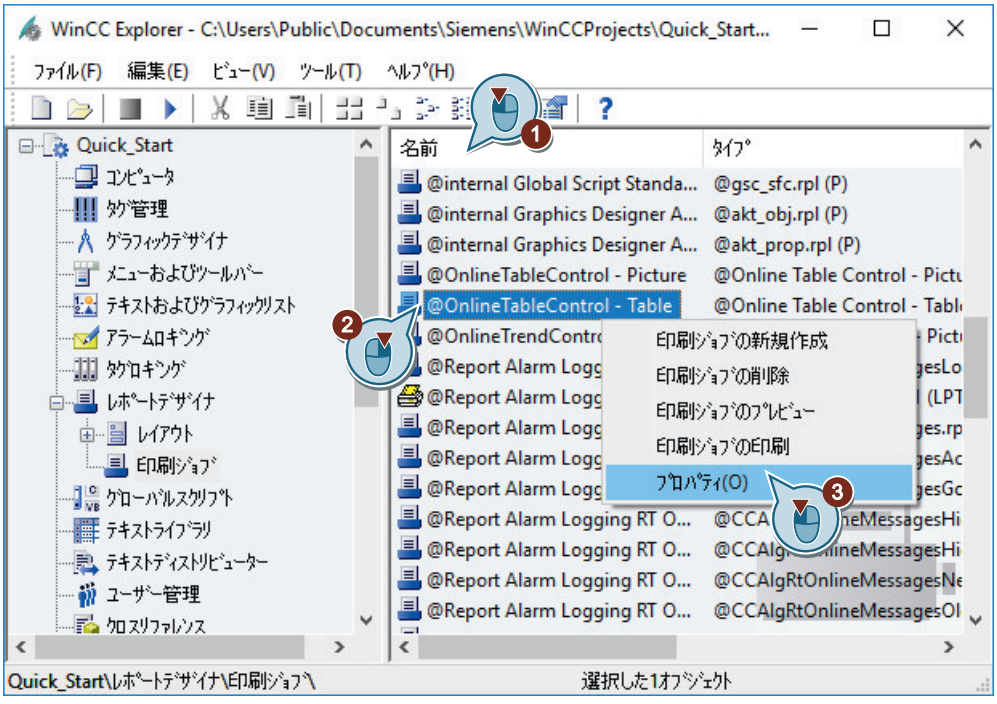
手順

1. WinCC エクスプローラで[レポートデザイナ]エディタの[印刷ジョブ]コンポーネントを選択します。

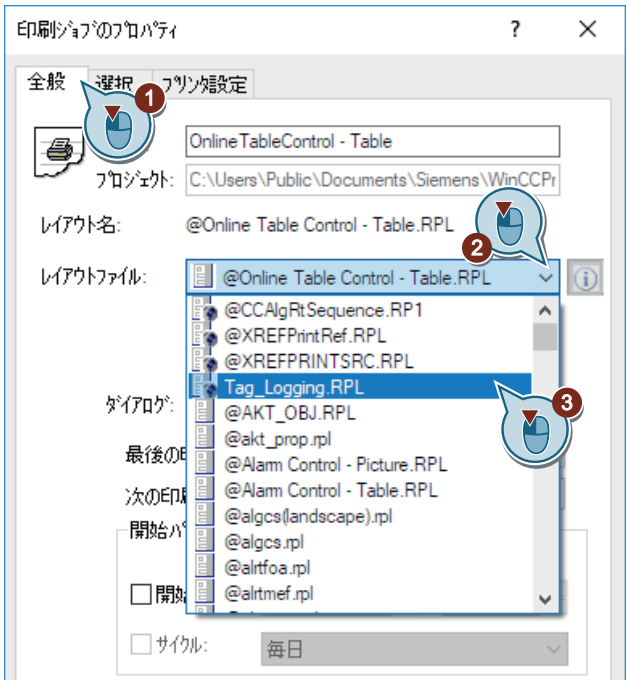


事前に定義された印刷ジョブが表示されます。

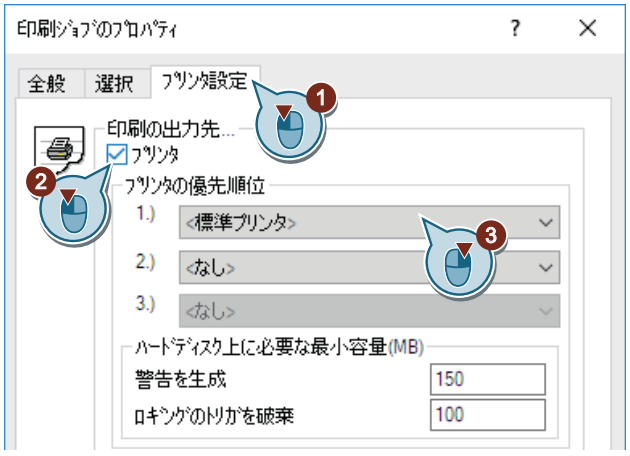
2. 印刷ジョブ"@OnlineTableControl - Table"を選び、[印刷ジョブのプロパティ]ダイアログを開きます。



3. 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。

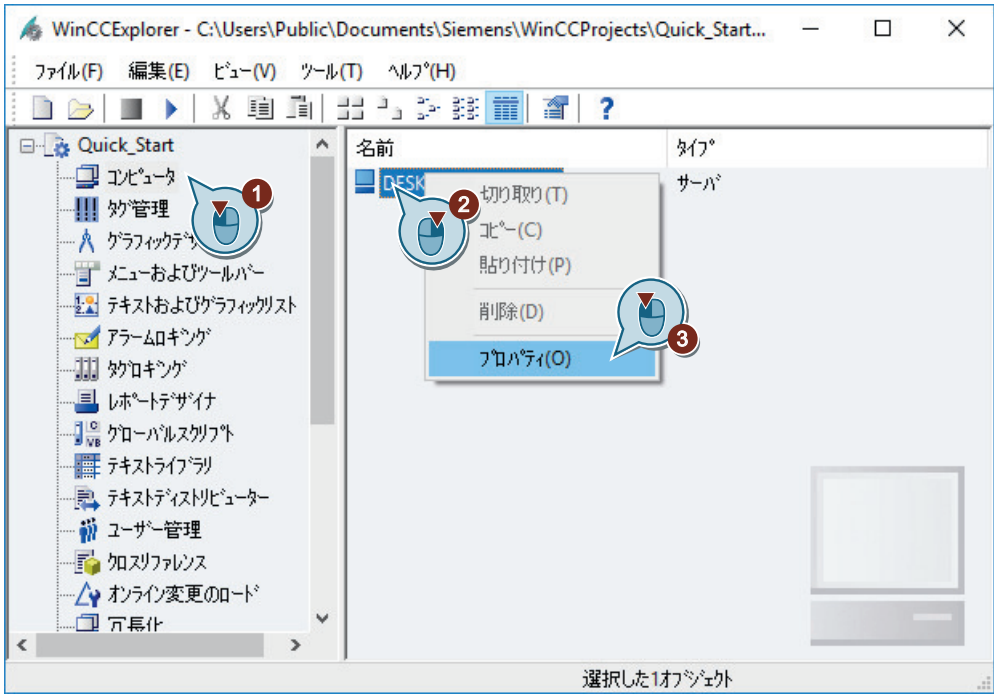
"Tag_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

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。

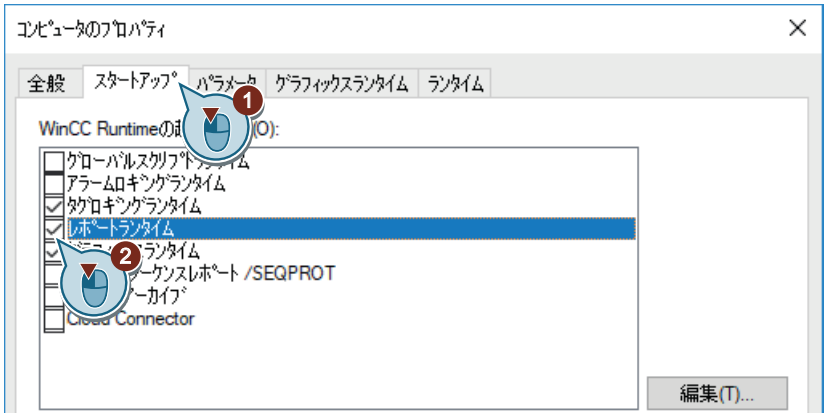
手順

1. [コンピュータのプロパティ]ダイアログを開きます。

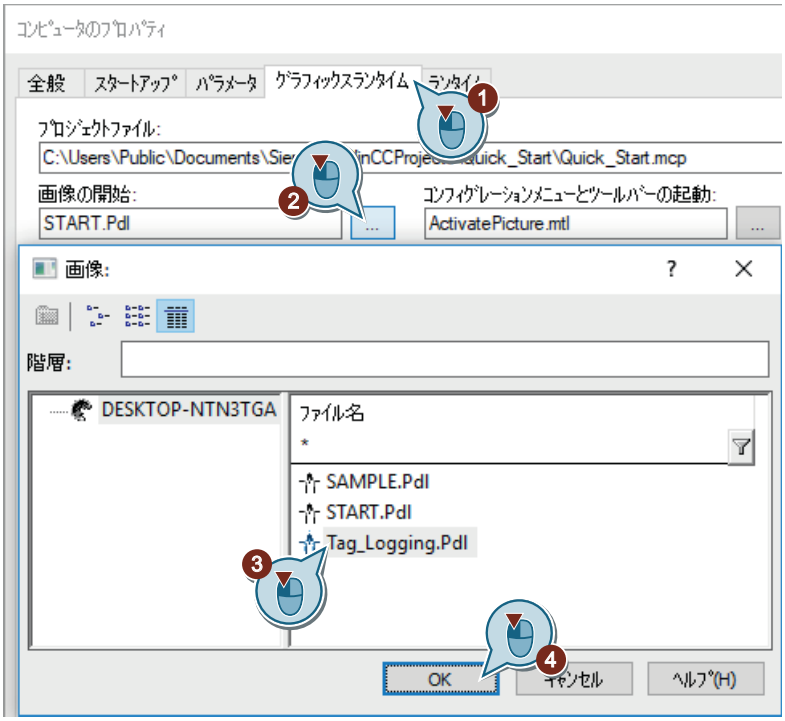


[コンピュータのプロパティ]ダイアログが開きます。

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。



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、ランタイムウィ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。

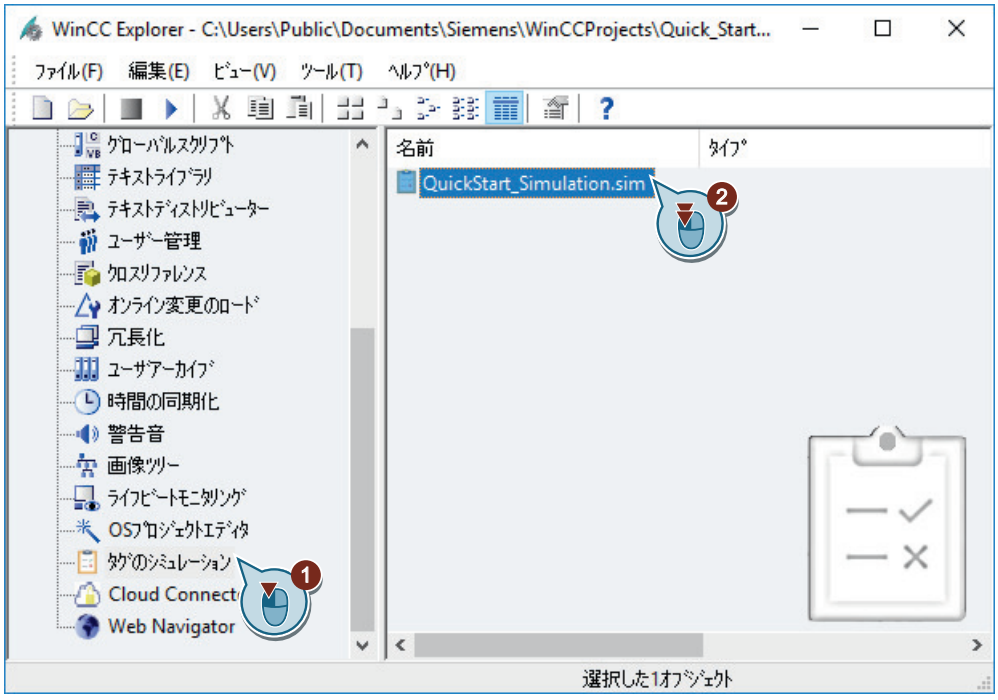
タグロギングランタイムがアーカイブ値を読み取り、[トレンド]ウィンドウおよび[テーブル]ウィンドウに転送します。値は、トレンドおよびテーブルとして出力されます。

必要条件

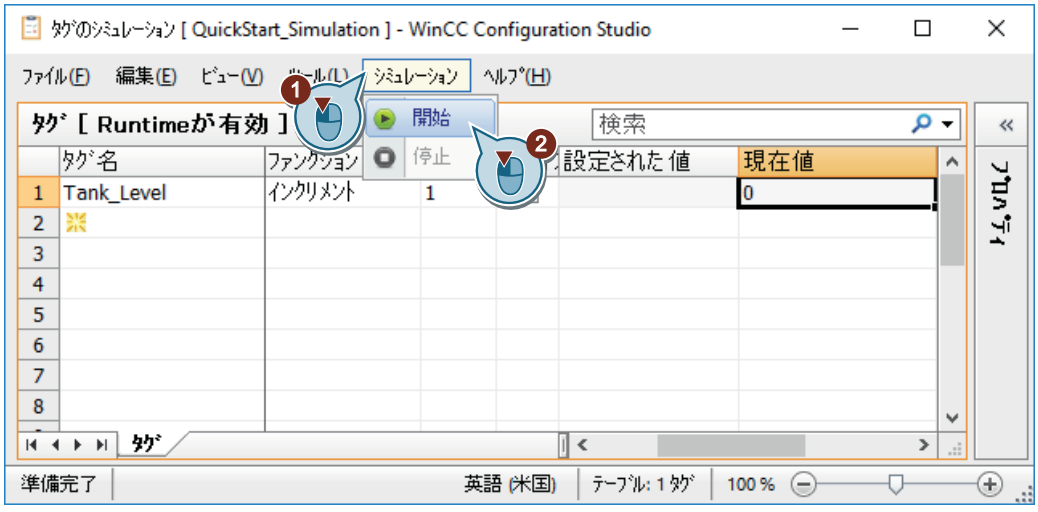
- ランタイムプロパティが定義されている。

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。

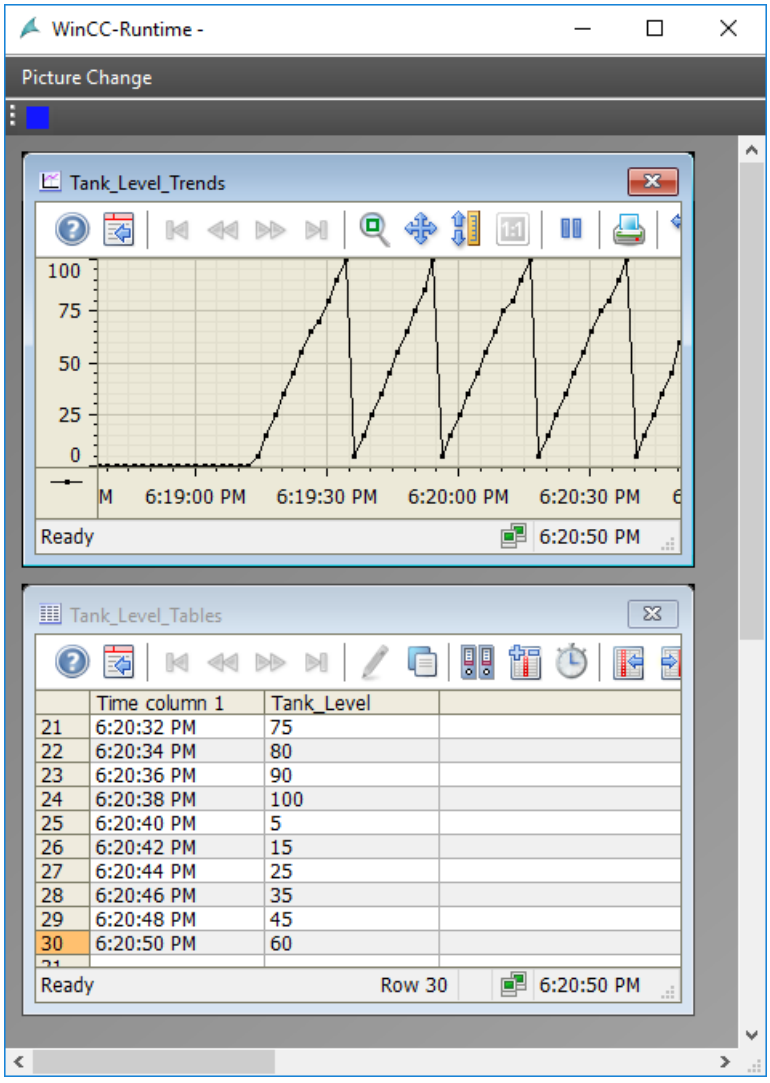


結果

"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。

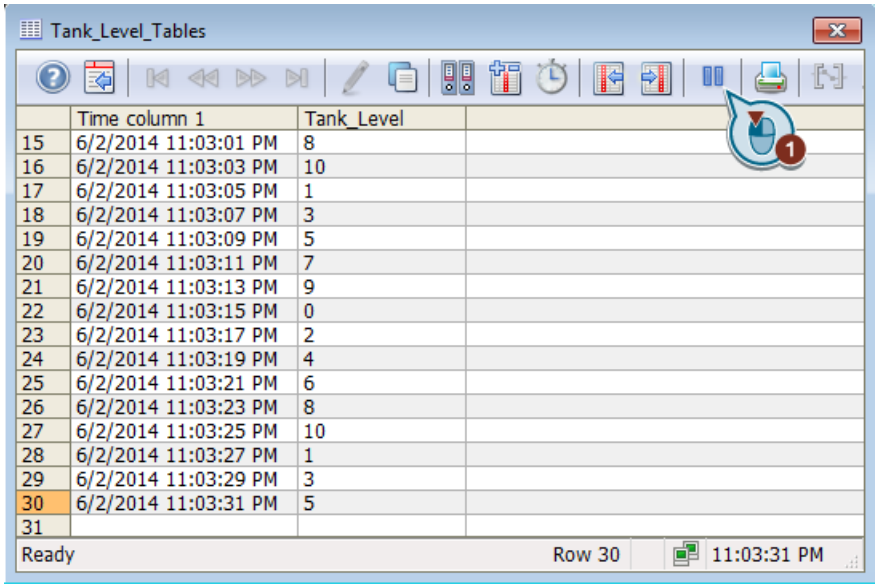
ログは、ページレイアウト"Tag_Logging.rpl"をベースにしています。

