

# SIEMENS

## SIMATIC HMI

### WinCC flexible 2008 Compact / Standard / Advanced


#### システムマニュアル


#### まえがき


WinCC flexibleの概要	1
WinCC flexibleエンジニアリングシステム	2
プロジェクトの取り扱い	3
タグの操作	4
画面の作成	5
アラームシステムの作成	6
接続の操作	7
レシピ管理システムの構造	8
タグのロギングと表示	9
レポートの取り扱い	10
ユーザー管理	11
システムファンクションとランタイムスクリプト	12
マルチ言語プロジェクトの構造	13
プロジェクト文書	14
モバイルワイヤレス	15
ジョブの計画;ジョブノケイカク	16
プロジェクトバージョンの管理	17
変更のロギング;ヘンコウノロギング	18
転送;テンソウ	19
STEP 7 でのWinCC flexibleの統合	20
付録	21

## 安全性に関する基準

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。

 <b>危険</b>
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。

 <b>警告</b>
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。

 <b>注意</b>
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 (安全警告サイン付き)。

<b>注意</b>
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 (安全警告サインなし)。

<b>通知</b>
回避しなければ、望ましくない結果や状態が生じ得る状況を示します (安全警告サインなし)。


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 (番号の低い) 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

装置/システムのセットアップおよび使用にあたっては必ず本マニュアルを参照してください。機器のインストールおよび操作は有資格者のみが行うものとします。有資格者とは、法的な安全規制/規格に準拠してアースの取り付け、電気回路、設備およびシステムの設定に携わることを承認されている技術者のことをいいます。

## 使用目的

以下の事項に注意してください。

 <b>警告</b>
本装置およびコンポーネントはカタログまたは技術的な解説に詳述されている用途のみ使用するものとします。また、Siemens 社の承認または推奨するメーカーの装置またはコンポーネントのみを使用してください。本製品は輸送、据付け、セットアップ、インストールを正しく行い、推奨のとおり操作および維持した場合にのみ、正確かつ安全に作動します。

## 商標

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

## 免責事項

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

# まえがき

## このマニュアルの目的

このマニュアルは WinCC flexible マニュアルの一部です。このマニュアルは、WinCC flexible で構成しているシステムの概要を記載しています。このマニュアルは、新しいプロジェクトの作成、システム構成で使用する手順、プロジェクトの HMI デバイスへの転送をサポートします。

このマニュアルは、新人、オペレータおよび WinCC flexible のコンフィグレーション、コミッショニング、インストール、サービスを行うコンフィグレーションエンジニアを対象としています。

WinCC flexible 内蔵のヘルプ、つまり WinCC flexible Information System には、詳細情報が格納されています。インフォメーションシステムには、取扱説明、例、参考情報が電子データで含まれています。

## 基本的知識の必要条件

このマニュアルを理解するには、オートメーション技術分野の一般的知識が必要です。

Windows 2000 または Windows XP オペレーティングシステムで動作する PC の使用経験も必要です。スクリプトを使用する詳細コンフィグレーションでは、VBA または VBS の知識が必要です。

## このマニュアルの適用範囲

このマニュアルは、WinCC flexible 2008 ソフトウェアパッケージに対して有効です。

## 情報体系の位置付け

このマニュアルは SIMATIC HMI マニュアルの一部です。以下に、SIMATIC HMI の情報種類の概要を記載します。

### ユーザーマニュアル

- 『WinCC flexible Micro』
  - WinCC flexible Micro エンジニアリングシステム(ES)に基づいて、エンジニアリングの基礎について説明します。
- 『WinCC flexible Compact/Standard/Advanced』
  - WinCC flexible Compact、WinCC flexible Standard、および WinCC flexible Advanced エンジニアリングシステム(ES)に基づいて、エンジニアリングの基礎について説明します。
- 『WinCC flexible Runtime』：
  - PC 上でランタイムプロジェクトをコミッショニングして操作する方法について説明します。

- 『WinCC flexible Migration』：
  - 既存の ProTool プロジェクトを WinCC flexible に変換する方法について説明します。
  - 既存の WinCC プロジェクトを WinCC flexible に変換する方法について説明します。
  - OP 3 から OP 73 または OP 73 micro へ、HMI 移行付き ProTool プロジェクトを移行する方法について説明します。
  - HMI 移行ツールを使用して、ProTool プロジェクトを OP7 から OP 77B または OP 77A へ移行する方法について説明します。
  - HMI 移行ツールを使用して、ProTool プロジェクトを OP17 から OP 177B へする移行方法について説明します。
  - HMI 移行付き ProTool プロジェクトを RMOS グラフィックデバイスから Windows CE デバイスへ移行する方法を説明しています。
- 『Communication』：
  - 『Communication』の第 1 部では、HMI デバイスの SIMATIC PLC への接続について説明しています。
  - 『Communication』の第 2 部では、HMI デバイスとサードパーティ PLC の接続について説明します。

#### オペレーティング命令

- SIMATIC HMI デバイスの操作説明書
  - OP 73、OP 77A、OP 77B
  - TP 170micro、TP 170A、TP 170B、OP 170B
  - OP 73micro、TP 177micro
  - TP 177A、TP 177B、OP 177B
  - TP 270、OP 270
  - TP 277、OP 277
  - MP 270B
  - MP 370
  - MP 377
- モバイル型 SIMATIC HMI デバイスの操作説明書
  - Mobile Panel 170
  - Mobile Panel 277
  - Mobile Panel 277F IWLAN
  - Mobile Panel 277 IWLAN
- SIMATIC HMI デバイスの操作説明書(コンパクト版)
  - OP 77B
  - Mobile Panel 170

## 入門書

- 『WinCC flexible for first time users』：
  - サンプルプロジェクトに基づいて、画面、アラーム、およびレシピの作成、および画面ナビゲーションの基礎についてステップバイステップ方式で概説します。
- 『WinCC flexible for advanced users』：
  - サンプルプロジェクトに基づいて、ログ、プロジェクトレポート、スクリプト、ユーザー管理の作成、多言語プロジェクト、および STEP 7 への組み込みの基礎についてステップバイステップ方式で概説します。
- WinCC flexible のオプション：
  - サンプルプロジェクトに基づいて、WinCC flexible Audit、Sm@rtServices、Sm@rtAccess、および OPC Server の各オプションの作成の基礎について順を追って説明します。

## オンラインによる入手

以下のリンクが、SIMATIC 製品およびシステムの種々の言語での技術マニュアルへのご案内いたします。

- SIMATIC ガイド技術マニュアル：  
[http://www.automation.siemens.com/simatic/portal/html\\_76/techdoku.htm](http://www.automation.siemens.com/simatic/portal/html_76/techdoku.htm)

## このマニュアル

このマニュアルの構成:

- WinCC flexible 入門 – 第 1 章
- WinCC flexible での作業 – 第 2 ~ 17 章
- HMI デバイスへのプロジェクトの転送 – 第 18 章
- STEP 7 での WinCC flexible の統合 – 第 19 章
- 付録 – 第 20 章

## 表記規則

コンフィグレーションソフトウェアとランタイムソフトウェアでは、命名表記規則が異なります。

- "WinCC flexible 2008"は、設定ソフトウェアを表します。
- "Runtime"は、HMI デバイス上で動作するランタイムソフトウェアを表します。
- "WinCC flexible Runtime"は、標準 PC または Panel PC 用 ビジュアル製品を表します。

一般的な文脈では、用語"WinCC flexible"が使用されます。他のバージョンとの区別が必要な場合には、"WinCC flexible 2008"のように、バージョン名が必ず使用されます。

マニュアルを読みやすくするため、以下のフォーマットが使用されます。

表記法	適用範囲
[画面の追加]	<ul style="list-style-type: none"> <li>ユーザーインターフェースで発生する専門用語。たとえば、ダイアログ名、タブ、ボタン、メニューコマンドなどが挙げられます。</li> <li>必要な入力。たとえば、限界値やタグ値が挙げられます。</li> <li>パス情報</li> </ul>
[ファイル 編集]	操作シーケンスです。たとえば、メニューコマンドやショートカットメニューコマンドが挙げられます。
<F1>、<Alt> + <P>	キーボード入力

このような注記には特に注意する必要があります。

### 注記

製品とその使用に関する重要情報が記載された注、または特別な注意を払う必要があるマニュアルの関連部分。

### 商標;シヨウヒョウ

HMI®
SIMATIC®
SIMATIC HMI®
SIMATIC ProTool®
SIMATIC WinCC®
SIMATIC WinCC flexible®

このマニュアルに記載される商標を示すその他の名称を第三者が自己の目的で使用することは、当該商標の所有者の権利を侵害する可能性があります。

### その他のサポート

#### 代理店と事務所

このマニュアルで触れられていない、該当製品の使用に関する疑問点については、お近くのシーメンスの代理店にお問い合わせください。

お問い合わせ先は、以下のサイトで確認できます。

<http://www.siemens.com/automation/partner>

SIMATIC 製品およびシステムに対する技術ドキュメントへの案内は、以下の場所で入手することができます。

<http://www.siemens.com/simatic-tech-doku-portal>

オンラインカタログおよびオンライン注文システムは、以下の場所で入手することができます。

<http://mall.automation.siemens.com>

## トレーニングセンター;トレーニングセンター

オートメーションシステムを習熟するために、さまざまなコースを提供します。お客様の地域のトレーニングセンターが、ドイツの D-90327 ニュルンベルクにある中央トレーニングセンターにお問い合わせください。

インターネット: <http://www.sitrain.com>

## テクニカルサポート

すべての A&D 製品のテクニカルサポートを受けることができます。

Web 上でのサポートリクエストフォーム:

<http://www.siemens.com/automation/support-request>

テクニカルサポートについての詳細は、次の Web サイトを参照してください。

<http://www.siemens.com/automation/service>

## インターネットによるサービスとサポート(Service & Support)

マニュアルに加えて、オンラインの総合的な知識ベースを提供しています。

<http://www.siemens.com/automation/service&support>

以下の内容が確認できます。

- お使いの製品についての最新の情報を提供するニュースレター。
- お使いのアプリケーションに該当するドキュメント。弊社のサービスおよびサポートデータベースの検索機能によってアクセスできます。
- フォーラム。世界中からのユーザーおよび技術者が考えを交換します。
- Automation & Drives 部門の現地担当提携会社の連絡先。
- オンサイトサービス、修復、スペアパーツに関する情報。"サービス"ページにはその他多くの情報が記載されています。



# 目次

まえがき .....	3
<b>1 WinCC flexibleの概要 .....</b>	<b>19</b>
1.1 SIMATIC HMIの入門 .....	19
1.2 WinCC flexibleシステムの概要 .....	20
1.2.1 WinCC flexibleの構成要素 .....	20
1.2.2 WinCC flexibleエンジニアリングシステム .....	21
1.2.3 WinCC flexible Runtime .....	23
1.2.4 使用可能なオプション .....	23
1.2.5 ライセンス .....	25
1.2.5.1 ライセンスおよびライセンスキー .....	25
1.2.5.2 ライセンスがない場合のWinCC flexible .....	26
1.3 自動化コンセプト;ジドウカコンセプト .....	27
1.3.1 WinCC flexibleでの自動化コンセプト .....	27
1.3.2 HMIデバイスへのリモートアクセス .....	31
1.3.3 自動アラーム配信 .....	32
1.3.4 分散HMI .....	33
1.3.5 パーソナルHMIデバイスのサポート .....	34
1.3.6 モジュールマシンのコンセプト .....	34
1.4 コンフィグレーションコンセプト .....	35
1.4.1 コンフィグレーションサポート .....	35
1.4.2 スケーラブルなコンフィグレーションツール .....	36
1.4.3 PLCに依存しないコンフィグレーション .....	37
1.4.4 使用 .....	37
1.4.5 インテリジェントツール .....	38
1.4.5.1 データの一括処理 .....	38
1.4.5.2 移動パスのコンフィグレーション .....	39
1.4.5.3 画面ナビゲーションのグラフィックコンフィグレーション .....	40
1.4.6 トータルインテグレートドオートメーション .....	41
<b>2 WinCC flexibleエンジニアリングシステム .....</b>	<b>43</b>
2.1 プログラミングインターフェースに関する基本原理 .....	43
2.2 WinCC flexibleユーザーインターフェース .....	44
2.2.1 WinCC flexibleユーザーインターフェースの要素 .....	44
2.2.2 メニューとツールバー .....	46
2.2.3 作業エリア .....	48
2.2.4 [プロジェクト]ウィンドウ .....	49
2.2.5 [プロパティ]ウィンドウ .....	50
2.2.6 ライブラリ .....	51
2.2.7 [出力]ウィンドウ .....	52
2.2.8 [オブジェクト]ウィンドウ .....	54
2.3 エディタ固有のオペレーティングエレメントの配置 .....	56
2.4 ウィンドウとツールバーの取り扱い .....	57
2.5 マウスの取り扱い .....	59

2.6	キーボードコントロール.....	61
2.7	WinCC flexibleの取り扱い.....	62
2.7.1	WinCC flexibleの取り扱い.....	62
2.7.2	プロジェクトの取り扱い.....	63
2.7.3	WinCC flexibleによる複数プロジェクトの編集.....	63
2.7.4	プロジェクトの機能範囲.....	64
2.7.5	エディタのプロパティ.....	65
2.7.6	エディタを開く.....	66
2.7.7	エディタ間の切り替え.....	68
2.7.8	オブジェクトリスト;オブジェクトリスト.....	70
2.7.9	ファンクションリスト.....	71
2.7.10	テキストリスト.....	73
2.7.11	グラフィックリスト.....	78
2.7.12	ヘルプの表示.....	83
2.7.13	WinCC flexibleのカスタマイズセットアップ.....	84
2.8	WinCC flexibleスタートセンター.....	85
<b>3</b>	<b>プロジェクトの取り扱い.....</b>	<b>87</b>
3.1	プロジェクトの取り扱いの基本.....	87
3.1.1	プロジェクトの取り扱い.....	87
3.1.2	プロジェクトの構成要素.....	89
3.2	プロジェクトのタイプ.....	89
3.2.1	プロジェクトのタイプ.....	89
3.2.2	プロジェクトのHMIデバイス依存性.....	90
3.2.3	複数のHMIデバイス用のプロジェクトの構築.....	93
3.2.4	複数のオペレーティングユニット上で使用するプロジェクトの作成.....	95
3.2.5	SIMOTIONとSTEP7 に統合されたWinCC flexible.....	96
3.3	複数言語の組み込み.....	98
3.4	プロジェクトの編集.....	100
3.4.1	プロジェクトの編集.....	100
3.4.2	プロジェクトの表示.....	102
3.4.3	[プロジェクト]ウィンドウでの作業.....	104
3.4.4	[オブジェクト]ウィンドウでの作業.....	106
3.4.5	既存のプロジェクトの移行.....	107
3.5	プロジェクトの変換.....	108
3.5.1	さまざまなWinCC flexibleバージョンを持つプロジェクト.....	108
3.5.2	さまざまなWinCC flexibleバージョンを持つプロジェクト間の相違点.....	109
3.5.3	HMIデバイスバージョンの区別.....	111
3.6	プロジェクトデータの再利用.....	112
3.6.1	コピーのメカニズム.....	112
3.6.2	フラットコピー.....	113
3.6.3	コピー.....	114
3.6.4	置換.....	115
3.6.5	ライブラリの使用.....	117
3.6.6	銘板の使用.....	117
3.7	クロスリファレンスの作業.....	118
3.8	再配線の概要.....	119
3.9	プロジェクト内部の検索と置換機能.....	119
3.10	WinCC flexible内の文書に関する基本原理.....	120

3.11	生成時の一貫性チェック .....	120
3.12	デバッグオブジェクト .....	121
3.13	プロジェクトの転送 .....	122
3.13.1	転送操作の基本原理 .....	122
3.13.2	プロジェクトのバック転送 .....	124
<b>4</b>	<b>タグの操作 .....</b>	<b>127</b>
4.1	基本 .....	127
4.1.1	タグの基本 .....	127
4.1.2	外部タグ .....	128
4.1.3	内部タグ .....	129
4.2	エレメントと基本設定 .....	130
4.2.1	タグエディタ;タグエディタ .....	130
4.2.2	タグおよび配列の基本設定 .....	131
4.3	タグの操作 .....	133
4.3.1	タグのプロパティ .....	133
4.3.2	外部タグを使用したPLCとの通信 .....	134
4.3.3	タグコンフィグレーションの変更 .....	136
4.3.4	タグの限界値 .....	136
4.3.5	タグの初期値 .....	137
4.3.6	ランタイム時のタグ値の更新 .....	137
4.3.7	データロギング .....	138
4.3.8	タグの線形スケーリング .....	139
4.3.9	タグの間接アドレス指定 .....	140
4.4	配列の基本 .....	141
4.5	配列の例 .....	143
4.6	サイクルの基本 .....	144
4.7	構造の操作 .....	145
4.7.1	基本構造 .....	145
4.7.2	構造エディタ .....	147
4.7.3	構造管理 .....	148
4.8	タグのインポート .....	150
4.8.1	タグのインポートとエクスポート .....	150
4.8.2	タグのエクスポートとインポートのための設定 .....	151
4.8.3	接続データのフォーマット .....	153
4.8.4	タグデータのフォーマット .....	155
<b>5</b>	<b>画面の作成 .....</b>	<b>159</b>
5.1	基本 .....	159
5.1.1	画面の基礎 .....	159
5.1.2	画面のHMIベース依存性 .....	161
5.1.3	"画面"エディタ .....	163
5.1.4	手順;テジュン .....	164
5.2	ナビゲーションシステムの作成 .....	165
5.2.1	ナビゲーションオプション .....	165
5.2.2	画面ナビゲーションシステムのグラフィックプログラミング .....	166
5.2.3	ナビゲーションコントロールの使用 .....	168
5.3	オブジェクトの取り扱い .....	169
5.3.1	オブジェクトの概要 .....	169
5.3.2	オブジェクトの編集オプション .....	173

5.3.3	複数のオブジェクトの位置変更とサイズ変更	174
5.3.4	外部グラフィック	175
5.3.5	オブジェクトグループ	177
5.3.6	グループプロパティの定義	178
5.4	ダイナミックな更新ファンクションの割り付けのオプション	179
5.5	ファンクションキーの取り扱い	180
5.6	レイヤーの利点	182
5.7	オブジェクトライブラリ	182
5.8	銘板の取り扱い	184
5.8.1	銘板の基礎	184
5.8.2	ダイナミックコントロールオプション	186
<b>6</b>	<b>アラームシステムの作成</b>	<b>187</b>
6.1	基本	187
6.1.1	プロセスとシステムアラームの表示	187
6.1.2	ユーザー定義のアラーム	188
6.1.2.1	利用可能なアラーム手順	188
6.1.2.2	アラームの確認	189
6.1.2.3	アラームクラス	190
6.1.3	システムアラーム	191
6.1.4	アラームの表示	192
6.1.4.1	HMIデバイスでのアラームの表示	192
6.1.4.2	アラーム表示のフィルタリング	193
6.1.4.3	アラームのロギングとレポート	194
6.1.4.4	アラーム編集のためのシステムファンクション	195
6.2	エレメントと基本設定	197
6.2.1	アラームの構成要素とプロパティ	197
6.2.2	アラームを設定するためのエディタ	199
6.2.2.1	エディタの基本原理	199
6.2.2.2	"ディスクリートアラーム"エディタ	201
6.2.2.3	"アナログアラーム"エディタ	202
6.2.2.4	"システムアラーム"エディタ	203
6.2.2.5	"アラームクラス"エディタ	204
6.2.2.6	"アラームグループ"エディタ	205
6.2.2.7	アラームシステムの基本設定	206
6.3	アラームの取り扱い	207
6.3.1	アラームのレポート機能	207
6.3.2	アラームナンバリング手順によるアラームの統合化	207
6.4	アラームロギング;アラームロギング	210
6.4.1	アラームロギングの基本原理	210
6.4.2	アラームロギング;アラームロギング	211
6.4.3	"アラームログ"エディタ	212
6.4.4	アラームログの基本設定	213
6.4.5	アラームロギング;アラームロギング	215
6.4.6	ログ済みアラームの画面への表示	215
6.4.7	アラームが格納されている*.csvファイルの構造	216
6.4.8	ODBCログデータベースへの直接アクセス	217
<b>7</b>	<b>接続の操作</b>	<b>219</b>
7.1	基本	219
7.1.1	通信の基本	219
7.1.2	通信の原則	220

7.2	エレメントと基本設定 .....	222
7.2.1	接続エディタ .....	222
7.2.2	接続のパラメータ .....	224
7.2.3	接続のエリアポインタ .....	225
7.3	接続とプロトコルの概要 .....	226
7.4	Ethernet接続 .....	229
7.5	HMIのSNMPおよびMIB .....	230
<b>8</b>	<b>レシピ管理システムの構造 .....</b>	<b>231</b>
8.1	基本 .....	231
8.1.1	レシピの基本原理 .....	231
8.1.2	レシピの使用例 .....	232
8.1.3	レシピの構造 .....	233
8.1.4	レシピの表示 .....	234
8.1.5	レシピデータレコードの転送 .....	236
8.1.6	レシピのコンフィグレーション .....	239
8.1.7	OP 77AおよびTP 177A専用の機能 .....	241
8.1.8	PLCによるレシピデータレコードの同期化 .....	243
8.2	エレメントと基本設定 .....	244
8.2.1	"レシピ"エディタ .....	244
8.2.2	レシピエレメント .....	245
8.2.3	レシピデータレコード .....	247
8.2.4	レシピ設定 .....	248
8.3	ランタイムでのレシピの表示と編集 .....	250
8.3.1	[レシピ]画面と[レシピ]ウィンドウ .....	250
8.3.2	[レシピ]ウィンドウ .....	251
8.3.3	[レシピ]ウィンドウのコンフィグレーションオプション .....	252
8.3.4	ランタイム中の[レシピ]ウィンドウの動作 .....	255
8.3.5	レシピ画面 .....	256
8.3.6	[レシピ]ウィンドウのオペレータ入力 .....	258
8.3.7	簡易[レシピ]ウィンドウのオペレータ入力 .....	260
8.3.8	レシピ構造の変更に対する応答 .....	263
8.4	シナリオ .....	264
8.4.1	シナリオ:ランタイム中のレシピデータレコードの入力 .....	264
8.4.2	シナリオ:手動製造シーケンス .....	265
8.4.3	シナリオ:自動製造シーケンス .....	267
<b>9</b>	<b>タグのロギングと表示 .....</b>	<b>269</b>
9.1	基本 .....	269
9.1.1	データロギングの基本原理 .....	269
9.1.2	WinCC flexibleでのデータロギング .....	269
9.1.3	トレンド .....	272
9.2	エレメントと基本設定 .....	274
9.2.1	"データログ"エディタ .....	274
9.2.2	データログの基本設定 .....	275
9.3	タブ値の記録 .....	277
9.4	ログ済みデータの出力;ログスミデータノシュツリョク .....	278
9.4.1	タグ値の画面への出力 .....	278
9.4.2	タグ値が格納されている*.csvファイルの構造 .....	279
9.4.3	ODBCログデータベースへの直接アクセス .....	280

<b>10</b>	<b>レポートの取り扱い</b> .....	<b>281</b>
10.1	レポートシステムの原理 .....	281
10.2	レポートのストラクチャ .....	282
10.3	エレメントと基本設定 .....	283
10.3.1	エディタ"レポート" .....	283
10.3.2	[ツールボックス]ウィンドウの使用 .....	285
10.4	レポートの取り扱い .....	286
10.4.1	レポートの作成 .....	286
10.4.2	レポートプロパティの調整 .....	288
10.4.3	レポート作成用のオブジェクト .....	289
10.4.4	レポートオブジェクトの使用 .....	292
10.5	アラームのレポート機能 .....	292
10.5.1	アラームのレポート機能 .....	292
10.5.2	アラームプロトコルに応じた出力パラメータの処理 .....	293
10.6	レシピのレポート機能 .....	296
10.6.1	レシピのレポート機能 .....	296
10.6.2	レシピレポート用の出力パラメータの編集 .....	297
10.7	レポートの出力 .....	300
<b>11</b>	<b>ユーザー管理</b> .....	<b>301</b>
11.1	ユーザー管理のアプリケーションの分野 .....	301
11.2	ユーザー管理の構造 .....	302
11.3	エレメントと基本設定 .....	303
11.3.1	"ユーザー"ユーザー管理 .....	303
11.3.2	"グループ"ユーザー管理 .....	304
11.3.3	ユーザー作業エリア .....	305
11.3.4	ユーザーグループ作業エリア .....	306
11.3.5	ランタイムセキュリティの設定 .....	307
11.4	ユーザー管理の使用 .....	309
11.4.1	SIMATICログオンを使用した一元的ユーザー管理 .....	309
11.4.2	ランタイム時のユーザー .....	311
11.4.3	[ユーザー]ウィンドウ .....	312
11.4.4	アクセスセキュリティ .....	314
<b>12</b>	<b>システムファンクションとランタイムスクリプト</b> .....	<b>315</b>
12.1	基本 .....	315
12.1.1	システムファンクションとランタイムスクリプト .....	315
12.1.2	システムファンクション .....	317
12.1.3	システムファンクションの使用 .....	319
12.1.4	スクリプト .....	320
12.1.5	スクリプトの使用 .....	321
12.2	ファンクションリストの使用 .....	322
12.2.1	ファンクションリストの基本原理 .....	322
12.2.2	ファンクションリストのプロパティ .....	323
12.3	エレメントと基本設定 .....	324
12.3.1	"スクリプト"エディタ .....	324
12.3.2	[スクリプト]エディタのプロパティ .....	326
12.3.3	基本設定 .....	329
12.4	スクリプトの作成 .....	331

12.4.1	タグへのアクセス	331
12.4.2	スクリプト内でのスクリプトとシステムファンクションの呼び出し	332
12.4.3	オブジェクトへのアクセス	334
12.4.4	タグとオブジェクトの同期化	335
12.4.5	スクリプトの保存	335
12.5	デバッグ;デバッグ	336
12.5.1	スクリプトのデバッグ	336
12.5.2	デバッグの統合	337
12.6	ランタイム中のファンクションのランタイム動作	341
12.6.1	ランタイム中のファンクションリストの実行	341
12.6.2	ランタイム中のスクリプトの処理	342
12.6.3	値の受渡しと戻し	343
12.6.4	VBSでのランタイム時のオブジェクトプロパティの変更	344
12.6.5	スクリプト内でのHMIデバイス依存システムファンクション	344
<b>13</b>	<b>マルチ言語プロジェクトの構造</b>	<b>345</b>
13.1	複数言語を扱う	345
13.2	WinCC flexible専用用語	346
13.3	言語設定	348
13.3.1	オペレーティングシステムの言語設定	348
13.3.2	アジア言語に対応したオペレーティングシステム設定	349
13.3.3	"プロジェクト言語"エディタ	349
13.4	複数の言語でのプロジェクトの作成	351
13.4.1	複数の言語でのプロジェクトの作成	351
13.4.2	エンジニアリングシステムのアジアおよび東洋言語に特有な特徴	352
13.4.3	エディタでプロジェクトテキストを変換する	353
13.4.4	"プロジェクトテキスト"エディタ	354
13.4.5	トランスレータでテキストを変換する	356
13.5	ディレクトリの取り扱い	358
13.5.1	ディレクトリの取り扱い	358
13.5.2	"システム辞書"エディタ	359
13.5.3	"ユーザー辞書"エディタ	360
13.6	言語依存グラフィックの使用	361
13.6.1	言語依存グラフィックの使用	361
13.6.2	"グラフィック"エディタ	362
13.7	ランタイム中の言語	363
13.7.1	ランタイム中の言語	363
13.7.2	言語切り替えのコンフィグレーション	364
13.7.3	ランタイム中のアジアおよび東洋言語に特有な特徴	364
<b>14</b>	<b>プロジェクト文書</b>	<b>367</b>
14.1	基本	367
14.1.1	プロジェクト文書	367
14.1.2	レイアウトのストラクチャ	368
14.2	レイアウトの使用	370
14.2.1	レイアウトの使用	370
14.2.2	プロジェクト文書用のレイアウトの編集	371
14.3	プロジェクトレポートの作成	373
14.3.1	プロジェクトレポート用のデータの選択	373
14.3.2	選択したオブジェクトのデータの出力	373

14.3.3	プロジェクト文書用オブジェクトの選択 .....	374
<b>15</b>	<b>モバイルワイヤレス .....</b>	<b>375</b>
15.1	基本原理 .....	375
15.1.1	Mobile Panel Wirelessのアプリケーションのフィールド .....	375
15.1.2	Mobile Panel Wirelessの動作 .....	377
15.2	エレメントと基本設定 .....	380
15.2.1	ゾーン .....	380
15.2.2	ゾーン作業エリア .....	381
15.2.3	有効な範囲 .....	382
15.2.4	[有効範囲]作業エリア .....	384
15.3	有効な範囲での作業 .....	385
<b>16</b>	<b>ジョブの計画;ジョブノケイカク .....</b>	<b>387</b>
16.1	スケジューラ of アプリケーションフィールド .....	387
16.2	ジョブとイベントを使用した作業 .....	388
16.3	エレメント .....	390
16.3.1	"スケジューラ"エディタ .....	390
16.3.2	[スケジューラ]エディタの作業エリア .....	391
<b>17</b>	<b>プロジェクトバージョンの管理 .....</b>	<b>393</b>
17.1	プロジェクトバージョン of アプリケーション .....	393
17.2	バージョン管理の基本 .....	394
17.3	トランク .....	395
17.4	ブランチ .....	396
17.5	エレメント .....	397
17.5.1	"プロジェクトバージョン"エディタ .....	397
17.5.2	Bedienelemente Projektversionen__ohneVersionsvergleich .....	399
17.5.3	バージョン管理作業エリア .....	399
17.5.4	[プロパティ]ウィンドウ .....	401
17.6	プロジェクトバージョンの取り扱い .....	402
17.6.1	バージョンの比較 .....	402
<b>18</b>	<b>変更のロギング;ハンコウノロギング .....</b>	<b>403</b>
18.1	変更ログ用のアプリケーション .....	403
18.2	プロジェクトの変更ログ .....	404
18.3	プロジェクトの変更ログ .....	405
18.4	バージョン管理下のプロジェクトの変更ログ .....	407
18.5	エレメント .....	408
18.5.1	"ログの変更"エディタ .....	408
18.5.2	ログオペレータ制御の変更 .....	409
18.5.3	ログ作業エリアの変更 .....	410
<b>19</b>	<b>転送;テンソウ .....</b>	<b>411</b>
19.1	基本 .....	411
19.1.1	転送操作の基本原理 .....	411
19.1.2	転送設定;テンソウセッテイ .....	413
19.1.3	USB経由の転送 .....	416
19.1.4	プロジェクトのバック転送 .....	417

19.2	HMIデバイス上でのファイルの管理	419
19.2.1	ProSave	419
19.2.2	HMIデータのバックアップ	420
19.2.3	オペレーティングシステムの更新	421
19.2.4	オーソリゼーションの転送	422
19.2.5	オプションのインストール	423
<b>20</b>	<b>STEP 7 でのWinCC flexibleの統合</b>	<b>425</b>
20.1	基本原理	425
20.1.1	統合されたプロジェクトの制限	425
20.1.2	統合されたプロジェクトの変換	426
20.1.3	STEP 7 への統合の基本原理	427
20.1.4	SIMATIC Managerの取り扱い	428
20.1.5	HW Configの取り扱い	429
20.1.6	接続を設定する	430
20.1.7	オブジェクトの取り扱い	431
20.1.8	統合されたプロジェクトの変換	433
20.1.8.1	STEP 7 に統合されたWinCC flexibleプロジェクトの変換	433
20.1.8.2	統合されたプロジェクトの現在のWinCC flexibleバージョンへの変換	435
20.1.8.3	統合されたプロジェクトの、以前のWinCC flexibleバージョンへの変換	436
20.1.9	PCステーションでのWinCC flexibleの統合	437
20.2	通信設定のコンフィグレーション	439
20.2.1	ルーティング経由の通信設定の構成	439
20.2.2	S7 ルーティング経由のプロジェクト転送	440
20.3	タグのコンフィグレーション	444
20.3.1	"タグ"エディタによるタグの設定	444
20.3.2	アプリケーションポイント経由のタグの接続	445
20.4	アラームのコンフィグレーション	446
20.4.1	アラームナンバリング手順によるアラームの統合化	446
<b>21</b>	<b>付録</b>	<b>449</b>
21.1	オープンソースソフトウェア	449
21.2	特徴	449
21.2.1	一般的な技術データ	449
21.2.1.1	対応するオペレーティングシステム	449
21.2.1.2	対応するデータベース	450
21.2.1.3	その他のソフトウェアバージョンサポート	450
21.2.1.4	推奨プリンタ	450
21.2.1.5	有効な文字	451
21.2.1.6	レシピの必要メモリ	451
21.2.1.7	特殊デバイスのレシピの必要メモリ容量	452
21.2.2	システム制限;システムセイゲン	454
21.2.2.1	システム制限	454
	索引	467



# WinCC flexible の概要

## 1.1 SIMATIC HMI の入門

### はじめに

プロセスがより複雑になり、マシンやプラントの機能性への要求が向上している環境で作業をするオペレータにとっては、最大限の透過性が不可欠です。Human Machine Interface(HMI)は、この透過性を提供します。

HMI システムは、人(オペレータ)とプロセス(マシン/プラント)間のインターフェースを表します。この PLC は、プロセスを制御する実際のユニットです。このため、オペレータと WinCC flexible(HMI デバイスに搭載された)間のインターフェース、および WinCC flexible と PLC 間のインターフェースが存在します。HMI システムは、以下の役割を引き受けます。

- プロセスのビジュアル化

プロセスは、HMI デバイス上でビジュアル化することができます。HMI デバイスの画面はダイナミックに更新されます。更新はプロセスの変化に基づき行われます。

- プロセスのオペレータ制御

オペレータは GUI を使用してプロセスを制御することができます。たとえば、オペレータは、制御用の参照値のプリセットやモーターの起動をすることができます。

- アラームの表示

設定値を超過したときなどプロセスの異常状態になると、アラームが自動的にトリガされます。

- プロセス値とアラームのアーカイブ

HMI システムは、アラームとプロセス値を記録することができます。この機能により、プロセスシーケンスをログに記録したり、以前の生産データを検索することができます。

- プロセス値およびアラームのロギング

HMI システムは、アラームおよびプロセス値のレポートを出力することができます。これにより、たとえばシフト終了時に生産データをプリントアウトすることができます。

- プロセスおよびマシンパラメータの管理

HMI システムでは、プロセスおよびマシンのパラメータをレシピに保存することができます。たとえば、生産する製品バージョンを変更するために、これらのパラメータを HMI デバイスから PLC に 1 つのパスでダウンロードすることができます。

## SIMATIC HMI

SIMATIC HMI は、さまざまなオペレータ制御や監視タスクのための、完全に統合された単一ソースのシステムを提供します。SIMATIC HMI を使用することで、プロセスを常に把握し、マシンやユニットを常に稼働しておくことができます。

簡単な SIMATIC HMI システムの例としては、マシンレベルで使用する小さなタッチパネルが挙げられます。

生産プラントの制御およびモニタに用いられる SIMATIC HMI システムは、最高のパフォーマンス範囲を実現します。これらのシステムには、高性能なクライアント/サーバーシステムが含まれます。

## SIMATIC WinCC flexible の統合

WinCC flexible は、快適で効率の良いエンジニアリングを伴う、将来を保証されたマシン指向の自動化コンセプトの HMI ソフトウェアです。WinCC flexible では、以下のメリットを組み合わせています。

- 簡単な操作
- 透過性
- 柔軟性

## 1.2 WinCC flexible システムの概要

### 1.2.1 WinCC flexible の構成要素

#### WinCC flexible エンジニアリングシステム

WinCC flexible エンジニアリングシステムは、すべての不可欠なコンフィグレーションタスクを処理するためのソフトウェアです。WinCC flexible edition では、SIMATIC HMI 機種ごとの HMI デバイスを組み込むかを決定します。

#### WinCC flexible Runtime

WinCC flexible Runtime は、プロセスをビジュアル化するためのソフトウェアです。WinCC flexible Runtime のプロセスモードで、プロジェクトを実行します。

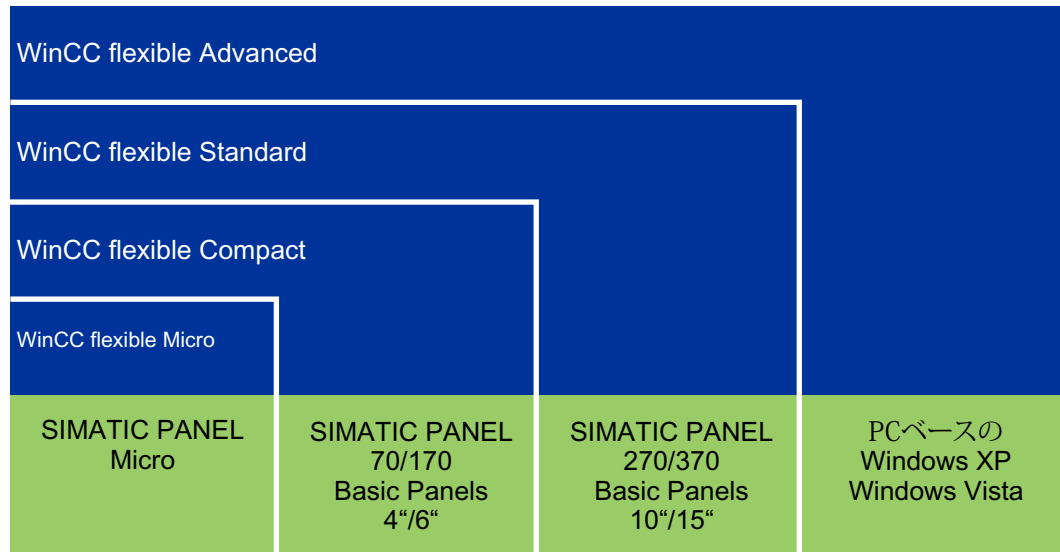
#### WinCC flexible のオプション

WinCC flexible のオプションを使用すると、WinCC flexible の標準的な機能を拡張することができます。各オプションを使用するには個々のライセンスが必要です。

## 1.2.2 WinCC flexible エンジニアリングシステム

### はじめに

WinCC flexible は、すべての設定タスクのためのエンジニアリングシステムです。WinCC flexible にはモジュラーデザインが採用されています。より上位のエディションを用いることで、サポートされるデバイスと WinCC flexible の機能性の範囲を拡張することができます。また、パワーパックによって上位エディションにいつでも移行できます。



WinCC flexible はマイクロパネルから簡単な PC でのビジュアル化に至るまでのパフォーマンス範囲をカバーします。そのため WinCC flexible の機能は、ProTool および TP Designer の機能に匹敵します。既存の ProTool のプロジェクトを WinCC flexible に統合することが可能です。

## 原理

WinCC flexible で新規のプロジェクトを作成したり、既存のプロジェクトを開くときに、WinCC flexible ワークベンチがプログラミングコンピュータの画面上で開きます。プロジェクトのストラクチャがビジュアル化され、プロジェクトは[プロジェクト]ウィンドウで管理されます。



WinCC flexible では、それぞれの設定タスクに対応する固有のエディタが用意されています。たとえば、"画面"エディタでは HMI デバイスの GUI を作成します。また、アラームを設定するには"ディスクリートアラーム"エディタを使用します。

プロジェクトに関係するすべてのプロジェクト設定データは、プロジェクトデータベースに格納されます。

## 他の WinCC flexible エディションへの移行

現在使用中の WinCC flexible のエディションによって、どの HMI デバイスを設定できるかが特定されます。現在使用中の WinCC flexible のエディションでサポートされていない HMI デバイスを設定するために、他の WinCC flexible のエディションに移行することができます。移行後もすべての既存の機能が使用可能です。

WinCC flexible Compact edition 以降は、WinCC flexible のエディションの更新にパワーパックを使用できます。

## WinCC flexible アジア エディション

WinCC flexible 2005 ASIA バージョンで使用できるのは、"Advanced"と"Standard"エディションのみです。

### 1.2.3 WinCC flexible Runtime

#### 原理

ランタイム中に、オペレータはプロセスの制御およびモニタをすることができます。これには、特に以下のタスクが含まれます。

- オートメーションシステムとの通信
- イメージの画面上でのビジュアル化
- セットポイント値の設定またはバルブの開閉などによるプロセスの操作
- プロセス値およびアラームイベントなどの、現在のランタイムデータのアーカイブ

#### WinCC flexible Runtime のパフォーマンス範囲

WinCC flexible Runtime では、ライセンスにの種類によって決定される一定数のプロセス変数(パワータグ)を、サポートします。

- WinCC flexible Runtime 128: 128 のプロセス変数をサポートします。
- WinCC flexible Runtime 512: 512 のプロセス変数をサポートします。
- WinCC flexible Runtime 2048: 2048 のプロセス変数をサポートします。

パワーパックを使用して、プロセス変数の数を増やすことができます。

### 1.2.4 使用可能なオプション

#### はじめに

以下のコンポーネントでオプションを使用できます。

- WinCC flexible エンジニアリングシステム
- PC ベース HMI デバイス上の WinCC flexible Runtime
- 非 PC ベース HMI デバイス

デバイス依存のランタイムオプションを HMI デバイスで操作できます。

各オプションにはライセンスが必要です。

#### WinCC flexible エンジニアリングシステムのオプション

WinCC flexible エンジニアリングシステムで使用可能なオプション:

WinCC flexible のオプション	機能	使用可能性
WinCC flexible /ChangeControl	バージョン管理および変更履歴	『WinCC flexible Compact/Standard/Advanced』

## WinCC flexible Runtime のオプション

オプションは、使用される対象システムによって異なります。PC ベースの HMI デバイスまたは非 PC ベースの HMI デバイス上の WinCC flexible Runtime には次のオプションが使用できます。

SIMATIC WinCC flexible RT オプション	機能	非 PC ベース HMI デバイス	SIMATIC パネル PC
WinCC flexible /Archives	ランタイム中のアーカイブ機能	Panel 270 から	x
WinCC flexible /Recipes	ランタイム中のレシピ機能	デバイス依存のオプションが使用できません。ライセンスは必要ありません。	x
WinCC flexible /Sm@rtAccess	異なる SIMATIC HMI システム間の通信同様、リモートコントロールおよびリモートモニタ	Panel 270 から	x
WinCC flexible /Sm@rtService	インターネット/イントラネットを介しての機械/プラントのリモート保守およびサービス	Panel 270 から	x
WinCC flexible /OPC-Server	HMI デバイスの OPC サーバーとしての使用	マルチパネル	x
WinCC flexible /ProAgent	ランタイム中のプロセス診断	Panel 270 から	x
WinCC flexible /Audit	FDA に従った相互作用のレポート	Panel 270 から	x

## 注記

Audit オプションを Sm@rtAccess、Sm@rtService で同時に使用することはできません。

## 注記

## Audit Viewer のインストール

Audit Viewer は、オフィス PC での外部分析の準備での、WinCC flexible / Audit オプションの監査トレイルの評価に便利なツールです。

Audit Viewer をインストールするには、製品 DVD のディレクトリ CD\_2/Support/Audit Viewer/Suport にある setup.exe を起動します。

Audit Viewer は、Windows XP でのみ動作します。

## 1.2.5 ライセンス

### 1.2.5.1 ライセンスおよびライセンスキー

#### 原理

すべての WinCC flexible のエディションはライセンスを必要とします。WinCC flexible の特定のエディションはライセンスがない場合には使用できる機能に制限があります。

- ライセンス

ライセンスは紙で配布されます。ライセンスによって、購入された WinCC flexible のエディションをコンピュータにインストールし使用する権利が与えられます。使用権に関する詳しい情報は、電子カタログ CA 01 で入手することができます。

- ライセンスキー

ライセンスキーはコピープロテクトがかかった別個の USB で配布されます。設定を行う間、ライセンスキーが保存されている USB を挿入しておく必要があります。

ライセンスキーの保存されている記憶媒体は、他のすべてのドキュメントで保存場所として指定されます。

#### WinCC flexible エンジニアリングシステムのライセンス

どのライセンスモデルが適用されるかは、WinCC flexible のエディションによって異なります。エディションは、購入してインストールしたライセンスキーによって有効化されます。

- WinCC flexible Micro: ライセンス契約
- 『WinCC flexible Compact/Standard/Advanced』:
  - USB 上の WinCC flexible Compact のライセンス契約およびライセンスキー
  - USB 上の WinCC flexible Standard のライセンス契約およびライセンスキー
  - USB 上の WinCC flexible Advanced のライセンス契約およびライセンスキー

#### WinCC flexible Runtime のライセンス

WinCC flexible Runtime は、ライセンス契約およびライセンスキーによって使用が許可されます。WinCC flexible Runtime のライセンスの種類によって、サポートされる変数の数が異なります:

- WinCC flexible Runtime 128: 128 のプロセス変数をサポートします。
- WinCC flexible Runtime 512: 512 のプロセス変数をサポートします。
- WinCC flexible Runtime 2048: 2048 のプロセス変数をサポートします。

#### オプションパッケージのライセンス

各オプションごとに、ライセンスおよびライセンスキーが USB で 1 つずつ提供されます。WinCC flexible エンジニアリングシステムは既に Runtime オプションパッケージの機能を含んでいます。ランタイムオプションの機能の設定のために、プログラミングコンピュータにライセンスは必要ありません。

### 1.2.5.2 ライセンスがない場合の WinCC flexible

#### 原理

ライセンスがない場合、WinCC flexible はデモモードで実行され、ソフトウェアの操作が制限されます。WinCC flexible エンジニアリングシステムと WinCC flexible Runtime のどちらでも、確認する必要があるアラームが画面に定期的に表示されます。

#### 新しいライセンスキーの要求

ライセンスキーの保存場所を破損または紛失した場合は、カスタマーサポートに連絡してください。ライセンスキーの保存場所には、フロッピーディスクまたは USB などがあります。

## 1.3 自動化コンセプト;ジドウカコンセプト

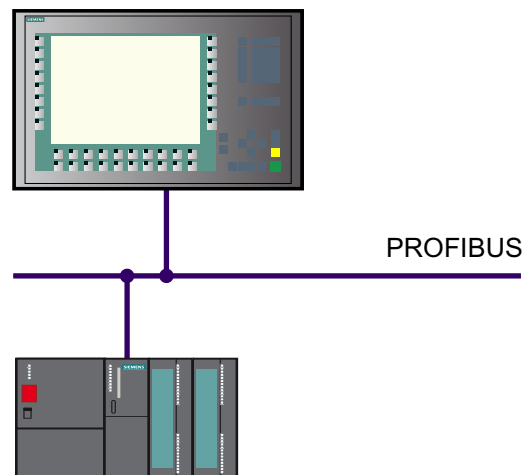
### 1.3.1 WinCC flexible での自動化コンセプト

#### はじめに

WinCC flexible は多数の異なる自動化コンセプトからなる設定をサポートしています。  
WinCC flexible では、デフォルトで次の自動化コンセプトを使用することができます。

#### 1 台の HMI デバイスを使用した制御

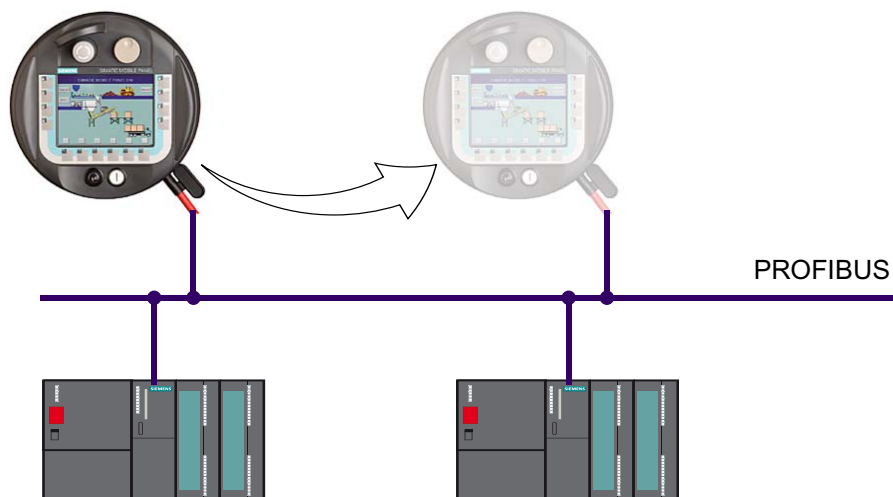
プロセスバスを介して PLC に直接接続された HMI デバイスは、シングルユーザーシステムとも呼ばれます。



シングルユーザーシステムは一般的に生産工程の近くで使用されますが、プロセスやシステムセクションの独立した一部分の操作やモニタに使用することもできます。

### 複数の HMI デバイスの PLC

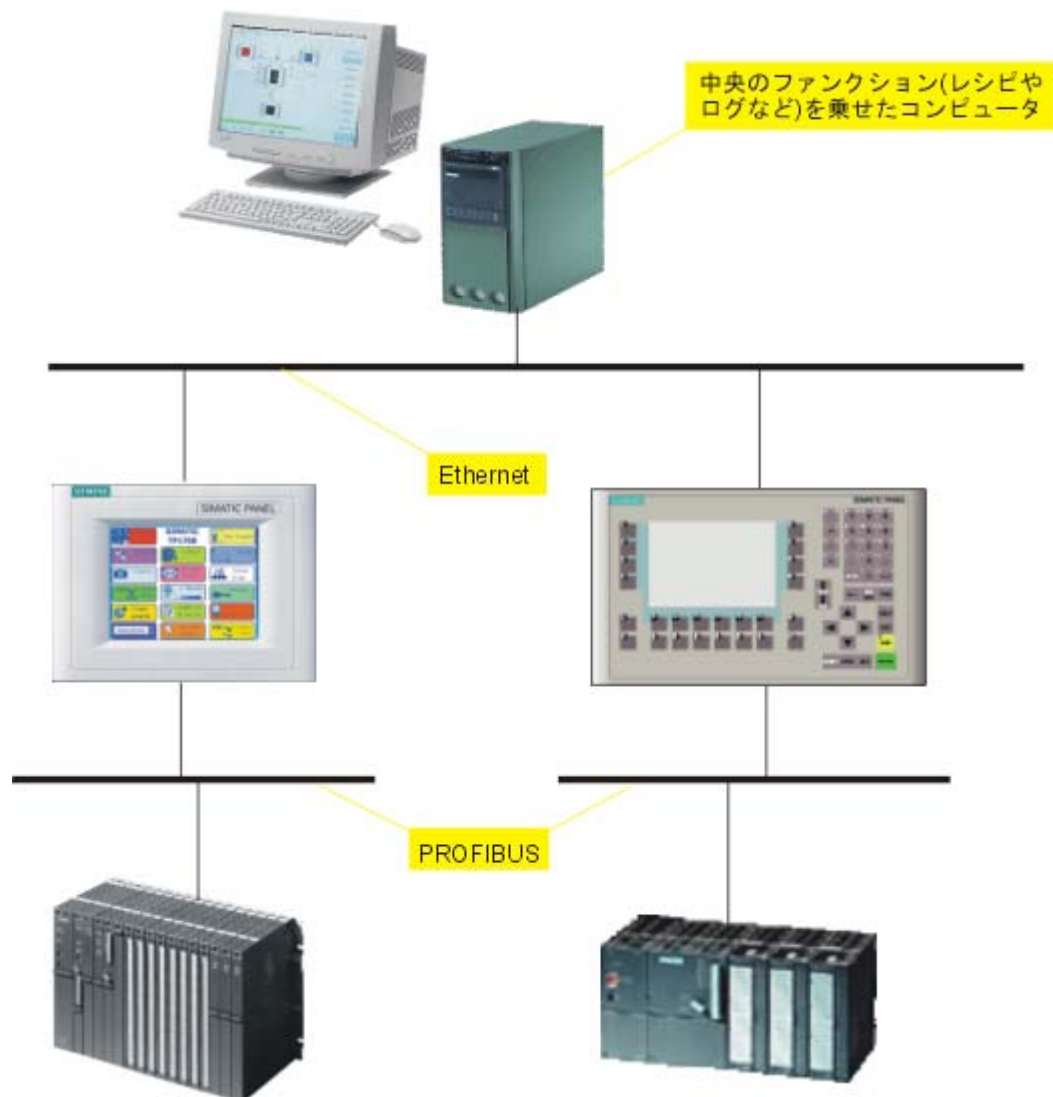
複数の HMI デバイスはプロセスバス(PROFIBUS や Ethernet など)で 1 台以上の PLC に接続されます。



このようなシステムは、たとえば、製造ラインに配置して、複数の箇所からプラントを操作します。

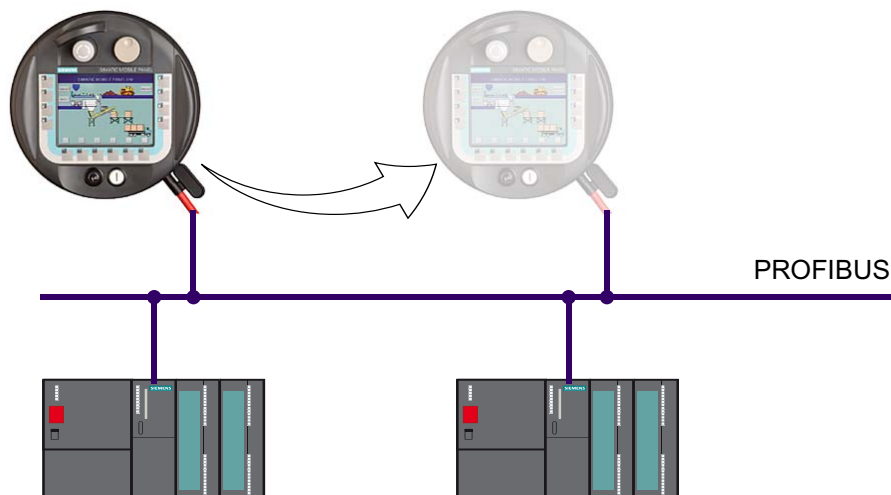
### 集中ファンクションを使用した HMI システム

HMI システムは Ethernet 経由で PC に接続されます。上位の PC はレシピ管理など、中心的な機能を担います。必要なレシピデータレコードは、下位の HMI システムから供給されます。



### モバイルユニットのサポート

モバイルユニットは、大規模な製造プラント、長い製造ライン、またはコンベヤ技術分野で主に使用されていますが、プロセスとの直接の視覚的な監視が必要なシステムでも使用することができます。操作対象のマシンには、Mobile Panel 170などを接続できるインターフェースがいくつか備えられています。



このため、オペレータや保守点検担当者が、現場ですぐに作業をすることができます。これにより、スタートアップ段階での正確なセットアップと位置決めが可能となります。修理の場合、モバイルユニットを使用することによりダウンタイムを短縮することができます。

## 1.3.2 HMI デバイスへのリモートアクセス

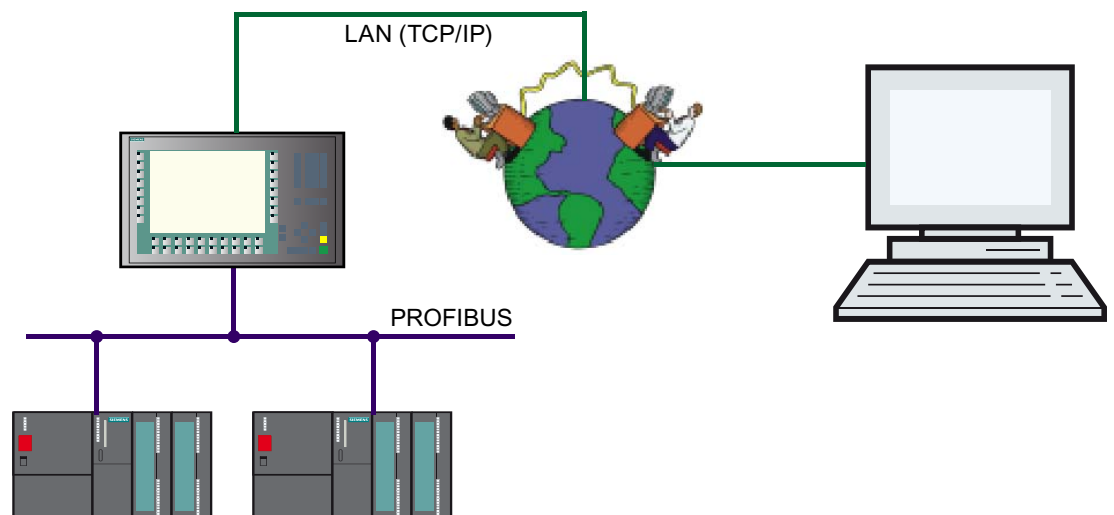
### はじめに

Sm@rtService オプションを使用することにより、ネットワーク(インターネット、LAN)を経由してワークステーションから HMI デバイスに接続することができます。

例:中規模の製造企業で、外部保守会社と保守契約を取り交わしているとします。修理が必要な場合、修理担当者が HMI デバイスにリモートでアクセスして、そのユーザーインターフェースを自分のワークステーションに直接表示することができます。このようにすると、修復済みプロジェクトをより迅速に転送することができ、それがマシンのダウンタイムの短縮にもつながります。

### アプリケーションの可能性

実行には、"Sm@rtService"オプションが必要です。



次のアプリケーションでは、ネットワークを経由したリモートアクセスが可能です。

- リモートオペレーションと監視
  - HMI デバイスを操作すれば、運転中のプロセスを監視することができます。
- リモート管理
  - プロジェクトは、ワークステーションから HMI デバイスに転送することができます。このようにして、プロジェクトを中央から更新することができます。
- リモート診断
  - 各パネルに HTML ページが提供されており、インストール済みのソフトウェア、バージョン、またはシステムアラームに、Web ブラウザを使用してアクセスすることができます。

### 1.3.3 自動アラーム配信

#### はじめに

マシンで故障が発生すると、コストがかかります。タイマーにサービス技術者へ送信されるアラームが予期せぬダウンタイムを最小化します。

例:供給ラインの汚染は、冷却水の流れを減少させます。値が設定された限界値を下回ると、HMI デバイスに警告が表示されます。この警告は同時に電子メールで修理担当者にも配信されます。

#### 原理

インプリメンテーションのために"Sm@rtAccess"オプションが必要です。アラームを電子メールで送信するには、HMI システムに電子メールサーバへのアクセス権がなければなりません。

電子メールクライアントは、イントラネットまたはインターネットを経由してアラームを送信します。自動アラーム配信により、マシンの状態が関係者全員(シフトの責任者やセールスマネージャなど)に適切なタイミングで確実に通知されます。

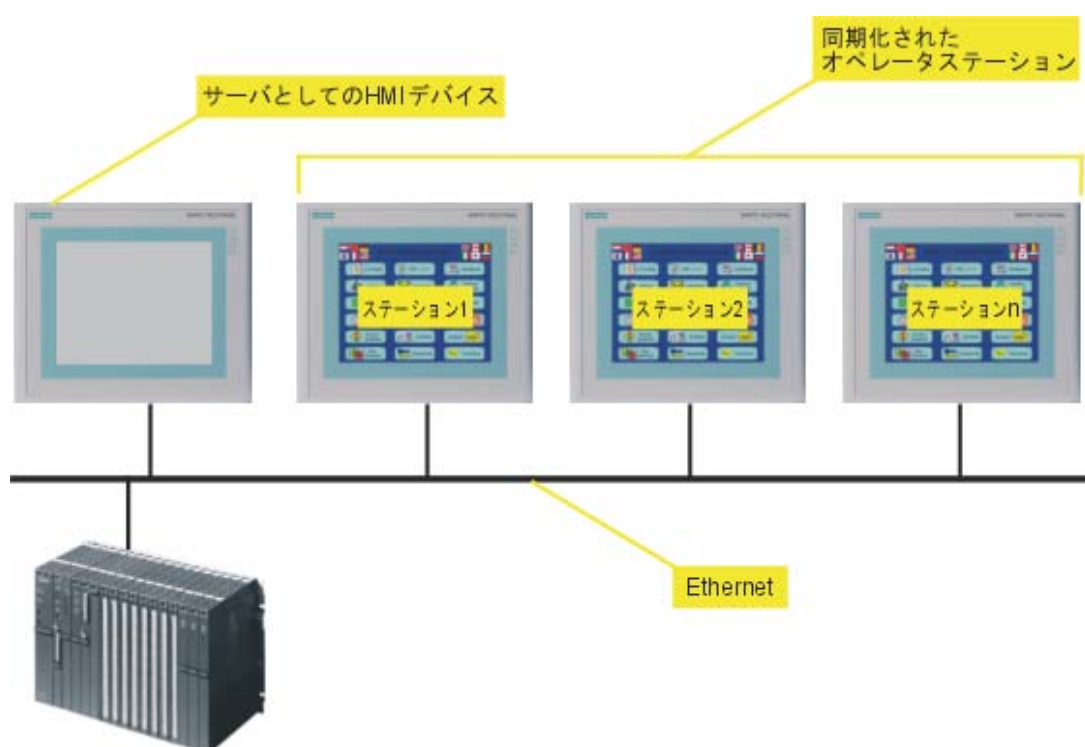
### 1.3.4 分散 HMI

#### はじめに

分散 HMI では、複数の同期化されたオペレーティングステーションから 1 台のマシンを操作することができます。すべてのオペレーティングステーションに同一のプロセス画面が表示されます。オペレーティングオーソリゼーションは、インテリジェントに転送されます。

#### 原理

実装するには、"Sm@rtAccess"オプションが必要です。



1 台の HMI デバイスのみが、サーバーとして、設定データおよびファンクションを格納しています。このサーバーは、他のオペレータデバイスから制御することができます。すべての HMI デバイスに同一の画面が表示されます。

### 1.3.5 パーソナル HMI デバイスのサポート

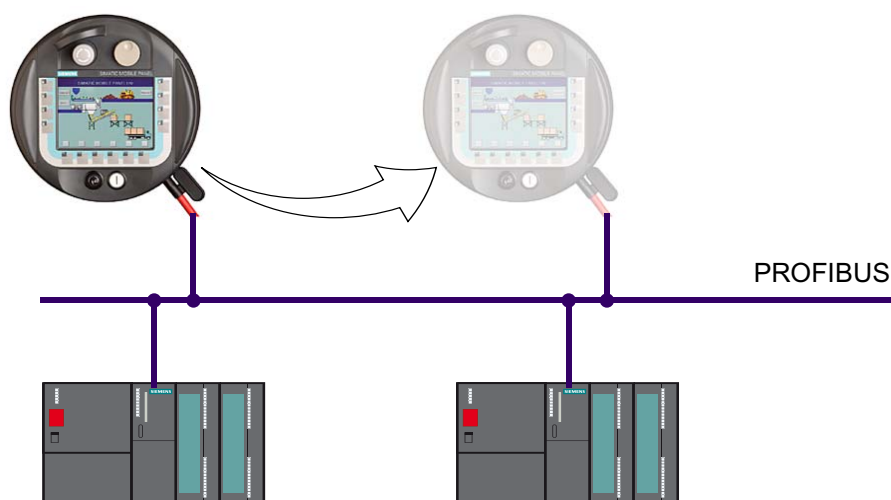
#### はじめに

パーソナル HMI デバイスへのサポートを使用すると、特殊 HMI デバイスとは無関係に、マシンを始動または保守点検することができます。

例: 製造現場には、複数の異なるマシンがあります。そのすべての稼動と保守点検を 1 人の技術員が責任を持っています。その技術員はパーソナル HMI デバイス(PDA、Personal Digital Assistant)を使用しており、この装置で全機械の保守点検業務を実行しています。

#### 原則

実装するには、"Sm@rtAccess"オプションが必要です。



パーソナル HMI デバイスはサーバーを接続して、プロセス画面を表示します。

### 1.3.6 モジュールマシンのコンセプト

#### はじめに

マシン、エレクトロニクスおよびソフトウェア用に承認およびテスト済みのモジュールが、個別のマシンのファンクションおよび複合マシンの作成のスピードアップのために開発されました。モジュールを導入すると、エラー頻度が下がりサイトでのスタートアップ時間を短縮することができ、全体的なコスト削減につながります。

PROFINet をベースにしたコンポーネントベースオートメーション(CBA)により、モジュールシステムおよびマシン組立ての自動化が簡単になります。

#### 原則

WinCC flexible では、モジュールの HMI ウィンドウを作成して、コントロールユニットと組み合わせて完全なモジュールを形成することができます。SIMATIC iMap リンクエディタでは、お互いのモジュールインターフェースをグラフィックを使ってリンクすることができます。通信リンクをプログラミングする必要はありません。モジュールのリンク情報と HMI パーツを使用して、WinCC flexible の視覚化のベースが自動的に生成されます。

## 1.4 コンフィグレーションコンセプト

### 1.4.1 コンフィグレーションサポート

#### はじめに

WinCC flexible を使用してユーザーインターフェースを作成し、マシンやプラントを操作およびモニタすることができます。WinCC flexible では、ソリューションベースのコンセプトとして設定作業のサポートを提供しています。たとえば、これは、大量データ、自動転送または、移動パスのインテリジェント設定の処理にまで関係することがあります。

#### エンジニアリングサポート

効率的な設定は時間と費用を削減します。WinCC flexible は、以下のアラーム手順をサポートしています。

- 対象のシステムに依存する設定  
設定中、選択した対象のシステムでサポートされているファンクションのみが表示されます。
- 使用する PLC に依存しないエンジニアリング  
異なるまたは数個のシステム用のプロジェクトを使用する場合は、プロジェクト内で HMI デバイスのみを切替えます。選択された HMI デバイスで、サポートされていない機能は表示されません。
- 参照オブジェクトの中央での変更  
中央のステーションで変更を行うとそれがプロジェクト全体に適用されます。
- 使用  
設定オブジェクトを再使用することにより、設定を簡単にし、全体的なコストを削減することができます。
- データの一括処理  
たとえば、1つのアクションを作成して、同じ値を持つ複数のタグや時系列昇順のアドレスを設定したとします。
- 画面ナビゲーションのグラフィック設定  
設定された画面のグラフィックオーバービューから画面階層を作成します。画面ナビゲーションに必要なオブジェクトは、自動生成されます。
- 移動パスの設定  
プロセス画面でのオブジェクトの移動をはっきりと見やすく設定します。
- 全体的に統合された自動化  
SIMATIC STEP7 および SIMOTION SCOUT の設定ユーザーインターフェースで、WinCC flexible を円滑に統合できる利点があります。

#### プログラミングユーザーインターフェースのカスタマイズ

WinCC flexible Workbench は、ウィンドウやツールバーを移動したり非表示にしたりしてカスタマイズすることができます。

## 1.4.2 スケーラブルなコンフィグレーションツール

## はじめに

WinCC flexible を使用して異なる HMI デバイスのプロジェクトを編集した場合、設定中に、そのファンクションの範囲が HMI デバイスに適用されます。HMI デバイスによって、使用できるファンクションは異なります。

## HMI デバイスのカスタムファンクション

カスタムデバイスファンクションにより、設定の効率化を図ることができます。それには、特定の HMI デバイスでサポートされるファンクションを設定するだけです。

プロジェクトウィンドウに表示されるエディタは、たとえば、選択した HMI デバイスでサポートされているファンクションをすばやく検出するのに使用できます。



異なる対象のシステムに対して1つのプロジェクトを使用することができます。対象のシステムが変更されると、プロジェクトデータのウィンドウだけが変更されます。対象のシステムが変更されても、作成されたオブジェクトは削除されません。対象のシステムでいくつかの機能がサポートされていない場合は、それらのオブジェクトが隠されて見えないだけです。

## 設定ユーザーインターフェースのカスタムセットアップ

WinCC flexible では、ウィンドウやツールバーの位置や動きをカスタマイズすることができます。これにより、特殊な必要条件を満たすように作業環境を作成できます。

WinCC flexible ワークベンチの設定は、Microsoft Windows にログオンしているユーザーに関連付けされます。プロジェクトを保存するときは、ウィンドウとツールバーの位置と動きも自動的に一緒に保存されます。

そのプロジェクトを再度開くと、ウィンドウおよびツールバーの位置と動作は、プロジェクトを最後に保存したときと同じ状態になっています。その作業環境を開くと、最後に閉じたときの設定と同じになっています。これは、他のプロジェクト企画者によって編集されたプロジェクトを開いたときも同じです。

### 1.4.3 PLC に依存しないコンフィグレーション

#### はじめに

WinCC flexible は目的のシステムに依存しない設定の作成をサポートします。

例: マシンに 3 つのオペレーションステーションがあるものとします。このオペレーティングステーションのうちの 1 台に接続された、平均的なパフォーマンスの HMI デバイスが 1 台あれば、十分に機能を果たします。

#### 原理

プロジェクトはこの HMI デバイスのために再作成する必要はありません。代わりにプロジェクトの HMI デバイスを置き換えるだけで充分です。

HMI デバイスでサポートされないファンクションは見えなくなります。

HMI デバイスを変更した場合は、解像度およびファンクションを大幅に変更することは避けるようにします。

### 1.4.4 使用

#### はじめに

設定オブジェクトを再使用すると、設定作業が楽になります。集中的な編集は、オブジェクトを変更する場合、設定作業を大幅に削減します。

#### 銘板

簡単な画面オブジェクトをフェースプレートとして結合し、複雑なオブジェクトを作成することができます。フェースプレートごとに、画面オブジェクトの変更可能なプロパティを定義することができます。ライブラリに保存されているフェースプレートの再利用により、中央の拠点からプロジェクト全体の変更を行うことができます。

## ライブラリ

すべての設定オブジェクトは、ライブラリで一括保存することができます。さらに、事前に作成された多数の画面オブジェクトが提供されているため、これを使用してプロセス画面を適切に設計することができます。

## テキストライブラリ

テキストライブラリを使用して、複数言語の全設定テキストを保存することができます。プロジェクトが複数言語で設定されている場合、テキストを自動的に変換することができます。

### 1.4.5 インテリジェントツール

#### 1.4.5.1 データの一括処理

## はじめに

一括データ管理により、複数のオブジェクトを同時に作成して、編集することができるようになります。コンフィグレーションがより効率的になり、時間とコストを削減することができます。

例: タグインベントリの一部が古いプロジェクトのものであり、そのインベントリのタグタイプが正しくないとします。WinCC flexible を使用すると、一度の手順ですべてのタグのタグタイプを変更することができます。

## 原理

特定のオブジェクト(タグなど)を作成および編集する際に、データの一括処理を利用することができます。

- 自動アドレス割り付け

プロセスのリンクにより作成された複数のタグがコントローラのメモリに連続して保存されている場合、タグごとにアドレス領域を自動的に増やすことができます。

- 複数の変更

タグタイプの変更やコントローラの変更など、複数のタグに同じ変更を加える場合、一度の手順で行うことができます。

### 1.4.5.2 移動パスのコンフィグレーション

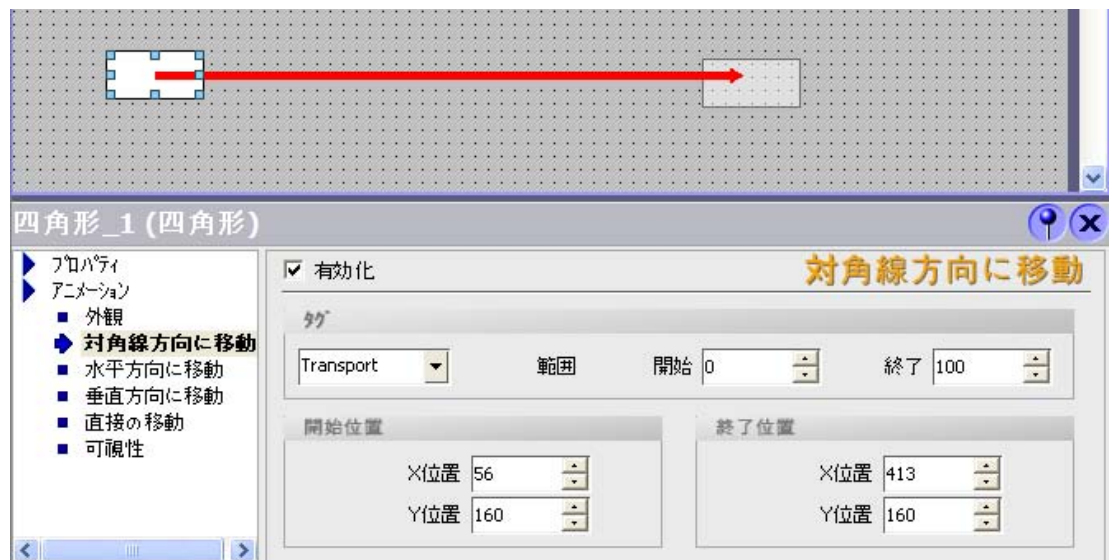
#### はじめに

ベルトコンベヤ上の製品の移動など、オブジェクトの移動を含むプロセスシーケンスも HMI デバイスに明確に表示することができます。

移動パスにより、プロセス画面に表示されるオブジェクト移動のコンフィグレーションが簡単になります。移動プロセスは画面上に図形式で表示されます。

#### 原理

オブジェクトの移動パスは、プロセス画面で定義します。移動パスは、開始ポイントと終了ポイントで構成され、タグが割り付けられます。このタグの値により、ランタイム時の移動パス上にあるオブジェクトの相対位置が定義されます。



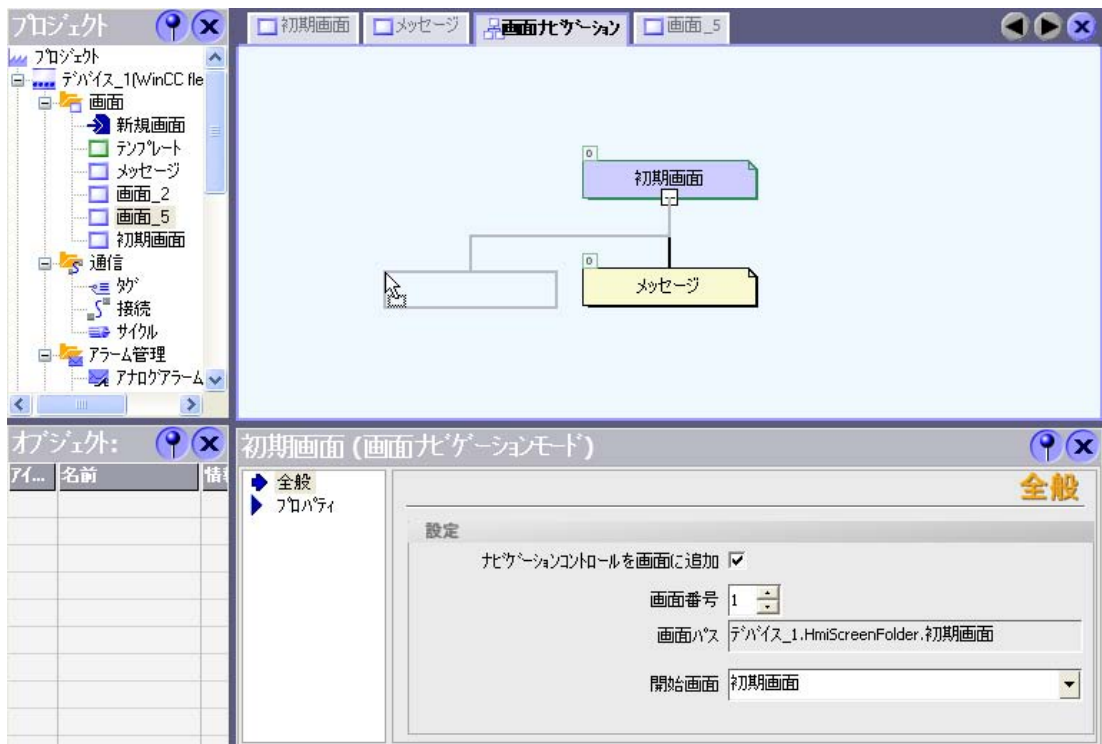
## 1.4.5.3 画面ナビゲーションのグラフィックコンフィグレーション

## はじめに

画面ナビゲーションとはプロセス画面用に作成された階層を意味します。画面ナビゲーションを使用すると、プロジェクトに、固定されたナビゲーション構造体が定義されます。オペレータは、ランタイムでナビゲーションコントロールを使用して構造体の様々な画面をナビゲーションすることができます。