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。

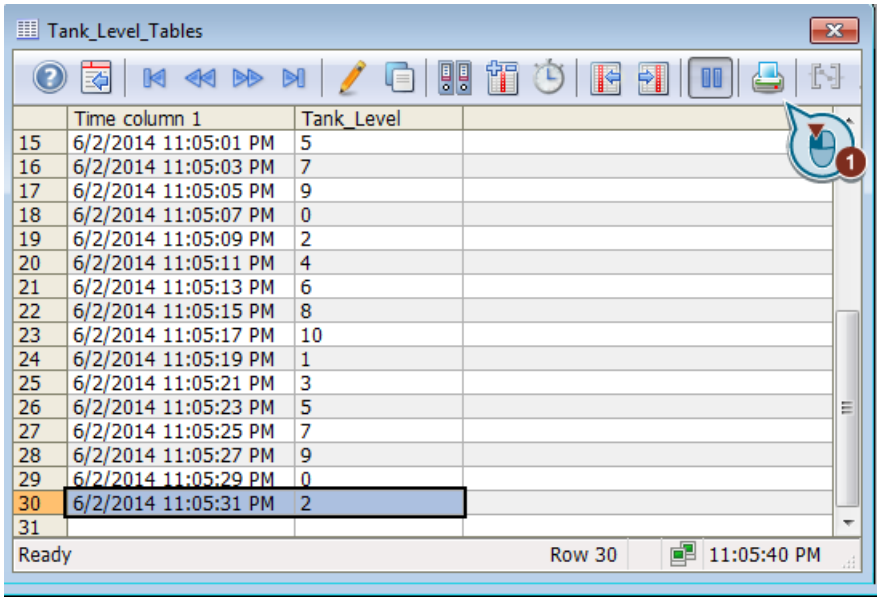
手順

1. ボタンによって、データ更新を停止します。



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。



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。

結果

ログが印刷されるだけです。

印刷されるログは、[テーブル]ウィンドウの現在の表示の値が文書化されるページによって構成されます。

メッセージの設定 9

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値および上限値を設定します。
  内部タグ"Tank_Level"の値が限度に違反すると、対応するアナログメッセージがトリガされ、ランタイムで表示されます。

9.2 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。

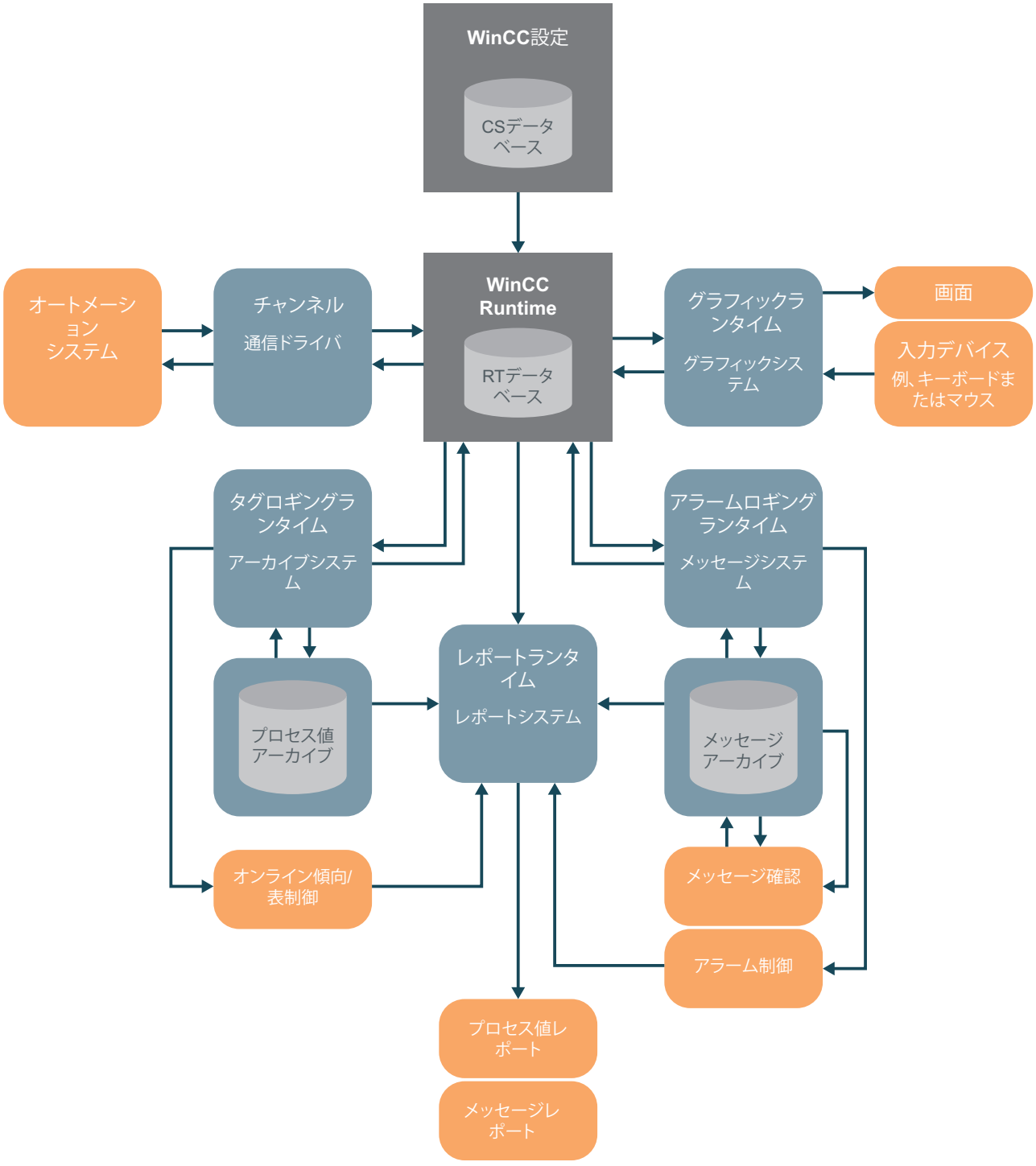
特定のステータスやプロセスにおける変化によって、メッセージシステムはメッセージを生成し、それらをランタイムのテーブルに出力します。メッセージは、危機的な状況を初期段階で認識し、ダウンタイムを回避するのに役立ちます。

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。

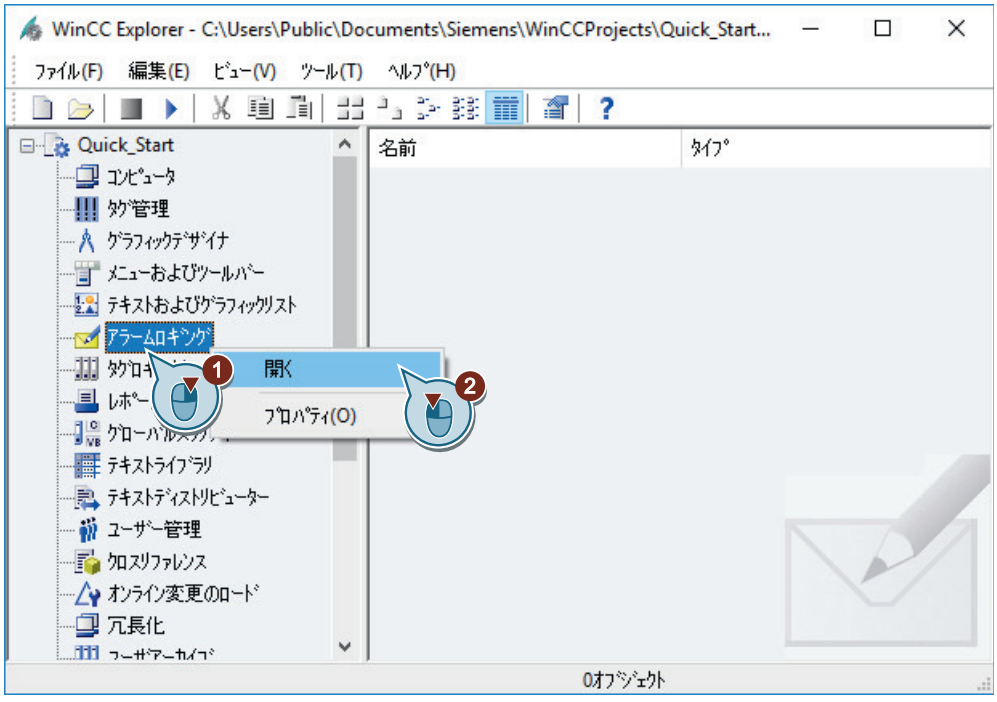
"アラームロギング"エディタで、"Quick_Start"プロジェクトに必要な全てのビットメッセージおよびアナログメッセージを設定します。

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。

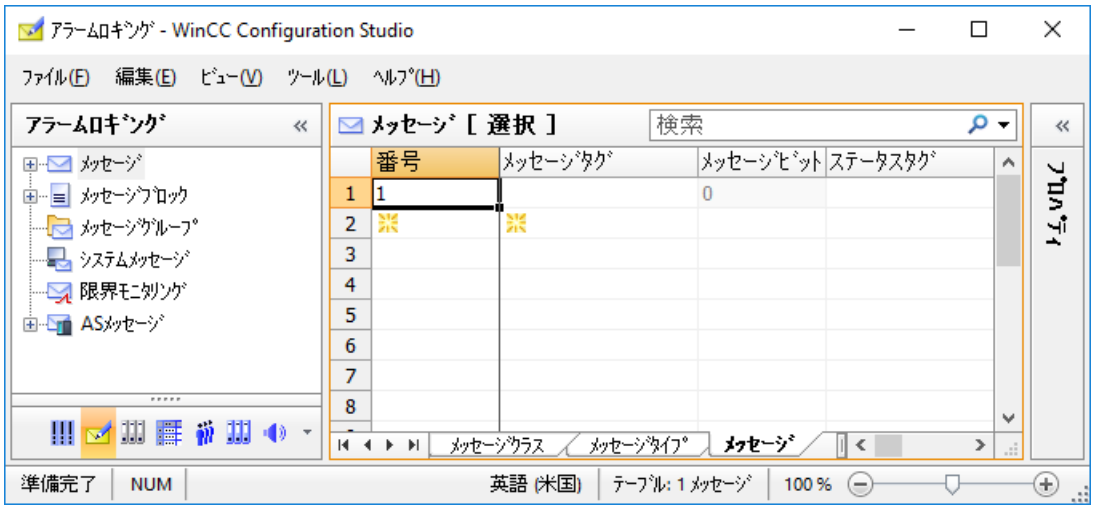


[アラームロギング]エディタが開きます。

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。


アラームロギング - WinCC Configuration Studio
ファイル(F) 編集(E) ビュー(V) ツール(L) ヘルプ(H)
アラームロギング
メッセージ
メッセージブロック
メッセージグループ
システムメッセージ
限界モニタリング
ASメッセージ
メッセージ [選択]
検索
番号
メッセージタグ
メッセージビット
ステータスタグ
1
0
メッセージクラス
メッセージタイプ
メッセージ
プロパティ
準備完了
NUM
英語 (米国)
テーブル: 1 メッセージ
100 %

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。

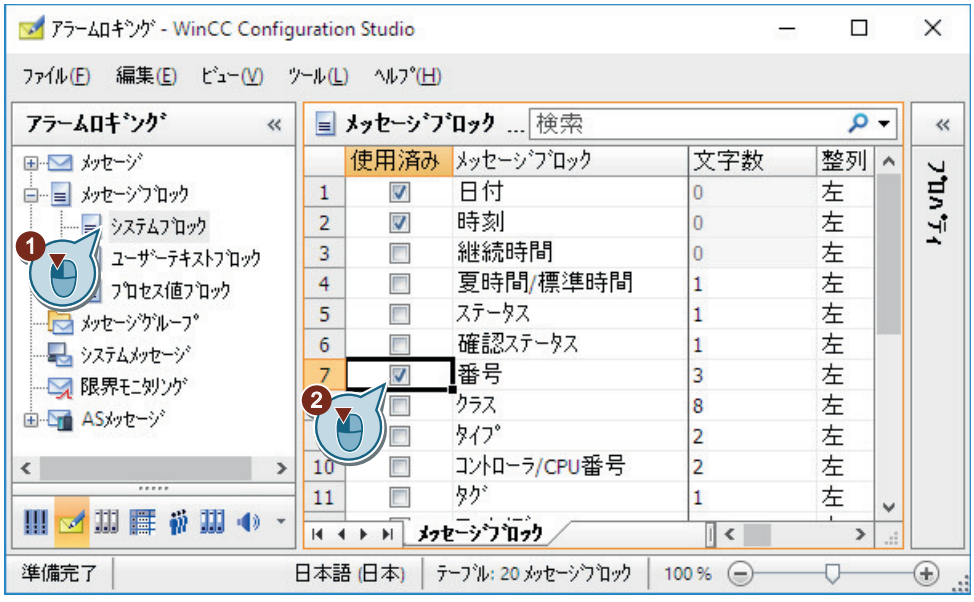
"Quick_Start"プロジェクトに使用されるユーザーテキストブロックの長さを変更します。

必要条件

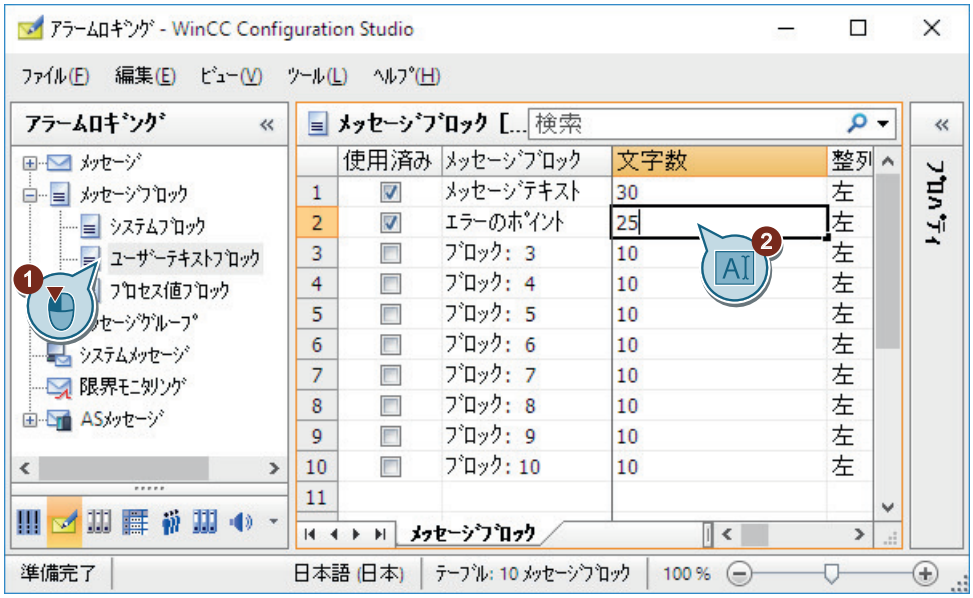
- [アラームロギング]エディタが開いている。

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"



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。

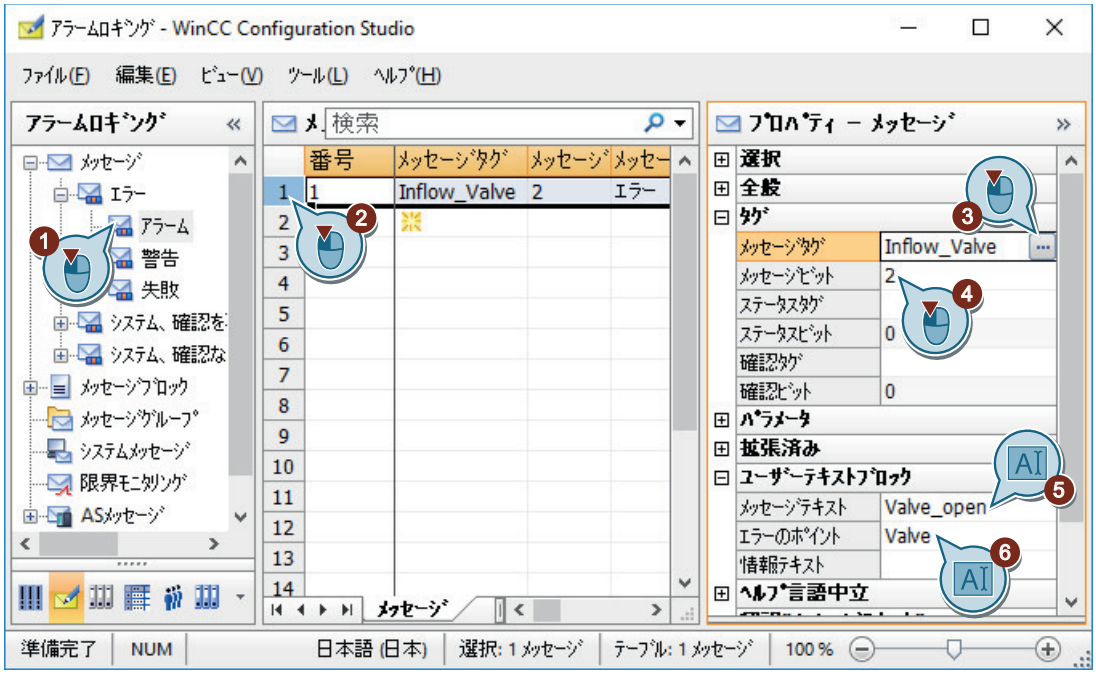
必要条件

- [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 45)」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。

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

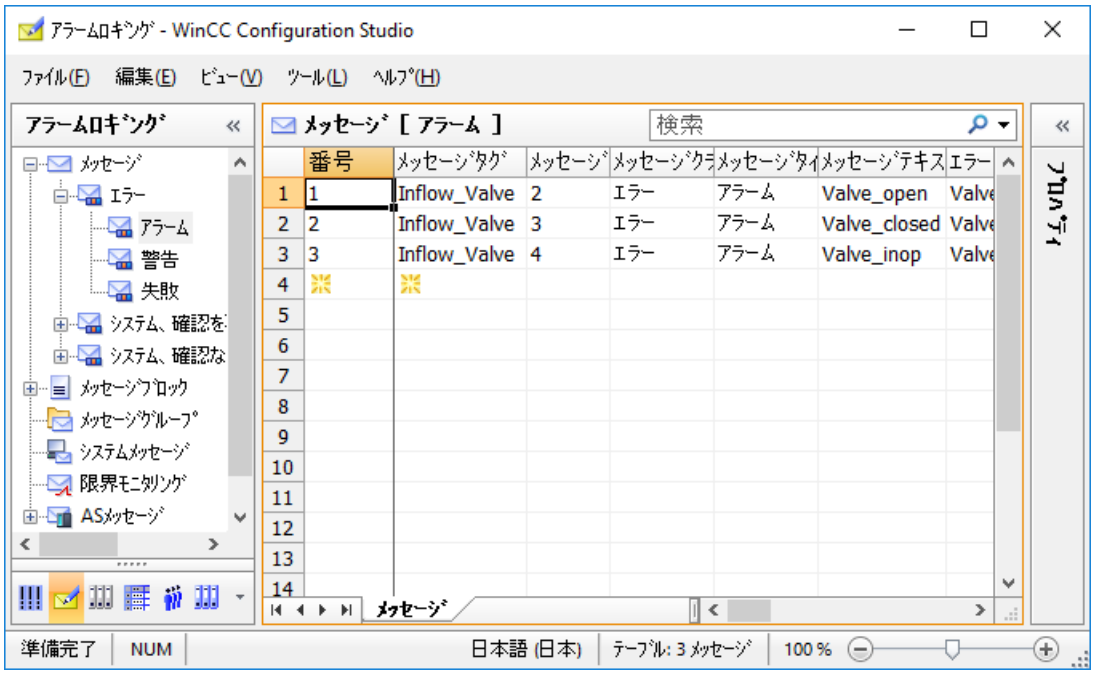
結果

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

作成されたディスクリートアラームは、[アラームロギング]エディタのテーブルウィンドウで表示されます。

ランタイムでのディスクリートアラームの出力はメッセージの設定で制御されています。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、ディスクリートアラーム"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。


アラームロギング - WinCC Configuration Studio
ファイル(F) 編集(E) ビュー(V) ツール(L) ヘルプ(H)
アラームロギング
メッセージ
エラー
アラーム
警告
失敗
システム、確認を
システム、確認な
メッセージブロック
メッセージグループ
システムメッセージ
限界モニタリング
ASメッセージ
メッセージ [アラーム]
検索
番号
メッセージタグ
メッセージ
メッセージクラ
メッセージタイ
メッセージテキス
エラー
1 1 Inflow_Valve 2 エラー アラーム Valve_open Valve
2 2 Inflow_Valve 3 エラー アラーム Valve_closed Valve
3 3 Inflow_Valve 4 エラー アラーム Valve_inop Valve
メッセージ
プロパティ
準備完了
NUM
日本語 (日本)
テーブル: 3 メッセージ
100 %

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。

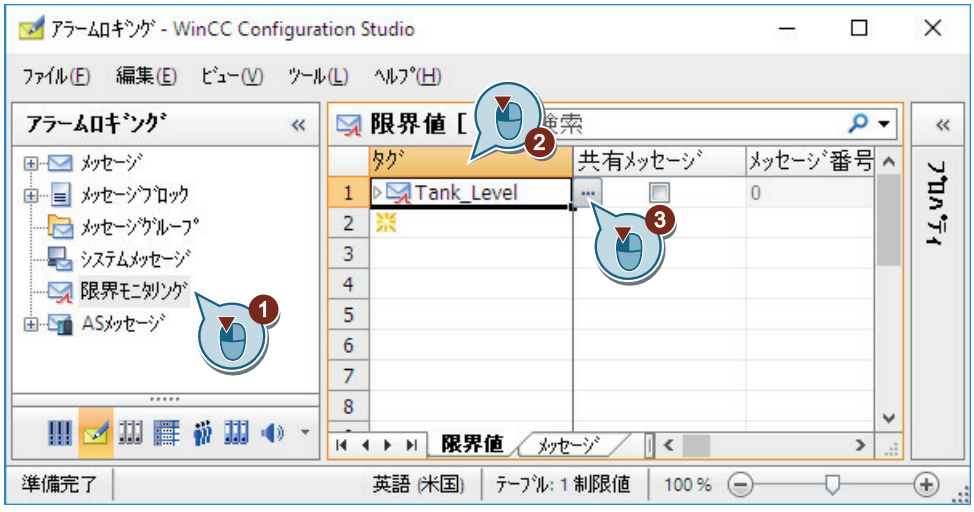
"Quick_Start"プロジェクトで、内部タグ"Tank_Level"の上限値と下限値を定義します。

必要条件

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

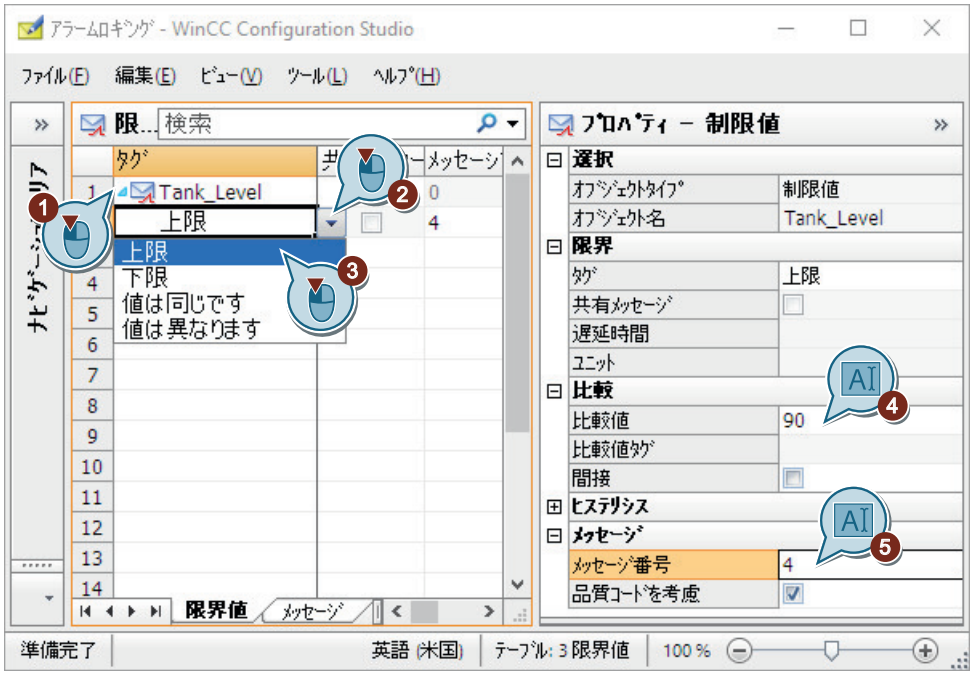
手順

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



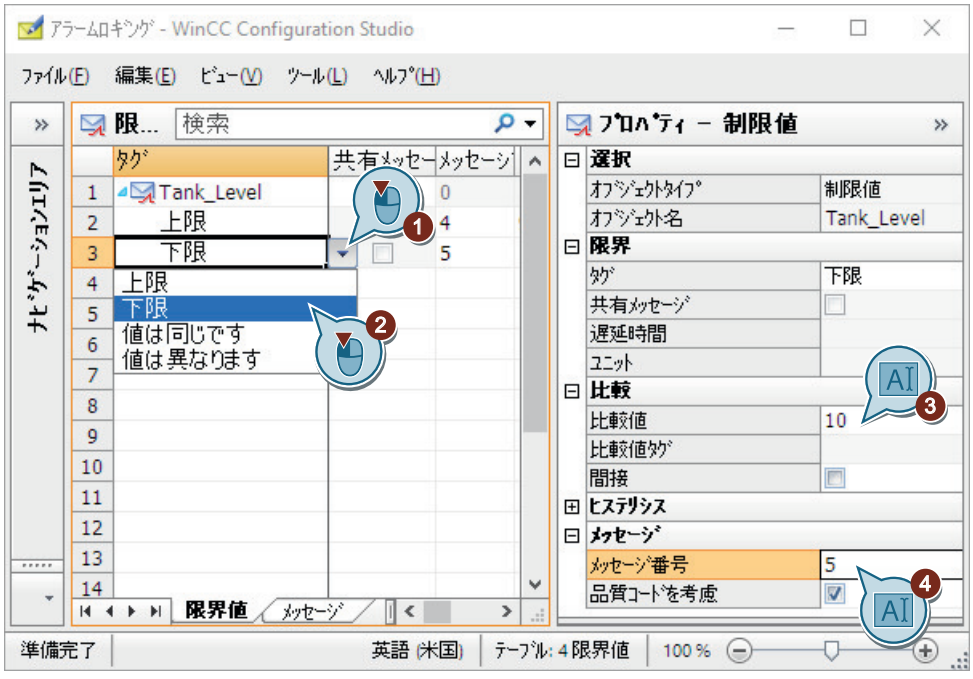
3. エントリを開くには、タグ名の前にある矢印をクリックします。
   テーブルペインのタグの下で[上限]エントリを選択します。

4. 上限値のプロパティを[プロパティ - 制限値]エリアまたはテーブルエリアに入力します。新しい、未使用のメッセージ番号を使用します。例、"4"。



5. 2 番目の制限値のデータを入力するには、タグの下の次の行で[下限]エントリを選択します。

6. 下限値のプロパティを[プロパティ - 制限値]エリアまたはテーブルエリアに入力します。新しい、未使用のメッセージ番号を使用します。例、"5"。

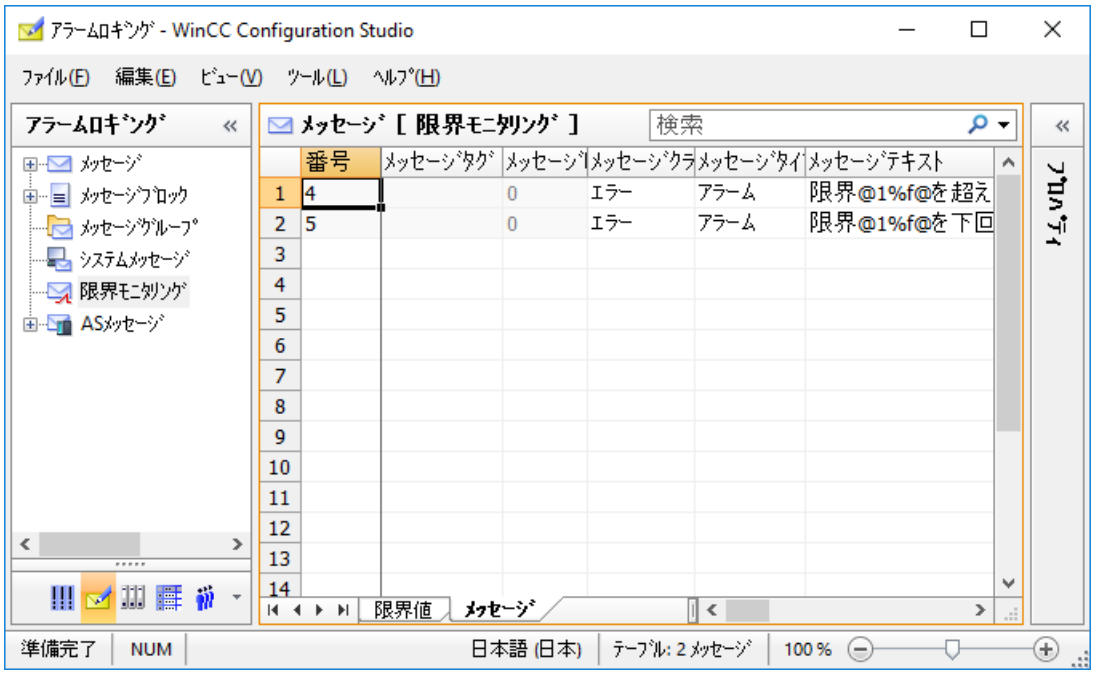


結果

内部タグ"Tank_Level"の制限値が定義されています。

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

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



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。

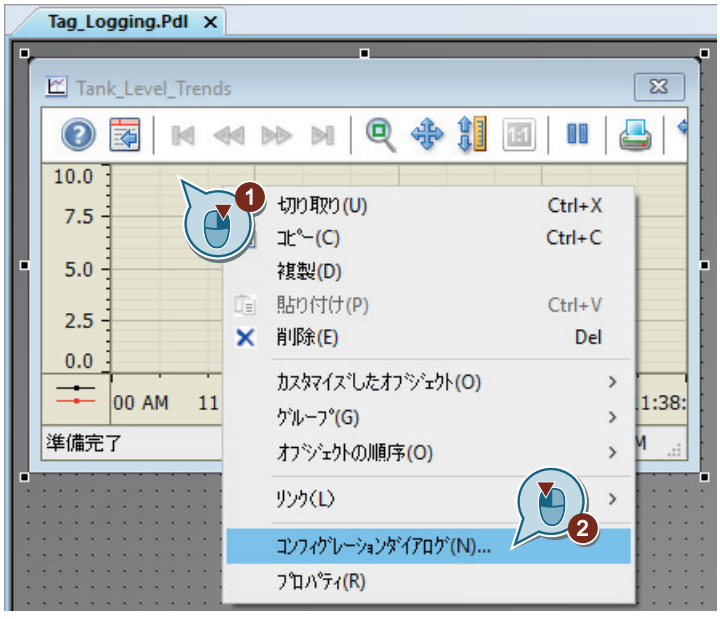
割り付けられたメッセージが、ランタイムで制限値違反のあるトレンド値に対するシンボルおよびツールヒントとして表示されます。

必要条件

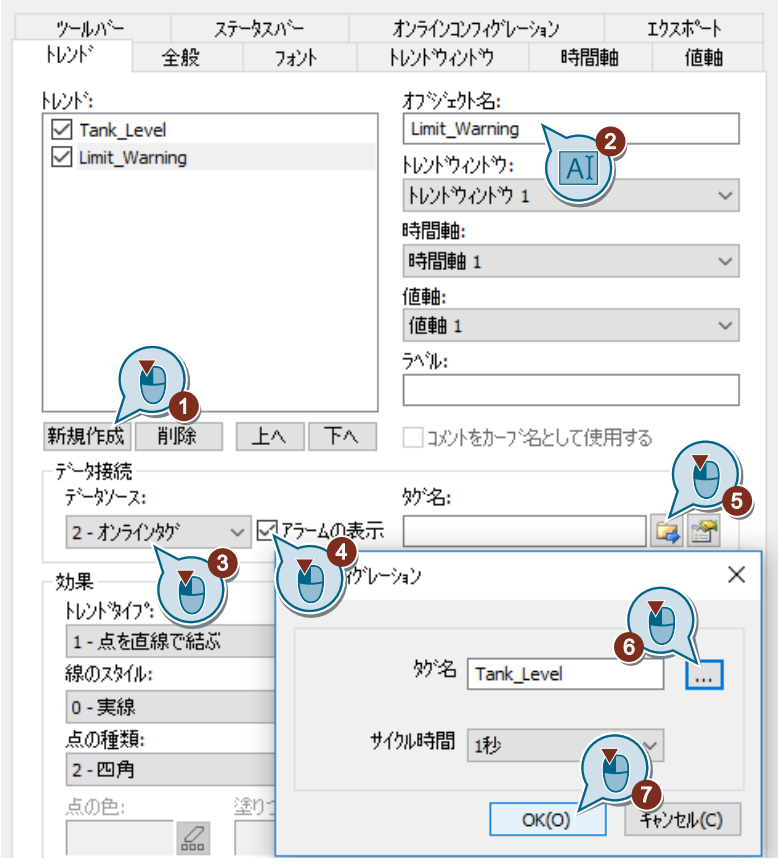
- グラフィックデザイナで"Tag_Logging.pdl"画像が開かている。
- アラームロギングで制限値のモニタリングを設定している。
  "限界値の設定 (ページ 194)"