## 原理

画面ナビゲーションエディタを使用して、ドラッグアンドドロップ操作により、画面の階層構造上の適切な位置に画面を配置します。階層に統合されていない画面間のリンクを作成することもできます。ナビゲーションボタンは、プロセス画面に貼り付けることができます。



ナビゲーション構造体を作成すると、次のような利点があります。

- プロジェクト全体にわたるナビゲーション構造体の概要を把握できる。
- プロセス画面間に直接リンクをすばやく作成できる。
- 基本的な画面ナビゲーションであれば自動作成できる。

## 1.4.6 トータルインテグレートドオートメーション

### はじめに

全自動ソリューションでは、WinCC flexible など HMI システムだけでなく、PLC、プロセスバスおよび周辺機器などその他のコンポーネントも含まれます。

WinCC flexible と、SIMATIC および SIMOTION 製品のコンポーネントを組み合わせると、特に洗練された統合システムを実現することができます。

- 一貫性のある設定およびプログラミング
- 一貫性のあるデータ保存
- 一貫性のある通信

### SIMATIC STEP 7 との統合

プロセスタグにより、PLC と HMI システム間の通信用リンクが提供されます。自動化を全体的に統合しないと、各タグを 2 回定義しなければなりません。PLC 用に 1 回、HMI システム用に 1 回必要です。

設定ユーザーインターフェースで SIMATIC STEP 7 の統合を行うことにより、エラーの発生頻度を抑えて設定作業を簡素化することができます。設定中、STEP7 アイコンテーブルと通信設定に直接アクセスされます。

- STEP 7 シンボルテーブルには、コントロールプログラムの作成中に定義されたデータベース定義(たとえば、アドレスやデータタイプ)が含まれます。
- 通信設定にはバスアドレスと PLC プロトコルが含まれます。通信設定は NetPro で作成されます。

### SIMOTION SCOUT との統合

SIMOTION SCOUT を WinCC flexible に統合すると、SIMATIC STEP 7 の統合の利点を活用できるだけでなく、SIMOTION-SCOUT のユーザーインターフェースの完全統合もできます。



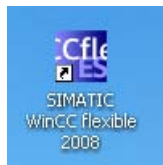
## WinCC flexible エンジニアリングシステム

### 2.1 プログラミングインターフェースに関する基本原理

#### 原理

WinCC flexible は、快適で効率の良いエンジニアリングを伴う、将来を保証されたマシン指向の自動化コンセプトの HMI ソフトウェアです。

選択した HMI デバイスがサポートしているすべてのファンクションにアクセスできます。WinCC flexible を開始するには、プログラムデバイスでデスクトップアイコンをクリックするか、Windows の[スタート]メニューからそのアイコンを選択します。



WinCC flexible では、常に 1 つのプロジェクトだけを開くことができます。WinCC flexible を必要な回数だけ開いて、複数のプロジェクトで同時に作業することができます。

---

#### 注記

WinCC では、同じプロジェクトで複数の HMI デバイスを作成することもできます。

---

## 2.2 WinCC flexible ユーザーインターフェース

### 2.2.1 WinCC flexible ユーザーインターフェースの要素

#### はじめに

WinCC flexible 作業環境はいくつかの要素で構成されます。要素の一部は固有のエディタにリンクされています。つまり、その要素は対応するエディタが動作中でない限り表示されないということです。

#### 注記

ダイアログボックスのレイアウトはコントロールパネルの表示設定により異なります。テキストは設定によっては切り詰められます。

ご使用の設定 PC のオペレーティングシステムを"標準サイズ(96 dpi)"に設定します。コントロールパネルから[表示]>[設定]>[詳細]>[全般]>[DPI 設定]を選択してこの設定を使用できます。

#### WinCC flexible の要素

WinCC flexible は以下の要素で構成されます。



## メニューとツールバー

メニューとツールバーを使用して、WinCC flexible で用意されているすべてのファンクションにアクセスできます。マウスポインタがファンクション上に移動されると、ツールヒントが表示されます。

## 作業エリア

プロジェクトオブジェクトは、作業エリアで編集されます。WinCC flexible のすべてのエレメントは、作業エリアの境界線に整列されます。作業エリアは例外として、ユーザー個別の要望に応じて、任意のエレメントを編成、作成、移動、非表示にすることができます。

## [プロジェクト]ウィンドウ

プロジェクトで使用可能な構成エレメントとエディタのすべてが、[プロジェクト]ウィンドウにツリー構造で表示されます。オブジェクトを構造化して保存できるように、各エディタのサブエレメントとしてフォルダが提供されています。さらに、画面、レシピ、プロトコルおよびユーザー辞書に対して作成されたオブジェクトへ直接アクセスできます。プロジェクトウィンドウで、HMI デバイスのデバイス設定、言語設定、バージョン管理にアクセスできます。

## [プロパティ]ウィンドウ

[プロパティ]ウィンドウは、画面オブジェクトの色などのオブジェクトプロパティを編集するために使用されます。[プロパティ]ウィンドウは特定のエディタでのみ使用可能です。

## ツールボックス

ツールボックスには、グラフィックオブジェクトやオペレータ制御エレメントなど、画面に追加できるオブジェクトの選択肢が含まれています。さらに、ツールボックスは、オブジェクトテンプレートとフェイスプレートの集合を含むライブラリを提供しています。

## ライブラリ

"ライブラリ"は、[ツールボックス]ウィンドウのエレメントです。"ライブラリ"は、画面オブジェクトテンプレートへのアクセスを提供します。ユーザーは、オブジェクトテンプレートの複数回使用または再利用によって、常に画面オブジェクトを追加して、プログラミングの効率を向上させることができます。ライブラリは、画面オブジェクト、タグなどの頻繁に使用するオブジェクトを保存するための中央データベースです。

## [出力]ウィンドウ

[出力]ウィンドウには、たとえば、プロジェクトテストの実行で生成されるシステムアラームが表示されます。

## [オブジェクト]ウィンドウ

[オブジェクト]ウィンドウには、[プロジェクト]ウィンドウから選択されたエリアのすべてのエレメントが表示されます。

### 注記

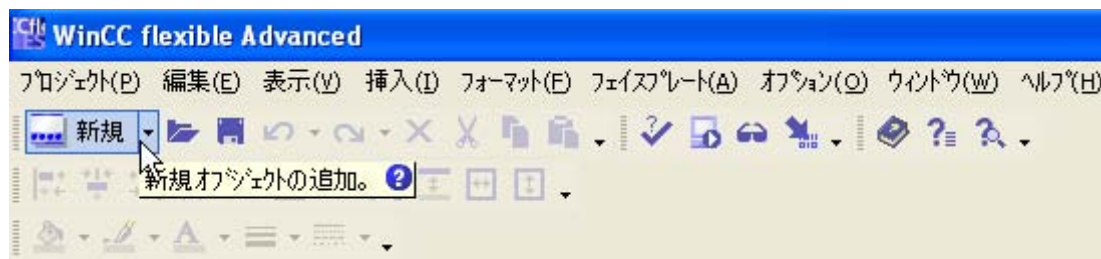
作業領域は例外として、[表示]メニューですべてのウィンドウを表示したり非表示にしたりすることができます。

## 2.2.2 メニューとツールバー

### はじめに

メニューとツールバーから、HMI デバイスを作成するのに必要なすべてのファンクションにアクセスできます。該当するエディタが有効になると、そのエディタに固有のメニューコマンドとツールバーが表示されます。

マウスポインタをコマンド上に移動すると、対応するツールヒントが表示されます。



### ツールバーの配置

メニューとツールバーは、新規プロジェクトを作成するときに、画面の上端に標準で位置付けられます。メニューとツールバーの位置は、Windows にログオンしているユーザーによって決定されます。マウスを使用してツールバーを移動すると、WinCC flexible を再起動したとき、ツールバーが最終の'終了'位置に復帰します。

## メニュー

WinCC flexible で使用可能なメニュー:

メニュー	簡単な説明
"プロジェクト"	プロジェクト管理用のコマンドが含まれています。
[編集]	クリップボードと検索機能のコマンドが含まれています。
[表示]	エレメントを開いたり閉じたりするコマンドや、ズーム/レイヤー設定用のコマンドが含まれています。閉じたエレメントを再度開くには、[表示]メニューを選択します。
[貼り付け]	新規のオブジェクトを貼り付けるためのコマンドが含まれています。
[フォーマット]	画面オブジェクトの整列とフォーマットのためのコマンドが含まれています。
[フェースプレート]	フェースプレートの作成と編集のためのコマンドが含まれています。
[ツール]	ユーザーインターフェース言語を変更し、WinCC flexible の基本設定などを構築するためのコマンドが含まれています。
[スクリプト]	同期およびスクリプトの構文チェックコマンドが含まれています。
[ウィンドウ]	他のウィンドウへの切り替えなど、作業エリアでの複数ウィンドウ管理のためのコマンドが含まれています。
[ヘルプ]	ヘルプ機能を呼び出すためのコマンドが含まれています。

メニューの利用可能性とコマンドの範囲は、使用している各エディタによります。

## ツールバー

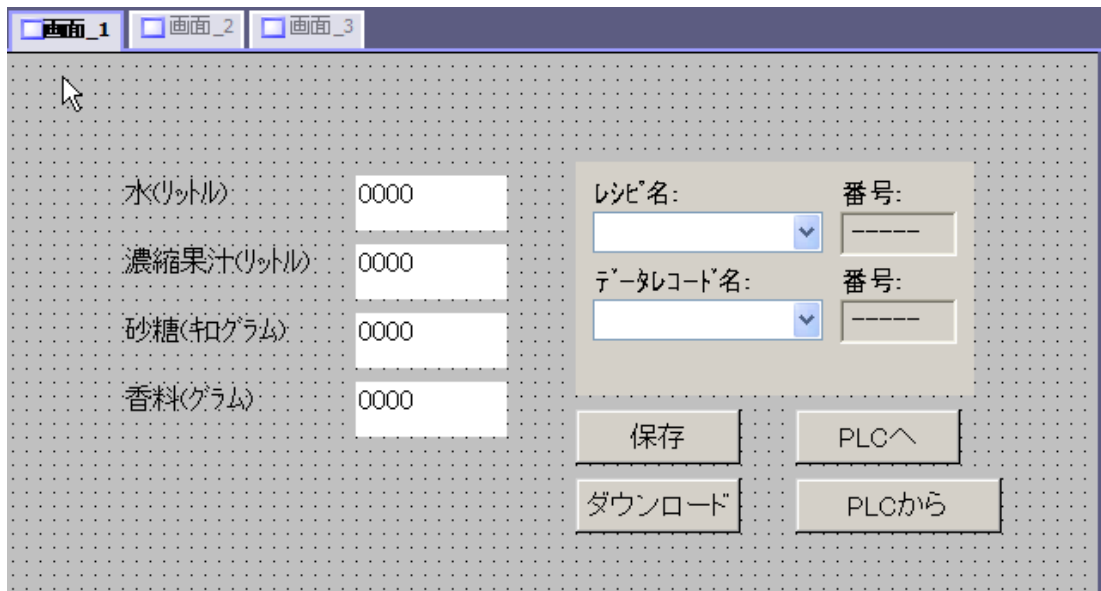
ツールバーは、頻繁に使用される重要なファンクションへの素早いアクセスを提供します。以下のツールバー設定オプションが使用できます。

- ボタンの追加と削除
- 位置の変更

### 2.2.3 作業エリア

#### はじめに

作業エリアは、タグなどのテーブルフォーマット、またはプロセス画面などのグラフィックフォーマットのプロジェクトデータを編集するために使用されます。



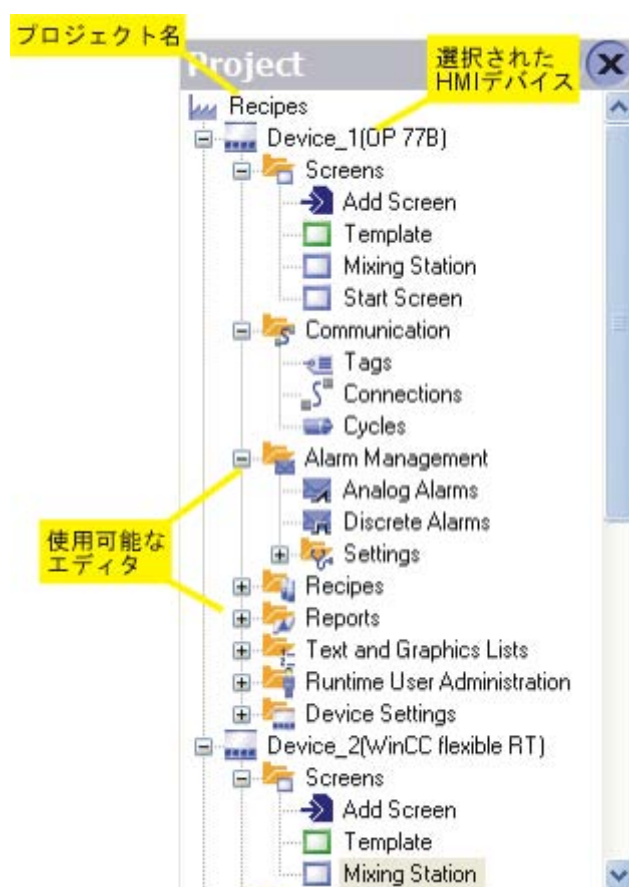
#### 説明

各エディタは、作業エリア上の別のタブコントロールで開きます。グラフィックエディタの場合は、各エレメントは別のタブコントロール上に表示されます。複数のエディタが同時に開いている場合は、1つのタブだけが有効です。別のエディタに移動するには、該当するタブをクリックします。最高 20 のエディタを同時に開くことができます。

## 2.2.4 [プロジェクト]ウィンドウ

### はじめに

[プロジェクト]ウィンドウはプロジェクトを編集する際の中央制御ポイントです。プロジェクトで使用可能な構成要素とエディタのすべてが、[プロジェクト]ウィンドウにツリー構造で表示されます。各エディタにはシンボルが割り付けられ、対応するオブジェクトを識別するために使用できます。選択された HMI デバイスでサポートされているエレメントのみがプロジェクトウィンドウに表示されます。プロジェクトウィンドウで、HMI デバイスのデバイス設定、言語設定、バージョン管理にアクセスできます。



### 説明

[プロジェクト]ウィンドウには、プロジェクトストラクチャが階層的に表示されます。

- プロジェクト
- HMI デバイス
- フォルダ
- オブジェクト

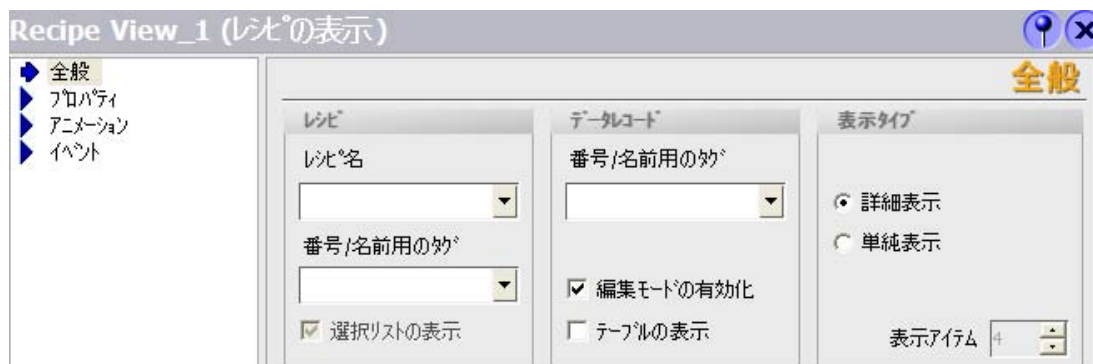
[プロジェクト]ウィンドウは、編集用のオブジェクトを作成して開くために使用されます。プロジェクトオブジェクトをフォルダに整理して、ストラクチャを作成できます。[プロジェクト]ウィンドウの取り扱いは、Windows Explorer の取り扱いと同じです。最も重要なコマンドから構成されるショートカットメニューは、すべてのオブジェクト用に使用可能です。

グラフィックエディタの元素は、[プロジェクト]ウィンドウと[オブジェクト]ウィンドウに表示されます。"テーブルエディタ"の元素は、[オブジェクト]ウィンドウのみ表示されます。

## 2.2.5 [プロパティ]ウィンドウ

### はじめに

[プロパティ]ウィンドウは、作業エリアから選択したオブジェクトのプロパティを編集するために使用されます。[プロパティ]ウィンドウの内容は、選択したオブジェクトに基づいています。



### 説明

[プロパティ]ウィンドウには、カテゴリに整理された選択済みオブジェクトのプロパティが表示されます。変更された値は、入力フィールドから抜け出た直後に有効になります。

無効なエントリは色付きの背景で強調表示されます。入力の修正を支援するためにツールヒントが表示されます。

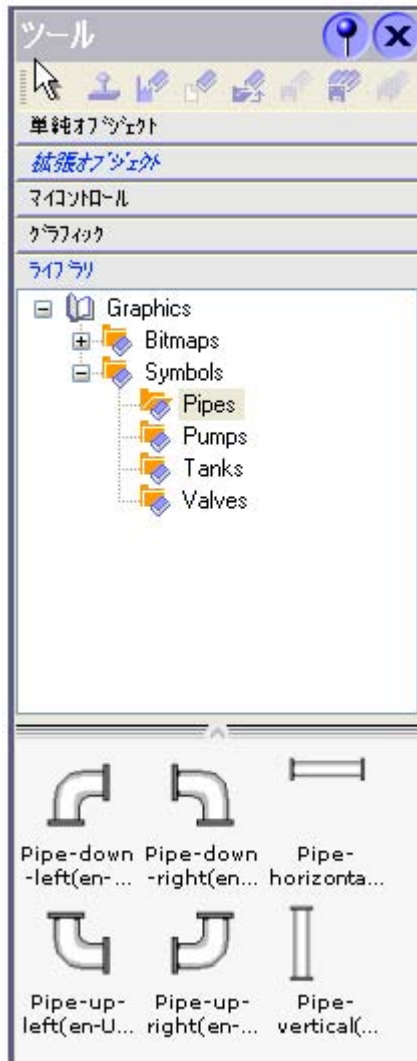
### 例

オブジェクトプロパティ"高さ"は、"Byte"変数に論理的にリンクされています。このタグタイプは、0~255の値の範囲があります。[プロパティ]ウィンドウの[高さ]入力ボックスに値[300]を入力すると、その値はボックスの入力を終了したときに、色付きの背景で強調表示されます。

## 2.2.6 ライブラリ

### はじめに

"ライブラリ"は、[ツールボックス]ウィンドウの要素です。ライブラリはツールボックスウィンドウの要素です。ライブラリは、頻繁に必要とされるオブジェクトを保存するための中央データベースです。ライブラリに保存されたオブジェクトは、一度だけ設定する必要があります。すると、そのオブジェクトを好きな回数だけ再利用できます。ユーザーは、オブジェクトテンプレートの複数回使用または再利用によって、常に画面オブジェクトを追加して、プログラミングの効率を向上させることができます。



### 説明

WinCC flexible は、グローバルライブラリとプロジェクトライブラリを区別します。

- 共有ライブラリ  
グローバルライブラリは、プロジェクトデータベースに保存されません。ファイルに書き込まれます。ファイルは、WinCC flexible のインストールディレクトリにデフォルトで保存されます。グローバルライブラリは、すべてのプロジェクトに利用できます。
- プロジェクトライブラリ  
プロジェクトライブラリは、データベースにプロジェクトデータと共に保存され、プロジェクトライブラリが作成されたプロジェクトでのみ使用可能です。

両方のライブラリにフォルダを作成して、フォルダに含まれているオブジェクト用のストラクチャを生成できます。さらに、プロジェクトライブラリからグローバルライブラリにエレメントを常にコピーすることができます。

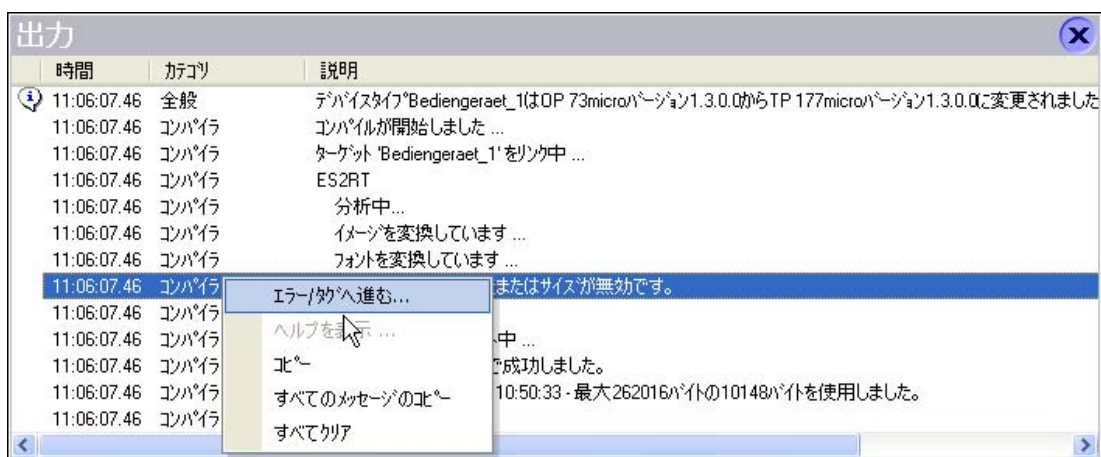
### 別のウィンドウでライブラリを開く

[ツールボックス]ウィンドウから別のウィンドウにライブラリをスワップすることができます。これを行うには、[ライブラリ]ウィンドウのショートカットメニューの[ツールボックスのライブラリ]コマンドを選択します。ライブラリを[ツールボックス]ウィンドウに復元するためにこのコマンドを再度選択します。

## 2.2.7 [出力]ウィンドウ

### はじめに

[出力]ウィンドウには、たとえば、プロジェクトテストの実行で生成されるシステムアラームが表示されます。



## 説明

[出力]ウィンドウには、システムアラームが発生する順序でそのシステムアラームが通常表示されます。このカテゴリは、システムアラームを生成した WinCC flexible モジュールを定義します。たとえば、一貫性チェック中には"Generator"カテゴリのシステムアラームが生成されます。

システムアラームをソートするには、対応する列のヘッダーをクリックします。ポップアップメニューを使用して、エラー箇所やタグにジャンプしたり、システムアラームのコピーや削除を行うこともできます。

[出力]ウィンドウには、最後のアクションのシステムアラームがすべて表示されます。新規のアクションが実行されると、前のシステムアラームはすべて上書きされます。別のログファイルから、古いシステムアラームを取り出すことができます。

## 2.2.8 [オブジェクト]ウィンドウ

### はじめに

フォルダまたはエディタが[プロジェクト]ウィンドウで選択された場合、その内容は[オブジェクト]ウィンドウに表示されます。

以下の図は、[プロジェクト]ウィンドウでの選択が[オブジェクト]ウィンドウでの表示にどのように影響を及ぼすかを示します。



## 説明

[オブジェクト]ウィンドウでオブジェクトをダブルクリックして、対応するエディタを開きます。ドラッグアンドドロップファンクションは、オブジェクトウィンドウに表示されているすべてのオブジェクトに利用できます。

サポートされているドラッグアンドドロップ動作を以下に示します。

- 作業エリアのプロセス画面にタグを移動: タグにリンクされている I/O フィールドを作成します。
- タグを既存の I/O フィールドに移動: タグと I/O フィールドの間の論理リンクを作成します。
- プロセス画面を作業エリアのほかのプロセス画面に移動: プロセス画面にリンクされている画面切り替えファンクション付きボタンを生成します。

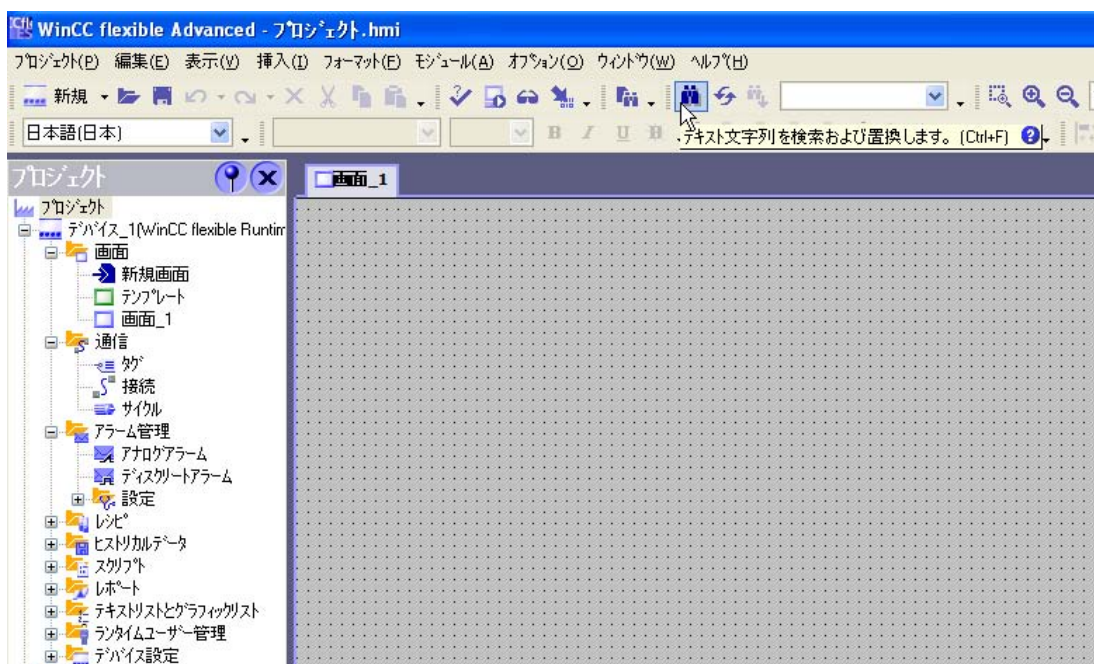
長いオブジェクト名は、[オブジェクト]ウィンドウで短縮されます。マウスポインタをオブジェクトに移動した後、フルネームがツールヒントとして表示されます。

多数のオブジェクトが利用できる場合、オブジェクトの最初の文字を入力するだけで、必要なオブジェクトを手早く見つけ出すことができます。

## 2.3 エディタ固有のオペレーティングエレメントの配置

### はじめに

エディタ固有のオペレーティングエレメントは、対応するエディタが動作中の作業エリアでのみ表示されます。



エディタ固有のオペレーティングエレメントには以下のものが含まれます。

- ツールバー
- ツールボックス
- メニューコマンド

### 配置

エディタ固有のツールバーのデフォルト位置は、右側または既存のツールバーの下です。

エディタ固有の[ツールボックス]ウィンドウのデフォルト位置は、右側の画面マージンです。

エディタ固有のコマンドは対応するメニューに追加されます。

エディタ固有のオペレーティングエレメントの位置は、個々の要求に合うように以前のセッションで配置変えた場合は、WinCC を次回起動したときに元に戻ります。

## 2.4 ウィンドウとツールバーの取り扱い









### はじめに

WinCC flexible では、フレームとツールバーのレイアウトをカスタマイズできます。作業領域を拡大するために、頻繁に使わない特定のフレームを非表示にすることができます。

[表示]メニューを使用して、フレームとツールバーのデフォルトレイアウトを回復することができます。

### 使用可能なオペレーティングエレメント

下記の表に、フレームとツールバーのオペレーティングエレメントおよびそれらの使用目的を示します。

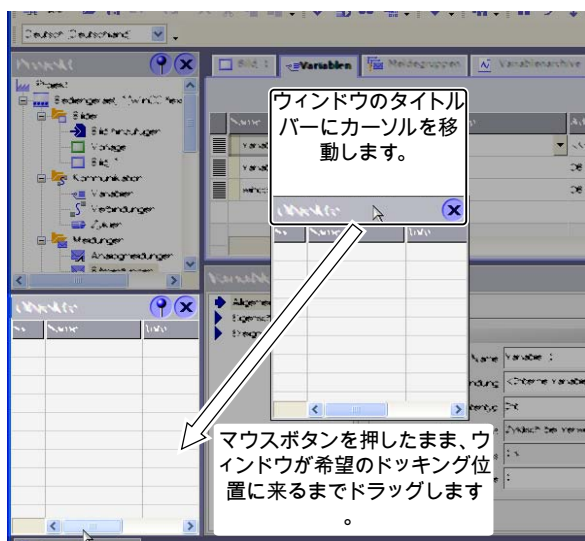
オペレータ制御エレメント	目的	使用場所
	フレームまたはツールバーを閉じる	フレームとツールバー(移動可能)
  	フレームとツールバーをドラッグアンドドロップを使用して移動、ドッキングさせる	フレームとツールバー(移動可能)
	ドラッグアンドドロップによってツールバーを移動する	ツールバー(ドッキングされた)
	ツールバーアイコンを追加、削除する	ツールバー(ドッキングされた)
	ウィンドウ用の自動非表示モードを有効にする。	フレーム(ドッキングされた)
	フレーム用の自動非表示モードを無効にする	フレーム(ドッキングされた)

### フレームとツールバーのドッキング

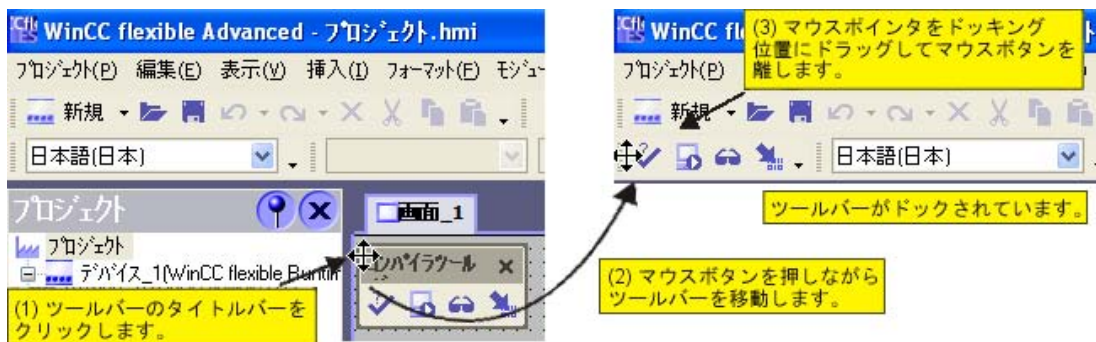
"ドッキング"とは、ウィンドウを WinCC flexible ワークベンチに統合することを言います。作業スペースを増やすために、ドッキングされたフレームを自動的に非表示にすることができます。

自由に移動可能なウィンドウをウィンドウの以下の位置にドッキングさせることができます。

- 上端
- 右端
- 下端
- 左端



任意の既存のツールバーにツールバーをドッキングさせることができます。

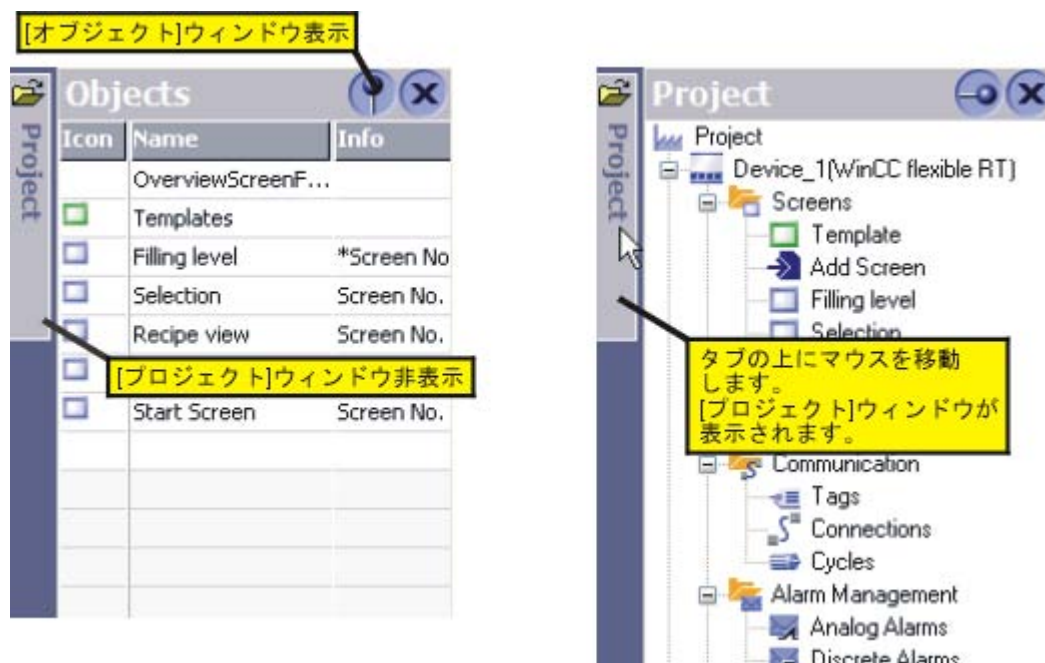


### 結合されたフレーム

1つのフレームを別のフレームと組み合わせることができます。各フレームは、結合されたフレームに別のタブで再表示されます。異なるフレームに変更するには、対応するタブをクリックします。

## ウィンドウを自動的に非表示にする

頻繁には必要としないウィンドウを自動的に非表示にすることができます。これにより、作業エリアが増えます。このウィンドウを画面に復元するには、そのタイトルバーをクリックします。



## 2.5 マウスの取り扱い



### はじめに

WinCC flexible では、作業は主にマウスを使用して完了します。このコンテキストでの重要なオペレーティング機能は、ドラッグアンドドロップ機能とショートカットメニューからのコマンドの呼び出しです。

### ドラッグアンドドロップ

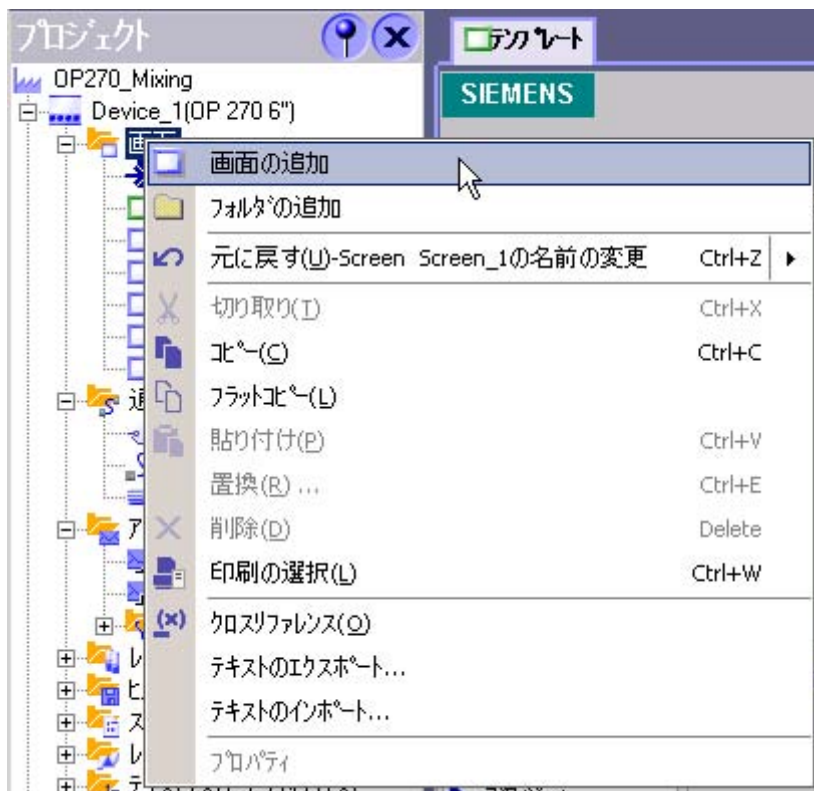
ドラッグアンドドロップにより、設定がずっと楽になります。たとえば、[オブジェクト]ウィンドウからプロセス画面などに変数をドラッグアンドドロップする場合、システムがそのタグに論理的にリンクされている I/O フィールドを自動的に生成します。画面切り替えを作成するには、必要なプロセス画面を作業エリアに表示されているプロセス画面にドラッグアンドドロップします。これにより、対応する画面変更機能が含まれるように設定されたボタンが生成されます。