手順

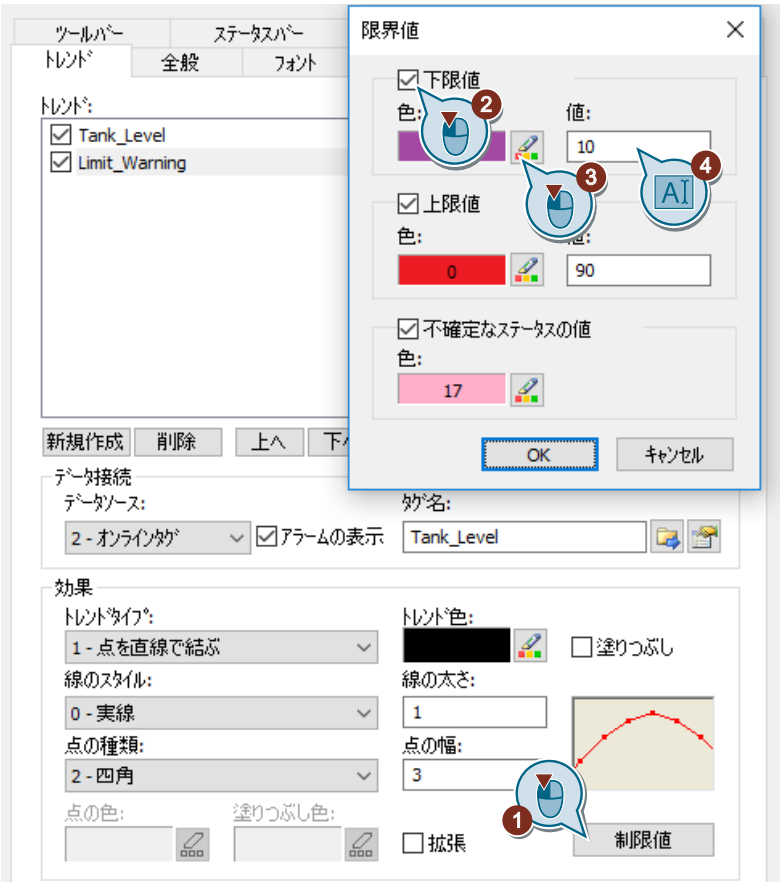
1. WinCC OnlineTrendControl [Tank_Level_Trends]を選択します。
2. ショートカットメニューを使用して、[WinCC オンライントレンドコントロールのプロパティ]設定ダイアログを開きます。



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



4. 下限のトレンドの色変更を設定します。



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

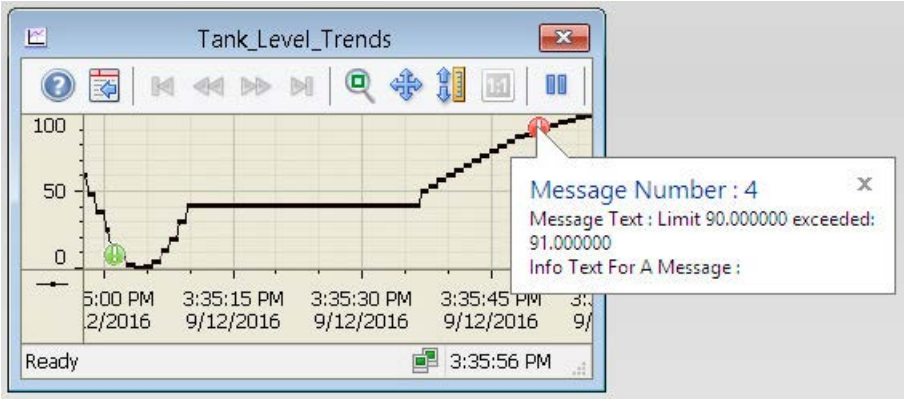
5. 上限および不特定なステータスの値のトレンドの色変更を設定します。

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しなくなると、メッセージは[発信]になります。
- ユーザーがメッセージを確認すると、メッセージは"確認済み"になります。

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

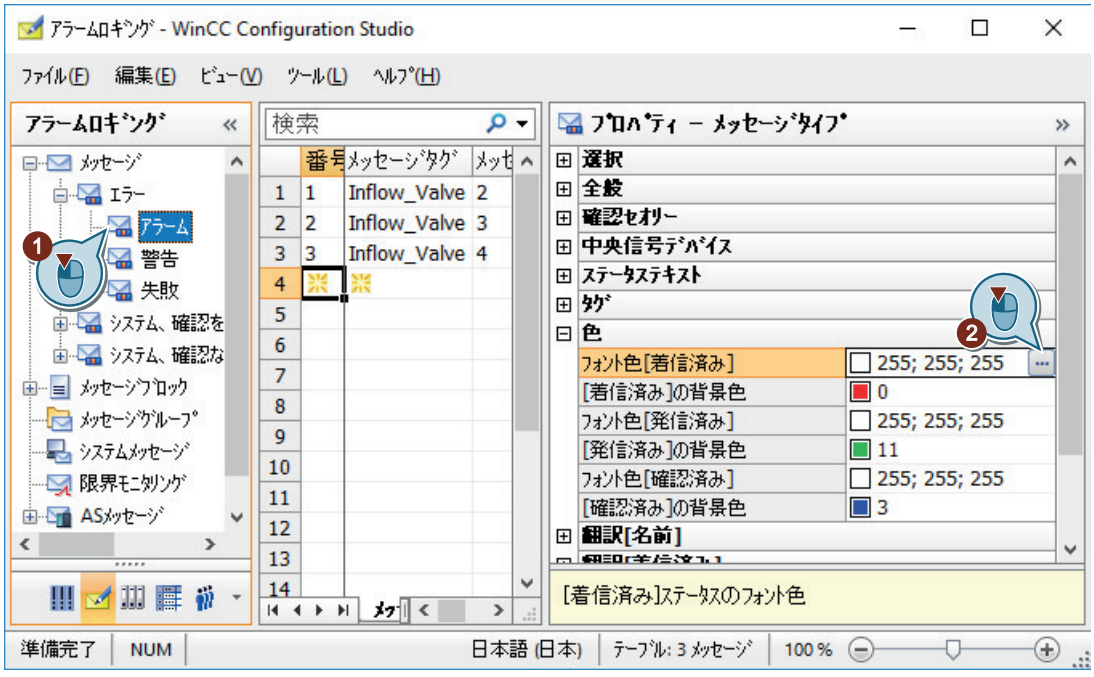
"Quick_Start"プロジェクトで、3 つの基本ステータスに対して異なるテキスト色と背景色を定義します。この定義は、メッセージクラス[エラー]のメッセージタイプ[アラーム]全体に適用されます。これは、設定を"Quick_Start"プロジェクトの全てのメッセージに適用します。

必要条件

- [アラームロギング]エディタが開いている。

手順

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



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

結果

メッセージステータス[着信]、[発信]および[確認済み]の表示色が定義されています。ランタイム中に、メッセージは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"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]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]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。

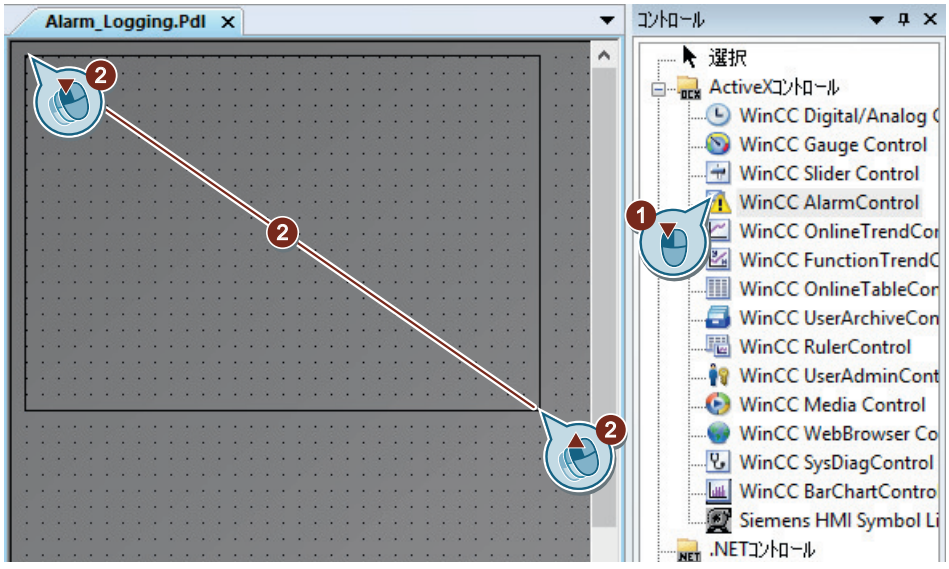
ランタイムで、メッセージはこれらのメッセージブロックで構成されます。

必要条件

- "Quick_Start"プロジェクトが開いている。

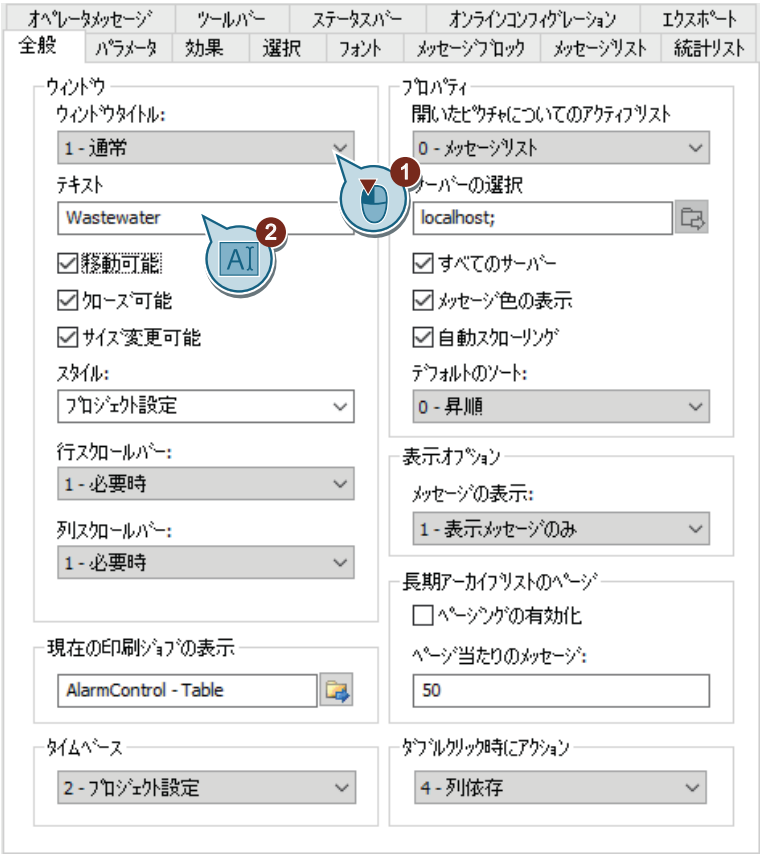
手順

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]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。

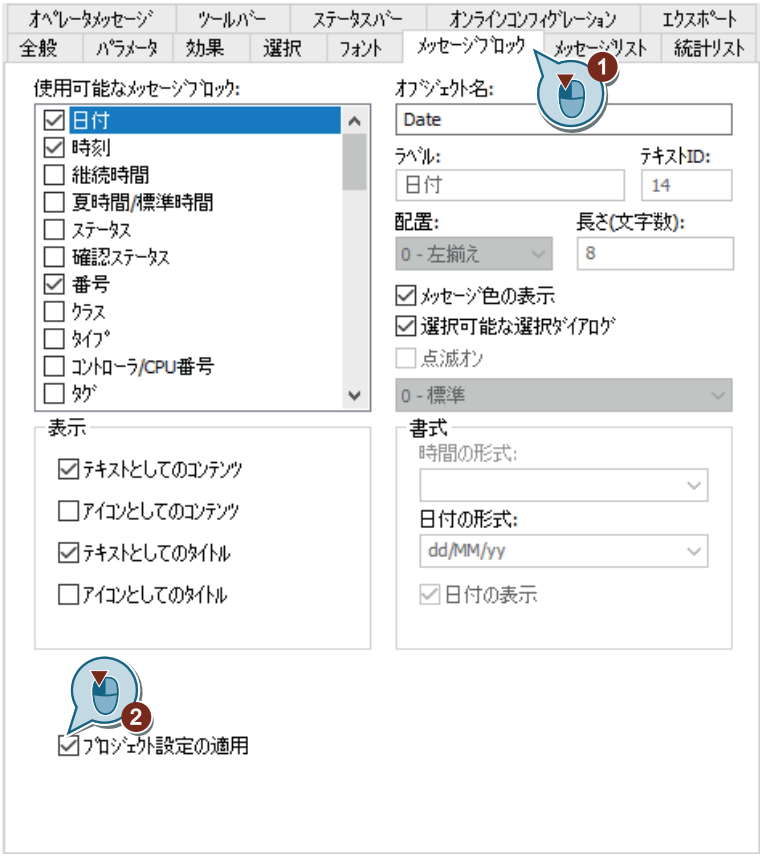


[WinCC AlarmControl プロパティ]ダイアログが開きます。

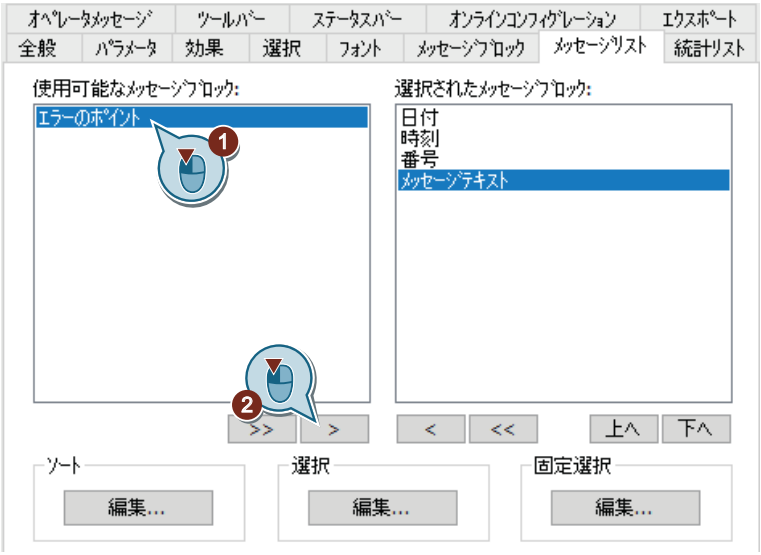
3. メッセージウィンドウの名前を指定します。



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。



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。

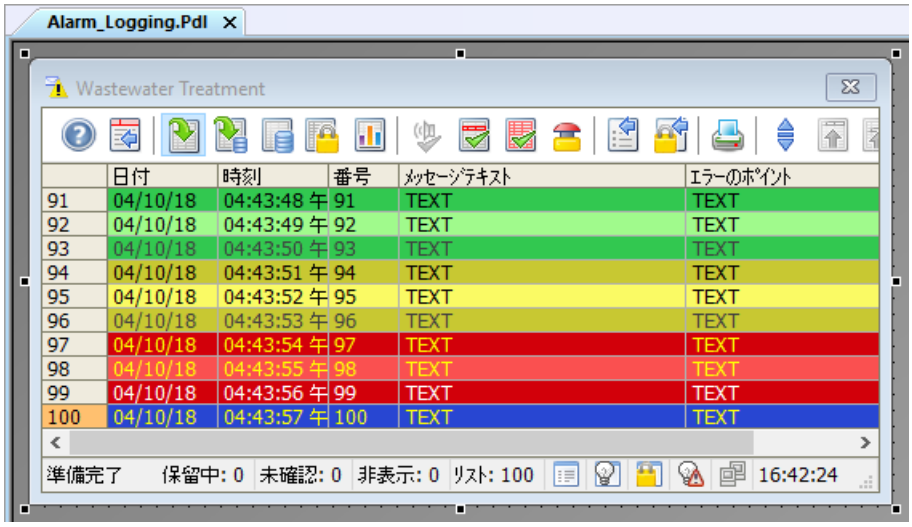
"Quick_Start"プロジェクトで設定したメッセージはランタイム中にこのウィンドウで表示されます。

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。



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

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

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

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

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

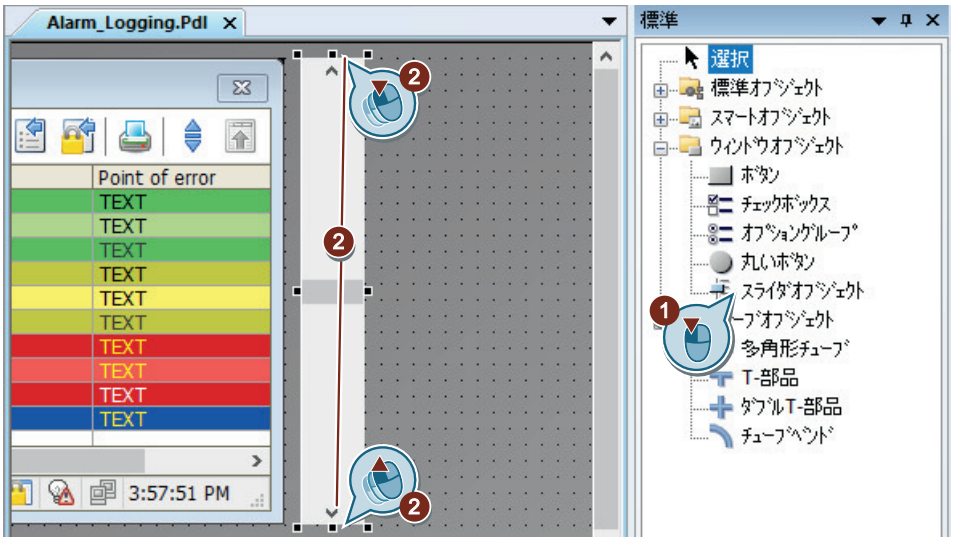
"Quick_Start"プロジェクトに対して、スライダオブジェクトのプリセットプロパティを変更します。[オブジェクトプロパティ]ウィンドウで、スライダオブジェクトの新しい名前を入力し、その高さを定義します。

必要条件

- "グラフィックデザイナ"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。

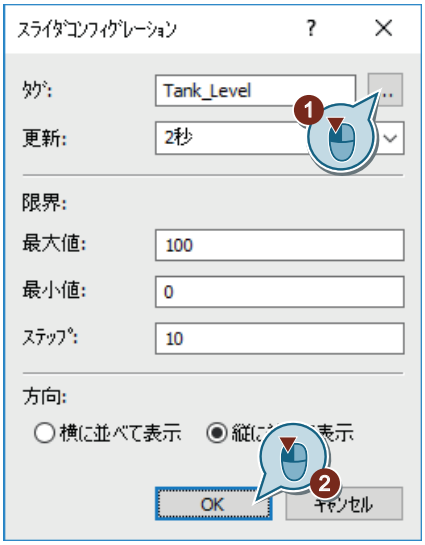
手順

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。

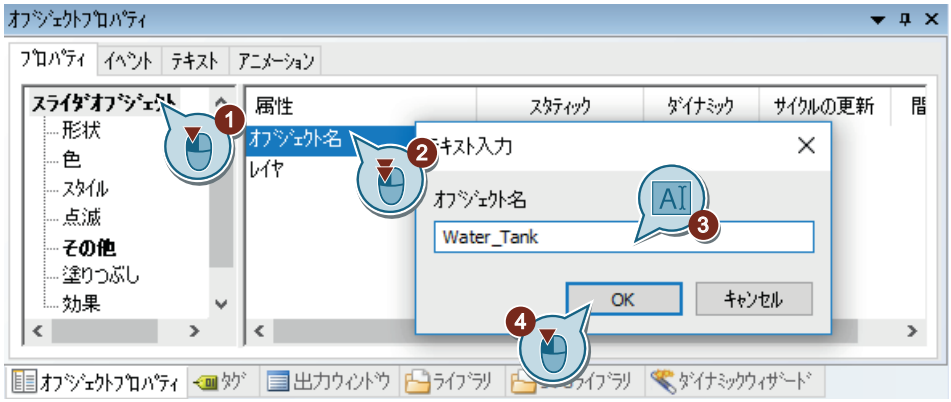


[スライダ設定]ダイアログが開きます。

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"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。

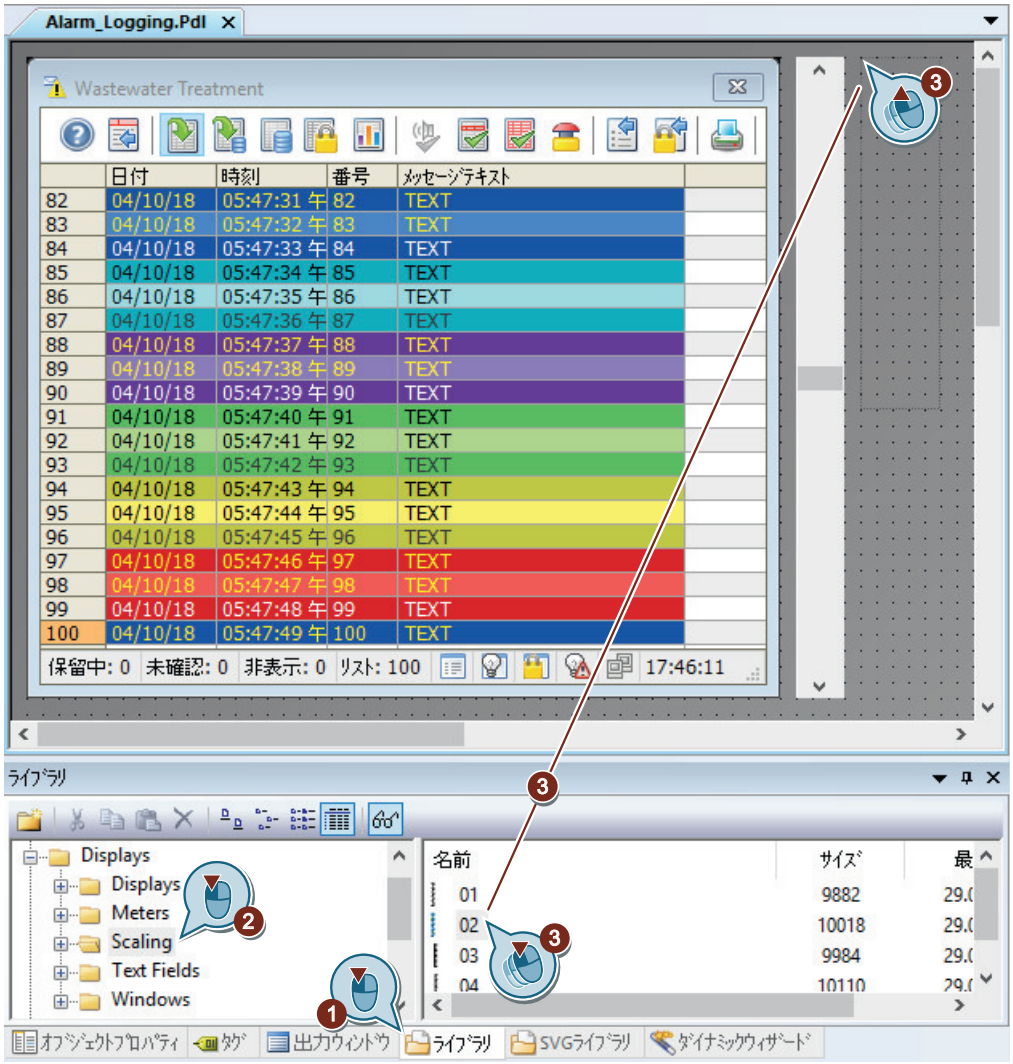
このスケールによって、スライダオブジェクト"Water_Tank"が想定できる値を表示します。スケール上のラインは、スライダオブジェクトの操作ステップに対応します。

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。

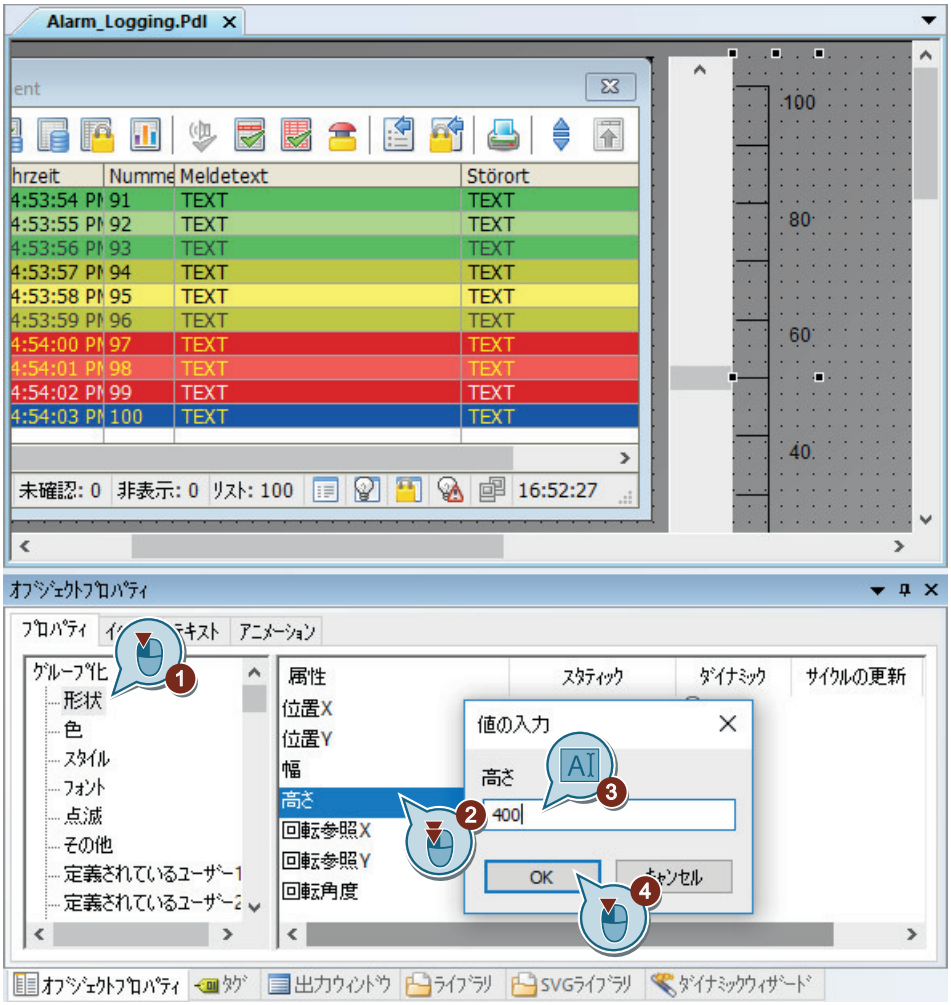
手順

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。

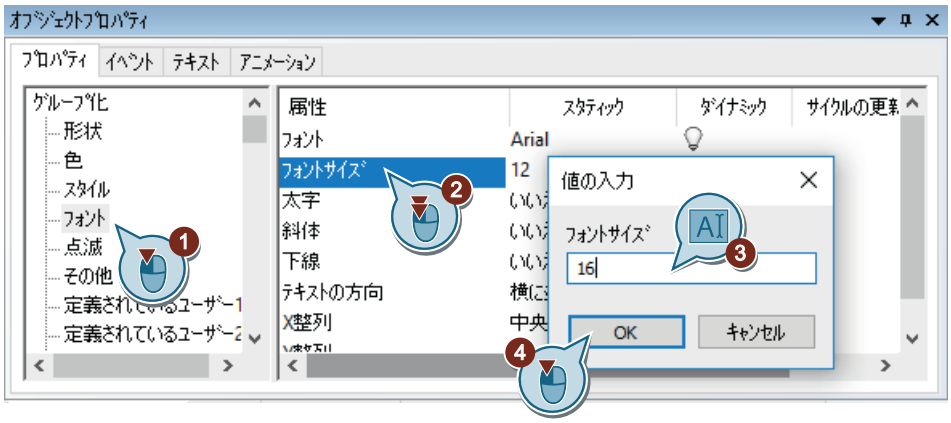


スケールは、プロセス画像"Alarm_Logging.pdl"に表示されます。

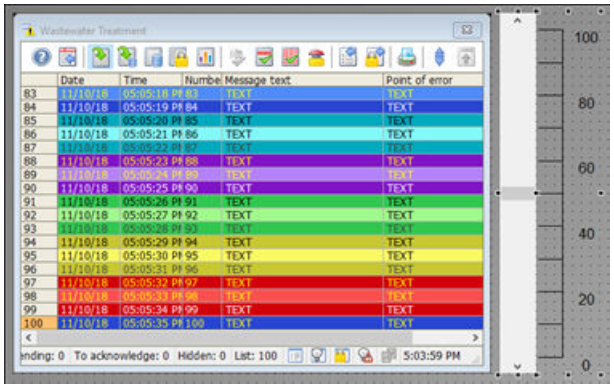
3. [オブジェクトプロパティ]に切り替えます。
   スケールの高さとして、[400]と定義します。



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。

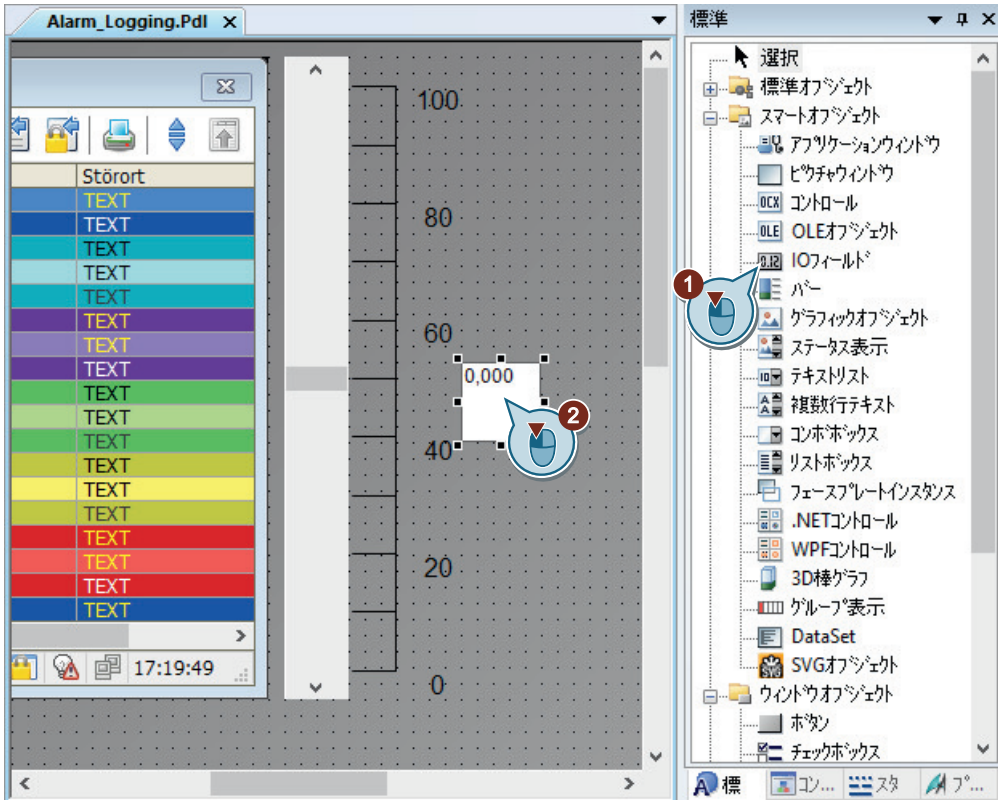
I/O フィールドは、ランタイム中にバイナリ値を内部[Inflow_Valve]タグに転送するために使用されます。

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。

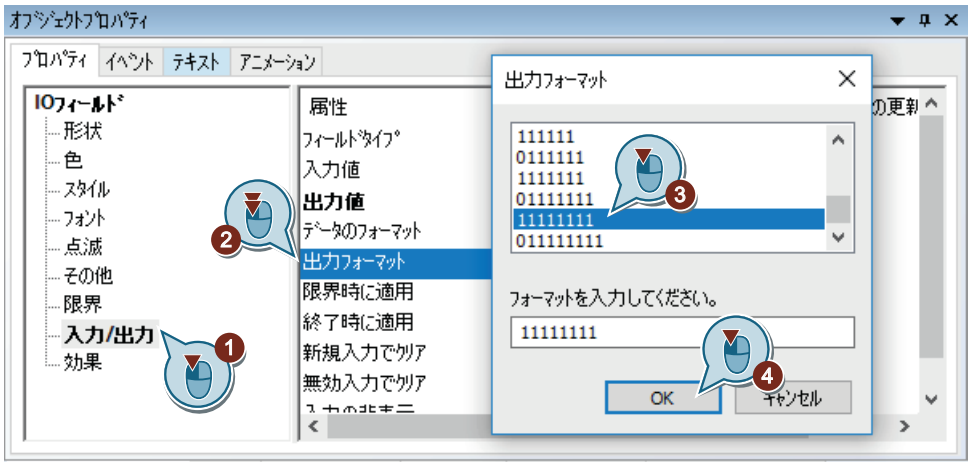
2. I/O フィールドを内部タグ"Inflow_Valve"にリンクします。

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]を閉じます。

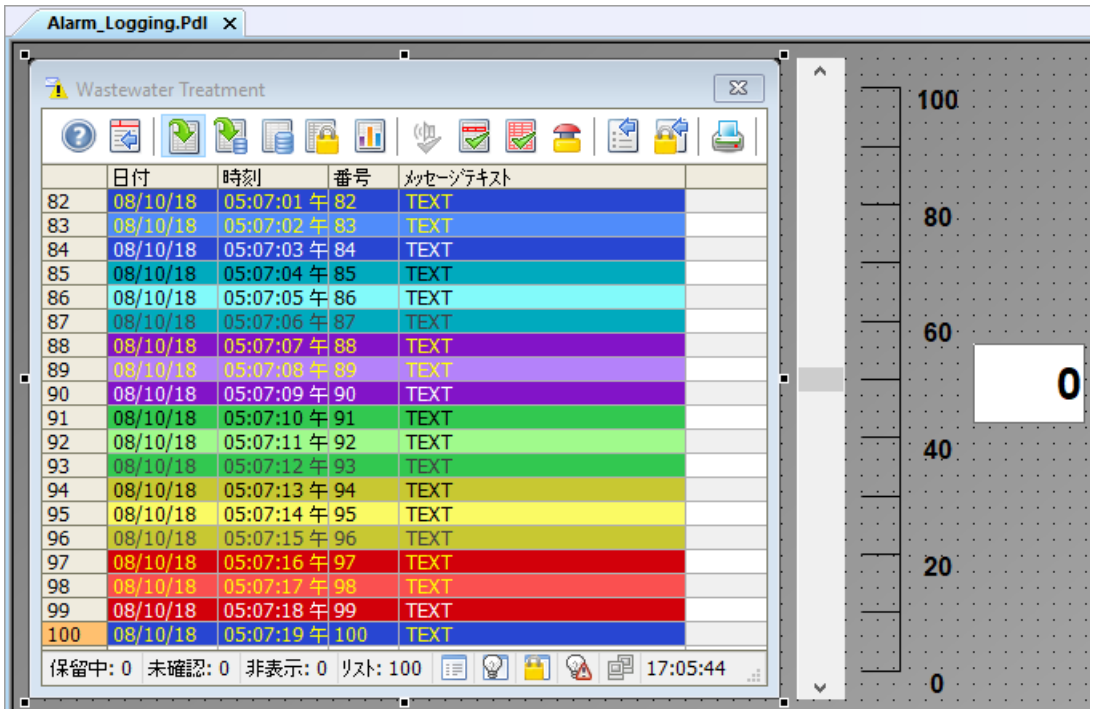
結果

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。


Alarm_Logging.Pdl
Wastewater Treatment
日付
時刻
番号
メッセージテキスト
保留中: 0
未確認: 0
非表示: 0
リスト: 100
17:05:44
100
80
60
40
20
0
0