ドラッグアンドドロップ機能は、[プロジェクト]ウィンドウと[オブジェクト]ウィンドウのすべてのオブジェクトに使用可能です。マウスポインタにより、保存先でドラッグアンドドロップがサポートされているかどうかを示されます。

-  ドラッグアンドドロップが可能です。
-  ドラッグアンドドロップは不可能です。

ショートカットメニュー

WinCC では、任意のオブジェクトを右クリックしてショートカットメニューを開くことができます。ショートカットメニューには、関係のある状況で実行できるコマンドが含まれています。



概要: マウス機能

機能	効果
左クリック	任意のオブジェクトを起動、または、メニューコマンドやドラッグアンドドロップなどのアクションを実行する。
右クリック	ショートカットメニューを開く。
ダブルクリック(マウスの左ボタン)	[プロジェクト]ウィンドウまたは[オブジェクト]ウィンドウでエディタを起動するか、フォルダを開く。
<マウスの左ボタン+ドラッグアンドドロップ>	[プロジェクト]ウィンドウにオブジェクトのコピーを生成する。
<CTRL+マウスの左ボタン>	[オブジェクト]ウィンドウからいくつかの個々のオブジェクトを次々と選択する。
<SHIFT+マウスの左ボタン>	[オブジェクト]ウィンドウでマウスを使って描画した四角形の枠内のすべてのオブジェクトを選択する。

## 2.6 キーボードコントロール

### はじめに

WinCC flexible には、必要なメニューコマンドを頻繁に実行するのに使用する多くのホットキーが用意されています。メニューには、ホットキーが関係のあるコマンドで使用可能であるかどうかを示されます。

WinCC では、Windows で提供されているすべての標準ホットキーも統合されます。

### 重要なホットキー

次の表には、WinCC flexible で使用する最も重要なホットキーが示されています。

ホットキー	効果
<Ctrl+Tab>/<Ctrl+Shift+Tab>	作業エリアで次の/前のタブを実行する。
<Ctrl+F4>	作業エリアの[アクティブ]ウィンドウを閉じる。
<Ctrl+C>	選択したオブジェクトをクリップボードにコピーする。
<Ctrl+X>	オブジェクトを切り取り、クリップボードにコピーする。
<Ctrl+V>	クリップボードに保存されたオブジェクトを挿入する。
<Ctrl+F>	[検索と置換]ダイアログボックスを開く。
<Ctrl+A>	有効なエリアのすべてのオブジェクトを選択する。
<ESC>	アクションをキャンセルする。

## 2.7 WinCC flexible の取り扱い

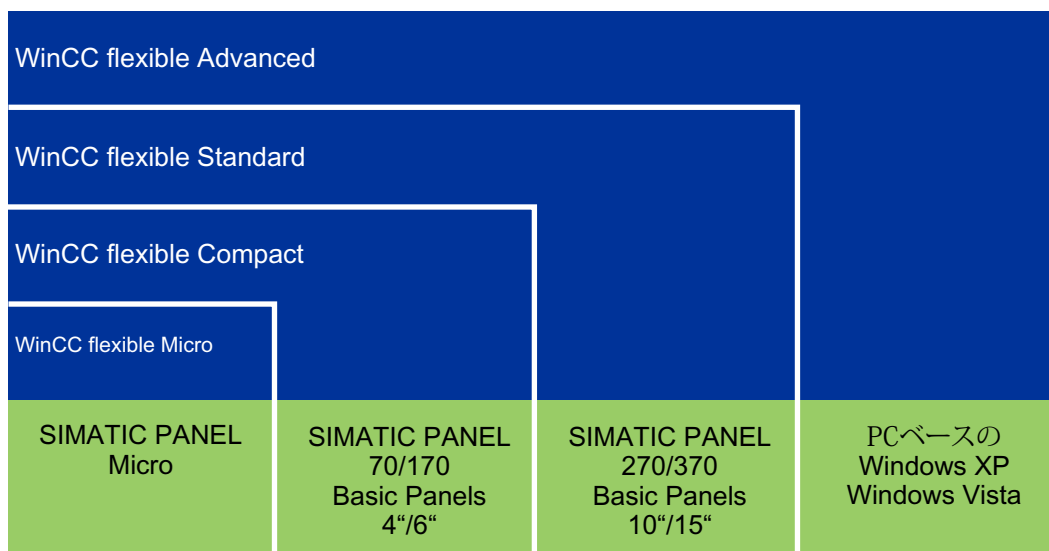
### 2.7.1 WinCC flexible の取り扱い

#### はじめに

WinCC flexible は、それぞれのコンフィギュレーションタスクに最適な状態に調整されているか、またはユーザーで調整できる、さまざまな拡張性のあるエンジニアリングシステムを提供します。各エディションは広範囲の HMI デバイスと機能をサポートしており、「標準」エディションを使用して、「マイクロ」エディションから HMI デバイスを作成することができます。また、パワーパックによって上位エディションにいつでも移行することができます。

#### 個々のエディションの機能範囲

WinCC flexible は、以下のエディションで使用可能です。



## 2.7.2 プロジェクトの取り扱い

### はじめに

WinCC flexible を使用してユーザーインターフェースを作成し、マシンやプラントを操作およびモニタすることができます。

さまざまなコンフィグレーションタスク用に、特殊なエディタが用意されています。コンフィグレーション情報はすべて、プロジェクトに保存されます。

### プロジェクトの作成またはロード

WinCC flexible の起動後は、ウィザードに従って新規プロジェクトの作成に必要な手順を実行することができます。例えば、ユーザーはプロジェクトの名前を入力し、HMI デバイスを選択するよう指示されます。

WinCC flexible が既に開いている場合は、[新規作成]コマンドを選択して新規プロジェクトを作成します。;WinCC flexible ガスデニヒライテイルバアイハ、シンキサクセイコマンドラセンタクシテシンキプロジェクトヲサクセイシマス状況によっては、ウィザードが表示されプロセスを通してガイドします。

既存のプロジェクトをロードするには、[プロジェクト]メニューから[開く]コマンドを選択します。

### デバイスベース依存性

WinCC flexible は、選択された HMI デバイスがサポートする機能しか実現しません。[プロジェクト]ウィンドウに、コンフィグレーションに使用できるエディタが表示されます。

### 移行

WinCC flexible で既存の ProTool または WinCC プロジェクトが開いている場合、データが変換されます。ユーザーは表示される手順に従って変換プロセスを実行します。変換の進捗状況も表示されます。

## 2.7.3 WinCC flexible による複数プロジェクトの編集

### 原理

WinCC flexible では、常に 1 つのプロジェクトだけを開いて編集することができます。例えば、複数のプロジェクトを一括してコピーする必要がある場合は、WinCC flexible を再起動してから必要なプロジェクトを開きます。

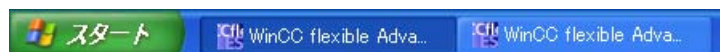
---

#### 注記

ProTool と WinCC flexible が両方とも PC にインストールされている場合、一度に開くことができるプログラムはどちらか一方に限られます。

---

複数の HMI デバイスを、各プロジェクトでパラレルにセットアップすることができます。開いている WinCC flexible は、それぞれ Windows のタスクバーに表示されます。



### 2.7.4 プロジェクトの機能範囲

#### はじめに

WinCC flexible を使って異なった HMI デバイスのプロジェクトを編集する場合、機能範囲はコンフィグレーションと一致しません。HMI デバイスによって、使用できるファンクションは異なります。

#### 原理;ゲンリ

使用できる機能範囲は、選択した HMI デバイスによって異なります。選択した HMI デバイスがサポートしている機能だけを組み込みます。この手順は、効率的なコンフィグレーションに有用です。

プロジェクトウィンドウに表示されるエディタは、たとえば、選択した HMI デバイスでサポートされているファンクションをすばやく検出するのに使用できます。

図は、[プロジェクト]ウィンドウによる 2 つの異なる HMI デバイスの機能範囲を示しています。



## 2.7.5 エディタのプロパティ

### はじめに

WinCC flexible では、それぞれの設定タスクに対応する固有のエディタが用意されています。WinCC flexible は 2 つの異なるエディタを区別しています。グラフィックエディタおよびテーブルエディタ 最高 20 のエディタを同時に開くことができます。

### グラフィックエディタ

スクリーンエディタのようなグラフィックエディタは、[プロジェクト]ウィンドウおよび[オブジェクト]ウィンドウの両方に属すエレメントを表示します。グラフィックエディタで作業エリアの各オブジェクトを開きます。

### テーブル形式エディタ

"タグ"エディタなどの"テーブル"エディタは、[オブジェクト]ウィンドウの関連するオブジェクトだけを表示します。オブジェクトを編集するためにテーブルエディタを開くと、すべての関連するオブジェクトが作業エリアにテーブル形式で表示されます。

### エディタのプロパティ

以下のプロパティは、すべてのエディタとそのオブジェクトに適用されます。

- コンテンツの変更

変更は入力フィールドを抜け出た直後から有効になり、プロジェクト全体に影響を及ぼします。変更の影響を受けたオブジェクトは、すべて自動的に更新されます。

例えば、スクリーンエディタを使用している場所でタグパラメータを変更した場合などは、変更内容は"タグ"エディタに直接反映されます。

- プロジェクトデータへの変更の受け入れ

変更されたプロジェクトデータは、プロジェクトの保存と同時にプロジェクトデータベースに転送されます。

- 作業手順を元に戻すまたはやり直し

どのエディタにも、ユーザーアクションを保存する内部リストが備わっています。これにより、すべてのアクションを復帰(元に戻す)または復元することができます。関連コマンドは、[編集]メニューにあります。エディタを閉じたりプロジェクトを保存すると、リストは削除されます。別のエディタに切り替えても、リストに保存されたアクションは影響を受けません。

---

### 注記

エディタを使用して画面をロードする時にエラーが発生した場合、グラフィックドライバが最新版でないことが原因の場合があります。

---

## 2.7.6 エディタを開く

### はじめに

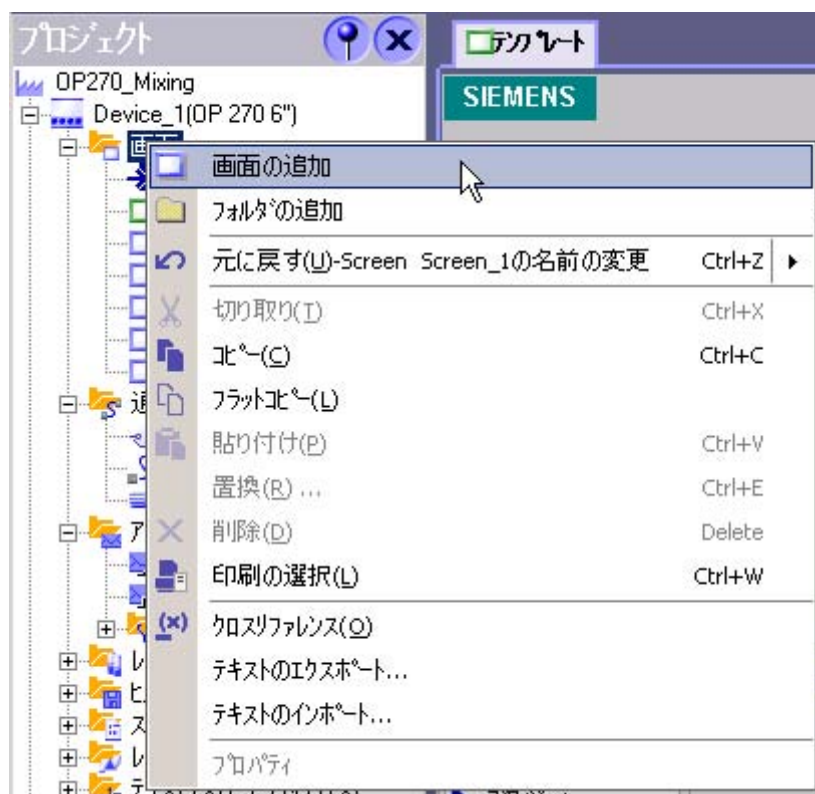
エディタの起動方法は、グラフィカルエディタ(たとえば、スクリーンエディタ)であるかテーブル形式エディタ(たとえば、タグエディタ)であるかによって異なります。最高 20 のエディタを同時に開くことができます。

### グラフィカルエディタを開く

グラフィカルエディタは、オブジェクトを新規作成するか既存のオブジェクトを開くと起動します。

オブジェクトを新規作成するには、以下の手順に従ってください。

1. 新規オブジェクトを追加する[プロジェクト]ウィンドウで、グラフィカルエディタを右クリックします。
2. 例として、ショートカットメニューの[画面の追加]を選択します。



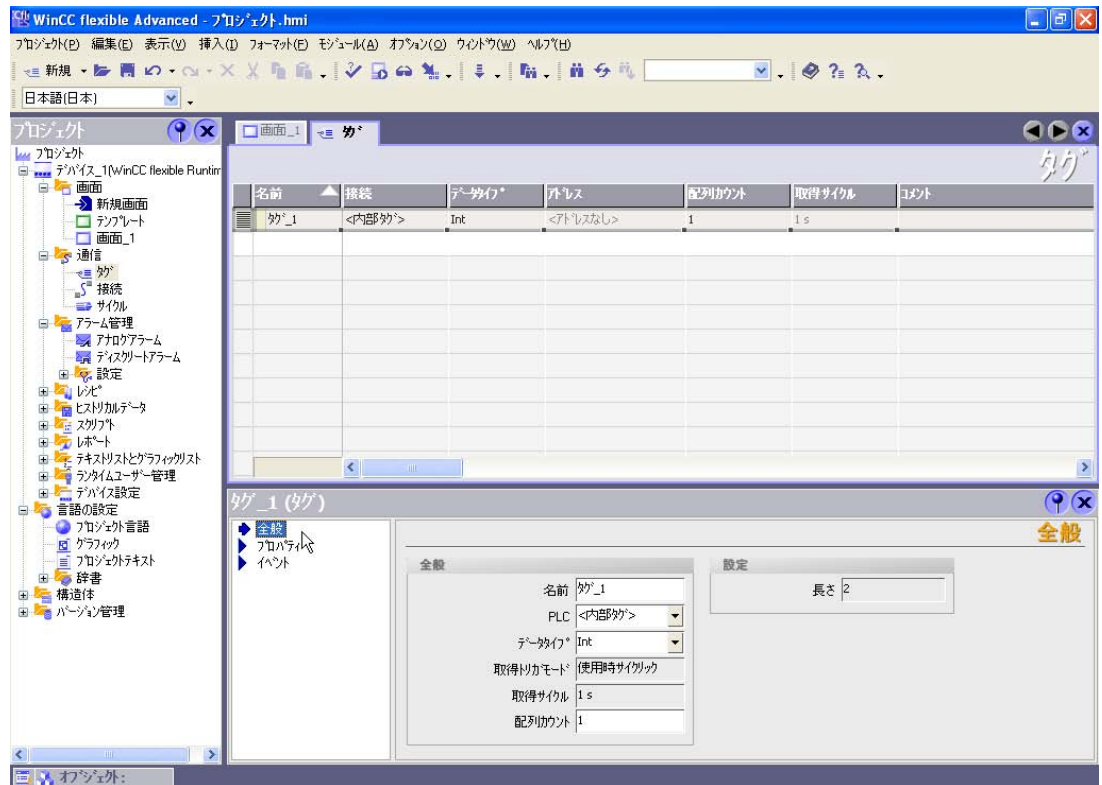
オブジェクト、ここでは画面が[プロジェクト]ウィンドウに作成され、作業エリアに表示されます。

3. 既存のオブジェクトを開くには、[プロジェクト]ウィンドウまたは[オブジェクト]ウィンドウでオブジェクトをダブルクリックします。

オブジェクト、例えば画面が作業エリアに表示されます。

## テーブル形式エディタを開く

テーブル形式エディタは、[プロジェクト]ウィンドウでテーブル形式エディタをダブルクリックして開きます。エディタが作業エリアに表示されます。



テーブル形式エディタは、ショートカットメニューを使って起動することもできます。テーブル形式エディタで既存の元素を開くには、[プロジェクト]ウィンドウでテーブル形式エディタを選択します。次に、[オブジェクト]ウィンドウで必要なオブジェクトをダブルクリックします。

## その他の方法

エディタをメニューから開くには、[挿入]メニューから[プロジェクト内の新規オブジェクト]を選択します。

## 2.7.7 エディタ間の切り替え

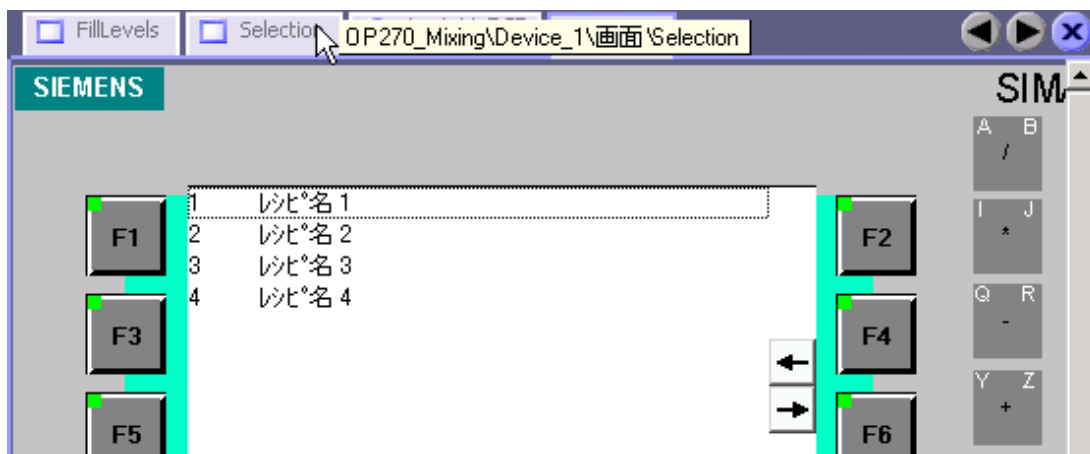
### はじめに

同時に複数のエディタまたはそれらのオブジェクトを WinCC flexible で開くことができますが、アクティブにできるのは 1 つのエディタの作業エリアだけです。

複数のエディタが開いている場合は、作業エリアのそれぞれのタブコントロールで切り替えます。

### タブコントロール

異なるエディタを選択するには、作業エリアで対応するタブをクリックします。テーブル形式エディタには、簡単に識別できるようタブにエディタの名前が表示されています。グラフィックエディタの場合は、カレントエレメントの名前が"Screen1"のように表示されます。

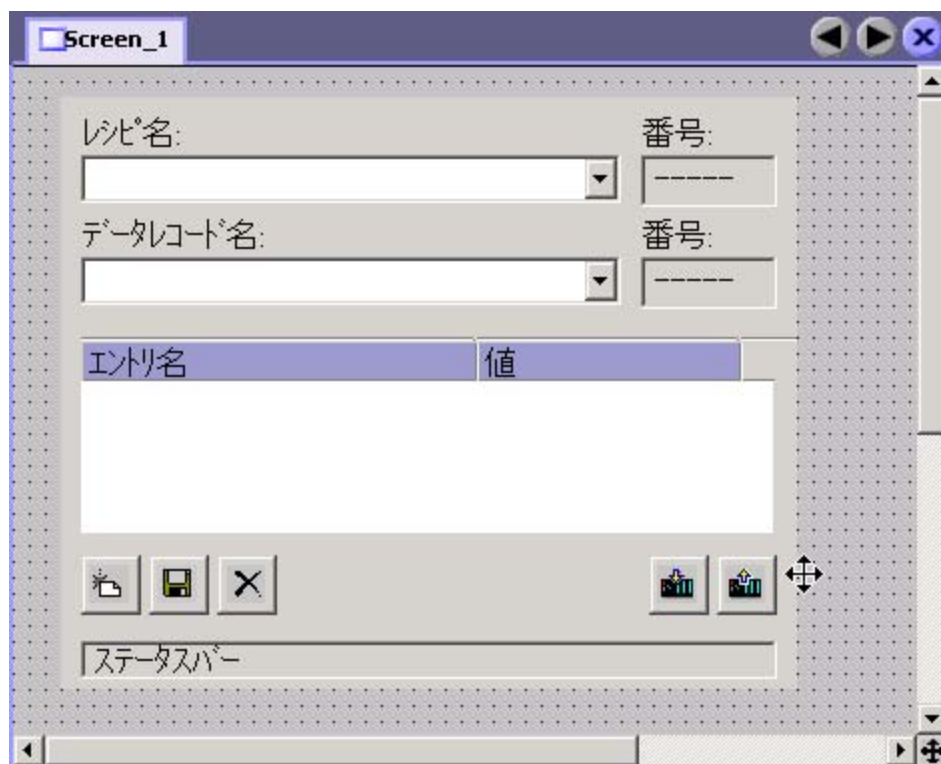


このツールヒントは、エディタによる HMI デバイスの設定の様子を示しています。


## ナビゲーション矢印

作業エリアのスペース不足ですべてのタブが表示されない場合は、作業エリア内のナビゲーション矢印が有効になります。

作業エリアに表示されていないタブコントロールにアクセスするには、対応するナビゲーション矢印をクリックします。



## エディタを閉じる

エディタを閉じるには、作業エリアで  シンボルをクリックします。

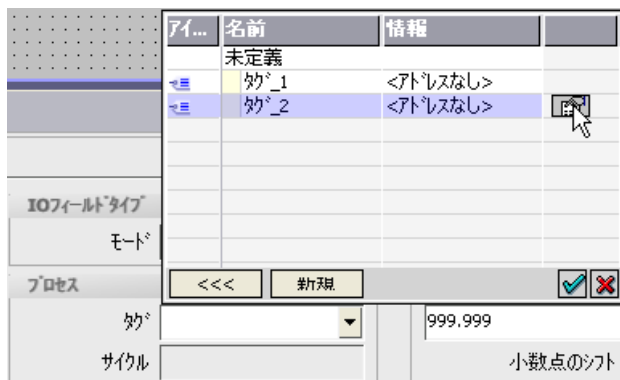
## 2.7.8 オブジェクトリスト;オブジェクトリスト

### はじめに


オブジェクトリストは、WinCC flexible でのコンフィグレーションタスクに役立つ機能です。オブジェクトリストを使用すると、必要なオブジェクトタイプの既存オブジェクトを検索したり、使用する場所で直接設定したりできます。また、オブジェクトリストを使用して、使用する場所で新規オブジェクトを作成することもできます。

### オブジェクトリストを開く

オブジェクトは通常[プロパティ]ウィンドウで編集されますが、テーブル形式エディタを使用しているときは、作業エリアでテーブルを直接編集することもできます。WinCC flexible でオブジェクトへの接続が必要な場合は、オブジェクト選択リストをクリックすると、オブジェクトリストが開きます。たとえば、グラフィックオブジェクトのタグを設定するときは、タグの選択フィールドをクリックします。選択フィールドからオブジェクトリストが開き、選択プロジェクト内でデータタイプが適した、使用できるタグがすべて表示されます。




オブジェクトリスト

必要なタグを選択します。  ボタンを押して、選択内容を確認します。

### オブジェクトリストの取り扱い

適したオブジェクトをプロジェクトで使用できない場合は、オブジェクトリストを使用して新規オブジェクトを作成します。新規オブジェクトを作成するには、オブジェクトリストで[新規作成]ボタンをクリックします。

新規オブジェクトが作成され、このオブジェクトを設定するための対応するダイアログが開きます。新規作成したオブジェクトを設定し、コンフィグレーションダイアログを閉じます。

オブジェクトリストから既存のオブジェクトを開いて設定することもできます。オブジェクトリストからオブジェクトを選択します。編集用の  アイコンが、右側の列に表示されます。そのアイコンをクリックします。オブジェクト編集用の対応するダイアログが開きます。オブジェクトのプロパティを編集してコンフィグレーションダイアログを閉じます。

## 2.7.9 ファンクションリスト

### はじめに

ファンクションリストはシステムファンクションまたはスクリプトのアタッチメントです。ファンクションリストを呼び出すとシステムファンクションやスクリプトが引き続いて実行されます。ファンクションリストを使用すると、イベントでシステムファンクションの実行をトリガできます。ファンクションリストは、画面オブジェクトやタグなどのオブジェクトのイベントを設定します。使用できるイベントは選択したオブジェクトによって異なります。イベントが発生するのは、プロジェクトがランタイムモードになっている場合に限ります。イベントには、たとえば、次のものがあります。

- タグの値変更
- 配列の値の変更 = 配列要素の値の変更
- ボタンを押す
- アラームの発生

イベントごとにファンクションリストを正確に作成できます。最高 16 のファンクションをファンクションリストに設定できます。ランタイム中に、作成したイベントが発生すると、ファンクションリストが一番上から一番下に向かって実行されます。待ち時間を避けるため、比較的長い実行時間をもつシステムファンクション(たとえばファイル操作など)は、同時に処理されます。前のシステムファンクションが完了していなくても、次のシステムファンクションを実行できます。

### ファンクションリストの設定

WinCC flexible において、ファンクションリストを作成するオブジェクトを含むエディタを開きます。マウスを使ってオブジェクトを選択します。[プロパティ]ウィンドウの中で、[イベント]グループでファンクションリストを作成したいイベントをクリックします。ファンクションリストは[プロパティ]ウィンドウで開きます。



ファンクションリスト

オブジェクトに対してファンクションが作成されていない場合は、ファンクションリストの1行目に「ファンクションなし」と表示されます。「ファンクションなし」と表示されたフィールドをクリックします。選択ボタンが表示されます。選択ボタンを使用し、使用できるシステムファンクションのリストを開きます。システムファンクションは、選択リストでカテゴリに応じて整理されています。

必要なシステムファンクションを選択します。



システムファンクション

システムファンクションにパラメータが必要な場合は、次の行の選択したシステムファンクションの後に「値なし」と表示されます。「値なし」と表示されたフィールドをクリックします。選択ボタンが表示されます。選択ボタンを使用し、オブジェクトリストを開いて必要なパラメータを選択します。



パラメータの選択

ファンクションリストにファンクションが作成されます。必要に応じてその他のファンクションを作成します。↓ボタンと↑ボタンを使用し、作成したファンクション/スクリプトのシーケンスを変更します。ファンクションを選択し、矢印ボタンをクリックしてファンクションを上下に移動します。ファンクションを削除するには、マウスを使用してファンクションにマークして<Del>キーを押します。

## 2.7.10 テキストリスト

### はじめに

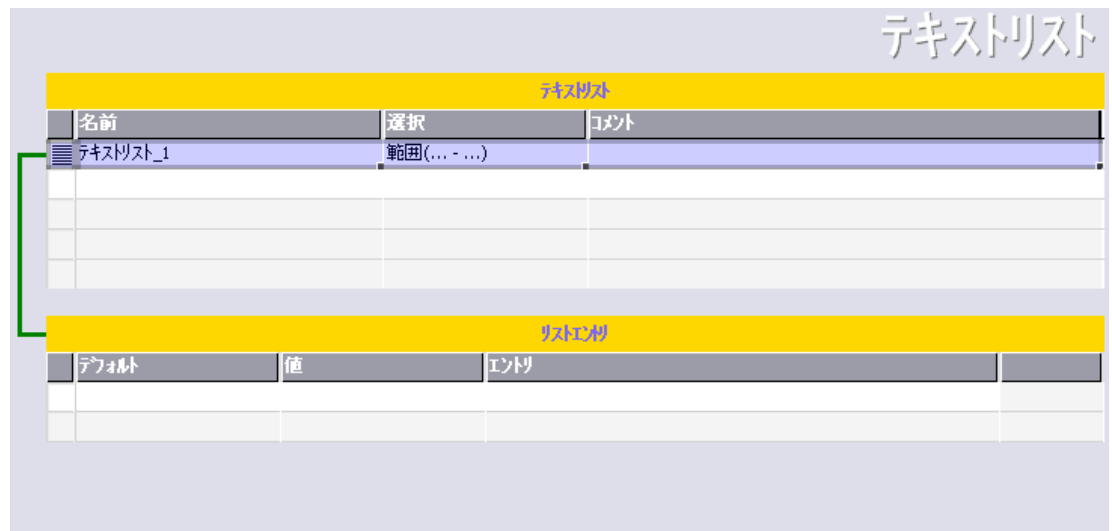
テキストリストで、タグの値が多数のテキストに割り付けられます。テキストリストは[テキストリスト]エディタで作成されます。タグへのテキストリストの添付は、シンボル IO フィールドなど、使用したオブジェクトに設定されます。テキストリストのアプリケーションエリアは以下のとおりです。

- シンボル IO フィールド付き選択リストの設定用
- ステータス依存ボタンラベルの設定用
- ディスクリットアラームかアナログアラームの値用テキスト出力の設定用
- レシピデータの値用テキスト出力の設定用

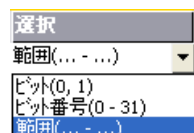
テキストリストのテキストは多国語で作成できます。ランタイム中、テキストは設定したランタイム言語で表示されます。

### テキストリストの設定

[プロジェクト]ウィンドウの[テキストリスト]エントリをダブルクリックし、"テキストリスト"エディタを開きます。エディタの最初の空白行をダブルクリックすると、新しいテキストリストが作成されます。



"選択"列をクリックしてドロップダウンリストを開きます。

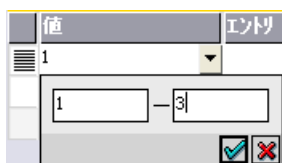


必要なテキストリストに対応するフィールドを選択します。これには、次のようなフィールドがあります。

- 範囲(... - ...) この設定を使用すると、タグの整数範囲または値範囲が、テキストリストからテキストエントリに割り付けられます。テキストエントリの数は自由に選択できます。エントリの最大数は HMI デバイスに応じて異なります。定義された領域外にタグの値がある場合に表示されるデフォルト値を設定します。
- ビット(0, 1) このように設定し、テキストリストのテキストエントリをバイナリタグの 2 つのステータスに割り付けます。バイナリタグのステータス用テキストエントリを作成できます。
- ビット番号(0 - 31) このエントリで、テキストリストのテキストエントリがタグのすべてのビットに割り付けられます。最大テキストエントリ数は 32 です。このタイプのテキストリストは、使用タグを 1 ビットのみでも設定できる場所ならどこでも使用できます。たとえば、シーケンサの実行時にシーケンシャルコントロールで使用できます。

最低重要度のビットセットとデフォルト値でビット数(0~31)の動作が決定されます。

テーブルの作業エリアにある「リストエントリ」によってテキストリストのテキストが作成されます。テーブルの最初の行をダブルクリックします。最初のテキストエントリが作成されました。ビット番号に使用される、割り付けられたタグのバイナリ値または値の範囲を"値"列で設定します。



"エントリ"列で出力に必要なテキストを入力します。他のテキストリスト、タグ、コントロールタグのエントリ用に、出力フィールドを挿入することもできます。コントロールタグは、テキストリストエントリを選択するタグです。

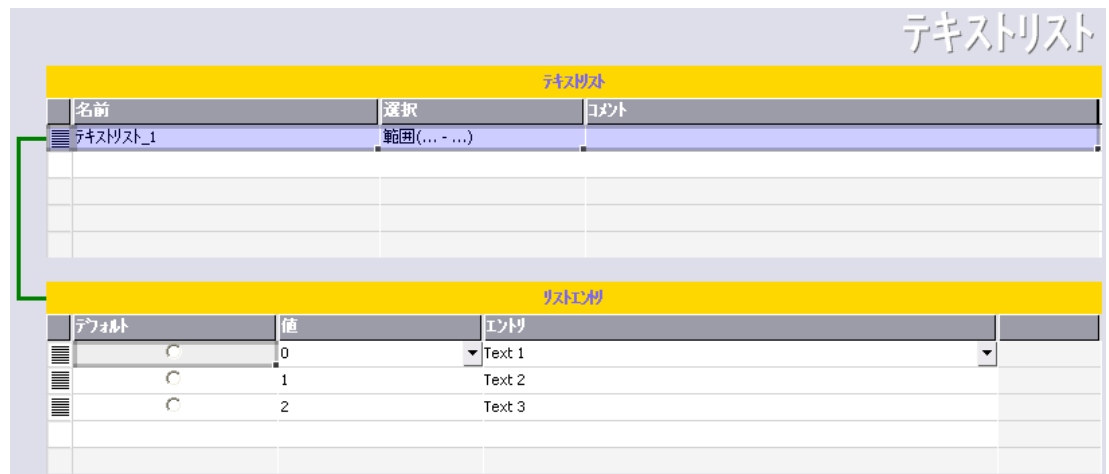
タグの出力用に、データタイプと表示フォーマットを選択します。コントロールタグには、10 進数、16 進数、2 進数のデータタイプのみが有効です。

テキストリストエントリの出力のフィールド長を指定します。

#### 注記

テキストリストエントリまたはタグへの参照を順に含むテキストリストエントリへの参照の数には、制限があります。

次のエントリを作成するには、テーブル内の次の空白行をダブルクリックします。



"値"列のエントリは、システムによって独自に割り付けられます。値を変更するには、対応するエントリをクリックします。ドロップダウンリストボックスを開き、必要な値と範囲値を入力します。

#### 注記

テキストリストを使用できるかどうかは、使用している HMI デバイスによって決まります。

### ビット数(0~31)の動作

[デバイス設定]エディタで[テキストおよびグラフィックリストのビット選択]が無効で、デフォルト値が設定されない場合、以下の標準応答が適用されます。1ビットのみがすべての設定ビットに構成されている場合は、設定済みビットに保存されたビットが表示されます。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	テキスト 3	-	テキスト 2	テキスト 1	-	-	-

重要度"4"のビットのみが設定されます。テキスト 2 は表示されます。

その他すべての場合は何も表示されません。

- たとえば、ビットが設定されません。
- たとえば、数ビットが設定されます。

## デフォルト値

そのような場合の空白表示を回避するためにデフォルト値を設定します。設定されたデフォルト値は、次の場合表示されます。

- [テキストおよびグラフィックリストのビット選択]が無効で、タグに正確に1ビットが設定されていない場合。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	テキスト3	テキスト2	テキスト1	-	-	-	テキスト0

2ビットが設定され、作成されます。設定済みデフォルト値が表示されます。

- [テキストおよびグラフィックリストのビット選択]が有効でビットが設定されていないが、最小値のビットが設定されていてテキストが設定されていない場合。

[リストエントリ]テーブルの[デフォルト]列のエントリをクリックしてデフォルト値を表示します。[デフォルト]を[値]として入力するか、または[プロパティ]ウィンドウの[全般]カテゴリの[設定]エリアで[デフォルト]チェックボックスを有効にすることもできます。

## 最低重要度ビットセット

[テキストおよびグラフィックリストのビット選択]が有効な場合、最小値で設定ビットに設定されたテキストが表示されます。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	テキスト3	-	テキスト2	-	テキスト1	-	-

最低重要度のセットビットは"2"です。テキスト1は表示されます。

テキストもデフォルト値も最低重要度ビットに設定されていない場合は、何も表示されません。デフォルト値が設定された場合は、表示されます。

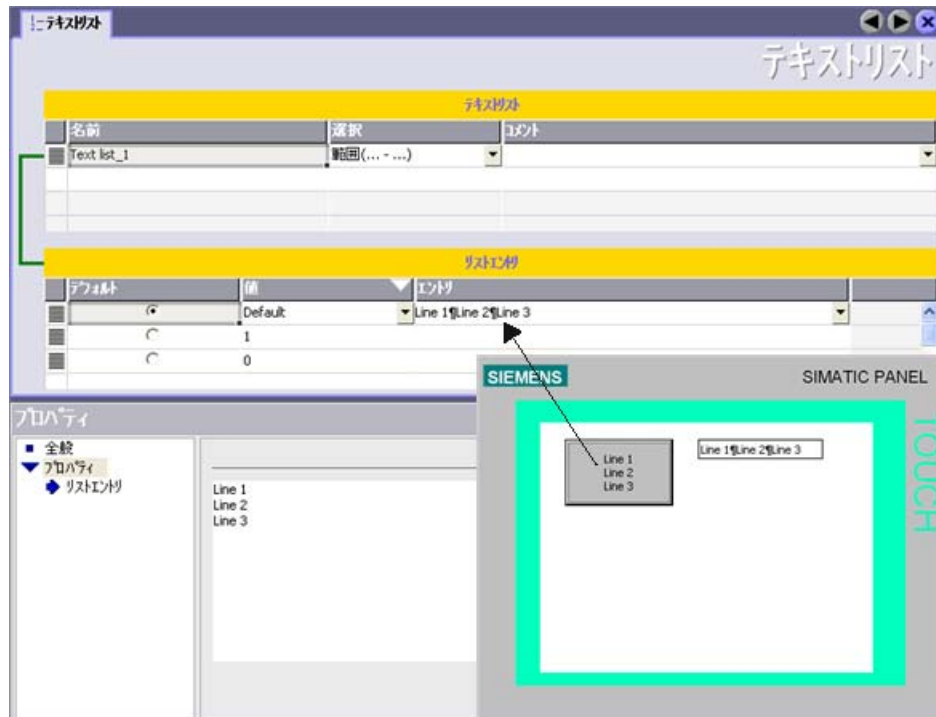
最下位ビットに割り付けられたテキストのみを視覚化するには、[デバイス設定]エディタの[ランタイム設定]エリアで、[テキストおよびグラフィックリストのビット選択]を有効にします。

下方互換性のために、この設定は標準では無効になっています。設定は HMI デバイスのすべてのテキストリストに適用されます。

## 複数行のテキストリストエントリ

"テキストリスト"エディタでは、改行記号は"<SHIFT>+<Return>"ショートカットを使って入力します。または、プロパティダイアログの"Return(リターン)"を使って改行を入力できます。改行は、行頭復帰記号"¶"を使って表示されます。

複数行のテキストリストエントリは、シンボル出力フィールドおよびボタン上でのみ、複数行に出力されます。メッセージ行やシンボル I/O フィールドなど、これ以外のすべての場合、複数行のテキストは行頭復帰記号"¶"を使って表示されます。



## 2.7.11 グラフィックリスト

## はじめに

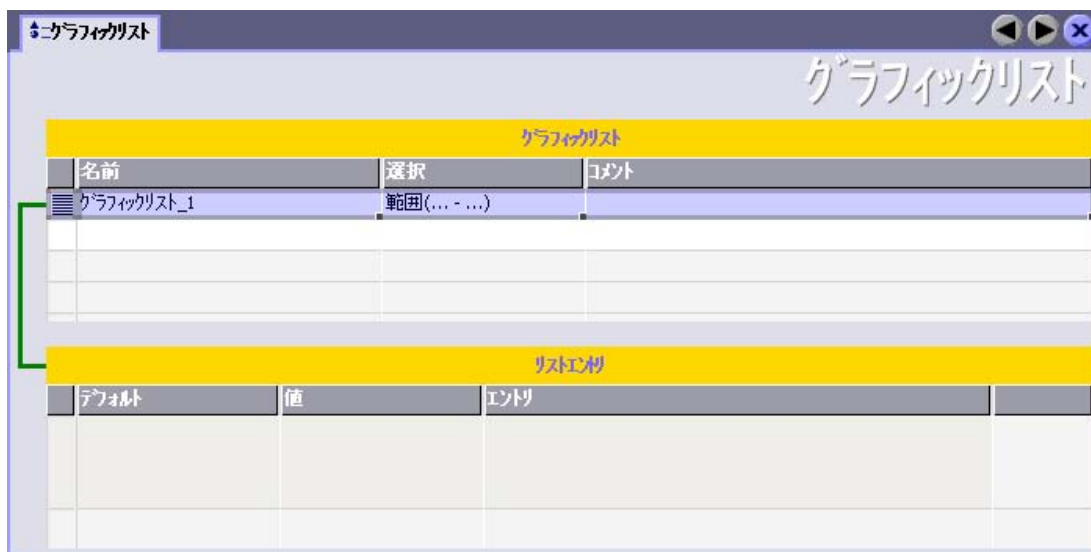
グラフィックリストで、タグの値が各種の画面またはグラフィックに割り付けられます。グラフィックリストは"グラフィックリスト"エディタで作成されます。タグへのテキストリストの添付は、シンボルグラフィックフィールドなど、使用されているオブジェクトに設定されます。グラフィックリストの適用範囲は以下のとおりです。

- グラフィック IO フィールド付きドロップダウンリストの設定用
- ステータス依存のボタン表示の設定用

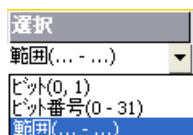
グラフィックリストのグラフィックは多言語で設定できます。ランタイム中に、グラフィックは設定したランタイム言語で表示されます。

## グラフィックリストの設定

[プロジェクト]ウィンドウの[グラフィックリスト]エントリをダブルクリックし、"グラフィックリスト"エディタを開きます。エディタの最初の空白行をダブルクリックすると、新しいグラフィックリストが作成されます。



"選択"列をクリックしてドロップダウンリストを開きます。



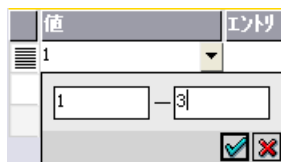
必要なグラフィックリストに対応するフィールドを選択します。次のようなフィールドがあります。

- 範囲(... - ...) この設定では、タグの整数または値範囲が、グラフィックリストからのグラフィックに割り付けられます。グラフィックの数は自由に選択できます。エントリの最大数は HMI デバイスに応じて異なります。  
定義された領域外にタグの値がある場合に表示されるデフォルト値を設定します。
- ビット(0, 1): この設定では、グラフィックリストからのグラフィックが、バイナリタグの2つのステータスに割り付けられます。バイナリタグの各ステータス用テキストエントリを作成できます。
- ビット番号(0 - 31): このエントリでは、グラフィックリストからのグラフィックが、タグのすべてのビットに割り付けられます。グラフィックエントリの最大数は 32 です。このタイプのグラフィックリストは、使用タグの1ビットだけ設定が許されている場所ならどこでも使用できます。たとえば、シーケンサの実行時のシーケンス制御に使用できます。

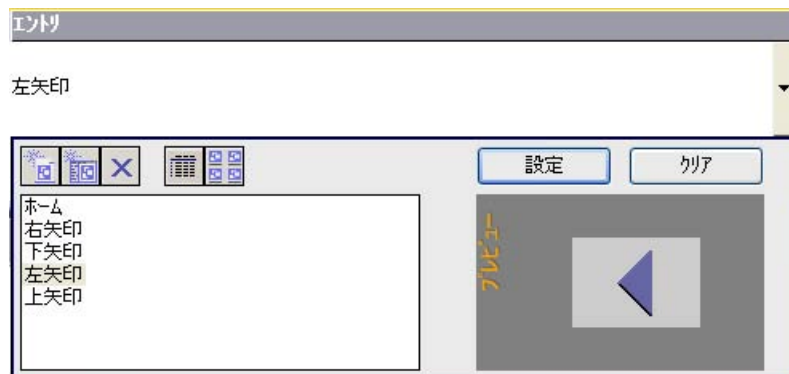
最低重要度のビットセットとデフォルト値でビット数(0~31)の動作が決定されます。

[リストエントリ]テーブルの作業エリアで、グラフィックリストのグラフィックを作成します。テーブルの最初の行をダブルクリックします。最初のグラフィックエントリが作成されます。

ビット番号に使用される、割り付けられたタグのバイナリ値または値の範囲を"値"列で設定します。



"エントリ"列で出力に必要な画像を入力します。



次のエントリを作成するには、テーブル内の次の空白行をダブルクリックします。



"値"列のエントリは、システムによって一意的に割り付けられます。値を変更するには、対応するエントリをクリックします。ドロップダウンリストボックスを開き、必要な値と値の範囲を入力します。

**注記**

グラフィックリストを使用できるかどうかは、使用している HMI デバイスによって決定されます。

## ビット数(0~31)の動作

デバイス設定の[テキストおよびグラフィックリストのビット選択]が無効で、デフォルト値が設定されていない場合は、次の標準応答が適用されます。すべての設定ビットから1つのみが構成された場合、設定済みビットに保存されたイメージが表示されます。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	イメ ジ 3	-	イメ ジ 2	イメ ジ 1	-	-	-

重要度"4"のビットだけが設定されます。イメージ 2 は表示されます。

その他すべての場合はサボテンイメージが表示されます。

- たとえば、ビットが設定されません。
- たとえば、数ビットが設定されます。

## デフォルト値

そのような場合にサボテン画像が表示されるのを回避するためにデフォルト値を設定します。デフォルト値に設定された画像は、次の場合に表示されます。

- [テキストとグラフィックリストのビット選択]が無効で、設定されたタグに正確に 1 ビットが設定されていない場合。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	イメ ジ 3	イメ ジ 2	イメ ジ 1	-	-	-	イメ ジ 0

2 ビットが設定され、作成されます。設定済みデフォルト値が表示されます。

- [テキストとグラフィックリストのビット選択]が有効で、ビットが設定されていないか、最小値のビットが設定され、グラフィックが設定されていない場合。

[リストエントリ]テーブルの[デフォルト]列のエントリをクリックしてデフォルト値を表示します。[デフォルト]を[値]として入力するか、または[プロパティ]ウィンドウの[全般]カテゴリの[設定]エリアで[デフォルト]チェックボックスを有効にすることもできます。

### 最低重要度ビットセット

[テキストとグラフィックリストのビット選択]が有効な場合、最小値で設定ビットに設定されたグラフィックが表示されます。

例

重要度	7	6	5	4	3	2	1	0
ビットセット	0	0	1	1	0	1	0	0
設定済み	-	イメ ジ 3	-	イメ ジ 2	-	イメ ジ 1	-	-

最低重要度のセットビットが"2"です。"イメージ 1"が表示されます。

テキストもデフォルト値も最低重要度ビットに設定されない場合は、サボテンが表示されません。いずれの設定済みデフォルト値を設定しても、このデフォルト値に設定されたイメージが表示されます。

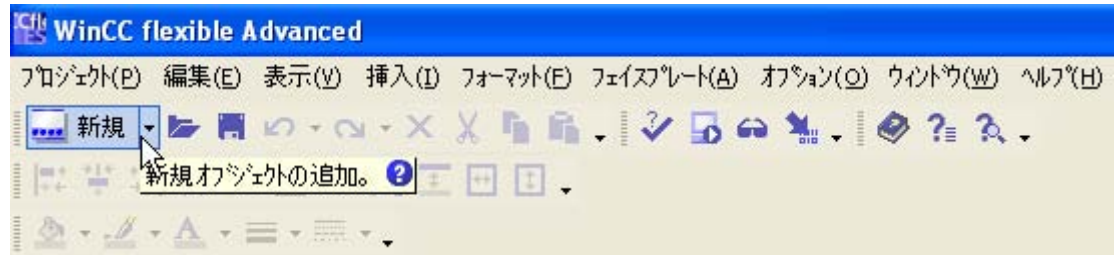
デバイス設定の[ランタイム設定]エリアで[テキストおよびグラフィックリストのビット選択]を有効にし、最下位ビットに割り当てられたテキストのみを視覚化します。

下方互換性のために、この設定は標準では無効になっています。設定は HMI デバイスのすべてのグラフィックリストに適用されます。

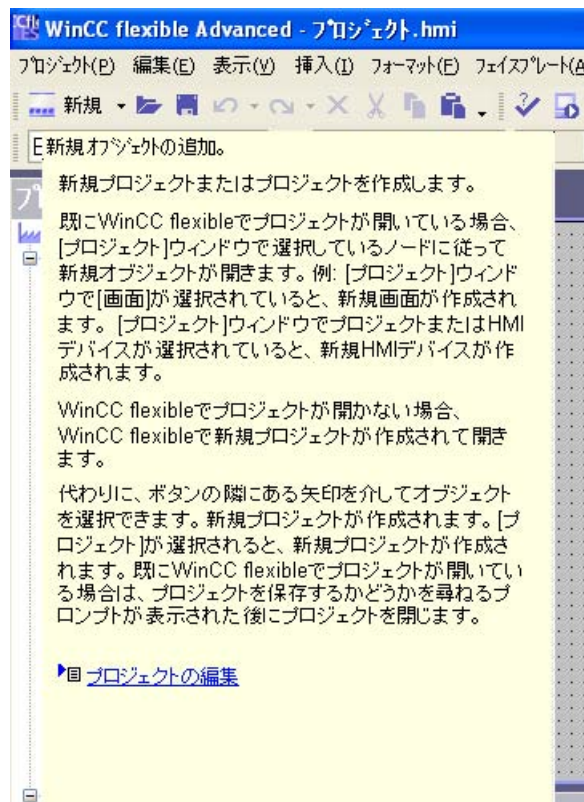
## 2.7.12 ヘルプの表示

### ショートカットヘルプ

任意のオブジェクト、アイコンまたはダイアログエレメント上にマウスポインタを置くと、ツールヒントが表示されます。



ツールヒントの隣に疑問符が表示される場合、そのユーザーインターフェースエレメントではショートカットヘルプが使用可能であることを意味します。簡単な説明のほかに追加説明を呼び出したい場合は、クエスチョンマークをクリックし、ツールヒントがアクティブであれば<F1>を押します。または、マウスカーソルをツールヒントに移動します。



追加説明には、オンラインヘルプの詳しい説明をユーザーが参照するための参照先が含まれています。

## オンラインヘルプ

[ヘルプ]コマンドメニューからオンラインヘルプにアクセスできます。[ヘルプ|目次]メニューコマンドを使用すると、WinCC flexible Information System が目次を表示して開きます。目次を使用して希望する項目を探し当てます。

あるいは、[ヘルプ|索引]メニューコマンドを選択します。WinCC flexible Information System が索引を表示して開きます。索引を使用して希望する項目を探し当てます。

WinCC flexible Information System 全体に対し全文検索を使用するには、[ヘルプ|検索]メニューコマンドを選択します。WinCC flexible Information System が検索タブを表示して開きます。希望する検索用語を入力します。

WinCC flexible Information System は Windows の[スタート]メニューから開くこともできます。タスクバーで[スタート|SIMATIC|WinCC flexible Help System]メニューコマンドを選択します。

別のウィンドウにオンラインヘルプシステムが開きます。

### 2.7.13 WinCC flexible のカスタマイズセットアップ

#### はじめに

WinCC flexible では、ウィンドウやツールバーの位置や動きをカスタマイズすることができます。これにより、特殊な必要条件を満たすように作業環境を作成できます。

#### 作業環境のユーザー依存性

WinCC flexible の表示は、Microsoft Windows でログオンしたユーザーにリンクしています。プロジェクトを保存するときは、ウィンドウとツールバーの位置と動きも自動的に一緒に保存されます。

次にプロジェクトを開くときには、保存時と同じプロジェクトステータスがロードされます。このように、作業環境は前回閉じた状態と同じ状態で開きます。これは、他のプロジェクト企画者によって編集されたプロジェクトを開いたときも同じです。

#### 作業環境のリセット

ウィンドウとツールバーの位置を元の状態にリセットすることができます。リセットするには、[表示]メニューの[レイアウトのリセット]を選択します。

## 2.8 WinCC flexible スタートセンター

### はじめに

WinCC flexible では加速的にプロジェクトを開くことをサポートしています。クイックスタートの場合、オペレーティングシステムがブートしている最中に、WinCC flexible のサービスの一部がバックグラウンドで起動します。WinCC flexible スタートセンターを操作するため、インストール中に、タスクバー通知エリア、いわゆるタスクバーのトレイエリアにシンボルが 1 つ作成されます。



スタートセンターシンボル

### WinCC flexible スタートセンター

タスクバーのトレイエリアに表示されているこのシンボルをマウスで右クリックすると、スタートセンターの操作用ポップアップメニューが開きます。このポップアップメニューには、次のメニューコマンドが表示されています。

メニューコマンド	機能
WinCC flexible の開始	WinCC flexible を起動して、プロジェクトウィザードを開きます。
[WinCC flexible スタートセンター]有効	WinCC flexible のクイックスタートを有効にします。オペレーティングシステムのスタートアップ中にクイックスタートに必要なサービスがロードされます。
[WinCC flexible スタートセンター]>[無効]	WinCC flexible のクイックスタートを無効にします。オペレーティングシステムのスタートアップ中にその他のサービスは起動されません。
ヘルプ	WinCC flexible スタートセンターに関するオンラインヘルプを開きます。
情報	ウィンドウが 1 つ開き、スタートセンターのバージョン情報が表示されます。
[終了]	スタートセンターを終了します。

### WinCC flexible の開始

WinCC flexible のインストール中に、WinCC flexible スタートセンターも自動的にインストールされ有効になります。コンピュータを再起動してスタートセンターを有効にします。[スタートセンター]ポップアップメニューの[WinCC flexible の実行]を選択して WinCC flexible を起動します。または、デスクトップアイコンを使用して WinCC flexible を起動します。WinCC flexible が起動して、プロジェクトウィザードが表示されます。その後、ウィザードの指示に従って進みます。

STEP 7 に組み込まれているプロジェクトのクイックスタートについては、「Start WinCC flexible in STEP 7 (STEP 7 での WinCC flexible の起動)」章を参照してください。

---

**注記**

WinCC flexible の初期起動時に以下リストのフォルダに対して書込み許可が必要です。

"Documents and Settings\All Users\Application data\Siemens AG\SIMATIC WinCC flexible"  
および"Documents and Settings\All Users\Application data\Siemens AG\SIMATIC WinCC  
flexible 2008\Caches"。

初期起動時にシステムでは、これらのフォルダに現在の WinCC flexible GUI 言語を使用した  
テンプレートファイルを生成します。新たにプロジェクト用 GUI 言語を初期設定する場  
合も書込み許可が必要です。

---

### スタートセンターの無効化

WinCC flexible スタートセンターを無効にするには、[WinCC flexible スタートセンター]ポ  
ップアップメニューで[WinCC flexible スタートセンター]>[無効]の順にメニューコマンドを  
選択します。次回にコンピュータを再起動したときに、バックグラウンドで WinCC flexible  
構成工元素が起動しなくなります。

## プロジェクトの取り扱い

### 3.1 プロジェクトの取り扱いの基本

#### 3.1.1 プロジェクトの取り扱い

##### WinCC flexible のプロジェクト

WinCC flexible 設定ソフトウェアは、マシンやシステムをコントロールするためのユーザーインターフェースの設定に使用されます。このシステムはパラメータ読み取り用の単純なテキスト表示ユニットが主ですが、たとえば、生産ライン用の複雑なオペレーティングステーションも含まれます。

WinCC flexible にはモジュラーデザインが採用されています。WinCC flexible のエディションによって、設定できる HMI デバイスのタイプが決まります。必要に応じて、WinCC flexible のエディションは、問題なく新しいエディションにアップグレードすることができます。

WinCC flexible プロジェクトでは、最大 8 つの HMI デバイスを設定できます。

WinCC flexible のプロジェクトには、プラントや HMI デバイスの設定データがすべて含まれています。設定データには以下のものが含まれます。

- プロセス画面(プロセスを表示します)。
- タグ(ランタイム時に PLC と HMI デバイスの間でデータを転送します)。
- アラーム(ランタイム時に操作の状態を表示します)。
- ログ(プロセス値とアラームを保存します)。

プロジェクトに関係するすべてのデータは、WinCC flexible に組み込まれているデータベース内に保存されます。

## 適用例

WinCC flexible は、たとえば、以下のようなシナリオで使用することができます。

- マシンレベルで恒久的に取り付けられた HMI デバイスの設定。

これは、HMI デバイスがシステムまたはマシンに永続的に取り付けられている、最も一般的なタイプの設定を示します。

- モバイルユニットの使用

モバイルユニットは、一般的に大規模な生産現場やライン、または材料搬送技術で使用されています。操作対象のマシンには、Mobile Panel 170などを接続できるインターフェースがいくつか備えられています。

このため、オペレータや保守点検担当者が、現場ですぐに作業をすることができます。これにより、スタートアップ段階での正確なセットアップと位置決めが可能となります。修理の場合、モバイルユニットを使用することによりダウンタイムを短縮することができます。

- 複数の HMI デバイスの設定

複数の HMI デバイスをシステムまたはマシンに接続することが可能です。そのため、システムを様々なポイントから操作することができます。WinCC flexible を使用することにより、タイプが異なっても同じプロジェクトデータを処理することができるものであれば、1つのプロジェクトで複数のユニットを設定することが可能です。

## Windows Explorer でのプロジェクトの管理

WinCC flexible は、プロジェクトを保存するときにハードディスクにプロジェクトデータを作成します。プロジェクト データベースは、拡張子\*.hmi を付けて、Windows のファイルマネージャに保存されます。ログファイル (\*\_log.ldf) は、各プロジェクト データベースに対して保存されます。このログファイルがないと、データの一貫性が保証されません。

他のファイルと同様に、プロジェクト データベースを Windows Explorer の対応するログファイルと一緒に、移動、コピー、および削除できます。ただし、コピー時と移動時に、データベースとログファイルを切り離すことはできません。

詳細については、「Windows Explorer でプロジェクトの管理」セクションの「プロジェクトの取り扱い」を参照してください。

## プロジェクトを開いた時の背景のロード

設定コンピュータの作業メモリが 1 GB 以上の場合、WinCC flexible プロジェクトを開いた後で背景ロードが始動します。背景ロードには、WinCC flexible でのパフォーマンス向上のための、スタートアップ時の WinCC flexible のさまざまなコンポーネントが含まれます。

背景ロードが完了するまで、エンジニアリングシステムの操作は実行されません。コンピュータの設定によっては、背景ロードは最長 1 分間かかることがあります。背景ロードが完了した後、エンジニアリングシステムの応答時間は通常に戻ります。

### 3.1.2 プロジェクトの構成要素

#### 原理

WinCC flexible のプロジェクトは、システムで操作やモニタが可能なすべての設定データにより構成されます。この設定データは、トピックカテゴリに応じて WinCC flexible に組み込まれます。各カテゴリは、個々のエディタで処理されます。

使用できるエディタは、WinCC flexible のエディションや、設定する HMI デバイスによって異なります。WinCC flexible の作業環境では、現在使用されている HMI デバイスでサポートされているエディタ名だけが表示されます。つまり、設定が簡単で分かりやすくなります。

プロジェクトのログを記録する場合、[プロジェクト名].hmi、[プロジェクト名]\_log.ldf ファイルをバックアップしていれば十分です。デルタコンパイルを続ける前に、[プロジェクト名].rt および [プロジェクト名]\_RT\_log.LDF ファイルもバックアップしてください。 "\*.fwx" および "\*.pwx" ファイルは、必要に応じて生成することができます。

## 3.2 プロジェクトのタイプ

### 3.2.1 プロジェクトのタイプ

#### 原理

WinCC flexible を使用してさまざまなタイプのプロジェクトを作成することができます。プロジェクトのタイプは、システム構成、システムまたはマシンのサイズ、システムまたはマシンに要求される表示能力および操作やモニタに使用される HMI デバイスによって決まります。

以下のプロジェクトタイプを WinCC flexible で作成することができます。

- 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  
単一の HMI デバイス用に使用されるプロジェクト
- マルチユーザープロジェクト; マルチユーザープロジェクト  
複数の HMI デバイスが組み込まれたプロジェクト
- 複数の HMI デバイス上で使用されるプロジェクト

#### シングルユーザープロジェクト

ほとんどの場合、HMI デバイスが 1 つだけ組み込まれます。設定フェーズの間、プロジェクトは常に選択中の HMI デバイスでサポートされているファンクション範囲を正しく表示します。

## マルチユーザープロジェクト

複数の HMI デバイスを使用してシステムを操作する場合は、WinCC flexible を使用して、複数の HMI デバイスを組み込むプロジェクトを作成することができます。このプロジェクトのタイプは、マシンまたはシステムが複数のポイントから制御されている場合などに使用されます。このとき、プロジェクトで共通のオブジェクトを使用することができます。この方法は、HMI デバイスごとにプロジェクトを作成する必要がなく、むしろ、すべての HMI デバイスを 1 つのプロジェクトで管理するという意味を持ちます。

WinCC flexible のプロジェクトは、システムで操作やモニタが可能なすべての設定データにより構成されます。組み込んだユニットごとに、サポートしているファンクションだけが表示されます。サポートされていないファンクションは表示されませんが、プロジェクトデータの構成要素の一部として残ります。

## 複数の HMI デバイス上で使用されるプロジェクト

特定の HMI デバイス用にプロジェクトを作成して、他の複数の HMI デバイス上にロードすることができます。HMI デバイス上にロードする際は、HMI デバイスがサポートしているデータのみがロードされます。

### 3.2.2 プロジェクトの HMI デバイス依存性

#### 原理

HMI デバイスのファンクションによって、WinCC flexible におけるプロジェクトの表示とエディタの機能範囲が決定されます。

#### オペレーティングユニットタイプの選択

プロジェクトを作成するときに最初の HMI デバイスタイプを選択します。HMI デバイスタイプは、HMI デバイスの[プロジェクト]ウィンドウショートカットメニューで変更できます。ハードウェアサポートパッケージを使用して、新しい HMI デバイスを遡ってインストールできます。

---

#### 注記

##### HMI デバイスのタイプの切り替え

- HMI デバイスタイプを切り替えた後も、設定されているデータはすべてプロジェクトファイル内に残ります。エンジニアリングシステムでは、ファンクションだけはまだ使用できるため、現在の HMI デバイスでサポートされている設定データに限り表示されます。これには、たとえば、ログ、レシピ、画面で使用可能なオブジェクト、使用可能なシステムファンクション、および使用可能な通信プロトコルなどが挙げられます。
  - WinCC flexible プロジェクトで HMI デバイスタイプを変更し、そのあと元に戻した場合、表示オブジェクトの高さや幅が 0 だという警告が出力されることがあります。これが起こるのは、より小さなディスプレイの HMI デバイスに変更した時にオブジェクトの高さや幅が 0 に設定されたためです。その後、より大きなディスプレイの HMI デバイスに変更しても、値 0 は意味あるサイズに変更されません。オブジェクトサイズを手動で修正してください。
-

## HMI デバイスに依存するファンクション

HMI デバイスタイプを切り換えるときに、ファンクション範囲が変わること以外に以下の点も考慮する必要があります。

- 表示色

表示色の少ない HMI デバイスに切り換えると、自動的に色が変更されます。表示色の少ない HMI デバイスに切り換え、その後表示色の多い HMI デバイスに戻すと、削減された色範囲は保持されます。

- ナビゲーションボタン

HMI デバイスタイプを変更した後で、プロジェクトを再コンパイルします。コンパイルによって、すべての画面が現在の HMI デバイスに確実に変換されます。

まず最初に生成しないで HMI デバイスを再変更すると、各画面でナビゲーションボタンが同一に表示されないことがあります。

- フォント

設定したフォントが HMI デバイス上で使用できない場合は、類似のフォントまたは組み込まれている"標準フォント"に置き換えられます。"標準フォント"は、選択した HMI デバイスによって決まります。

- 異なるフォントサイズの文字セット

OP 73micro、OP 73、OP 77A、TP 177micro、TP 177A などの HMI デバイス用のプロジェクトでは、異なるフォントサイズを使わないでください。各フォントサイズの文字セットが HMI デバイスにロードされます。データを生成する場合は、出力ウィンドウをチェックし、文字セットによって使用される HMI メモリリソースを決定します。

- フォントサイズ

HMI デバイス上にテキストを表示するには、小さな Windows フォントを使用してください。大きな Windows フォントを使用すると、ディスプレイのサイズによってはテキストが完全に表示されないことがあります。

OP 77A、TP 177micro、および TP 177A の HMI デバイスに 20 ポイント以上のフォントサイズを使用すると、パフォーマンスに影響を与える場合があります。効果的に設定するには、次の事項に留意してください：

- プロジェクト全体で、大きな文字にはすべて同じフォントを使用します。これには Arial や Tahoma が最適です。
- 72 ポイントを超えるフォントサイズを使用しないでください。

- 解像度

HMI デバイスを低い解像度に切り替えるには、2つの方法があります。すべての画面オブジェクトは、自動的にサイズ調整されます。すべての画面オブジェクトのサイズが変わらないオプション。表示可能な画面に重なっている画面の下側や右側のオブジェクトは表示されません。これらの非表示オブジェクトを表示するには、画面背景を選択し、ショートカットメニューで[隠れたオブジェクトの表示]を選択します。すると、ダイアログが開きます。このダイアログで、個々のオブジェクトまたは全オブジェクトを選択して[OK]を押せば、画面の表示エリアにこれらのオブジェクトを移動できます。

#### 注記

ディスプレイサイズが 6 インチ未満の各 HMI デバイスは、幅は同じですが高さが異なります。このため、HMI デバイスを交換したら、自動スケーリング機能をオフにする必要があります。幅は変わらないため、自動スケーリング機能は、オブジェクトの高さだけ変更します。この結果、オブジェクトが歪んでしまいます。自動スケーリング機能を無効または有効にするには、[オプション]設定]の順にメニューコマンドを選択します。すると、[設定]ダイアログが開きます。この[設定]ダイアログの[画面エディタ]グループで[画面エディタの設定]をクリックします。[新規 HMI デバイスに画面と画面オブジェクトを合わせる]オプションを有効または無効にします。

- DP 設定と HMI

HMI を DP 設定と交換しても、DP 設定は変更されません。互換性のない DP スレーブを使用している場合など、DP 設定が一致しなければ、Runtime で HMI を操作することはできません。HMI を交換する場合は、必ず DP 設定をチェックし、HW Config に適切な DP モジュールがあることをチェックしてください。

- 使用できないオブジェクトのタブシーケンス

HMI デバイスでサポートされていない設定済みのオブジェクトは表示されません。表示されなくなったオブジェクトのタブシーケンス番号は、タブシーケンスの次のオブジェクトに転送されます。システムファンクション ActivateScreenByNumber を設定するときタブシーケンス番号を使用した場合は、オブジェクト番号を変更されたタブシーケンス番号に適合させます。

### HMI デバイス上のオペレーティングシステムバージョンの選択

HMI デバイスを新たに設定する場合、WinCC flexible が自動的に最新のオペレーティングシステムバージョンを選択します。

旧オペレーティングシステムバージョンを備えた HMI デバイス上で新規バージョンを使用する場合には、該当するファームウェアバージョンのイメージをその HMI デバイスに転送する必要があります。WinCC flexible は、サポートされている HMI デバイスに必要なイメージを提供します。詳細については、「Operating System Transfer (オペレーティングシステム転送)」章を参照してください。

互換性維持のために旧オペレーティングシステムバージョンを使用する必要がある場合、WinCC flexible プロジェクトを旧バージョンに変換する必要があります。変換中、HMI デバイスのバージョンは、自動的に旧バージョンに設定されます。詳細については、「プロジェクトの変換」の章を参照してください。

現在のオペレーティングシステムバージョンを備えた HMI デバイス上で旧バージョンを使用する場合には、該当するファームウェアバージョンのイメージをその HMI デバイスに転送する必要があります。WinCC flexible は、サポートされている HMI デバイスに必要なイメージを提供します。詳細については、「Operating System Transfer (オペレーティングシステム転送)」章を参照してください。

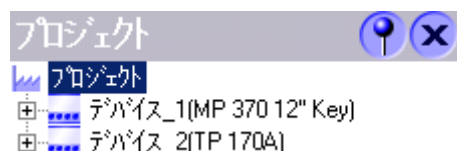
WinCC flexible で使用される HMI デバイスのバージョンに関する情報は、インターネット上の記事 ID 21742389 の FAQ にあります。

HMI デバイスのバージョンに関する情報へのリンク

### 3.2.3 複数の HMI デバイス用のプロジェクトの構築

#### 原理

WinCC flexible の[簡易]エディション以降を使用すれば、複数の HMI デバイスを組み込むプロジェクトを構成することができます。



プロジェクトウィンドウで HMI デバイスの削除、コピー(複数のプロジェクトも)および名前の変更ができます。

#### 適用例

このタイプの構成は、たとえば複数の HMI デバイスで操作するような大規模なシステム用に設計されたプロジェクトで使用されます。

#### グローバルデータと HMI デバイス固有のデータ

複数の HMI デバイスが組み込まれたプロジェクト内には、特定の HMI デバイスで使用可能なデータやオブジェクトと、プロジェクト全体で使用可能なデータやオブジェクトがあります。

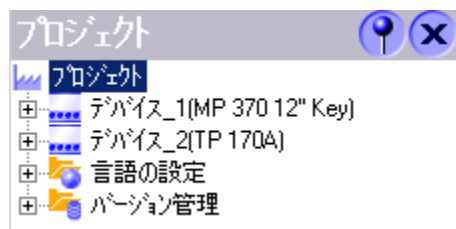
- HMI デバイス固有のデータ

プロジェクト内で特定の HMI デバイスに関連するデータを個別に設定することができます。HMI デバイス固有のデータとオブジェクトは、[プロジェクト]ウィンドウで、画像、通信、レシピまたはログなどのエントリ"デバイス"の下に表示されます。



- グローバルプロジェクトデータ

グローバルプロジェクトデータは、プロジェクト全体のすべての HMI デバイスに適用されます。このデータは、[プロジェクト]ウィンドウで、"言語"や"バージョン管理"などの"デバイス"オプションと同じレベルのすべてのデータとオブジェクトに適用されます。



### プロジェクト内の HMI デバイスの数

最高 5 台の HMI デバイスを 1 つの WinCC flexible プロジェクト内で設定できます。WinCC flexible は WinCC flexible 2005 などの 5 台以上のデバイスのプロジェクトを開くことをサポートしていますが、それより多くのデバイスを作成することはできません。

### 3.2.4 複数のオペレーティングユニット上で使用するプロジェクトの作成

#### 原理

単一のプロジェクトを作成して、複数の HMI デバイス上にロードすることができます。

#### 適用例

この構築のタイプは、タイプが似ていてパフォーマンスの異なる複数のオペレーティングユニットで使用されます。

#### 設定の特徴

以下のようにして、異なる HMI デバイス用のプロジェクトを使用します。

- 特定のオペレーティングユニットタイプでプロジェクトを作成します。通常は、最もファンクション範囲の小さいオペレーティングユニットを採用します。
- [プロジェクト]ウィンドウで、このオペレーティングユニット用の構築データをコピーします。
- プロジェクト内のオペレーティングユニットタイプを切り換えて、他のオペレーティングユニットに対する適合性をテストします。

特に以下の点に注意する必要があります。

- オペレーティングユニットタイプを切り換えた後も、設定されているデータはすべてプロジェクトファイル内に残ります。ただし、使用中のオペレーティングユニットでサポートされている構築データだけが表示されます。これには、エディタ、オブジェクトおよびオブジェクトプロパティが関係します。
- WinCC flexible は、オペレーティングユニットのファンクション範囲だけでなく、その限界もチェックします。たとえば、HMI デバイス上で使用可能なタグの数に制限がある場合は、その HMI デバイスにプロジェクトを転送したとき、または、ランタイムでプロジェクトをテストしたときに、対応するエラーメッセージが表示されます。
- HMI のタイプを変更するときは、HMI バージョンを必ず指定してください。ランタイムおよびシミュレーションは、新しい HMI デバイスのバージョンを設定しなければ、開始できません。WinCC flexible で使用される HMI デバイスのバージョンに関する情報は、インターネット上の記事 ID 21742389 の FAQ にあります。

HMI デバイスのバージョンに関する情報

### 3.2.5 SIMOTION と STEP7 に統合された WinCC flexible

#### はじめに

WinCC flexible 動作は、WinCC flexible Compact エディションから起動した SIMATIC STEP 7 と SIMOTION SCOUT で操作できます。統合することによって以下のような利点があります。

- タグおよびテキストを WinCC flexible のプロジェクトにインポートできます。
- プロセス接続中に直接、SIMATIC STEP 7 シンボルと SIMOTION SCOUT シンボルに直接アクセスできます。
- アラーム設定に含まれるテキストや属性を、WinCC flexible にインポートできます。
- 構築データを共有することによって、構築作業のオーバーヘッドが削減されます。

SIMOTION SCOUT で統合操作をするための条件は、SIMATIC STEP 7 と SIMOTION SCOUT が設定コンピュータ上にインストールされていることです。以下にインストールシーケンスを示します。

1. SIMATIC STEP 7
2. SIMOTION SCOUT
3. WinCC flexible

#### SIMATIC STEP 7 に統合された WinCC flexible

WinCC flexible をインストールする際、ユーザーは、WinCC flexible を SIMATIC STEP 7 に統合するかどうかを定義します。SIMATIC STEP 7 を設定インターフェースに統合すると、以下の利点があります。

- 耐故障性の向上
- 変更作業の減少
- 構築作業の減少

設定中に、SIMATIC STEP 7 から SIMATIC STEP 7 シンボルテーブル、データエリア、およびコントローラに直接アクセスします。アイコンテーブルには、PLC プログラムの作成中に定義したデータポイント定義(たとえば、アドレスやデータタイプ)が指定されています。

WinCC flexible プロジェクトツリーは、SIMATIC Manager のプロジェクトツリーでミラーリングされます。ただし、オブジェクトは、別の WinCC flexible アプリケーションで WinCC flexible ES ユーザーインターフェースを使って編集します。

SIMATIC STEP 7 の使用方法の詳細については、STEP 7 のマニュアルに記載されています。

## SIMOTION SCOUT に統合された WinCC flexible

WinCC flexible のインストール時に、WinCC flexible を SIMOTION SCOUT に統合するかどうかをユーザーが定義します。

SIMOTION SCOUT と接続する HMI デバイスは、SIMOTION SCOUT の作業環境で構築されます。

WinCC flexible と SIMOTION SCOUT を同じ設定コンピュータ上にインストールすることによって、WinCC flexible が SIMOTION SCOUT の作業環境に統合されます。そして、SIMOTION SCOUT または WinCC flexible の環境から発生したすべてのタスクに対して単一の作業環境で作業を実行します。

WinCC flexible のプロジェクトは、SIMOTION SCOUT のプロジェクトツリー内の 1 つのノードとして表示されます。プロジェクトに設定されたすべての操作ユニットは、プロジェクトツリーのサブエントリに表示されます。WinCC flexible エディタは、SCOUT ユーザーインターフェースの SCOUT エディタと並行して開きます。

SIMOTION SCOUT の使用方法の詳細については、「SIMOTION SCOUT」のマニュアルに記載されています。

## 統合されたプロジェクトを開く

前のバージョンからの WinCC flexible プロジェクトが WinCC flexible 2008 で最初に開かれると、デフォルトで変換が開始されます。変換には時間がかかることがあります。大規模なプロジェクトの場合は、数時間かかることもあります。

プロジェクトを SIMATIC Manager で直接開くと、変換の進捗度が画面に表示されます。大規模なプロジェクトの場合、ここで表示されるのは実際の進捗度ではありません。この場合は、コンピュータの CPU の使用率が高ければ、変換がまだ実行中であることを示します。

変換プロセスには割り込まないでください。

## 3.3 複数言語の組み込み

### 複数言語の組み込み

WinCC flexible を使って、複数言語でプロジェクトを構築することができます。WinCC flexible は、ランタイムで表示されるテキストを使用したほとんどすべてのオブジェクトの複数言語構成をサポートしています。