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。

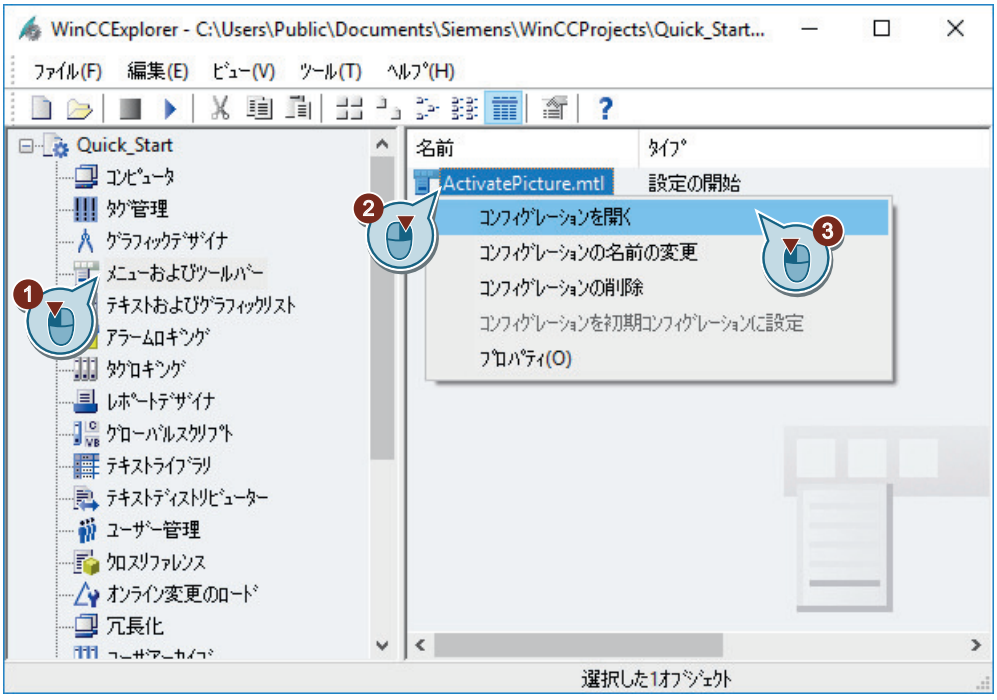
変更するプロセス画像の名前を、[ユーザーデータ]フィールドに入力します。

必要条件

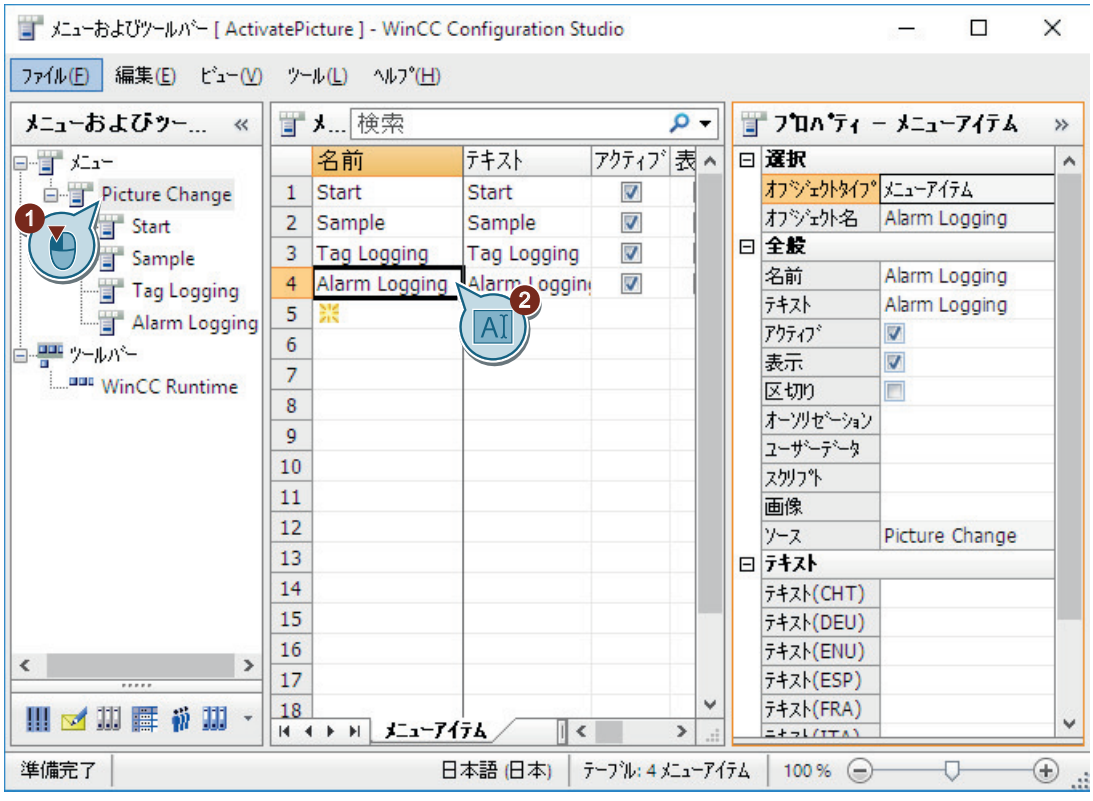
- プロシージャ"ActivatePicture(ByVal PictureName)"が作成されている。
- カスタマイズされた[画像変更]メニューがプロセス画像"START.pdl"、"SAMPLE.pdl"および"Tag_Logging.pdl"用に作成されていること。

手順

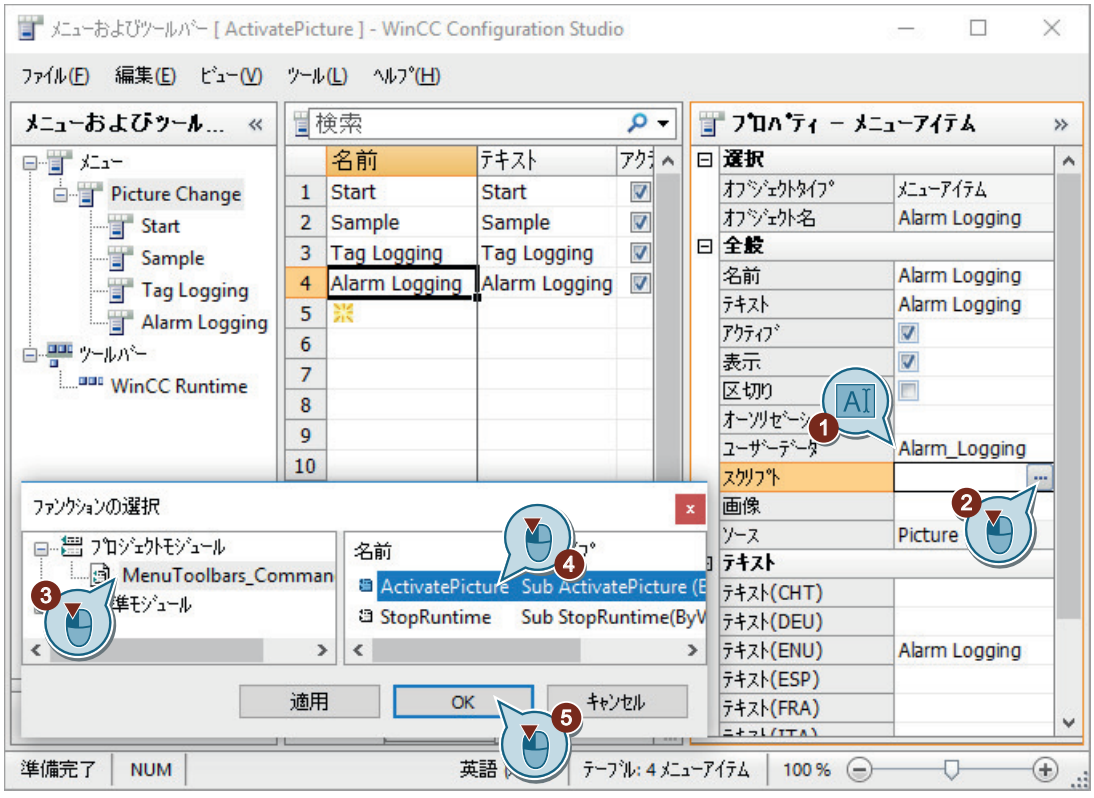
1. 保存された設定[ActivatePicture.mtl]を[メニューとツールバー]エディタで開きます。



2. [アラームロギング]メニューコマンドを作成します。



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



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

結果

[アラームロギング]メニュー項目を[画像変更]メニューに追加しました。

ランタイムでメニューエントリを使用して、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。

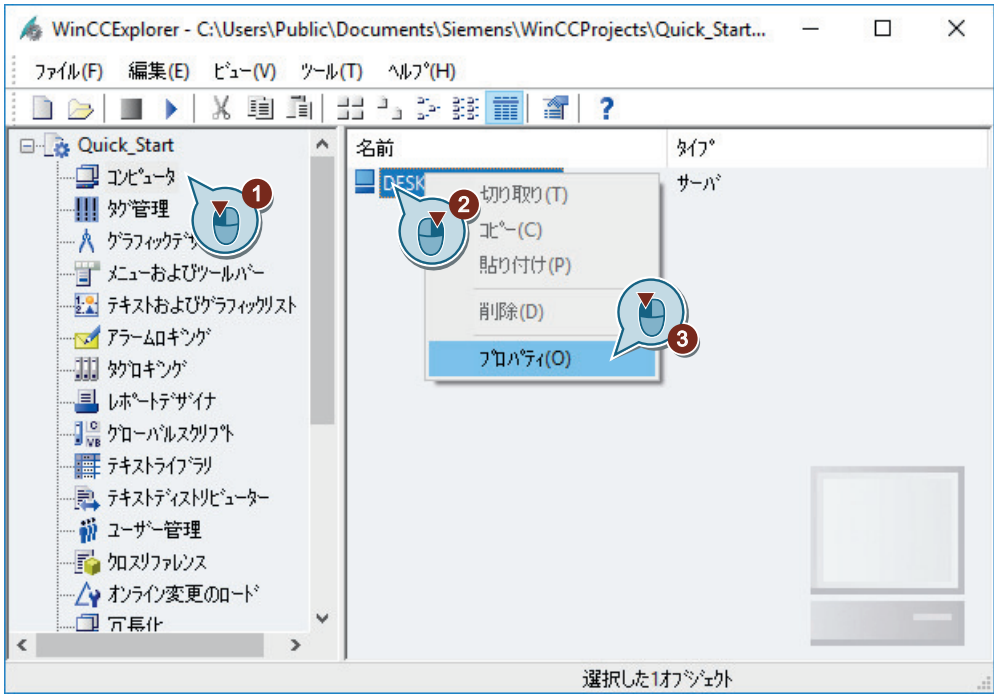
"Alarm_Logging.pdl"プロセス画像を、ランタイムウィンドウの開始画像として設定します。

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。

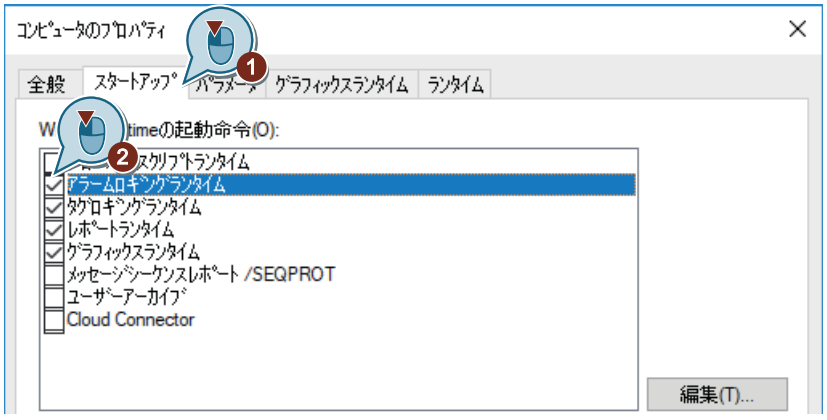
手順

1. [コンピュータのプロパティ]ダイアログを開きます。

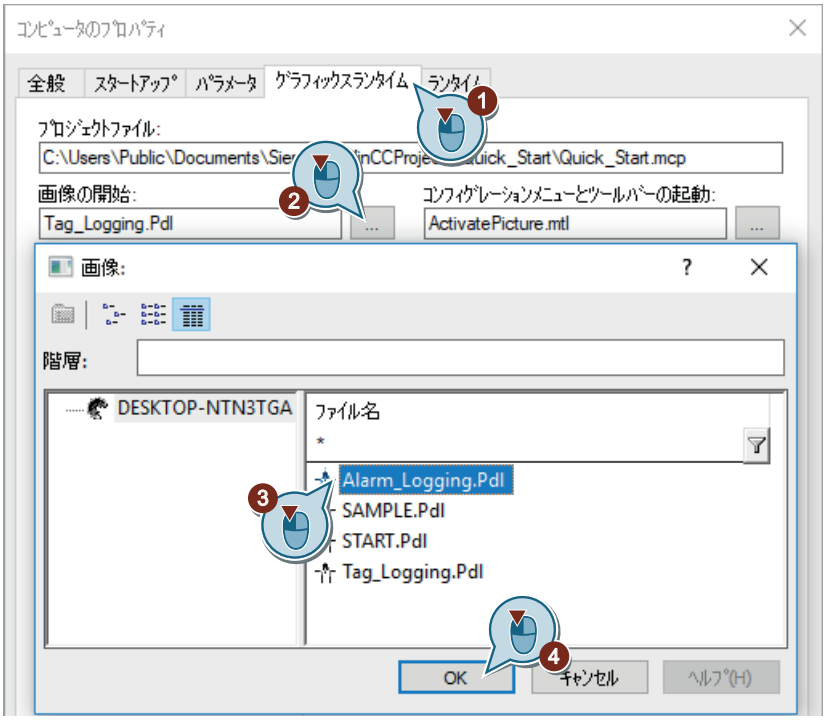


[コンピュータのプロパティ]ダイアログが開きます。

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。



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。

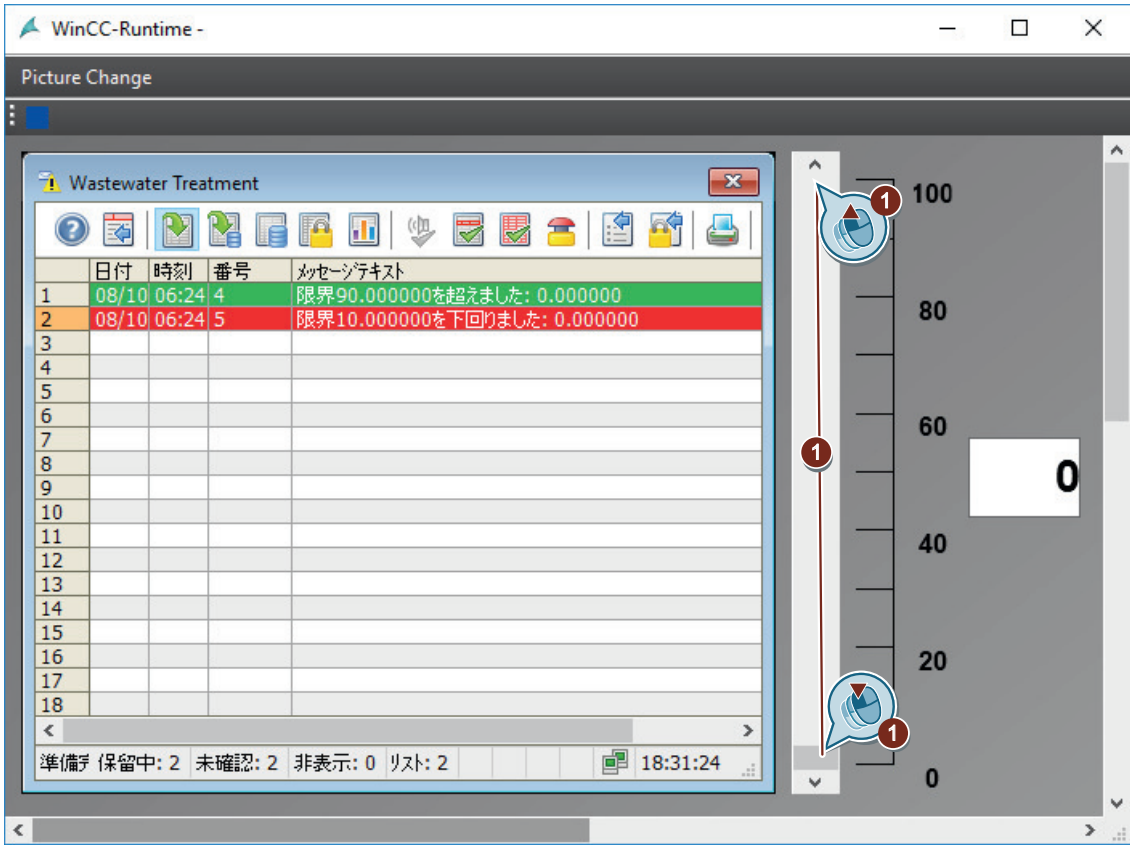
メッセージウィンドウのメッセージの表示色は、メッセージステータスによって変化します。