WinCC flexible は、オペレーティングシステムにインストールされているすべての言語の構築に使用することができます。

WinCC flexible は、エディタでのダイレクトテキスト入力に加え、プロジェクトを変換するための使いやすいエクスポートとインポート機能を提供しています。この機能は、テキストの大部分を共有する大規模なプロジェクトを構築する場合に役立ちます。

以下のエディタを使用して WinCC flexible 内のテキストを変換します。

ツールバー	簡単な説明
プロジェクト言語	プロジェクトテキスト用の言語の管理
言語とフォント	ランタイムで使用する言語とフォントの定義
プロジェクトテキスト	言語依存のプロジェクトテキストの管理
グラフィックブラウザ	ローカルなグラフィックの管理
辞書	システム辞書とユーザー辞書の管理

### WinCC flexible の複数言語ユーザーインターフェース

たとえば国籍の異なるエンジニアが 1 つのプロジェクト設定で作業している場合などに、設定中に WinCC flexible で GUI 言語を変更できます。WinCC flexible のインストール時に、使用する可能性のある言語を選択します。



WinCC flexible では、デフォルトの GUI 言語として英語がインストールされます。インストール可能な言語を以下に示します。

- 西ヨーロッパ言語

ドイツ語

スペイン語

イタリア語

フランス語

- アジア言語

日本語

中国語

台湾語

韓国語

---

**注記**

WinCC flexible 2008 CHINA では、GUI 言語として簡体字中国語が提供されています。

---

## 3.4 プロジェクトの編集

### 3.4.1 プロジェクトの編集

#### オブジェクトとエディタ

WinCC flexible で作成と編集が可能なオブジェクトを以下に示します。

- 画面

画面は、"スクリーン"エディタで作成と編集を行います。"画面ナビゲーション"エディタで画面間のナビゲーションを定義することができます。

- フェイスプレート

フェイスプレートは、プロジェクトで必要なときに使用することができるオブジェクトのグループです。フェイスプレートはライブラリに保存されます。

- グラフィックリスト

グラフィックリストでタグの値が多数のグラフィックに割り当てられます。グラフィックリストは、"グラフィックリスト"エディタで作成され、"グラフィック I/O フィールド"オブジェクトで表示されます。

- テキストリスト

テキストリストでタグの値が多数のテキストに割り当てられます。テキストリストは、"テキストリスト"エディタで作成され、"シンボル I/O フィールド"オブジェクトで表示されます。

- 言語依存のテキストとグラフィック

WinCC flexible を使用して複数の言語でプロジェクトを作成することができます。

- プロジェクト言語エディタは、プロジェクトが動作する言語を管理するために使用します。
- "プロジェクトテキスト"エディタは、主に言語依存のテキストの管理と変換を行うために使用します。
- "グラフィック"エディタは、言語依存のグラフィックを管理するために使用します。
- "ユーザー辞書"エディタは、プロジェクトテキストを変換する辞書の作成と管理を行うために使用します。"システム辞書"エディタは、WinCC flexible に統合されたシステム辞書を表示するために使用します。

- タグ

タグは、"タグ"エディタで作成と編集を行います。

- サイクル

定期的に発生する WinCC flexible 内のイベントを組み込むことができます。時間間隔は、"サイクル"エディタで定義します。

- アラーム

アラームは、"アナログアラーム"エディタと"ディスクリートアラーム"エディタで作成と編集を行います。

- ログ  
"アラームログ"エディタを使用して、システムで発生したオペレーティングステータスと不具合を記録するためにアラームを記録します。  
"データログ"エディタを使用して、プロセス値のコンパイル、処理および記録を行います。
- プロトコル  
"レポート"エディタを使用して、ユーザーがランタイム中にアラームやプロセス値を印刷するためのレポートを作成します。
- スクリプト  
WinCC flexible には、カスタムスクリプトを使用してプロジェクトをダイナミックに作成するオプションが備わっています。スクリプトは、"スクリプト"エディタで管理します。

以下のタスクを WinCC flexible で実行することができます。

タスク	エディタ
コントローラの構築	接続
ユーザー、ユーザーグループの設定とランタイム中の操作に対するユーザー権限の割り付け	ランタイムユーザー管理
タスクに関連するジョブの管理。ジョブは 1 回以上実行することができます。	スケジューラ
開始画面、使用言語などのデバイスの設定	デバイス設定
プロジェクトのバージョンの管理	バージョン管理

## ユニット依存性とエディタ

WinCC flexible の[プロジェクト]ウィンドウにおけるプロジェクトの表示とエディタのファンクション範囲は、選択した HMI デバイスによって決まります。HMI デバイスで使用可能なオブジェクトとエディタについては、マニュアルを参照してください。

## "テーブル"エディタと"イメージ"エディタ

スクリーンエディタのようなグラフィックエディタは、[プロジェクト]ウィンドウおよび[オブジェクト]ウィンドウの両方に属すエレメントを表示します。グラフィックエディタで作業エリアの各オブジェクトを開きます。

"タグ"エディタなどの"テーブル"エディタは、[オブジェクト]ウィンドウの関連するオブジェクトだけを表示します。オブジェクトを編集するためにテーブルエディタを開くと、すべての関連するオブジェクトが作業エリアにテーブル形式で表示されます。

### 注記

#### OP 73 から OP 77 HMI デバイスに変更するためのフォントサイズ

HMI デバイスを変更すると、アラームウィンドウなどの画面オブジェクトのフォントサイズが変更されます。[オプション]>[設定]>[スクリーンエディタ]の[スクリーンエディタの設定]メニューの[デバイスが変わったときに画面とオブジェクトを合わせる]オプションを無効にします。

### 3.4.2 プロジェクトの表示

#### 原理

プロジェクトで使用可能な構成要素とエディタのすべてが、[プロジェクト]ウィンドウにツリー構造で表示されます。

#### [プロジェクト]ウィンドウ内のプロジェクトの表示

使用可能なすべてのエディタが、[プロジェクト]ウィンドウのプロジェクトノードの下に表示されます。プロジェクト内のオブジェクトは、さまざまなエディタを使って編集することができます。

オブジェクトを構造化して保存できるように、各エディタのサブエレメントとしてフォルダが提供されています。さらに、画面、レシピ、スクリプト、ログおよびレポートに対して、作成されたオブジェクトへの直接アクセスが行えます。



[プロジェクト]ウィンドウの表示は、プロジェクトを作成するときに選択した HMI デバイスによって決まります。選択した HMI デバイスでサポートされているエディタだけが表示されます。たとえば、"TP170A"が組み込まれた場合は、"TP170A"がログ機能を持っていないために"ログ"エディタが使用できません。

[プロジェクト]ウィンドウで編集するプロジェクトオブジェクトを選択します。これを実行するには、該当するオブジェクトをダブルクリックします。該当のエディタが開きます。

## [オブジェクト]ウィンドウにおけるオブジェクトの表示

[オブジェクト]ウィンドウは、[プロジェクト]ウィンドウで選択したフォルダとエディタの内容と関連情報を表示します。[オブジェクト]ウィンドウは、デフォルトで[プロジェクト]ウィンドウの下に表示されます。

[オブジェクト]ウィンドウは、次の3つの列で構成されています。

- オブジェクトタイプ
- [名前]  
オブジェクトの名前
- "情報"  
構築プランナーが入力したコメントなどの簡単な情報



以下のアイコンによって、[オブジェクト]ウィンドウにオブジェクトが表示されます。

シンボル	簡単な説明	シンボル	簡単な説明
	図		データログ
	タグ		アラームログ
	アナログアラーム		プロトコル
	ディスクリートアラーム		接続
	システムアラーム		サイクル
	アラームクラス		テキストリストおよびグラフィックリスト
	アラームグループ		ユーザー

シンボル	簡単な説明	シンボル	簡単な説明
	レシピ		ユーザーグループ
	タスク		

### [オブジェクト]ウィンドウによる作業

[オブジェクト]ウィンドウでオブジェクトをダブルクリックして、対応するエディタを開きます。

さらに、[オブジェクト]ウィンドウに表示されたすべてのオブジェクトに対してドラッグアンドドロップ動作を実施することができます。サポートされているドラッグアンドドロップ動作を以下に示します。

- タグを作業エリアのプロセス画面に移動 これでタグにリンクされた I/O フィールドが作成されます。
- プロセス画面を作業エリアのほかのプロセス画面に移動 関係するプロセス画面用に[画面切替]ボタンを作成します。

### 3.4.3 [プロジェクト]ウィンドウでの作業

#### 原理

[プロジェクト]ウィンドウにプロジェクトを表示して編集することができます。

[プロジェクト]ウィンドウで実行可能なアクションを以下に示します。

- ダブルクリック
- ショートカットメニューでのコマンドの選択
- ドラッグアンドドロップ動作

#### ダブルクリック

[プロジェクト]ウィンドウでフォルダをダブルクリックすれば、フォルダが開きます。

[プロジェクト]ウィンドウでエディタ("タグ"エディタなど)またはオブジェクト(画面など)をダブルクリックすれば、エディタが開きます。

## ショートカットメニュー

オブジェクトまたはフォルダ上にポインタを移動してマウスの右ボタンをクリックすれば、それぞれのショートカットメニューが表示されます。ショートカットメニューでは、以下のアクションを使用できます。

アクション	説明
[エディタを開く]	エディタを開きます。
[フォルダの追加]	サブフォルダを作成します。サブフォルダを作成することにより、トピックに基づいてオブジェクトをソートできます。
[削除]	選択したオブジェクトまたはフォルダを削除します。
[名前の変更]	選択したオブジェクトまたはフォルダの名前を変更できます。
[元に戻す]	直前のプロセスに復帰します。
[切り取り]	クリップボード内のオブジェクトまたはフォルダをクリップボードにコピーして、画面から削除します。
[コピー]	クリップボード内のオブジェクトまたはフォルダをコピーします。
[貼り付け]	クリップボードに保存されたオブジェクトを挿入します。
[印刷の選択]	選択したオブジェクトまたはフォルダを印刷します。
[クロスリファレンス]	選択されたオブジェクトやフォルダのすべての使用場所を表示します。
[プロパティ]	選択したオブジェクトまたはフォルダのプロパティを表示します。

## ドラッグアンドドロップ動作

以下のアクションにドラッグアンドドロップを使用できます。

- エディタへのオブジェクトの挿入  
 [プロジェクト]ウィンドウからイメージをドラッグして別の画面にドロップします。その時、クリックすると画面の内容が最初の画面に戻るボタンがその画面に割り付けられます。
- オブジェクトのサブフォルダ内への移動またはコピー  
 [プロジェクト]ウィンドウにオブジェクトとサブフォルダが表示されていれば、ドラッグアンドドロップ操作でオブジェクトをサブフォルダに移動したり、コピーしたりすることができます。

### 3.4.4 [オブジェクト]ウィンドウでの作業

#### 原理

[オブジェクト]ウィンドウは、オブジェクトの概要を提供します。

[オブジェクト]ウィンドウで実行可能なアクションを以下に示します。

- ダブルクリック
- ショートカットメニューでのコマンドの選択
- ドラッグアンドドロップ

#### ダブルクリック

[オブジェクト]ウィンドウでフォルダをダブルクリックすれば、フォルダが開きます。

[オブジェクト]ウィンドウでオブジェクト(画面など)をダブルクリックすれば、そのオブジェクトのエディタが開きます。

#### ショートカットメニュー

ショートカットメニューでは、以下のアクションを使用できます。

アクション	説明
[エディタを開く]	エディタを開きます。
[フォルダの追加]	サブフォルダを作成します。サブフォルダを作成することにより、トピックに基づいてオブジェクトをソートできます。
[削除]	選択したオブジェクトまたはフォルダを削除します。
[名前の変更]	選択したオブジェクトまたはフォルダの名前を変更できます。
[元に戻す]	直前のプロセスに復帰します。
[切り取り]	クリップボード内のオブジェクトまたはフォルダをクリップボードにコピーして、画面から削除します。
[コピー]	クリップボード内のオブジェクトまたはフォルダをコピーします。
[貼り付け]	クリップボードに保存されたオブジェクトを挿入します。
[印刷の選択]	選択したオブジェクトまたはフォルダを印刷します。
[クロスリファレンス]	選択されたオブジェクトやフォルダのすべての使用場所を表示します。
[プロパティ]	選択したオブジェクトまたはフォルダのプロパティを表示します。

#### ドラッグアンドドロップ

以下のアクションにドラッグアンドドロップを使用できます。

- エディタへのオブジェクトの挿入

ドラッグアンドドロップを使用して、[オブジェクト]ウィンドウからオブジェクトをドラッグして、そのオブジェクトを編集可能なエディタにドロップすることができます。この応用例が、画面に対するタグのリンク付けです。[オブジェクト]ウィンドウからタグをドラッグして画面にドロップすれば、I/O フィールドが自動的に作成されます。

- オブジェクトのサブフォルダ内への移動またはコピー

[オブジェクト]ウィンドウにオブジェクトとサブフォルダが表示されていれば、ドラッグアンドドロップ操作でオブジェクトをサブフォルダに移動したり、コピーしたりすることができます。

### 3.4.5 既存のプロジェクトの移行

#### ProTool および WinCC からの移行プロジェクト

ProTool または WinCC で作成したプロジェクトを WinCC flexible で開くこともできます。インストールした WinCC flexible のエディションが、定義した HMI デバイスをサポートしていれば、これらのプロジェクトは自動的に変換されます。

"HmiProjects"タイプのファイルの代わりに、以下のタイプの 1 つを[開く]ダイアログで開きます。

- ProTool プロジェクト

プロジェクトを開くときに、すべてのデータが変換されます。その後では、このプロジェクトは、WinCC flexible プロジェクトとしてしか保存することができません。

- WinCC プロジェクト

WinCC バージョン 6.2 のプロジェクトのみ、非常に限られた範囲内で WinCC flexible に移行できます。

## 3.5 プロジェクトの変換

### 3.5.1 さまざまな WinCC flexible バージョンを持つプロジェクト

#### はじめに

WinCC flexible は、以前の製品バージョンの WinCC flexible で作成したプロジェクトを、自動的に現行のバージョンに変換します。この機能性により、以前のバージョンで作成したプロジェクトを、現在の WinCC flexible バージョンで編集できます。WinCC flexible では、プロジェクトを以前の製品バージョンで保存することもまた可能です。変換したプロジェクトは、選択した以前の製品バージョンでのみ編集可能です。

#### 必要条件

[バージョンを付けて保存]機能がインストールされていること。詳細はインストール説明書を参照してください。

#### 変換オプション

WinCC flexible では、プロジェクトを使用可能な製品バージョンに変換できます。以下の変換オプションが使用できます：

プロジェクト作成手段	バージョン変更
WinCC flexible 2008	WinCC flexible 2007
WinCC flexible 2007	WinCC flexible 2005 SP1
WinCC flexible 2005 SP1 HF7	WinCC flexible 2007 WinCC flexible 2005 WinCC flexible 2004 SP1
WinCC flexible 2005 SP1	WinCC flexible 2007 WinCC flexible 2005 WinCC flexible 2004 SP1
WinCC flexible 2005	WinCC flexible 2007 WinCC flexible 2005 SP1 WinCC flexible 2004 SP1
WinCC flexible 2004 SP1	WinCC flexible 2007 WinCC flexible 2005 SP1 WinCC flexible 2005
WinCC flexible 2004 HF2	WinCC flexible 2007 WinCC flexible 2005 SP1 WinCC flexible 2005

プロジェクトを開くと、システムはプロジェクトを、以前のバージョンの WinCC flexible から現行のバージョンに変換します。"バージョンを指定して保存"コマンドを使用すると、プロジェクトを以前のバージョンで保存できます。

WinCC flexible 2005 SP1 以降のバージョンでは STEP 7 や SIMOTION Scout へ統合されるプロジェクトの変換に対応しています。詳細は"統合プロジェクトの変換"の章を参照してください。

#### ProAgent フォルダは、WinCC flexible では表示されません。

ProAgent エレメントを使ってプロジェクトの変換全体を実行する前に、ProAgent オプションをインストールします。対応するカスタマイズ済みのインストールを開始します。

## 3.5.2 さまざまな WinCC flexible バージョンを持つプロジェクト間の相違点

### はじめに

さまざまなバージョンの WinCC flexible のプロジェクトを編集するときには、多数ある特殊な機能に注意してください。以前のバージョンの WinCC flexible に保存するプロジェクトには、新しい機能は限られた範囲だけに使用できます。以前のバージョンの WinCC flexible の設定を実行するときは、希望する対象バージョンがサポートしている機能だけを使用できます。より最新の WinCC flexible バージョンの新しい機能は、以前のバージョンの WinCC flexible ではサポートされていません。

### 変換したプロジェクトのシステム要件

変換したプロジェクトを編集するには、変換中に選択したバージョンの WinCC flexible が必要です。指定したバージョンのサービスパックを、インストールする必要があります。オペレーティングシステムに必要なサービスパックとシステム要件にも注意してください。システム要件に関する情報については、各バージョンの WinCC flexible の文書を参照してください。無効となった HMI デバイスに関しては、ハードウェアサポートパッケージが、WinCC flexible の対応するバージョンに対して利用可能であるかを確認してください。

### ライブラリの HMI デバイス

WinCC flexible を使用して現在のバージョン以前のバージョンでプロジェクトを保存する場合、システムでは、選択した WinCC flexible バージョンに合致する HMI デバイスバージョンを自動的に選択します。ライブラリ内の HMI デバイスが旧型 HMI デバイスバージョンに、自動的に調整されることはありません。この種類の HMI デバイスを古い WinCC flexible バージョンで使用する場合、デバイス変更のダイアログで HMI デバイスを追加してから、HMI デバイスのバージョンを合わせる必要があります。

### アラームクラス群の表示

WinCC flexible を古いバージョンで保存する場合、古いバージョンの WinCC flexible の"システムイベント"エディタでは、アラームクラスはシステムイベントに割り当てられていません。この問題は、エンジニアリングシステムでの表示に固有のもので、設定を変更しないでプロジェクトを Runtime に実行することはできません。

### 画面ナビゲーション設定

バージョン WinCC flexible 2005 以降から WinCC flexible 2004 SP1 バージョンに WinCC flexible プロジェクトを保存する場合、画面ナビゲーションにバージョン WinCC flexible 2004 SP1 からのデフォルト設定が使用されます。それ以前のデフォルト設定を使用すると、設定された画面ナビゲーションのレイアウトが変更される場合があります。たとえば、キーボードデバイス用の「ファンクションキーの調整」の設定と追加する行や列の設定がなくなりました。

プロジェクトをバージョン WinCC flexible 2005 またはそれ以前の WinCC flexible のバージョンで保存すると、ナビゲーションバーで画面オブジェクトの背面色が使用されません。

### 変換したプロジェクトのスクリプト

プロジェクトを異なるバージョンの WinCC flexible で保存する場合、その異なるバージョンでの機能範囲の違いにより、スクリプトに矛盾が生じる可能性があります。

スクリプトの矛盾を取り除くには、"スクリプト"エディタで変換したバージョンのスクリプトを開いてから、"Check syntax"ファンクションを実行します。

### 変換したプロジェクトでの OPC 接続

変換したプロジェクトでは、OPC XML ラッパーを用いて、OPC サーバー名と OPC 接続のアイテム名を手動で合わせる必要があります。

バージョン	OPC Server 名	アイテム名
WinCC flexible 2004 SP1	n OPC.Siemens.Xml 1 OPC.Siemens.XML など	Win CC Flexible RT<@[タグ名] Win CC Flexible RT<@>S7_Bool_Tag_out など
WinCC flexible 2005 の場合	OPC.Siemens.XML	[プレフィックス]:Win CC Flexible RT<@[タグ名] 1:Win CC Flexible RT<@>S7_Bool_Tag_out など

### ユーザーパスワード

WinCC flexible 2007 以降では特殊文字を含むユーザーパスワード設定オプションに対応しています。特殊文字を含むパスワードはそれ以前の WinCC flexible では対応していません。

WinCC flexible 2007 以降のバージョンのプロジェクトをそれより古いバージョンに変換すると、特殊文字を含むパスワードは、デフォルト値"100"にリセットされます。変換の完了後に特殊文字なしの新しいパスワードを定義します。

### 機能の違い

WinCC flexible プロジェクトを以前のバージョンで保存し、その以前のバージョンで HMI デバイスに転送する場合、機能を完全に固定したり、機能の強化を行うと機能の差異が発生する可能性があります。

### 3.5.3 HMI デバイスバージョンの区別

#### はじめに

機能が拡張されているため、多数のさまざまな HMI デバイスバージョンが、種々のバージョンの WinCC flexible で利用できます。WinCC flexible プロジェクトを異なるバージョンに変換するときは、使用する HMI デバイスに合った HMI デバイスバージョンを使用してください。

#### HMI デバイスバージョンの選択

WinCC flexible を使用して現在のバージョン以前のバージョンでプロジェクトを保存する場合、システムでは、選択した WinCC flexible バージョンに合致する HMI デバイスバージョンを自動的に選択します。新しい HMI デバイスは現在のバージョンの WinCC flexible だけでサポートされており、以前のバージョンの WinCC flexible プロジェクトでは使用できません。前のバージョンの WinCC flexible で利用できない HMI デバイスを含む、前のバージョンのプロジェクトを保存しようとする時、システムは対応するダイアログボックスを提示します。[OK]を押してこのダイアログを確認すると、[デバイスタイプの変更]ダイアログボックスが開きます。ダイアログボックスには、選択した WinCC flexible のバージョンでサポートされているすべての HMI デバイスを、表示します。適切な HMI デバイスを選択します。プロジェクトを選択した HMI デバイスに変換し、保存します。ダイアログで[キャンセル]を選択すると、保存操作がキャンセルされて変換は実行されません。

現在のバージョンの WinCC flexible で以前のバージョンのプロジェクトを変換すると、変換している間に HMI デバイスのバージョンは自動的に現在のバージョンに変換されます。プロジェクトを転送する前に、現在のオペレーティングシステムのバージョンを、HMI デバイスにインストールする必要があります。詳細については、「Operating System Transfer (オペレーティングシステム転送)」章を参照してください。

現在のバージョンの WinCC flexible を使用すると、様々な HMI デバイスバージョンを設定できます。既存のプロジェクトを変更する前に、HMI デバイスのバージョンを選択します。HMI デバイスのバージョンを変換しても、既存の設定データはすべてプロジェクトファイル内に含まれています。エンジニアリングシステムでは、ファンクションだけはまだ使用できるため、現在の HMI デバイスの選択したバージョンでサポートされている設定データに限り表示されます。詳細については、「Project HMI device dependency (プロジェクト HMI デバイス依存)」章を参照してください。

## 3.6 プロジェクトデータの再利用

### 3.6.1 コピーのメカニズム

#### 概要

WinCC flexible で、オブジェクトをクリップボード内にコピーして、別の HMI デバイス上でコピーしたオブジェクトを、同一プロジェクト内または別のプロジェクトに挿入します。WinCC flexible で異なるコピープロセスを理解するには、2 つの異なるタイプのオブジェクト接続を区別する必要があります。

"参照"は、2 つのオブジェクト間の非排他的な接続を指定します。参照オブジェクトは、追加オブジェクトでも参照される可能性があります。タグまたはログは、プログラムの異なるオブジェクトなどで使用されます。

お互いに排他的な関係を持ったオブジェクトがある参照と比較すると、"親子関係"であるといえます。親子関係は、その参照に画面とオブジェクトなどが含まれていたり、または画面フォルダに画面が含まれていたりします。

コピー用に 2 つの異なるコピーメカニズムを使用できます。

選択したオブジェクトと子オブジェクトには、コマンド[フラットコピー]を使用してコピーします。フラットコピー中は、参照はコピーされません。

コマンド[コピー]は、子オブジェクトおよび既存の参照と一緒に、選択したオブジェクトをコピーします。

---

#### 注記

##### 多言語オブジェクトのコピー

多言語オブジェクトを異なるオブジェクトにコピーした場合、ターゲットプロジェクトで有効化されているプロジェクト言語のテキストオブジェクトのみが含まれます。コピーを転送するときに、該当するテキストオブジェクトを含めるには、ターゲットプロジェクトですべてのプロジェクト言語を有効化します。

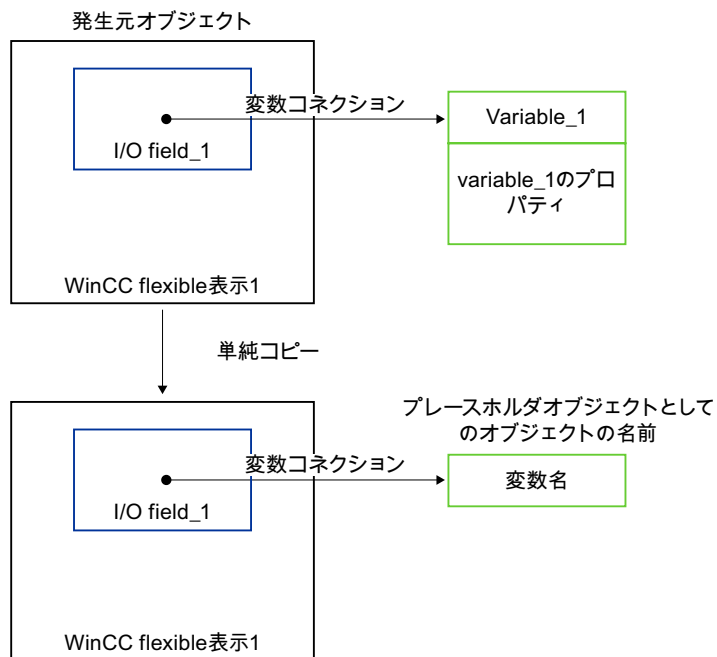
---

## 3.6.2 フラットコピー

### フラットコピー

[フラットコピー]は、親子関係を含むそれらのオブジェクトを完全にコピーします。画面をコピーする場合は、含まれているオブジェクトをすべてコピーします。[フラットコピー]を使用すると、[参照を開く]のように参照が単純にコピーされます。プレースホルダオブジェクトをコピーすると、参照の名前のみを保存します。

#### フラットコピーの例



WinCC flexible の画面には、タグ接続付き I/O フィールドが表示されています。フラットコピーは、WinCC flexible の画面と IO フィールドが親子関係であるため、IO フィールドが含まれている WinCC flexible の画面をコピーします。参照を処理しているという理由で、タグのオブジェクト名だけがコピーされます。

### オブジェクトの挿入

クリップボード内にコピーしたオブジェクトは、同じオブジェクトの別のデバイス内、または異なるオブジェクト内に挿入される可能性があります。同じ名前を使用したオブジェクトが既にターゲットプロジェクトに存在する場合、コピーしたオブジェクトが、連続番号を使用した同じ名前でも保存されます。同じ名前および連続番号を持った複数のオブジェクトが存在する場合は、その次の使用可能な番号が使われます。子オブジェクトと同様に親オブジェクトも挿入されます。参照のオブジェクト名だけが挿入されます。有効なプロパティを持つ同じ名前の参照がターゲットオブジェクトに存在する場合、既存の参照はコピー済みのオブジェクトに接続されます。

## オブジェクトの置換

ターゲットプロジェクトでの既存のオブジェクトは、クリップボードにコピーしたオブジェクトと置換される可能性があります。コピーしたオブジェクトの全内容は、フラットコピー時にターゲットプロジェクトで既存のオブジェクトの内容と比較されます。同じ名前を持った既存のオブジェクトは置換されます。含まれている子オブジェクトも置換されます。フラットコピー中には、参照はコピーされません。参照のオブジェクト名だけが挿入されます。有効なプロパティを持つ同じ名前の参照がターゲットオブジェクトに存在する場合、既存の参照はそのときコピーしたオブジェクトに接続されます。

### 3.6.3 コピー

#### メニューコマンド[コピー]

オブジェクトの広範囲のコピーを作成するには、メニューコマンド[コピー]を使用します。[フラットコピー]を使用すると、親子関係を持つオブジェクトと同様に参照もコピーされます。

#### フラットコピーの例

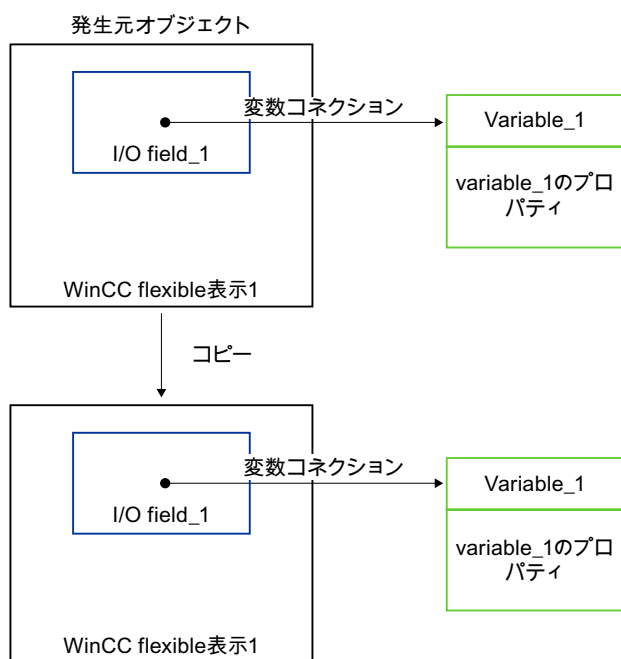


図 3-1 コピー

WinCC flexible の画面には、タグの添付とともに I/O フィールドが含まれています。メニューコマンド[コピー]を使用するコピー時には、そのプロパティが含まれている画面、IO フィールドおよびタグがコピーされます。

## オブジェクトの挿入

クリップボード内にコピーしたオブジェクトは、同じオブジェクトの別のデバイス内、または異なるオブジェクト内に挿入される可能性があります。同じ名前を使用したオブジェクトが既にターゲットプロジェクトに存在する場合、コピーしたオブジェクトが、連続番号を使用した同じ名前で保存されます。同じ名前および連続番号を持った複数のオブジェクトが存在する場合は、その次の使用可能な番号が使われます。親オブジェクト、子オブジェクトおよび参照が挿入されます。コピーが参照を含む場合、システムはターゲットオブジェクトで既存の参照を持つこれらの参照を比較します。既存の参照が適合する機能を持っている場合、それらは挿入したオブジェクトに接続されます。既存の参照が適合しない場合、コピーした参照がコピーされます。

## オブジェクトの置換

ターゲットプロジェクトでの既存のオブジェクトは、クリップボードでコピーしたオブジェクトと置換される可能性があります。広範囲のコピーをする時に、コピーしたオブジェクトの全内容は、ターゲットプロジェクトで既存のオブジェクトの内容と比較されます。同じ名前を持った既存のオブジェクトは置換されます。含まれている子オブジェクトは置換されます。コピーが参照を含む場合、システムはターゲットオブジェクトで既存の参照を持つこれらの参照を比較します。置換する必要がある既存の参照を決定してダイアログが開きます。置換する必要がある参照のタイプをダイアログで選択します。

### 3.6.4 置換

#### アプリケーション

WinCC flexible の[置換]コマンドを使用して、WinCC flexible プロジェクトに ProAgent プロジェクトのオブジェクトなどの新しいオブジェクトや更新済みオブジェクトを統合できます。

[置換]コマンドを使用すると、オブジェクトが適用されるか、他のプロジェクトから更新されます。プロジェクト構造の調整なしにオブジェクトを有効にできます。

オブジェクトには、画面などの個別オブジェクトまたは HMI デバイスの構成全体を適用できます。

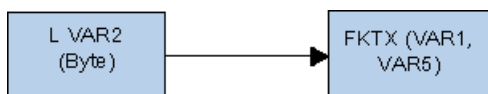
#### プロセス

オブジェクトの広範囲のコピーを作成するには、メニューコマンド[コピー]を使用します。コピー中は、親子関係を持つオブジェクトと同様に参照もコピーされます。コピーを挿入するには、[置換]メニューコマンドを使用します。

ターゲットプロジェクトに同一名のオブジェクトが存在しないか、オブジェクト名が確認されます。同じ名前を持った既存のオブジェクトは置換されます。含まれている子オブジェクトも置換されます。ターゲットプロジェクトのレシピに 50 エントリが存在し、ソースプロジェクトのレシピに 500 エントリが存在します。ターゲットプロジェクトのレシピを置換すると、ターゲットオブジェクトにないエントリが設定されます。ターゲットプロジェクトにすでに存在するエントリの場合、設定はソースプロジェクトの設定で置き換えられます。

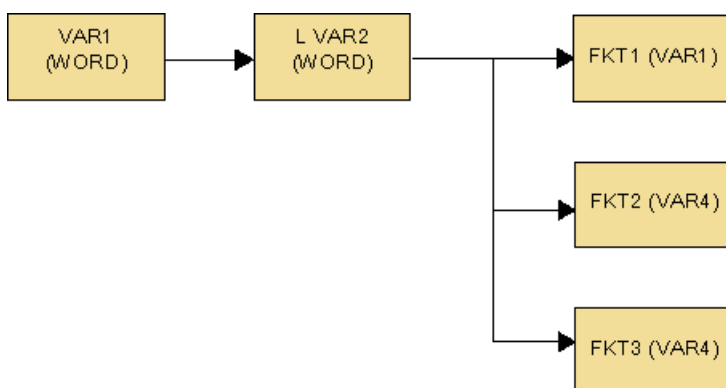
コピーが参照を含む場合、システムはターゲットオブジェクトで既存の参照を持つこれらの参照を比較します。ダイアログボックスを使って、どの参照を置換するか設定できます。

### ソースプロジェクト



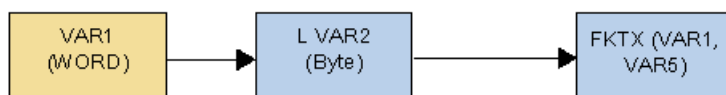
VAR2 タグは、BYTE データタイプで、FKTX 関数を参照します。

### ターゲットプロジェクト



VAR1 タグには、WORD データタイプの制限タグ VAR2 があります。制限タグは、"FKT1"、"FKT2"、"FKT3"の 3 つの関数を参照します。

### 結果



制限タグ VAR2 は、ソースプロジェクトの VAR2 で置き換えられます。ターゲットプロジェクトの参照は、置換によって変更されません。不足している FKTX 参照がターゲットプロジェクトに追加されます。挿入されたオブジェクトからの VAR5 参照タグもターゲットプロジェクトに挿入されます。

#### 注記

置換を使ってオブジェクトを挿入する場合、次の点に注意してください。

- 挿入できるオブジェクトインスタンス数は HMI デバイスによって異なります。
- ターゲットプロジェクトの関数スコープがソースプロジェクトより小さい場合、すべてのオブジェクトが適用されないことがあります。
- HMI デバイスの設定全体を置換した場合、プロセスを元の戻すことはできません。ただし、個別オブジェクトの置換は元に戻すことができます。
- ターゲットプロジェクトの置換時にエラーが発生した場合、プロジェクトの生成時に報告されます。
- [コピー]と[置換]のメニューコマンドを使用して、銘板とそのインスタンスを置換できません。詳細については、「銘板のコピー」を参照してください。

### 3.6.5 ライブラリの使用

#### 原理

ライブラリによって、オブジェクトの複数使用が可能になります。ライブラリを使用すれば、単純なグラフィックから複雑なモジュールまで、すべてのオブジェクトタイプを保存することができます。

WinCC flexible は、タスクごとのライブラリを提供しています。

- 共有ライブラリ

グローバルライブラリは、プロジェクトから独立してファイルシステム内にファイルとして(デフォルトでは、WinCC flexible のインストールディレクトリに)保存されます。グローバルライブラリは、すべてのプロジェクトで使用可能です。

- プロジェクトライブラリ

プロジェクトライブラリは、データベースにプロジェクトデータと共に保存され、プロジェクトライブラリが作成されたプロジェクトでのみ使用可能です。

この2つのライブラリ間でオブジェクトを交換することができます。

#### ライブラリ内のオブジェクト

グラフィックオブジェクト、画面、アラームおよびタグなどのドラッグアンドドロップ操作で移動することができるオブジェクトはすべて、ライブラリに保存することができます。

他のオブジェクトを参照しているオブジェクトをライブラリに保存するときに、参照オブジェクトもライブラリに保存するかどうかを選択することができます。参照オブジェクトとして、たとえば I/O フィールドに対するタグが挙げられます。

#### ライブラリの構成

以下の構成オプションがライブラリ用に提供されています。

- オブジェクトを整理するためのフォルダの作成

- ライブラリオブジェクトの表示の変更

たとえば、小さいアイコンまたは名前の付いていないライブラリオブジェクトを表示することができます。

- ライブラリオブジェクトの複数言語構成

### 3.6.6 銘板の使用

#### 原理

銘板は事前コンフィグレーションオブジェクトの集合です。銘板は利用可能な画面オブジェクト数を拡大し、コンフィグレーションの作業量を減少させます。銘板は銘板デザイナーで作成、編集されます。

このエディタを使用して、使用時に作成される銘板プロパティを定義します。このプロパティは、含まれているオブジェクトのプロパティになります。

さらに、銘板が使用されているすべての場所は、"銘板"エディタで集中的に管理することができます。銘板変更後は、銘板が使用されるすべての場所または選択した場所のみを更新することができます。

## 3.7 クロスリファレンスの作業

### 原理

"クロスリファレンスリスト"エディタによって、タグや画面などの特定のオブジェクトの使用ポイントを設定したり、直接それらのポイントにスキップしたりできます。

### クロスリファレンスのユーザーインターフェース



アイコンをダブルクリックしてオブジェクトを開きます。

### ショートカットメニューを使用して実行する可能性のあるクロスリファレンスの作業

[使用方法へ]コマンドを使用して、プロジェクト内の使用場所に直接スキップします。または、使用ポイントアイコンをダブルクリックします。

"クロスリファレンス"エディタで表示を変更することができます。表示を変更するために、以下のコマンドを使用できます。

- すべて折りたたむ  
すべてのオブジェクトの使用場所リストを非表示にするには、[すべて折りたたむ]コマンドを使用します。
- すべてを展開  
[すべてを展開]コマンドを使用すれば、使用場所のリストを元に戻すことができます。

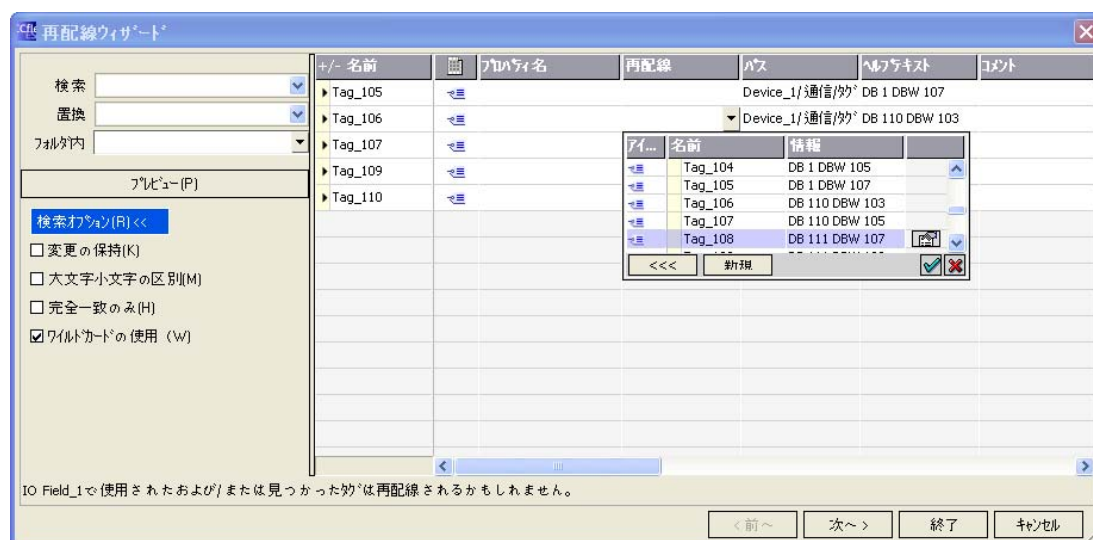
"印刷"コマンドを使用して、クロスリファレンスリストを印刷します。

## 3.8 再配線の概要

### はじめに

オブジェクトのタグリンクを変更できます。"再配線"ウィザードを使用して、新しいタグを、たとえば、画面の I/O フィールドへ割り当てます。"再配線"ウィザードでタグを検索し、下記の場所を再配線します：

- [プロジェクト]ウィンドウで選択したオブジェクト内
- 作業エリアで選択したオブジェクト内



再配線には、以下のオプションがあります。

- "再配線"欄で個別に再配線
- "検索"と"置換"フィールドを使用した再配線

## 3.9 プロジェクト内部の検索と置換機能

### 原理

WinCC flexible は、文字列やオブジェクトの検索と置換を実行することができます。

- 作業エリアの文字列を検索および置換できます。この場合は、[作業エリア内の検索]および[プロジェクト内の置換]ダイアログを使用します。
- プロジェクト全体またはプロジェクトの一部内でオブジェクトを検索できます。この場合は、[プロジェクト内の検索]ダイアログを使用します。

## 3.10 WinCC flexible 内の文書に関する基本原理

### 原理

プロジェクト文書を使用して、構築データの概要を確認することができます。

プロジェクト文書は以下のような形で提供されます。

- 画面に表示
- PDF や HTML などのファイルとして出力。
- プリンタに出力

プロジェクトデータの一部だけをプロジェクト文書に使用したい場合は、対応するオブジェクトを選択します。

## 3.11 生成時の一貫性チェック

### はじめに

設定中、入力されたデータの整合性を自動的にチェックします。


例:

"バイト"データタイプで作成されたタグは、0 から 255 の値が想定されています。255 以上の初期値を入力しようとするすると拒否されます。また、値の有効範囲を表示したメッセージが表示されます。

たとえば、整合性テストで値の有効範囲が確認されるため、構築フェーズにおける不正な入力が表示されます。

I/O フィールドにタグが割り当てられていない場合など、不完全な設定は入力時に確認されません。生成中に指定が確認されます。出力ウィンドウにアラームが出力されます。

### 生成時の一貫性チェック

設定データの一貫性を確認するには、 をクリックするか、[プロジェクト]>[ジェネレータ]>[生成]メニューコマンドを選択して、生成開始します。プロジェクトが作成されます。プロジェクト内の不正なポイントが、[出力]ウィンドウにリスト表示されます。出力ウィンドウのエントリを使って、エラーの原因に直接飛ぶことができます。エラーリストの上から下まで対処します。

生成中に常に再生成されるのは設定変更のみです。エンジニアリング時に設定に多数の変更を加えた場合、保存場所のサイズが適時増大されます。たとえば、プロジェクトの一時ファイルを削除してプロジェクト保存場所のサイズを削減し、プロジェクト終了後のメモリリソースの負荷を軽減します。[オプション]>[一時ファイルの削除]のメニューコマンドを選択して、増大したプロジェクトデータベースを必要最低限度のサイズに削減します。

[一時ファイルの削除]のメニューコマンドは、すべての一時プロジェクトデータを削除します。

コンパイル済みプロジェクトファイルを復元するには、生成を実行します。一時データ削除の完了後に初期生成を実行すると、プロジェクト全体を再コンパイルします。".\*fwx"と".\*srt"は完全に再作成されます。プロジェクトの大きさに応じて、このプロセスには時間がかかる場合があります。

[すべての再構築...]コマンドを選択しても、プロジェクト全体のコンパイルを開始できます。

## すべての再構築

商用プロセスにプロジェクトをリリースする前に[すべての再構築]コマンドを実行して、プロジェクト全体を再コンパイルします。

また、現在のエンジニアリングセッションでの差分データのコンパイル所要時間を削減するために、適切な周期で[すべての再構築]コマンドを実行することをお勧めします。

[プロジェクト]>[コンパイラ]>[すべての再構築]メニューコマンドを選択して、フルコンパイルを実行してください。

複数の HMI デバイスを設定している場合、[すべての再構築]コマンドを実行すると、[生成する HMI デバイスの選択]ダイアログが開きます。

このダイアログから生成する HMI デバイスを選択します。複数選択も可能です。

## 3.12 デバッグオブジェクト

### シミュレータを使用したプロジェクトのテスト

シミュレータによって、プロジェクトを構築コンピュータ上で直接シミュレートすることができます。すべての設定可能な HMI デバイス用にプロジェクトをシミュレートできます。シミュレーションプログラムは、WinCC flexible と共にインストールされた独立したツールです。タグおよびエリアポインタ用の値を設定することにより、シミュレータで設定の結果をテストすることができます。

タグ値のシミュレーションは、シミュレーションテーブルを介して実行するか、実際の PLC とのシステム通信によって実行することができます。

### Windows CE デバイスによるプロジェクト検証

Windows CE HMI デバイス向けプロジェクトのシミュレーション時に次の注意事項に留意してください。プロジェクトに VBS オブジェクトを含むスクリプトを作成する場合、シミュレータでランタイム時にエラーメッセージを生成できます。

原因：VBS オブジェクトによっては、オペレーティングシステムに依存するものがあります。Windows CE HMI デバイス向けプロジェクトは、Windows XP などの Windows オペレーティングシステム上でシミュレートします。この場合、シミュレーションに Windows XP スクリプトエンジンが使用されます。Windows CE だけでサポートされている VBS オブジェクトがスクリプトに含まれている場合、該当するエラーメッセージが表示されます。

オンラインヘルプ「Windows CE 用 VBScript」では「Windows 用 VBScript」のリストと異なるファンクションリストが提供されます。

## 3.13 プロジェクトの転送

### 3.13.1 転送操作の基本原則

#### 転送

転送操作とは、プロジェクトを実行する HMI デバイスに完全なプロジェクトファイルを転送することを言います。

[プロジェクト]>[コンパイラ]>[コンパイル]または[プロジェクト]>[コンパイラ]>[すべての再構築]メニューコマンドを選択して、設定後のプロジェクトの整合性を確認します。

---

#### 注記

##### すべての再構築

商用プロセスにプロジェクトをリリースする前に[すべての再構築]コマンドを実行して、プロジェクト全体を再コンパイルします。

また、現在のエンジニアリングセッションでの差分データのコンパイル所要時間を削減するために、適切な周期で[すべての再構築]コマンドを実行することをお勧めします。

[プロジェクト]>[コンパイラ]>[すべての再構築]メニューコマンドを選択して、フルコンパイルを実行してください。

複数の HMI デバイスを設定している場合、[すべての再構築]コマンドを実行すると、[生成する HMI デバイスの選択]ダイアログが開きます。

このダイアログから生成する HMI デバイスを選択します。複数選択も可能です。

---

一貫性チェックの完了後に、システムはコンパイル済みのプロジェクトファイルを生成します。このプロジェクトファイルにはプロジェクトと同じ名前が割り付けられますが、拡張子は".fwx"になります。コンパイルされたプロジェクトファイルを、設定した HMI デバイスに転送します。

---

#### 注記

診断メッセージのために fwx ファイルのサイズはかなり大きくなる場合があります。fwx ファイルのサイズが大きすぎて HMI デバイスへ転送できない場合、アラーム設定で診断メッセージを無効にしてください。

---

プロジェクトデータを転送するには、HMI デバイスを設定コンピュータに接続する必要があります。HMI デバイスが PC の場合、ディスクなどのデータ媒体を使用しても転送操作を実行できません。

\*.pwx が見つからない場合にプロジェクトを再コンパイルすると、データの転送中にエラーメッセージが受信されます。

## 基本的な手順

1. WinCC flexible プロジェクトに、個々の HMI デバイスに応じた転送設定を入力します。
2. プロジェクトを転送する HMI デバイスで転送モードを入力します。
3. 設定コンピュータから HMI デバイスに、コンパイルしたプロジェクトファイルを転送します。プロジェクトファイルは、転送設定で選択されている各チェックボックスに対応する HMI デバイスすべてに転送されます。

## 転送モード

転送操作中、HMI デバイスは"転送モード"になっている必要があります。HMI デバイスのタイプに基づいて、転送モードは次のように有効になります。

- Windows CE システム

初めて作動するとき、HMI デバイスは自動的に転送モードで起動します。

HMI デバイスの設定メニューでこの転送オプションが有効になっている場合、転送操作を追加起動するたびに HMI デバイスが転送モードに切り替わります。

転送オプションが無効になっている場合、HMI デバイスを再起動して、[スタート]メニューで転送アプレットを呼び出します。または、プロジェクト内の Change Operating Mode システムファンクションを設定します。

- PC

HMI デバイスが、プロジェクトがまだ格納されていない PC の場合、最初の転送操作の前に"RT Loader"で転送モードを手動で有効にする必要があります。

HMI デバイスで転送モードを設定する方法の詳細については、マニュアルを参照してください。

---

### 注記

#### PROFIBUS 上での MP 377 を使ったオペレーティングシステムの転送

PROFIBUS で使用できる画像サイズとポーレートのため、PROFIBUS を使った MP 377 での画像転送には 1 時間かかることがあります。

USB またはイーサネット経由でオペレーティングシステムまたはイメージを転送します。

---

## HMI デバイスバージョン

プロジェクトをオペレータデバイスに転送すると、設定したオペレーティングシステムバージョンが、HMI デバイス上のバージョンに対応しているかどうかチェックされます。バージョンが違くと、転送が中止されて、メッセージが表示されます。WinCC flexible プロジェクトまたは HMI デバイスのオペレーティングシステムバージョンが異なる場合、次の可能性が考えられます。

- HMI デバイス上のオペレーティングシステムの更新。

詳細については、「Operating System Transfer (オペレーティングシステム転送)」章を参照してください。

### 3.13.2 プロジェクトのバック転送

#### はじめに

転送時、圧縮ソースデータファイルを、コンパイルしたプロジェクトファイルと一緒に HMI デバイスに転送できます。HMI デバイスからコンフィグレーションコンピュータにプロジェクトをバック転送するには、このソースデータファイルが必要になります。統合されたプロジェクトのアップロードは、サポートされていません。

#### バック転送の用途

通常、転送操作中、HMI デバイスに転送されるのは実行可能プロジェクトだけです。オリジナルのプロジェクトデータは設定デバイス上に残り、プロジェクトを将来さらに開発したり、エラーの分析に使用できます。

ただし、外部記憶媒体付きの Windows CE デバイス上及び PC 上には、コンパイルされたプロジェクトファイルだけでなく、プロジェクトの圧縮されたソースデータファイルも保存できます。このデータファイルを後ほど使用して、設定コンピュータにソースデータファイルをバック転送すれば、HMI デバイスまたはデバイスからプロジェクトを復旧できます。

#### 利点:

オリジナルの組込みデバイスが使用できない、または組込みデバイスのプロジェクト用のソースファイル(\*.hmi)が使用できない場合でも、バック転送操作によって既存プロジェクトを分析、変更することができます。

---

#### 注記

WinCC flexible を使用しても、HMI デバイスからコンフィグレーションコンピュータ上に ProTool プロジェクトのソースデータファイルをバック転送できます。その後で、ProTool プロジェクトを WinCC flexible プロジェクトに移行できます。

WinCC flexible でサポートされていないオペレーティングデバイス用に作成された ProTool プロジェクトのソースデータは、ProTool を使ってコンフィグレーションコンピュータにバック転送する必要があります。ProTool プロジェクトを保存します。それから、WinCC flexible を使って移行を実行します。

---

## バック転送の必要条件

- ソースデータファイルを HMI デバイスに転送できるのは、コンパイルしたプロジェクトファイルの転送操作の一部としてだけです。各 HMI デバイスの転送設定で[バック転送を有効にする]チェックボックスを選択しておく、ソースデータファイルが、コンパイルしたプロジェクトファイルと一緒に HMI デバイスに転送されます。
- 圧縮ソースデータファイルを保存するには、HMI デバイス上に、十分な記憶容量のメモリが装備されている必要があります。バック転送操作のためのソースデータファイルを Windows CE デバイス上に供給するには、このデバイスに外部メモリカードを実装する必要があります。HMI デバイスにメモリカードが実装されていない場合、またはメモリ容量が不十分な場合には、転送が中断します。ただし、コンパイルされたプロジェクトファイルは事前にその全体が転送されるため、そのデータでランタイムを起動することができます。

オペレーティングデバイスに対してイーサネット接続が利用可能で、大きなプロジェクトのソースデータをバック転送用に保存しなければならない場合は、オペレータデバイスのメモリカード以外の保存場所としてネットワークドライブを選択することができます。これによって、保存場所の問題が解消されます。

- WinCC flexible 内でプロジェクトが開いていない場合、バック転送操作を実行する前に、[通信設定]ダイアログで、バック転送用ソースデータファイルを割付ける HMI デバイスとロード方法を選択する必要があります。

WinCC flexible でプロジェクトが開いている場合、選択した HMI デバイスからバック転送操作が実行されます。この場合、WinCC flexible の[転送設定]ダイアログで、HMI デバイスに選択した転送モードが適用されます。

## 転送とバック転送

転送操作にソースファイルが指定されている場合、プロジェクトはソースフォーマット (\*.hmi) から圧縮され、\*.pdz ファイルとして HMI デバイスの外部記憶媒体に転送されるか、PC に直接転送されます。

バック転送操作では、\*.pdz ファイルが設定コンピュータに保存されます。バック転送中に WinCC flexible でプロジェクトが開いていた場合は、そのプロジェクトを保存して閉じるように要求されます。その後で、バック転送されたプロジェクトが WinCC flexible の中で解凍されて開きます。バック転送されたプロジェクトを保存するときに、名前を付ける必要があります。

### 注意

WinCC flexible は、オペレーティングユニット上のソースデータファイルが、そのデバイス上で実行しているプロジェクトに実際に所属しているかどうかチェックできません。その間、ソースデータファイルを組み込まない転送操作を実行すると、旧プロジェクトデータがまだ HMI デバイス上に存在している可能性があります。状況によっては、現在動作中のプロジェクトにデータが一致しない状態になる可能性があります。

### 注記

転送時間を出来るだけ短くするためには、出来れば小さな及び中くらいのサイズの設定に対してバック転送プロセスを使用してください。

大型のプロジェクトファイルには、次のオプションがあります: たとえば、プロジェクトマネージャの backup ファンクションを使用して、圧縮\*.arj ファイルとしてプロジェクトファイルを CF カード上に転送します。



## タグの操作

### 4.1 基本

#### 4.1.1 タグの基本

##### はじめに

外部タグを使用すると、HMI デバイスと PLC など、オートメーションプロセスの構成要素間の通信(データ交換)が可能になります。

##### 原理

外部タグは、PLC 内の定義されたメモリロケーションのイメージです。HMI デバイスと PLC の両方からこの保存先に対して、読取りおよび書込みアクセスができます。

外部タグは PLC 内の保存先のイメージであるため、使用できるデータタイプは HMI デバイスに接続されている PLC によって異なります。

STEP 7 または SIMOTION Scout に統合して構成する場合は、外部タグを作成するとき、PLC のプログラミング中に作成されたすべてのタグに直接アクセスすることができます。

分散システムで作業を行う場合は、OPC インターフェースを使って別の HMI システムのタグに直接アクセスすることができます。

##### データタイプ;データタイプ

基本データタイプは、すべての設定で使用可能です。

さらに、接続がすでに確立されている PLC 専用の外部タグの別のデータタイプを使用することもできます。

基本のデータタイプと S7 コントローラへの接続用のデータタイプの詳細リストは、「S7 に接続する場合のデータタイプ」に記載されています。別の PLC への接続に使用できるデータタイプに関する情報は、それぞれの通信ドライバのマニュアルを参照してください。

また、構造体を作成して、独自のデータタイプを作成することもできます。

---

##### 注記

外部タグの他に、HMI デバイスと PLC 間の通信にエリアインジケータを使用できます。"接続"エディタのエリアインジケータをセットアップして実行することができます。エリアインジケータに関する詳細情報は、「通信」を参照してください。

---

## 4.1.2 外部タグ

### はじめに

外部タグを使用すると、HMI デバイスと PLC の間など、オートメーションプロセスの構成要素間で通信(データ交換)が可能になります。

### 原理

外部タグは、PLC 内の定義されたメモリロケーションのイメージです。HMI デバイスと PLC の両方からこの保存先に対して、読取りおよび書込みアクセスができます。

外部タグは PLC 内の保存先のイメージであるため、使用できるデータタイプは HMI デバイスに接続されている PLC によって異なります。

外部タグを作成する時、STEP 7 または SIMOTION Scout に統合して構成する場合は、PLC のプログラミング中に作成されたすべてのタグに直接アクセスすることができます。

### データタイプ

基本データタイプは、すべてのコンフィグレーションで使用可能です。

さらに、接続がすでに確立されている PLC 専用の外部タグ用の別のデータタイプを使用することもできます。

基本のデータタイプと S7 コントローラへの接続用のデータタイプの詳細リストは、「S7 に接続する場合のデータタイプ」に記載されています。別の PLC への接続に使用できるデータタイプに関する情報は、それぞれの通信ドライバのマニュアルを参照してください。

---

### 注記

外部タグの他に、HMI デバイスと PLC 間の通信にエリアインジケータを使用できます。"接続"エディタのエリアインジケータをセットアップして実行することができます。エリアインジケータに関する詳細情報は、「通信」を参照してください。

---

### スクリプトおよびファンクションのタグ値の更新

スクリプトとシステム機能は常に、外部タグの値をランタイムメモリから取得します。Runtime の開始時に、現在値が PLC から読み取られ、Runtime メモリに書き込まれます。次に、タグ値が更新されてサイクルタイムを設定します。ファンクションとスクリプトは、前のスキャンサイクルチェックポイントで PLC から読み取られたタグ値に、最初にアクセスします。

### 4.1.3 内部タグ

#### はじめに

内部タグは、PLC には一切接続されません。

#### 原理

内部タグは、HMI デバイス上のメモリに保存されます。そのため、この HMI デバイスでだけ内部タグへの読み書きアクセスが可能です。たとえば、内部タグをローカルの計算を実行するために作成します。

内部タグでは、すべての基本データタイプを使用することができます。データタイプの詳細リストは、「基本データタイプ」に記載されています。

## 4.2 エlementと基本設定

### 4.2.1 タグエディタ;タグエディタ

#### はじめに

タグエディタでタグを作成、コンフィグレーションできます。

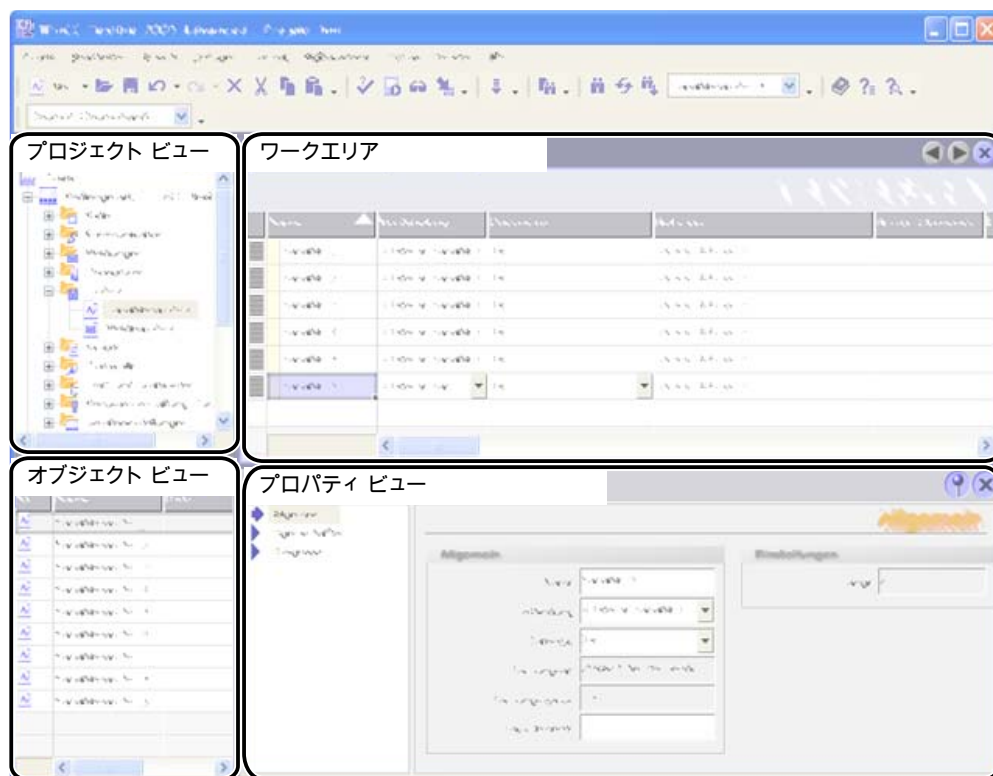
#### 開く

タグエディタを開くには、新しいタグを作成するか、[オブジェクト]ウィンドウに表示されているタグを編集します。

または、[プロジェクト]ウィンドウ内の[タグ]エントリをダブルクリックしても、タグエディタを開くことができます。

#### レイアウト;レイアウト

タグエディタは、フォルダ内のすべてのタグを表示します。



タグエディタ

## 作業エリア;サギョウエリア

すべてのタグは、作業エリアのテーブルに表示されます。テーブルのセル内にあるタグのプロパティは、編集可能です。列のヘッダーをクリックして、列のエントリによるテーブルのソートを実行することができます。

ニーズに合った列を選択して設定することができます。作成している HMI デバイスによっては、使用できない列もあります。プロジェクトが保存されると、作成されている列の選択は常に保存されます。この選択は、Microsoft Windows のログインに使用するユーザー名とリンクされます。

## [プロパティ]ウィンドウ

このダイアログボックスで、タグを設定します。[プロパティ]ウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。

[プロパティ]ウィンドウの左側には、さまざまなプロパティカテゴリを選択できるツリー構造があります。選択したプロパティカテゴリをコンフィグレーションするためのフィールドは、[プロパティ]ウィンドウの右側にあります。

## 4.2.2 タグおよび配列の基本設定

### はじめに

テーブル形式エディタでタグおよび配列エlementのプロパティを設定できます。この設定は、対応する[プロパティ]ウィンドウでも実施できます。

それぞれの[プロパティ]ウィンドウには、タグエディタと同じ情報および設定が表示されます。

### [プロパティ]ウィンドウの構造

すべての[プロパティ]ウィンドウには、さまざまなプロパティカテゴリを選択できるツリー構造があります。現在選択されているプロパティカテゴリのフィールドは、[プロパティ]ウィンドウの右側に表示されます。

### タグの[プロパティ]ウィンドウ

次のものから、タグの[プロパティ]ウィンドウで選択したタブ用のプロパティを設定できます。

ツリー構造内のエントリ	フィールド
[全般]	[名前] [接続] [データタイプ] [取得タイプ] [取得サイクル] [配列数] [長さ]
[プロパティ]	
[アドレス指定] (外部タグの場合のみ)	[アイコン](統合設定のみ) [エリア] [DB] [DBW]
[限界値]	[上限値 - 無効] [上限値定数] [上限値 - タグ] [下限値 - 無効] [下限値定数] [下限値 - タグ] 限界チェック アラームの生成
"LinearScaling" (外部タグの場合のみ)	[PLC 終了値] [PLC 初期値] [HMI デバイス - 終了値] [HMI デバイス - 開始値]
[基本設定]	[更新コード] [連続更新]
[コメント]	コメント入力用のテキストフィールド
[多重化]	間接アドレス指定の有効化および無効化。
[ロギング]	[データログ] [サンプリングのタイプ] [ロギングサイクル]
[ロギング限界]	[定数上限値] [タグ上限値] [上限値 - 限界値なし] [定数下限値] [タグ下限値] [下限値 - 限界値なし]
[イベント]	
[上限値超過]	上限値を超過した場合に処理されるファンクションのリスト

ツリー構造内のエントリ	フィールド
[値の変更]	プロセス値の変更に処理されるファンクションリスト。 1を超える値が[配列エレメント]列に入力された場合は、"値の変更"イベントだけが使用できます。
[下限値超過]	値が下限値を下回った場合に処理されるファンクションのリスト

## 4.3 タグの操作

### 4.3.1 タグのプロパティ

#### はじめに

WinCC flexible では、一部のプロパティはすべてのタグに対して設定できます。コンフィグレーション内でのタグの使用方法はプロパティによって決まります。

#### 原理

タグの場合、以下のプロパティの設定が可能です。

- [名前]
 

すべてのタグには、選択可能な名前があります。ただし、名前はタグフォルダ内で1回だけ実行可能です。
- PLC への[接続]およびタグの[ロギングサイクル]
 

外部タグの場合、これらのタグは PLC 内のメモリロケーションを表すため、HMI デバイスが接続される PLC を指定する必要があります。タグで使用可能なデータタイプおよび PLC メモリ内のアドレスは、PLC のタイプによって決まります。

さらに、タグを更新する頻度を指定する必要があります。
- [データタイプ]および[長さ]
 

タグのデータタイプによって、タグに保存される値のタイプ、内部的な保存方法、およびタグが保有できる上限値の範囲が決まります。

データタイプの簡単な例には、整数を保存するための"Int"や文字列を保存するための"String"があります。"Integer"データタイプのタグ用の値には、先行して0を入力できません。

"String"または"StringChar"タイプのテキストタグでは、バイト単位でタグの"長さ"も設定できます。その他のデータタイプではすべて、"長さ"の値は固定されています。
- [配列数]
 

同じタイプの配列エレメントのタグをまとめることができます。配列エレメントは、連続したメモリロケーションに保存されます。

主に配列タグは、同じ形式のデータを大量に扱う場合(カーブバッファやレシピの定義など)に使用します。

- [コメント]  
プロジェクトのより正確なドキュメント化を行うために、各タグのコメントを入力できます。
- [限界値]  
各タグの上限および下限値の範囲を使用して、値の範囲を指定できます。タグに保存されるプロセス値が制限範囲に入った場合、アラームメッセージを送信できます。プロセス値がこの値の範囲から外れた場合、メッセージを送信するためのファンクションリストを処理できます。
- [初期値]  
すべてのタグについて、初期値を指定できます。タグは、ランタイムの起動時にこの値に設定されます。このようにして、定義された状態でプロジェクトを毎回起動することができます。
- [ロギング]および[ロギング限界]  
文書化および後の評価を容易にするために、データを異なるログに保存することができます。  
ロギングの頻度とモードを設定できます。  
さらに、WinCC flexible では、指定されたロギング限界の範囲内または範囲外にあるデータへのロギングを制限できます。  
  
タグの作成時に設定されたプロパティはすべて、後にタグが使用されるオブジェクトリストで修正できます。  
  
例: タグを作成し、その限界値を設定します。このタグを IO フィールドにリンクします。タグの作成時に設定された限界値は、後に IO フィールドを設定するオブジェクトリストで修正できます。

### 4.3.2 外部タグを使用した PLC との通信

#### はじめに

外部タグは、HMI デバイスと PLC 間でのデータ交換に使用されます。

#### 原理

外部タグは、PLC 内の定義されたメモリロケーションのイメージです。HMI デバイスと PLC の両方から、この保存先に対して読取りおよび書込みアクセスができます。

HMI デバイスが PLC 上のデータにアクセスできることが、タグを作成する時にどのプロパティを使用可能にするかに影響します。以下のタグのプロパティによる構成が可能かどうかは、HMI デバイ스에接続される PLC によって決まります。

- [アドレス指定]
- [データタイプ]

線形スケーリングを使用して、設定の要求に合うように外部タグの値の範囲を調整できます。

## アドレス指定;アドレスシテイ

WinCC flexible で外部タグを作成する場合、PLC プログラムで保有しているアドレスと同じアドレスを指定する必要があります。これにより、HMI デバイスと PLC は同じメモリロケーションにアクセスすることが可能になります。

### 注記

統合された設定環境で外部タグを作成する場合、STEP 7 または SIMOTION Scout を使用して、PLC がプログラムされたときに作成されたシンボルテーブル内のアイコンに直接アクセスできます。この場合、タグを表すアイコンのみを選択する必要があります。それ以降の設定は、PLC プログラムに従って WinCC flexible によって行われます。

## [データタイプ]

外部タグは PLC 内の指定された保存先のイメージであるため、使用できるデータタイプは HMI デバイスに接続されている PLC によって異なります。

基本のデータタイプと S7 コントローラへの接続用のデータタイプの詳細リストは、「S7 に接続する場合のデータタイプ」に記載されています。別の PLC への接続に使用できるデータタイプに関する情報は、それぞれの通信ドライバのマニュアルを参照してください。

通信ドライバ A から通信ドライバ B に変更すると、次のような特別な状況で不一致の原因になる場合があります：

- データタイプを認識したが、アドレス範囲が確認できない。  
この場合、そのデータタイプでのデフォルトのアドレス範囲を使用します。
- 適切なデータタイプがない。  
このような状況では、通信ドライバのデフォルトのデータタイプが使用されます。

## 取得サイクル;シユトクサイクル

HMI デバイスが外部タグのプロセス値を読取るタイミングは、取得サイクルによって決まります。通常、タグがプロセス画面に表示されるか、またはロギングされる限り、この値は定期的に更新されます。取得サイクルによって定期的な更新の間隔が設定されます。デフォルトの取得サイクルを選択するか、またはユーザー固有のサイクルを定義することができます。

外部タグは、またプロセス画面の表示に依存することなく、たとえば、タグ機能用の値の変更をトリガすることにより、更新することができます。頻繁に読み込みを行うと、通信負荷が増大することに注意してください。

## 線形スケーリング

数値のデータタイプの線形スケーリングを、設定できます。PLC の外部タグ用データを WinCC flexible プロジェクトの特定の値の範囲にマッピングすることができます。

例: コントローラはインチを前提としてますが、ユーザーは、長さの値をセンチメートルで入力します。入力された値はコントローラに渡される前に、自動的に変換されます。線形スケーリングを使用して、PLC 上の値の範囲[0 ..0.100]を HMI デバイス上の値の範囲 [0 ..0.254]にマッピングできます。

### 4.3.3 タグコンフィグレーションの変更

#### はじめに

タグはプロジェクト要件の変更に合わせて、いつでも変更できます。

#### 原理

WinCC flexible では、タグコンフィグレーションの変更方法が複数あります。

- タグエディタ;タグエディタ

複数のタグを表形式で参照する必要がある場合は、タグエディタを使用してタグを設定します。タグエディタでは、複数のタグのプロパティを比較して調整したり、プロパティでタグをソートすることが可能です。

- オブジェクトリスト;オブジェクトリスト

タグが使用されている場所で直接タグを修正する場合は、オブジェクトリストを使用します。この場合、オブジェクトリストで変更するタグの横にある[...]ボタンをクリックします。次に、[プロパティ]ウィンドウでタグを設定できます。

タグのプロパティを変更したことで、別のプロパティとの矛盾が生じた場合は、そのタグが色付きで強調表示されます。これは、このタグのタイプをサポートしていない異なる PLC にタグを接続した場合などに起こります。

### 4.3.4 タグの限界値

#### はじめに

数値タグの値の範囲を定義できます。

#### 原理

数値タグの上限および下限値の範囲を使用して、値の範囲を指定できます。

タグのプロセス値が限界範囲に含まれる場合、警告などのアナログアラームを送信できます。

プロセス値が値の範囲を超える場合、これをアナログアラームメッセージまたはファンクションリストをトリガするように設定できます。設定された値の範囲外の値をタグに入力すると、入力が拒否され、この値は入力されません。

---

#### 注記

限界値を超えた場合に送信されるアナログアラームメッセージは、アナログアラームエディタを使用して変更できます。

---

#### 適用例;テキヨウレイ

タグの値が危険な範囲に入った場合に、限界値を使用してオペレータへの警告などを行います。

### 4.3.5 タグの初期値

#### プロジェクトの開始時におけるタグの値

数値タグについて、初期値を設定できます。タグは、ランタイムの起動時にこの値に事前設定されます。このようにして、プロジェクトを定義した状態で開始できます。

外部タグの場合、初期値は PLC またはオペレータ入力によって書き込まれるまで、HMI デバイス上に表示されます。

#### 適用例

IO フィールドをデフォルト値に事前設定できます。IO フィールドにリンクされているタグの初期値に目的のデフォルト値を入力します。

### 4.3.6 ランタイム時のタグ値の更新

#### はじめに

タグにはランタイム時に変化するデータが含まれています。値の変化は、内部タグと外部タグにおいて異なる方法で処理されます。

#### 原理

タグの初期値を設定すると、タグはランタイムの起動時にこの値に設定されます。タグの値はランタイム時に変化します。

ランタイム時にタグの値を変更するオプションは、以下のとおりです。

- "SetValue"など、システムファンクションの実行による
- IO ボックスに対してなど、オペレータの入力による
- スクリプト内の値の割り付け
- PLC の外部タグでの値の変更

#### 外部タグの値の更新

##### 外部タグの値の更新方法

- 取得サイクル後の更新

通常、タグが画像に表示またはロギングされている限り、タグは取得サイクル後に更新されます。取得サイクルにより、HMI 上のタグ値更新用の更新サイクルが決定されます。デフォルトの取得サイクルを選択するか、またはユーザー固有のサイクルを定義することができます。

- [周期的連続]設定が有効な場合

この設定が有効な場合、現在開いている画面にタグが表示されていなくても、タグはランタイム時に更新されます。このファンクションは、値の変更イベント時にファンクションリストをトリガするように設定されているタグなどに対して設定されます。

本当に更新する必要があるタグにだけ[周期的更新]設定を使用します。頻繁に読取り操作を行うと通信の負荷が増加します。

### 4.3.7 データロギング

#### はじめに

ランタイム中にタグ値をログに保存し、後で評価できます。タグのロギング用に値が保存されるログを指定して、その頻度および特定の値範囲にあるタグ値のみを保存するのかどうかを指定する必要があります。

---

#### 注記

データロギングの主な目的は、外部タグの値をロギングすることです。ただし、内部タグの値もロギング可能です。

---

#### 原理

データロギングにはいくつかのステップがあります。

- データログの作成および設定

データログの作成時には、以下を定義する必要があります。

- 名前、サイズ、保存先などの全般設定
- ランタイム開始時の動作
- ログが満杯になったときの動作

- タグのロギングの設定

すべてのタグにデータログを指定することができます。このログは、ランタイム中のタグの値を記録し、ログされた時間や値などその他の情報も記録します。

さらに、タグの値をいつ、どの程度の頻度でロギングするかを定義することができます。後者を実行するには、以下のオプションが必要です。

- "要求時"

タグ値は、LogTag システムファンクションを呼び出してログされます。

- "変化時"

オペレータデバイスでタグの値の変化が検出されるとすぐに、タグ値がログされます。

- "連続サイクリック"

タグ値は定期的な間隔でロギングされます。WinCC flexible で使用可能な標準サイクルに加えて、標準サイクルに基づいてユーザー独自のサイクルを追加することができます。

さらに、ロギングの対象を許容範囲内または許容範囲外の値に制限することができます。この方法では、後で行う別の分析用に、異なるログへ個別にタグ値を配信できます。

タグ"要求時"をロギングする場合、タグ"連続サイクリック"または"変化時"がロギングされるシーケンスログにこのタグをロギングすることはできません。要求時のロギングがめったに起こらない場合、シーケンスログは、たとえば周期的にログされる値によって完成されるなどして、次のシーケンスログがセットアップされます。そのあと要求時にロギングされたタグへのアクセスがあると、このタグは現在のシーケンスログがアクセスされるランタイム中に表示できなくなります。対処法として、めったにログされないタグについては別のタグログをセットアップします。