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。

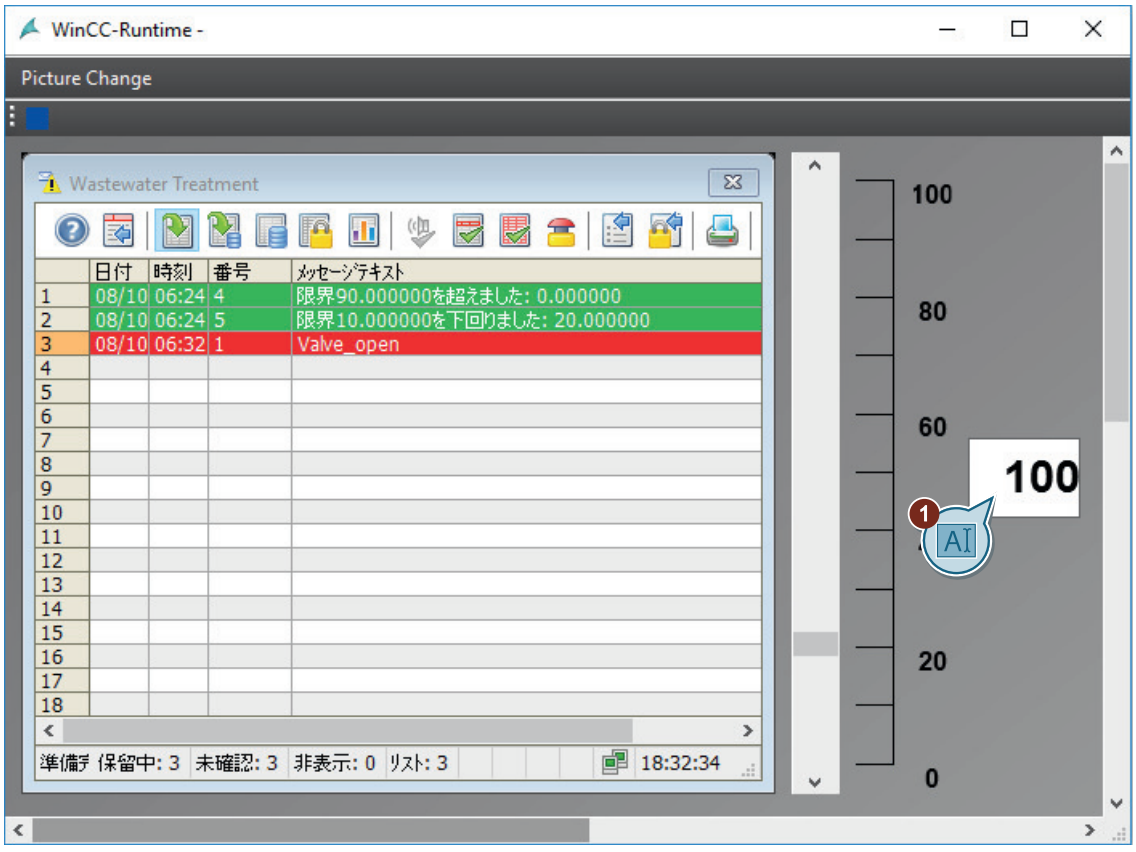
手順

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_Logging.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。

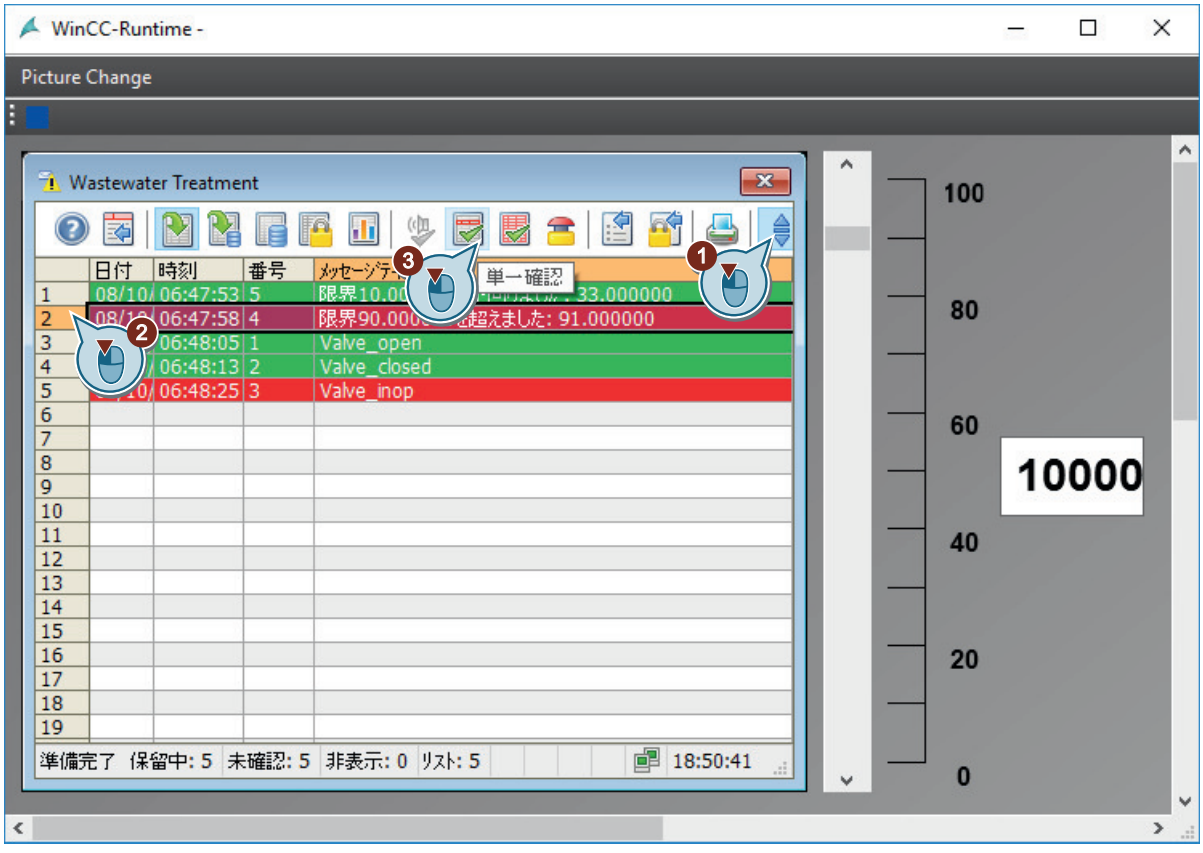


タグ値の 2 番目のビットが設定されます。
ディスクリートアラーム"Valve_open"が表示されます。

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。

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。

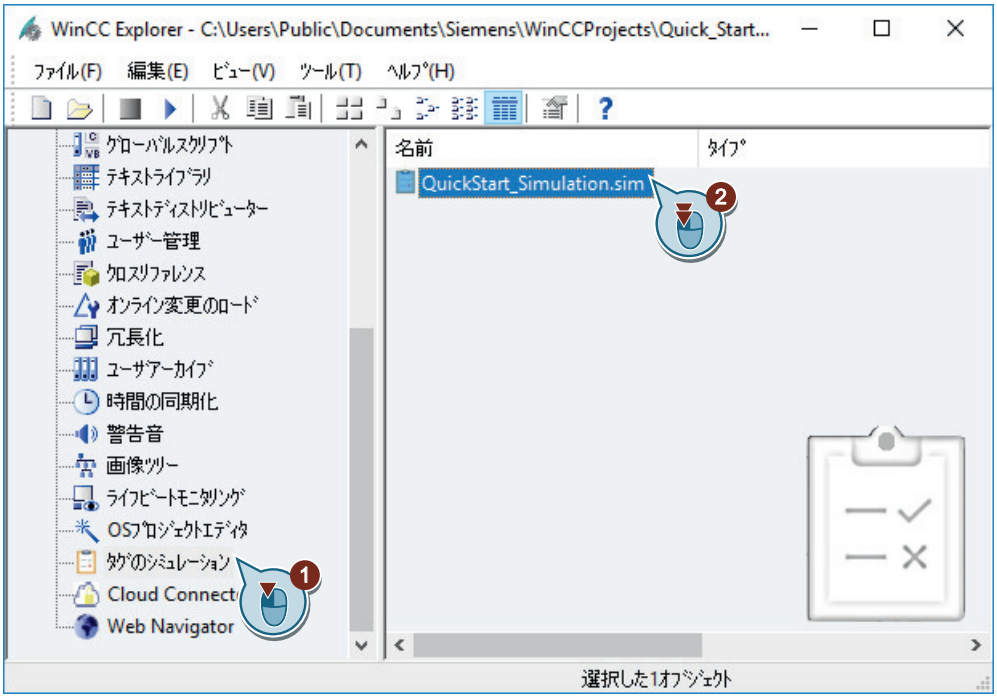
メッセージが、メッセージウィンドウに表示されます。メッセージステータスは、異なる表示色によってマークされます。

必要条件

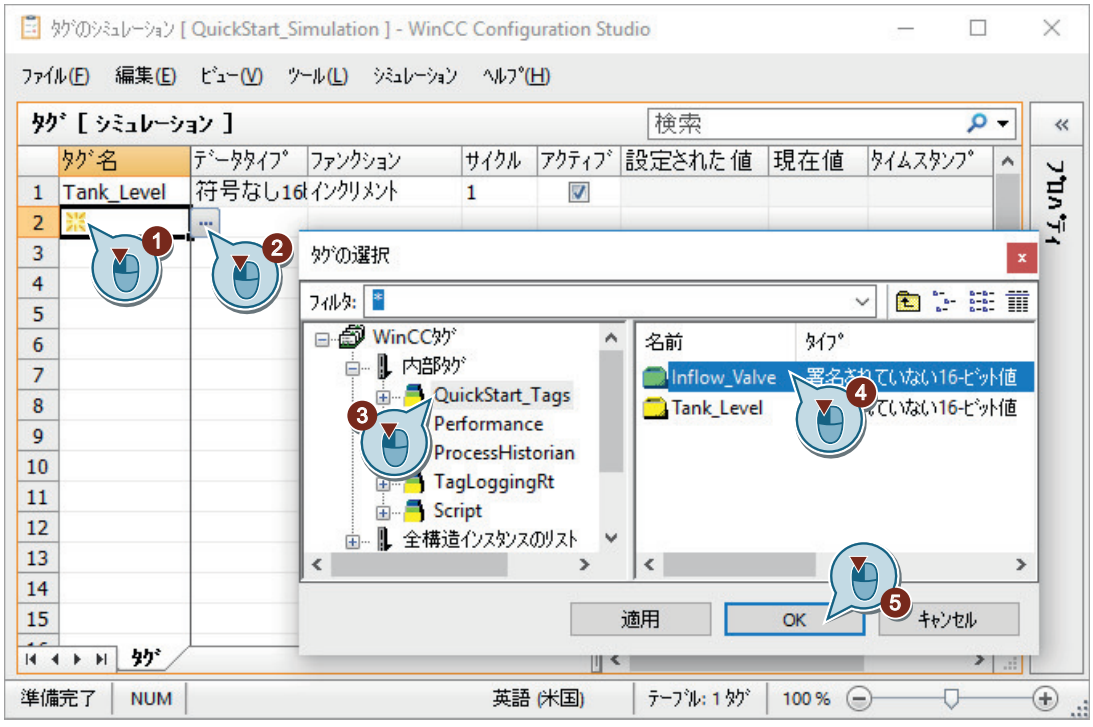
- ランタイムプロパティが定義されている。

手順

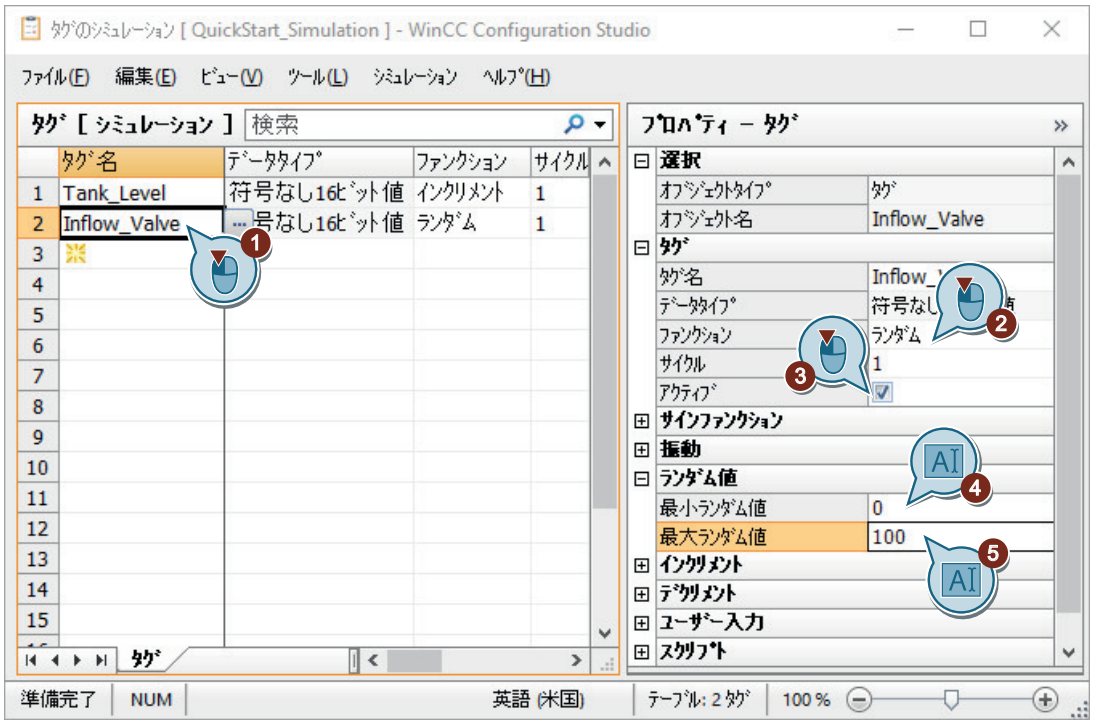
1. [QuickStart_Simulation.sim]設定を[タグシミュレーション]エディタで開きます。



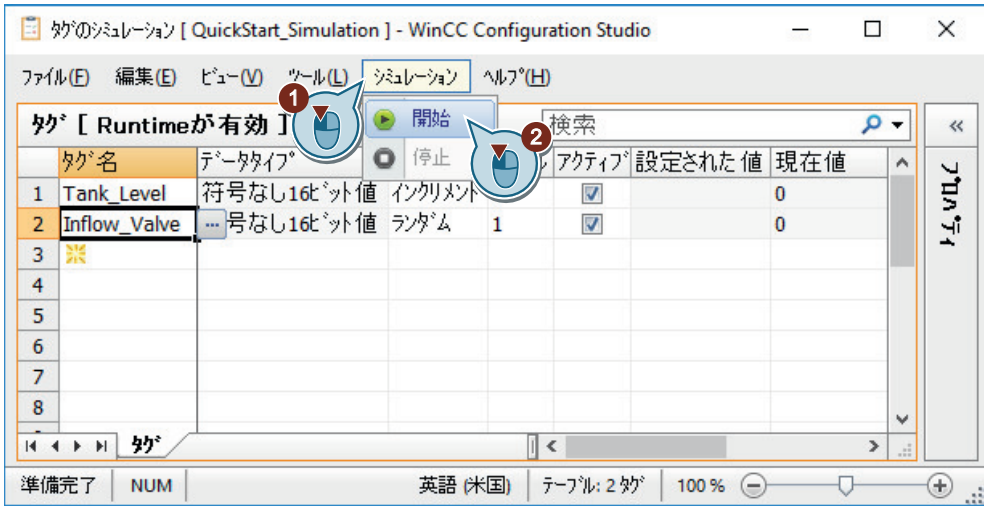
2. タグ選択ダイアログを開き、内部タグ[Inflow_Valve]をタググループ[QuickStart_Tags]から選択します。



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

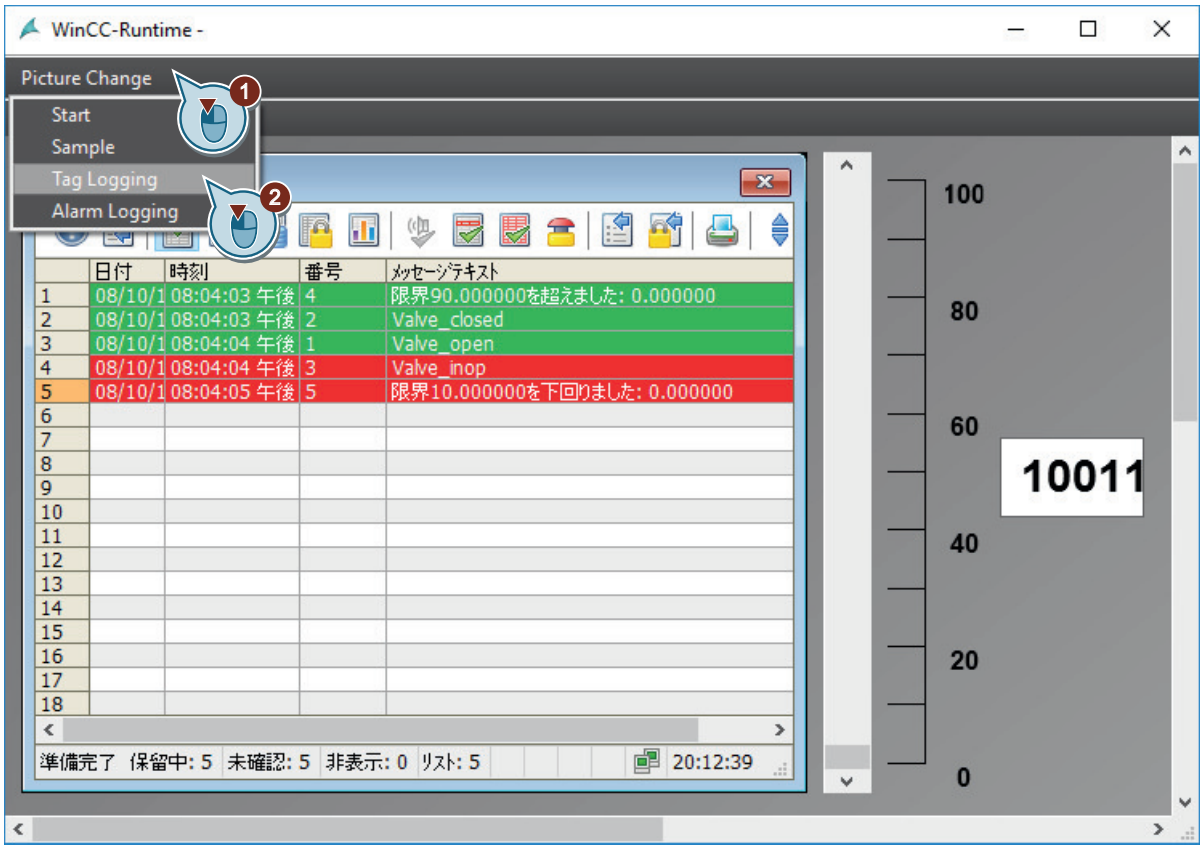


4. メニューバーで[ファイル] > [保存]コマンドを使用して設定を保存します。
5. ▶ボタンを使用して、WinCC エクスプローラで WinCC Runtime を有効にします。
6. [タグシミュレーション]エディタでシミュレーションを起動します。

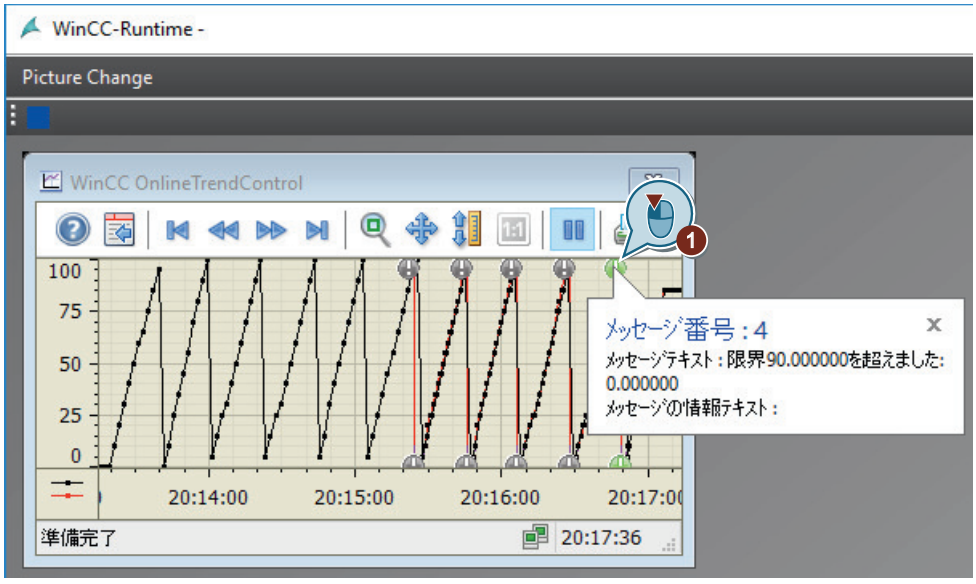


7. "Alarm_Logging.pdl"プロセス画像で、シミュレーション値の出力を確認します。
シミュレーション値の変化が、メッセージウィンドウにどのように影響するかを監視します。

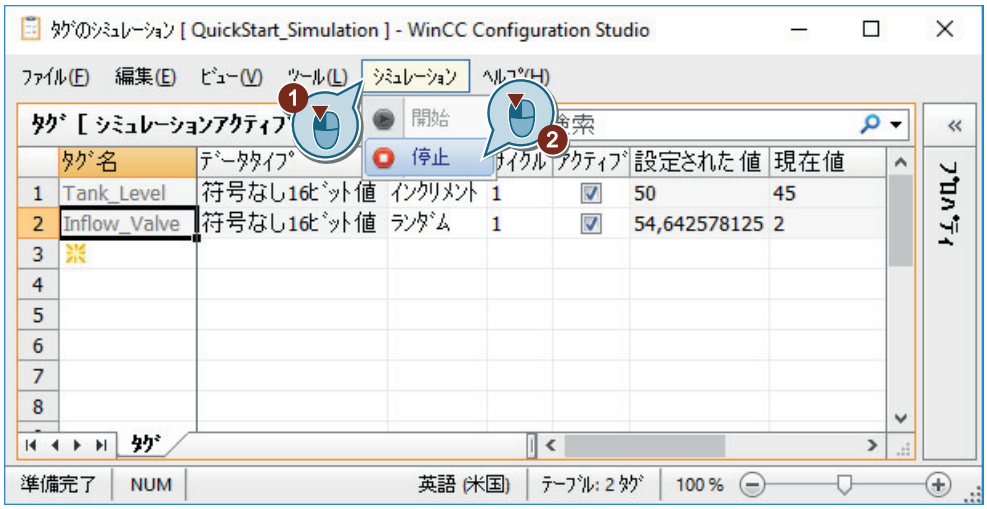
8. "Tag_Logging.pdl"プロセス画像に切り替えます。



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



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]エディタのライブラリ

[グラフィックデザイナ]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

[グラフィックデザイナ]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]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

[グラフィックデザイナ]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, 167

[

[テーブル]ウィンドウ, 131
　WinCC OnlineTableControl, 131
　設定, 131
[トレンド]ウィンドウ, 124
　WinCC OnlineTrendControl, 124
　設定, 124

A

Alarm_Logging.pdl, 205

I

I/O フィールド, 66, 71, 205
　ダイナミック化, 71, 216
　フォントプロパティの定義, 216
　更新の定義, 71
　出力フォーマットの定義, 71, 216
　挿入, 71, 216
　入力の制限, 71

M

MPI, 27, 32

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, 27
SAMPLE.pdl, 52, 64
　作成, 52
　編集, 64
START.pdl, 52
　作成, 52
　編集, 64
SVG オブジェクト
　ダイナミック化, 67
SVG ライブラリ, 56

T

Tag_Logging.pdl, 124
Tag_Logging.rpl, 149

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, 205
WinCC OnlineTableControl, 124, 131
　挿入, 131
WinCC OnlineTrendControl, 124
　メッセージの表示, 197
　挿入, 124
WinCC Runtime, 13
　プロパティの指定, 92, 139, 171, 225
　開始画像, 139
WinCC エクスプローラ, 13
WinCC タグシミュレータ, 99, 142, 174, 232
　シミュレーションタイプの定義, 232
　開始, 99, 142, 232
Windows オブジェクト, 55

ア

アーカイブ, 111, 118
　プロセス値アーカイブ, 111, 118
　値, 111
アーカイブサイクル, 116
　時間の割り付け, 121
　時間の設定, 116
アーカイブシステム, 112
　タグロギング, 112
　タグロギングランタイム, 112
アーカイブタグ, 118
　作成, 118
　名前の変更, 121
アイコン, 11
アドレス指定, 37
　プロセスタグ, 37

アナログアラーム, 181
ランタイムでの表示, 197
制限値の設定, 194
設定, 194
アラームウィンドウ, 205
WinCC AlarmControl, 205
設定, 205
アラームロギング, 181, 182
開始, 184
アラームロギングランタイム, 182
有効, 225

ウ

ウィンドウ属性, 92
有効, 92

エ

エディタ, 13, 26, 50
アラームロギング, 13, 184
グラフィックデザイナ, 50, 55
タグロギング, 112, 114
タグ管理, 26
ページレイアウトエディタ, 152, 154
レポートデザイナ, 13, 146, 149

オ

オブジェクト, 49, 50, 56, 152
I/O フィールド, 205, 216
WinCC AlarmControl, 205
WinCC OnlineTableControl, 124, 131
WinCC OnlineTrendControl, 124
Windows オブジェクト, 55
スケール, 213
スタティックオブジェクト, 49, 152
スタティックテキスト, 61
スマートオブジェクト, 55
スライダオブジェクト, 205, 210
ダイナミックオブジェクト, 49
タグテーブル, 157
プロジェクト名, 161
挿入, 55
操作可能なオブジェクト, 49
標準オブジェクト, 55
オブジェクトパレット, 55
Windows オブジェクト, 55
コントロール, 55
スマートオブジェクト, 55
標準オブジェクト, 55
オブジェクトプロパティ, 67, 71, 210, 213
印刷余白, 154
オブジェクトプロパティダイアログ, 157

カ

カバーシート, 152
出力の無効化, 154

ク

グラフィックシステム, 50
グラフィックデザイナ, 50
グラフィックランタイム, 50
グラフィックデザイナ, 50, 55
オブジェクトパレット, 49
プロセス画像の設定, 49
ライブラリ, 55
グラフィックランタイム, 50
有効, 92

コ

コントロール, 55
WinCC AlarmControl, 205
WinCC OnlineTableControl, 124, 131
WinCC OnlineTrendControl, 124
コンポーネント, 13
WinCC, 13
アーカイブシステム, 112
グラフィックシステム, 50
メッセージシステム, 182
レポートシステム, 146

シ

システムオブジェクト, 152
プロジェクト名, 161
ページ番号, 164
システムダイアログ
使用, 104
システムブロック, 186
シミュレーション, 99
WinCC タグシミュレータ, 99, 142, 174, 232
シミュレーションタイプ, 99, 142, 174, 232
開始, 99, 232
終了, 232
シミュレーションタイプ, 232

ス

スケール, 43, 213
サイズの定義, 213
フォントサイズの定義, 213
プロセスタグ, 43
挿入, 213
スタティックテキスト, 61
フォントサイズを変更する, 61
挿入, 61
スタティックレベル, 152
開く, 161
ステータス, 190
バルブ, 181
メッセージ, 203
スマートオブジェクト, 55
スライダオブジェクト, 205
サイズの定義, 210
ダイナミック化, 210
色の定義, 210
挿入, 210
名前の定義, 210

タ

ダイナミックオブジェクト, 49, 152
ダイナミック化, 66, 67
I/O フィールド, 71, 216
スライダオブジェクト, 210
タイプ変換, 37
プロセスタグ, 37
タイマ, 116
作成, 116
設定, 116
タグ, 26
アーカイブタグ, 118
スケーリング, 43
プロセスタグ, 26, 37, 43
リンク, 67
作成, 37, 45
内部タグ, 26, 45
タググループ, 26
作成, 37
タグテーブル, 157
挿入, 157
タグロギング, 112
開始, 114
タグロギングランタイム, 112
有効, 139
タグ管理, 26
タググループ, 37
プロセスタグ, 26, 37
内部タグ, 26, 45
タグ接続, 66
直接タグ接続, 66

チ

チャンネル, 25
SIMATIC S7-1200、S7-1500 チャンネル, 27
挿入, 27
チャンネルユニット, 25, 32
MPI, 27
挿入, 27

テ

ディスクリートアラーム, 189
作成, 190
データタイプ, 37, 45
プロセスタグ, 37
符号なし 16 ビット値, 45
符号付き 16 ビット値, 37
データ交換, 26
プロセスタグ, 26
テキストフィールド, 61
テスト, 99
プロジェクト, 99, 142, 174, 232

フ

フォントサイズ, 61
変更, 61, 71, 213
フッター, 152
ページ番号の挿入, 164
編集, 164
プリンタ, 167
印刷ジョブの定義, 167
プロジェクト, 17, 96, 99
WinCC の起動, 18
シングルユーザープロジェクト, 18
テスト, 99, 142, 174, 232
作成, 18
無効化, 177
有効, 96, 142, 174, 228
プロジェクト設計ソフトウェア, 13
WinCC, 13
プロジェクト名, 161
サイズの変更, 161

再位置決め, 161
挿入, 161
プロセス, 13
モニタリング, 13, 181
視覚化, 13, 49, 50
操作, 13, 23, 25, 26
プロセスタグ, 26
アドレス指定, 37
スケーリング, 43
タイプ変換, 37
データタイプ, 37
作成, 37
プロセス画像, 49
Alarm_Logging.pdl, 205
SAMPLE.pdl, 52
START.pdl, 52
Tag_Logging.pdl, 124
ダイナミック化, 66
作成, 52
設定, 49, 124, 205
編集, 55, 64
保存, 61, 64
名前の変更, 52
プロセス値, 23, 26, 111
アーカイブ, 111, 116
ウィンドウ, 111
シミュレーション, 99, 142, 174, 232
入力, 116
プロセス値アーカイブ, 111
サイズの定義, 121
メモリの場所の定義, 121
作成, 118
値の出力, 145
編集, 121
保存, 121
プロセス値ブロック, 186

へ

ページレイアウト, 149, 152, 154
Tag_Logging.rpl, 149
カバーシート, 152
スタティックレベル, 152
ダイナミックレベル, 152, 157
フッター, 152
プロパティの指定, 154
ページサイズ, 154
ヘッダー, 152
レポート内容, 152, 157
印刷余白, 154
開く, 154
最終ページ, 152
作成, 149
編集, 152
保存, 164
名前の変更, 149
ページレイアウトエディタ, 146, 152
開始, 154
ページ番号, 164
サイズの変更, 164
再位置決め, 164
挿入, 164
ヘッダー, 152, 161
プロジェクト名の挿入, 161
編集, 161

マ

マニュアル類, 111
値, 145

メ

メッセージ, 181
アナログアラーム, 181
ディスクリートアラーム, 181
メッセージステータス, 203
メッセージステータスの色の定義, 203
確認, 228
制限値モニタリング, 194
設定, 181
保存, 194
メッセージクラス, 186
定義, 186
メッセージシステム, 182
アラームロギング, 182
アラームロギングランタイム, 182
メッセージステータス, 190, 203
メッセージブロック, 186
システムブロック, 186
プロセス値ブロック, 186
ユーザーテキストブロック, 186
定義, 186
表示の有効化, 205
メッセージリスト, 228

ユ

ユーザーテキストブロック, 186

ラ

ライブラリ, 55, 56
グローバルライブラリ, 56
プロジェクトライブラリ, 56
開く, 56
ランタイム, 13
ランタイムウィンドウ, 92
ウィンドウ属性, 92
開始画像, 92, 139, 171, 225
ランタイムコンポーネント, 13
アーカイブシステム, 112
グラフィックシステム, 50
メッセージシステム, 182
レポートシステム, 146
ランタイムシステムダイアログ, 104
ランタイムソフトウェア, 13
WinCC, 13
ランタイムデータ, 145, 146

リ

リニアスケーリング, 43
例, 43

レ

レイアウト, 146
レイヤ, 152
スタティックレベル, 152, 164
ダイナミックレベル, 152
レポートシステム, 146
レポートデザイナ, 146
レポートランタイム, 146
レポートデザイナ, 146
ページレイアウトエディタ, 146
レイアウト, 146
印刷ジョブ, 146
行レイアウトエディタ, 146
レポートランタイム, 146
有効, 171
レポート内容, 152, 157

ロ

ログ, 145, 146
カバーシート, 154
プロセスアーカイブからの値のロギング, 145
ログの内容の決定, 157
印刷, 177
最終ページ, 152

印

印刷, 177
プロセス値のログ, 177
印刷ジョブ, 145, 146, 167
@OnlineTableControl - 表, 167
ページレイアウトのリンク, 167
開く, 167
出力媒体の定義, 167
定義, 167

開

開く, 18, 52
[ページサイズ]ダイアログ, 154
オブジェクトプロパティダイアログ, 157, 164
スタティックレベル, 161
ダイナミックレベル, 157
プロジェクト, 18
プロセス画像, 52, 64
ライブラリ, 56
レポート内容ページ, 157
印刷ジョブ, 167
開始, 18
WinCC, 18
WinCC タグシミュレータ, 142
アラームロギング, 184
タグロギング, 114
ページレイアウトエディタ, 154
開始画像, 92, 139, 171
定義, 92, 225

確

確認, 228
メッセージ, 228

更

更新サイクル
I/O フィールド, 71

行

行レイアウトエディタ, 146

最

最終ページ, 152

作

作成, 32, 37, 45, 124
[テーブル]ウィンドウ, 131
[トレンド]ウィンドウ, 124
アナログアラーム, 194
アラームウィンドウ, 205
タイマ, 116
タググループ, 37
ディスクリートアラーム, 190
フッター, 164
プロジェクト, 17, 18
プロセスタグ, 37
プロセス画像, 52
プロセス値アーカイブ, 118
ページレイアウト, 149
ヘッダー, 161
接続, 32
内部タグ, 45

実

実例, 56
スケール, 213

取

取得サイクル, 116
時間の割り付け, 121
時間の設定, 116

出

出力, 145
プロセス値のログ, 145

色

色, 203
メッセージステータスの色の定義, 203

制

制限値, 194
下限値, 194
上限値, 194
定義, 194
制限値モニタリング, 194
ランタイムでのメッセージの表示, 197
制限値の設定, 194

接

接続, 25, 32
作成, 32

設

設定, 23, 55, 124
[テーブル]ウィンドウ, 131
[トレンド]ウィンドウ, 124
アラームウィンドウ, 205
プロセス画像, 55, 124, 205
プロセス値アーカイブ, 118, 121
メッセージ, 181, 189, 194
取得サイクルおよびアーカイブサイクルの時間, 116
通信, 23
設定コンポーネント, 50, 112, 146, 182
アーカイブシステム, 112
グラフィックシステム, 50
メッセージシステム, 182
レポートシステム, 146
設定データ, 145, 146

値

値, 23, 145
アーカイブ, 111
シミュレーション, 99
プロセス値, 23, 26, 111
出力, 145
値の範囲, 43
プロセスタグのスケーリング, 43

通

通信, 23
チャンネル, 25, 27
チャンネルユニット, 25, 27

プロセスタグ, 26, 37
接続, 25, 32
設定, 23
通信ドライバ, 25
SIMATIC S7-1200、S7-1500 チャンネル, 27

貼

貼り付け, 27
I/O フィールド, 71, 216
SIMATIC S7-1200、S7-1500 チャンネル, 27
WinCC AlarmControl, 205
WinCC OnlineTableControl, 131
WinCC OnlineTrendControl, 124
スケール, 213
スタティックテキスト, 61
スライダオブジェクト, 210
タグテーブル, 157
チャンネル, 27
チャンネルユニット, 27
プロジェクト名, 161
ページ番号, 164
ライブラリからのオブジェクト, 56

内

内部タグ, 26
データタイプ, 45
作成, 45

標

標準オブジェクト, 55
スタティックテキスト, 61

表

表示色, 203
メッセージステータスの色の定義, 203
表示要素, 71, 111, 210
値, 71, 111, 210

符

符号なし 16 ビット値, 45
符号付き 16 ビット値, 37

編

編集, 55, 167
スタティックテキスト, 61
フッター, 164
プロセス画像, 55, 64
プロセス値アーカイブ, 121
ページレイアウト, 152
印刷ジョブ, 167

保

保持, 142, 177
データの更新, 177
保存, 64, 164
プロセス画像, 61, 64
プロセス値アーカイブ, 121
ページレイアウト, 164
メッセージ, 194

無

無効化
カバーシートの出力, 154
プロジェクト, 177

名

名前の変更, 52, 121
アーカイブタグ, 121
プロセス画像, 52

有

有効, 96
アラームロギングランタイム, 225
ウィンドウ属性, 92
グラフィックランタイム, 92
タグロギングランタイム, 139
プロジェクト, 96, 142, 174, 228
リニアスケーリング, 43
レポートランタイム, 171