- ログ済みタグ値のそれ以外の処理

ログ済みプロセスタグ値は、たとえば[トレンド]ウィンドウなど WinCC flexible プロジェクトで、または Excel など別のアプリケーションを使用して直接評価できます。

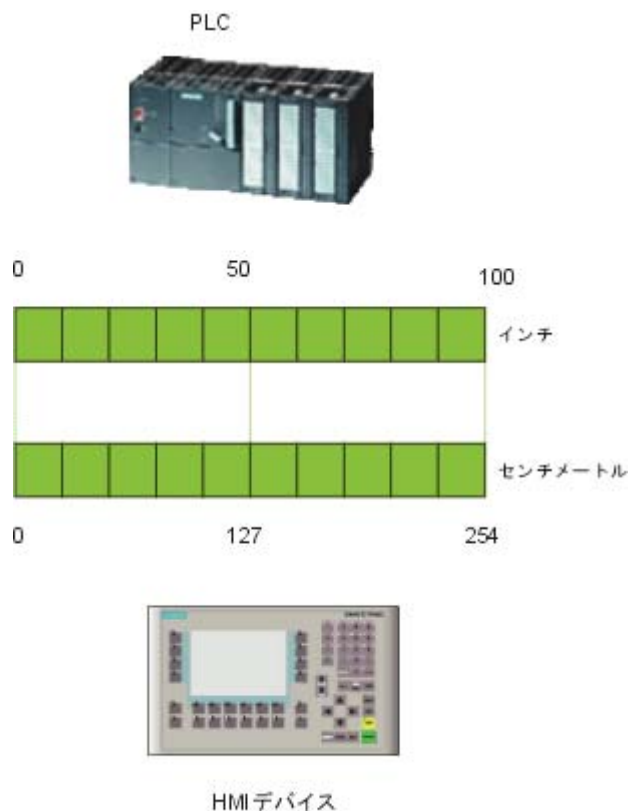
### 4.3.8 タグの線形スケーリング

#### はじめに

数値のデータタイプは線形スケーリングで処理できます。PLC の外部タグ用データを WinCC flexible プロジェクトの特定の値範囲にマッピングすることができます。

#### 原理

タグに線形スケーリングを適用するには、HMI デバイスと PLC にそれぞれ 1 つの値の範囲を指定する必要があります。値の範囲は互いに直線的にマッピングされます。



HMI デバイスからのデータが外部タグに書き込まれると直ちに、PLC の値の範囲に自動的にマッピングされます。HMI デバイスからのデータが外部タグから読み込まれると直ちに、対応する変換が逆方向に実行されます。

#### 注記

システムファンクション LinearScaling および InverseLinearScaling を使って、プロセス値を自動的に変換することもできます。

## 適用例

コントローラーはインチを前提としてますが、ユーザーは、長さの値をセンチメートルで入力します。入力された値はコントローラーに渡される前に、自動的に変換されます。線形スケーリングを使用して、PLC上の値の範囲[0 ..0.100]を HMI デバイス上の値の範囲 [0 ..0.254]にマッピングできます。

### 4.3.9 タグの間接アドレス指定

#### 原理

マルチプレクスでは、間接アドレス指定のタイプは、使用されるタグがランタイムで最初に決まります。タグのリストは、マルチプレクスタグのために定義されます。該当するタグが、ランタイム時にタグのリストから選択されます。タグの選択は、インデックスタグの値によって決まります。

ランタイム時に、システムは最初にインデックスタグの値を読み込みます。次に、タグリストの対応する場所にある指定されたタグにアクセスします。

#### 適用例;テキヨウレイ

間接アドレス指定を使用すると、以下の設定が可能です。

オペレータが選択リストからいずれかのマシンを選択します。オペレータの選択に従って、選択したマシンのデータが出力フィールドに表示されます。

このようなシナリオをコンフィグレーションするために、シンボルの IO フィールド用のインデックスタグをコンフィグレーションします。IO フィールドのマルチプレクスタグをコンフィグレーションします。選択したリストの構造を反映するには、マルチプレクスタグのタグリストを設定します。

オペレータが別のマシンを選択すると、インデックスタグの値が変わります。この場合、新しいインデックス値によって示されるタグリスト(マルチプレクスタグ)にあるタグの内容が選択フィールドに表示されます。

## 4.4 配列の基本

### 定義

同一データタイプの配列データを連続的に配列して、アドレス空間内でアドレスを指定すれば、インデックスを使用してこれらのデータにアクセスできます。"0"で始まる整数インデックスを使用して、配列エレメントのアドレスを指定します。配列タグでプロパティを設定して、同一プロパティを配列エレメントに指定します。

Sy...	Name	Info
[-]	▼ ArrayTag	DB 1 DBD 0
[-]	ArrayTag[0]	DB 1 DBD 0
[-]	ArrayTag[1]	DB 1 DBD 4
[-]	ArrayTag[2]	DB 1 DBD 8
[-]	ArrayTag[3]	DB 1 DBD 12
[-]	ArrayTag[4]	DB 1 DBD 16
[-]	ArrayTag[5]	DB 1 DBD 20
[-]	ArrayTag[6]	DB 1 DBD 24
[-]	ArrayTag[7]	DB 1 DBD 28
[-]	ArrayTag[8]	DB 1 DBD 32
[-]	ArrayTag[9]	DB 1 DBD 36
[-]	ArrayTag[10]	DB 1 DBD 40
[-]	ArrayTag[11]	DB 1 DBD 44
[-]	ArrayTag[12]	DB 1 DBD 48
[-]	ArrayTag[13]	DB 1 DBD 52
[-]	ArrayTag[14]	DB 1 DBD 56
[-]	ArrayTag[15]	DB 1 DBD 60
[-]	ArrayTag[16]	DB 1 DBD 64
[-]	ArrayTag[17]	DB 1 DBD 68
[-]	ArrayTag[18]	DB 1 DBD 72
[-]	ArrayTag[19]	DB 1 DBD 76
[-]	ArrayTag[20]	DB 1 DBD 80
[-]	ArrayTag[21]	DB 1 DBD 84
[-]	ArrayTag[22]	DB 1 DBD 88
[-]	ArrayTag[23]	DB 1 DBD 92
[-]	ArrayTag[24]	DB 1 DBD 96
[-]	ArrayTag[25]	DB 1 DBD 100
[-]	ArrayTag[26]	DB 1 DBD 104
[-]	ArrayTag[27]	DB 1 DBD 108
[-]	ArrayTag[28]	DB 1 DBD 112
[-]	ArrayTag[29]	DB 1 DBD 116
[-]	ArrayTag[30]	DB 1 DBD 120
[-]	ArrayTag[31]	DB 1 DBD 124
[-]	ArrayTag[32]	DB 1 DBD 128
[-]	ArrayTag[33]	DB 1 DBD 132
[-]	ArrayTag[34]	DB 1 DBD 136
[-]	ArrayTag[35]	DB 1 DBD 140
[-]	ArrayTag[36]	DB 1 DBD 144
[-]	ArrayTag[37]	DB 1 DBD 148
[-]	ArrayTag[38]	DB 1 DBD 152
[-]	ArrayTag[39]	DB 1 DBD 156
[-]	ArrayTag[40]	DB 1 DBD 160
[-]	ArrayTag[41]	DB 1 DBD 164
[-]	ArrayTag[42]	DB 1 DBD 168
[-]	ArrayTag[43]	DB 1 DBD 172
[-]	ArrayTag[44]	DB 1 DBD 176
[-]	ArrayTag[45]	DB 1 DBD 180
[-]	ArrayTag[46]	DB 1 DBD 184
[-]	ArrayTag[47]	DB 1 DBD 188
[-]	ArrayTag[48]	DB 1 DBD 192
[-]	ArrayTag[49]	DB 1 DBD 196
[-]	ArrayTag[50]	DB 1 DBD 200
[-]	ArrayTag[51]	DB 1 DBD 204
[-]	ArrayTag[52]	DB 1 DBD 208
[-]	ArrayTag[53]	DB 1 DBD 212
[-]	ArrayTag[54]	DB 1 DBD 216
[-]	ArrayTag[55]	DB 1 DBD 220
[-]	ArrayTag[56]	DB 1 DBD 224
[-]	ArrayTag[57]	DB 1 DBD 228
[-]	ArrayTag[58]	DB 1 DBD 232
[-]	ArrayTag[59]	DB 1 DBD 236
[-]	ArrayTag[60]	DB 1 DBD 240
[-]	ArrayTag[61]	DB 1 DBD 244
[-]	ArrayTag[62]	DB 1 DBD 248
[-]	ArrayTag[63]	DB 1 DBD 252
[-]	ArrayTag[64]	DB 1 DBD 256
[-]	ArrayTag[65]	DB 1 DBD 260
[-]	ArrayTag[66]	DB 1 DBD 264
[-]	ArrayTag[67]	DB 1 DBD 268
[-]	ArrayTag[68]	DB 1 DBD 272
[-]	ArrayTag[69]	DB 1 DBD 276
[-]	ArrayTag[70]	DB 1 DBD 280
[-]	ArrayTag[71]	DB 1 DBD 284
[-]	ArrayTag[72]	DB 1 DBD 288
[-]	ArrayTag[73]	DB 1 DBD 292
[-]	ArrayTag[74]	DB 1 DBD 296
[-]	ArrayTag[75]	DB 1 DBD 300
[-]	ArrayTag[76]	DB 1 DBD 304
[-]	ArrayTag[77]	DB 1 DBD 308
[-]	ArrayTag[78]	DB 1 DBD 312
[-]	ArrayTag[79]	DB 1 DBD 316
[-]	ArrayTag[80]	DB 1 DBD 320
[-]	ArrayTag[81]	DB 1 DBD 324
[-]	ArrayTag[82]	DB 1 DBD 328
[-]	ArrayTag[83]	DB 1 DBD 332
[-]	ArrayTag[84]	DB 1 DBD 336
[-]	ArrayTag[85]	DB 1 DBD 340
[-]	ArrayTag[86]	DB 1 DBD 344
[-]	ArrayTag[87]	DB 1 DBD 348
[-]	ArrayTag[88]	DB 1 DBD 352
[-]	ArrayTag[89]	DB 1 DBD 356
[-]	ArrayTag[90]	DB 1 DBD 360
[-]	ArrayTag[91]	DB 1 DBD 364
[-]	ArrayTag[92]	DB 1 DBD 368
[-]	ArrayTag[93]	DB 1 DBD 372
[-]	ArrayTag[94]	DB 1 DBD 376
[-]	ArrayTag[95]	DB 1 DBD 380
[-]	ArrayTag[96]	DB 1 DBD 384
[-]	ArrayTag[97]	DB 1 DBD 388
[-]	ArrayTag[98]	DB 1 DBD 392
[-]	ArrayTag[99]	DB 1 DBD 396

[オブジェクト]ウィンドウ

### 利点

同一プロパティの複数の配列エレメントは、単一配列タグでアドレス指定できます。次に、設定の他のタグに各配列エレメントを使用できます。特定の限界があります。

- 配列タグに対応していない HMI デバイスもあります。

### 適用例


配列タグは以下の状況で使用できます。

- プロファイルトレンドでプロセス値をグループ化するには、様々な時点で取得したプロセス値をトレンドにマッピングします。
- トレンドでグループ化した特定値へアクセスするには、インデックスタグを設定するなどして、すべてのプロファイルトレンド値を出力します。
- 連続ビット数を使用してディスクリートアラームを設定します。
- マシンデータレコードを単一タグに保存します。

### ランタイムのライセンスルール

配列エレメント数に関係なく、WinCC flexible Runtime で配列を 1 つのパワータグとして数えます。

## 特殊機能

 <b>警告</b>
<b>システム負荷の増加と性能の劣化</b> 読取りか書込みのために、単一の配列エレメントにアクセスする場合でも、該当する配列タグのすべての配列エレメントへアクセスすることになります。大容量配列のデータを PLC とやりとりする場合は、基本データタイプに比べて長時間かかります。これによって通信に過負荷がかかり、結果として損傷する恐れがあります。


### 例:

- データタイプ"Real"の配列エレメント 100 個を構成する配列タグを設定したとします。
- 4 バイトの長さを持つ配列エレメント 1 つを変更する場合、100×4 の容量が PLC に書き込まれます。

### スクリプトでの使用

性能に関する同様の理由により、配列の変更は常に、内部一時配列を使用してください。

1. スクリプト開始時に PLC 配列を内部配列にコピーします。
2. スクリプトで内部配列を処理する間、PLC へのデータ転送でかかる負荷を解消できます。

 <b>注意</b>
<b>配列タグのデータ矛盾</b> 配列エレメント 1 つを変更する場合に、時間 t1 で配列全体が読み取られます。修正された配列エレメントが配列で置換されます。t1 から t3 の時間でその配列が PLC に再び書込みされます。時間 t2 内で配列が再び変更された場合、時間 t3 に時間 t2 で変更された値は時間 t1 の値で上書きされます。そのため時間 t3 の配列データに矛盾が生じます。  HMI デバイスと PLC から同一配列タグへ同時に値を書き込みしないように常に注意してください。PLC でレシピデータレコードの同期転送を使用して、配列タグを同期します。

## 4.5 配列の例

### はじめに

配列タグで複数タグをグループ化して、100 個の配列エレメントを含む構造などの、データ構造を形成します。配列エレメントは、設定のどこでも使用できるタグです。他にも配列タグは以下の状況で使用できます。

- [アラーム]エディタの場合
- [レシピ]エディタの場合
- アドレスの多重化用
- [トレンド]ウィンドウの場合

### 例

配列タグを相当する数の配列エレメントで設定して、同一データタイプを持つ複数のタグを処理でき、エンジニアリング時間が短縮されます。

1. 多重化インデックスタグなどで配列エレメントに間接的にアクセスできます。
2. これらのインデックスタグを使用して、配列エレメントの操作とモニタできます。
3. 配列タグを使用して、複数のレシピエレメントを作成すれば、対応する配列エレメントと自動的に相互接続されます。
4. レシピを使用して PLC と配列タグを同期します。

## 4.6 サイクルの基本

### はじめに

サイクルは定期的に行われるプロジェクトシーケンスを制御するために使用します。共通の用途は、取得サイクル、ロギングサイクル、および更新サイクルです。WinCC flexible で事前定義されたサイクルの他に、独自のサイクルを定義することもできます。

### 原理

ランタイム時に定期的に行われるアクションは、サイクルによって制御されます。サイクルの一般的な用途には、外部タグの取得、データのロギング、および画面の更新などがあります。

- 取得サイクル

HMI デバイスが PLC から外部タグのプロセス値を読み取るタイミングは、取得サイクルによって決まります。取得サイクルは、プロセス値の変更度合いに合わせて設定します。たとえば、オープンの温度は電気ドライブの速度よりはるかにゆっくりと変化します。

取得サイクルの設定が低すぎると、処理の通信負荷が必要以上に高くなります。

- ロギングサイクル

データがログデータベースに保存されるタイミングは、ロギングサイクルによって決まります。ロギングサイクルは、必ず取得サイクルの整数倍になります。

- 更新サイクル

画面が更新される頻度は、更新サイクルによって決まります。

サイクルで可能な最小値は、プロジェクトで使用される HMI デバイスによって決まります。HMI デバイスのほとんどで、この値は 100 ms です。他のすべてのサイクルの値は必ず、最小値の整数倍になります。

WinCC flexible で事前定義された標準サイクルがプロジェクトの必要条件を満たさない場合、独自のサイクルを定義することができます。ただし、こうしたカスタムサイクルは標準サイクルを基準とする必要があります。

### 適用例

定期的に処理を記録する場合や保守間隔を通知する場合などにサイクルを使用します。

## 4.7 構造の操作

### 4.7.1 基本構造

#### はじめに

構造によって、1個の論理ユニットを形成する多数のさまざまなタグを束ねます。構造はプロジェクトグローバルデータであり、プロジェクト全体のすべての HMI デバイ스에適用されます。

#### 原理

たとえば、多様なバルブの状態は 4 個のタグで再現できます。



名前	接続	データ型*	配列助数
Valve_activated	<内部タグ>	Byte	1
Valve_closed	<内部タグ>	Byte	1
Valve_defect	<内部タグ>	Byte	1
Valve_opened	<内部タグ>	Byte	1

構造により、さまざまな状態を 1 つにできます。構造では、各タグに対して構造エレメントを設定します。構造エレメントには、プロパティを割り当てます。このプロパティは、そこから作成されるタグが含むべきもの、データタイプなどです。



名前	データ型*	配列助数	オフセット
opened	Int	1	0
defect	Int	1	2
activated	Int	1	4
closed	Int	1	6

生成した構造には、タグや銘板のダイナミックプロパティを、データタイプとして割り当てます。選択できる構造が、標準データタイプに合わせて[データタイプ]リストボックスに表示されます。

#### 注記

構造や構造データタイプのタグは、銘板だけで操作可能です。

## ランタイムでのライセンス規制

構造データタイプの外部タグを WinCC flexible ランタイムで銘板インスタンスに使用する場  
合、構造エレメントはそれぞれ 1 個のタグとしてカウントされます。

### 例

[画面エディタ]で 2 個の画面を作成しました：画面 1 と画面 2

3 つの銘板インスタンスが画面 1 に挿入され、4 つの銘板インスタンスが画面 2 に挿入され  
ます。個々の銘板インスタンスは、構造データタイプの 1 個の外部タグとリンクしています。  
構造には 10 個の構造エレメントがあります。

画面 1: 3 個の銘板インスタンス × 10 個の構造エレメントは、30 個の外部タグ = 30 個のパ  
ワータグに相当します。

画面 2: 4 個の銘板インスタンス × 10 個の構造エレメントは、40 個の外部タグ = 40 個のパ  
ワータグに相当します。

WinCC flexible ランタイムには、両画面で 70 個のパワータグがあります。これは必要でな  
い構造エレメントにも適用されます。

## 4.7.2 構造エディタ

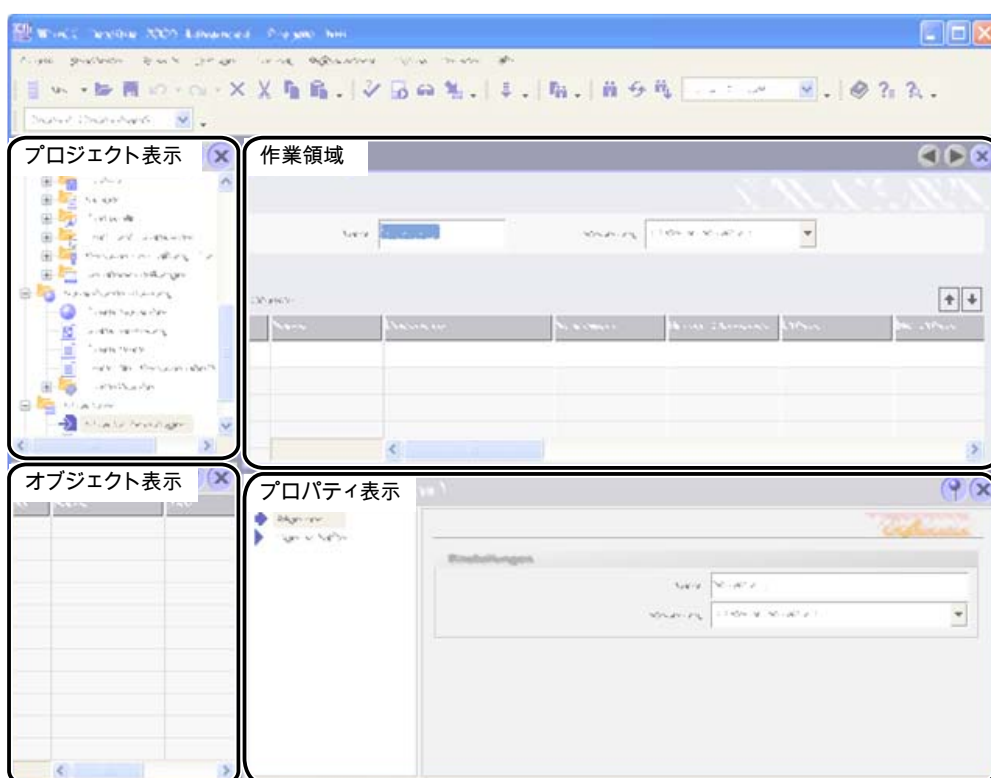
### はじめに

[構造エディタ]で、構造と構造エレメントを作成します。構造はプロジェクトグローバルデータであり、プロジェクト全体のすべての HMI デバイスに適用されます。

### 開く

[プロジェクト]ウィンドウで、[構造]グループの[構造の追加]をダブルクリックします。[プロジェクト]ウィンドウに新しい構造が作成され、作業エリアが新しい構造で開きます。

### 構造



### 作業エリア

構造エレメントは、作業エリアにテーブル形式で表示されます。テーブルセルを使用して、構造エレメントのプロパティを編集します。テーブルを列のエントリでソートするには、単純にその列ヘッダーをクリックします。

テーブルの列を表示または非表示にできます。これを実行するには、テーブルヘッダーのポップアップメニューで、エントリを有効または無効にします。

### [プロパティ]ウィンドウ

[プロパティ]ウィンドウで、構造エレメントをコンフィグレーションします。[プロパティ]ウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。

### 4.7.3 構造管理

#### 構造の名前の変更

1. プロジェクトウィンドウの"構造"で、名前を変更する構造を選択します。
2. ショートカットメニューから[名前の変更]を選択します。
3. 新規の名前を入力します。
4. <ENTER>キーを押します。

オプションとして、<F2>キーを押しても構造の名前を変更できます。

#### 構造のコピーと挿入

1. プロジェクトウィンドウの"構造"で、コピーする構造を選択します。
2. ツールバーの[コピー]アイコンをクリックします。
3. ツールバー上の[貼り付け]アイコンをクリックします。

オプションとして、[編集]メニューまたはショートカットメニューの[コピー]と[貼り付け]のコマンドを使うことができます。これらのファンクションは、<CTRL+C>および<CTRL+V>のキー操作でも実行することができます。

オリジナル構造のコピーはプロジェクトウィンドウで生成され、作業エリアに挿入されます。構造の名前には、連続する番号が割り振られます。

#### 構造の削除

構造を削除するときは、その構造の構造エレメントも一緒に削除されます。

1. プロジェクトウィンドウの"構造"で、削除する構造を選択します。
2. ツールバーの[削除]アイコンをクリックします。削除する構造が銘板で使用中の場合、ダイアログが表示されます。そのダイアログは、その構造を削除するかを確認します。

オプションとして、[編集]メニューまたはポップアップメニューの[削除]コマンドを使うことができます。<DEL>キーを使用することもできます。

構造を削除するときは、その構造の構造エレメントも一緒に削除されます。エントリは割り当てられたタグに残っています。[タグエディタ]で、背景に色が付いていることは、削除によってデータに矛盾が生じたことを表しています。[タグエディタ]でデータタイプを変更します。

#### 構造エレメントの名前の変更

1. [プロジェクト]ウィンドウの[構造]で必要な構造を選択します。  
[構造エディタ]にその構造が開きます。
  2. 名前を変更する構造エレメントを選択します。
  3. ショートカットメニューから[名前の変更]コマンドを選択します。
  4. 新規の名前を入力します。
  5. <Enter>キーを押します。
- または<F2>ファンクションキーを押して構造の名前を変更することもできます。

## 構造エレメントの編集

1. [プロジェクト]ウィンドウの[構造]エリアで必要な構造をダブルクリックします。  
[構造]エディタでその構造が開きます。
2. 編集する構造エレメントを強調表示します。
3. データタイプなど、構造エレメントのプロパティを変更します。

構造タグのプロパティを変更する場合、構造エレメントのプロパティを必ず変更する必要があります。

## 構造エレメントのコピーと挿入

1. [プロジェクト]ウィンドウの[構造]エリアで必要な構造をダブルクリックします。  
[構造]エディタでその構造が開きます。
2. コピーする構造エレメントを強調表示します。
3. [コピー]ツールバーアイコンをクリックします。
4. [貼り付け]ツールバーアイコンをクリックします。

ショートカットメニューから[コピー]や[貼り付け]コマンドを使用することもできます。キーストロークの<CTRL+C>と<CTRL+V>も利用できます。

オリジナルの構造エレメントのコピーを、作業エリアに挿入します。構造エレメントの名前には、連続する番号が割り振られます。

## 構造エレメントの削除

1. [プロジェクト]ウィンドウの[構造]エリアで必要な構造をダブルクリックします。  
[構造]エディタでその構造が開きます。
2. 削除する構造エレメントを強調表示します。
3. ツールバー上の[削除]アイコンをクリックします。

ショートカットメニューや[編集]から[削除]コマンドも使用できます。<DEL>キーも使用できます。

## 4.8 タグのインポート

### 4.8.1 タグのインポートとエクスポート

#### はじめに

WinCC flexible には、プロジェクトからタグをエクスポートして、このデータを別のプロジェクトにインポートするオプションがあります。このインポートとエクスポートの機能は、作業負担を軽減します。新しいタグを作成する代わりに、以前のプロジェクトで作成したタグ、または PLC プログラムからエクスポートしたタグを使用します。

#### タグのエクスポート

エクスポート時に、タグ情報が CSV ファイルに書き込まれます。タグ名、タグタイプに加えて、タグのこれ以外のプロパティもエクスポートされます。

#### タグのインポート

タグデータを完全にインポートするには、ファイルが 2 つ必要です。このファイルの 1 つに、コントロールとのインターフェースの情報が含まれます。これ以外のファイルには、タグからの情報が含まれます。

最初に、インターフェースデータを含むファイルをインポートします。次に 2 番目のファイルをインポートするとき、データタイプと PLC メモリアドレスが同時に確認されます。インポートを開始する前に、同じ名前での既存の接続やタグを上書きするかどうか指定できます。

タグデータは、インターフェースデータを含むファイルなしで、インポートできます。以下の例に、インポート全体を示します。

#### 外部データソース用に CSV ファイルを準備

PLC プログラムのタグデータの準備に、アプリケーションを使用できます。Tag Converter という 1 つのアプリケーションが、製品 DVD の <CD\_3\サポート\Tag Converter> から使用できます。

このアプリケーションの最新バージョンは、インターネットの次のアドレスからダウンロードできます。ダウンロードエリアへ

Tag Converter を使用して、以下の通信ドライバからデータを変換します。

- Allen-Bradley
- GE FANUC
- Modicon MODBUS
- STEP 7
- Telemecanique

## 4.8.2 タグのエクスポートとインポートのための設定

### はじめに

タグのインポートには[CSV インポート]ダイアログが使用できます。インポートファイルを適切に解釈できるように、このダイアログで必要な設定を実行します。

#### 注記

WinCC flexible 2008 では、CSV ファイルのインポートのメニュー構造が変更されています。

### HMI デバイスへのタグのインポート

[CSV インポート]ダイアログを開くには、[プロジェクト]ウィンドウから必要な HMI デバイスを選択して、ショートカットメニューコマンド[CSV のインポート]を選択します。別の方法としては、HMI デバイスを選択して、メニューコマンド[プロジェクト]>[インポート/エクスポート]>[CSV インポート]を選択します。



[ファイルのインポート]列でインポートファイルの場所を入力するか、保存先に移動して必要なファイルを選択します。この各インポートファイルは、\*.csv フォーマットになっている必要があります。インポートが開始される前に、ファイル名を確認します。

[上書き]チェックボックスを有効にしている場合、以前の接続またはタグが同じ名前の場合、インポート中に上書きされます。[既存の接続/タグの上書き]チェックボックスを無効にしている場合、WinCC flexible 内の名前が同じ各接続とタグはインポートされません。

[オプション]列の[...]ボタンをクリックして、インポート設定を定義します。新しいダイアログボックスが開きます。



[リスト区切り記号]フィールドでは、各接続とタグの個々のパラメータを区切るのに使用する文字を選択します。詳細については、「Format of the Connection Data for the Import (インポートに応じた接続データのフォーマット)」と「Format of the Tag Data for the Import (インポートに応じたタグデータのフォーマット)」を参照してください。

テキスト修飾子を使用して、テキストと文字列を識別します。引用符で囲まれた各文字は、テキストと見なされます。たとえば、インポート用コントロール文字として使用する文字が指定されたテキストをインポートする場合、これらの文字を引用符で囲む必要があります。引用符をテキスト修飾子として使用します。別の文字は使用できません。

小数点や桁区切り記号を指定すれば、数値データを識別できます。各フィールドに表示される文字のリストから選択します。これらの区切り記号に引用符は使用できません。

#### 注記

リスト、小数点、桁区切りには、異なる区切り記号を使用します。これらの区切り記号は同じであってははいけません。

[フォルダ区切り記号の使用]オプションを使用すれば、タグ名でフォルダツリーを作成できます。フォルダツリーは WinCC flexible で作成され、タグはこのフォルダに保存されます。[フォルダ区切り記号]フィールドで、フォルダツリーの区切り記号を選択します。

例:

タグ名が、"Folder1\Tag\_01"とします。フォルダ区切り記号は"\"です。WinCC flexible では、[プロジェクト]ウィンドウの[通信/タグ]下で[Folder1]が作成され、このフォルダに[Tag\_01]が保存されます。

### 4.8.3 接続データのフォーマット

#### はじめに

このセクションでは、インポートに使用する接続データが格納されたファイルに必要なフォーマットについて説明します。この接続データファイルは、\*.csv フォーマットになっている必要があります。

#### 接続データのフォーマット

各接続は、インポートファイルの個別の行に指定されます。接続データを格納するインポートファイルのフォーマットは、次のようになっている必要があります。

<接続名><リスト区切り子>

<通信ドライバ名><リスト区切り子>

<コメント><改行(行頭復帰+改行)>

#### エントリの意味

リストエントリ	説明
[接続名]	接続の設定名を指定します。タグのインポートファイルで、対応するエントリと一致させるのにこのエントリが必要です。[名前]のリストエントリを空にすることはできません。名前には、アポストロフィ(')を指定できません。
リスト区切り子	リスト区切り子は、リスト内の個々のエントリを区切ります。インポート用ダイアログで、リスト区切り文字に使用する文字を選択できます。次の各文字を使用できます。"タブ"、"セミコロン";、"コンマ";、"ピリオド"。必要なら、選択フィールドを使用して、表示されている文字以外の文字を選択します。
通信ドライバの名前。	WinCC flexible で使用する通信ドライバの名前を指定します。WinCC flexible で使用する名前と正確に一致している必要があります。次の各名前を使用できます。 <ul style="list-style-type: none"> <li>• Allen Bradley DF1</li> <li>• Allen Bradley DH485</li> <li>• Allen Bradley E/IP C.Logix</li> <li>• GE Fanuc SNP</li> <li>• LG GLOFA-GM</li> <li>• Mitsubishi FX</li> <li>• Mitsubishi protocol 4</li> <li>• Modicon MODBUS</li> <li>• Modicon Modbus TCP/IP</li> <li>• Omron Hostlink/Multilink</li> <li>• OPC</li> <li>• SIMATIC S5 AS511</li> <li>• SIMATIC S5 DP</li> <li>• SIMATIC S7200</li> <li>• SIMATIC S7300/400</li> <li>• SIMATIC 500/505 serial</li> <li>• SIMATIC 500/505 DP</li> <li>• SIMOTION</li> <li>• SIMATIC HMI HTTP プロトコル</li> <li>• Telemecanique Uni-Telway</li> </ul>

リストエントリ	説明
コメント	接続に関するコメントです。コメントで使えるのは、最大で 256 文字までです。
改行	改行(行頭復帰+改行)は、1つの接続のエントリと、次の接続のエントリとを区切ります。

### 接続用インポートファイルのフォーマット

接続用インポートファイルのフォーマットは、次のとおりです。

connection, "SIMATIC S7 300/400", connection example

たとえば、コンマは、リスト区切り子として使用できます。リストエントリが空の場合、リスト区切り子が2つ連続します。行でエントリの定義を終了したとき、行の終わりにリスト区切り子を入力する必要はありません。

#### 注記

インポートファイルの一例が、WinCC flexible DVD の「CD\_3\Support\Tag Import」フォルダにあります。

### インポートファイルの編集

インポートファイルを、MS Excel またはテキストエディタで編集できます。

MS Excel を起動して、[ファイル]メニューから[開く]を選択します。[ファイルの種類]リストから、"テキストファイル(\*.prn、\*.txt、\*.csv)"のタイプを選択します。インポートファイルを、MS Excel でダブルクリックして開かないでください。そのデータ構造が壊れて、インポートできなくなります。

簡易テキストエディタでインポートファイルを開いて、データ構造が正しいか確認します。

#### 4.8.4 タグデータのフォーマット

##### はじめに

このセクションでは、インポートに使用するタグデータが格納されたファイルに必要なフォーマットについて説明します。このタグデータファイルは、\*.csv フォーマットになっている必要があります。

##### タグデータのフォーマット

各タグは、インポートファイルの個別の行に指定されます。タグデータを格納するインポートファイルのフォーマットは、次のようになっている必要があります。

<タグ名><リスト区切り子>  
 <接続名><リスト区切り子>  
 <タグアドレス><リスト区切り子>  
 <データタイプ><リスト区切り子>  
 <タグの長さ(バイト)><リスト区切り子>  
 <列数><リスト区切り子>  
 <取得タイプ><リスト区切り子>  
 <取得サイクル><リスト区切り子>  
 <上限><リスト区切り子>  
 <付加上限><リスト区切り子>  
 <付加下限><リスト区切り子>  
 <下限><リスト区切り子>  
 <線形スケーリング><リスト区切り子>  
 <スケーリング上限 PLC><リスト区切り子>  
 <スケーリング下限 PLC><リスト区切り子>  
 <スケーリング上限 HMI デバイス><リスト区切り子>  
 <スケーリング下限 HMI デバイス><リスト区切り子>  
 <初期値><リスト区切り子>  
 <ID 更新><リスト区切り子>  
 <コメント><改行(行頭復帰+改行)>

##### エントリの意味

リストエントリ	説明
[タグ名]	タグの設定名を指定します。フォルダ区切り文字を使用して設定したフォルダツリーを指定した後にタグ名を指定できます。たとえば、Foldername1\Foldername2\tagname と指定できます。[フォルダ区切り記号の使用]チェックボックスを有効にしている場合、WinCC flexible へのインポートを実行すると、フォルダツリーが作成されます。[名前]のリストエントリを空にすることはできません。名前には、アポストロフィ(')を指定できません。

## タグの操作

### 4.8 タグのインポート

リストエントリ	説明
リスト区切り子	リスト区切り子は、リスト内の個々のエントリを区切ります。リスト区切り子を、インポートダイアログボックスで選択できます。次の各文字を使用できます。"タブ"、"セミコロン";、"コンマ";、"ピリオド"。必要に応じて選択フィールドを使用し、表示されている文字以外の文字を選択します。
接続名	接続の設定名を指定します。接続のインポートファイルで対応するエントリと一致させるのにこのエントリが必要です。各外部タグには、接続名に有効なエントリを指定する必要があります。接続に名前を指定しないと、内部タグが作成されます。
[タグアドレス]	PLCでタグアドレスを指定します。タグアドレスは、WinCC flexible で使用するタグアドレスと正確に一致する必要があります。たとえば、"DB 1 DBW 0"とし、"DB1, DBW0"とはしません。内部タグでは、このタグアドレスは空になっています。
データタイプ	タグのデータタイプを指定します。使用している通信ドライバによって、使用できるデータタイプが違ってきます。次に、可能な値の例を挙げます。Char、Byte、Int、UInt、Long、ULong、Float、Double、Bool、String、DateTime、Word、Dint、DWord、Real、StringChar、Timer、Counter、Date、Date and time、Time of day、ASCII、+/-DEC、DEC、LDEC、+/-LDEC、IEEE、BIN、4/8/12/16/20/24/28/32ビットブロック、+/- Double、+/- int、16ビットグループ、short、KF、KH、KM、KY、KG、KS、KC、KT、Bit in D、Bit in W、DF、DH、IEEE-Float、USInt、SInt、UDInt、Dint、time、BCD4、BCD8など。種々の通信ドライバに使用できるデータタイプの詳細については、マニュアルの「通信」セクションを参照してください。
[タグの長さ(単位: バイト)]	タグの長さ(単位: バイト)を指定します。このエントリは、通常、文字列タグにだけ使用されます。他のデータタイプの場合、このエントリは空です。
[配列番号]	タグの配列エレメントの番号を定義します。この値を使用すれば配列を定義できます。このエントリが空になっている場合、この値は"1"に設定されます。
[取得トリガモード]	タグの取得トリガモードを指定します。取得トリガモードは、番号で示されます。 1=オンデマンド 2=サイクリック、使用される場合(デフォルト値) 3=連続サイクリック
取得サイクル	タグの取得サイクルを指定します。取得サイクルは、WinCC flexible の取得サイクルと正確に一致している必要があります。値は言語とは無関係で、どの言語でも同じであることが必要です。初期値は1秒です。タグの取得トリガモードが"オンデマンド"になっている場合、取得サイクルは定義されていません。ユーザー定義の取得サイクルを、インポートするファイルに対して、あらかじめ作成しておく必要があります。
上限値 低上限値 高下限値 下限値	これらの各限界値は、数値にだけ設定できます。限界値には、定数値だけ使用できます。変数は使用できません。限界値のデフォルト値は"限界なし"です。各限界値には、次の条件が適用されます。 "上限" ≥ "低上限" ≥ "高下限" ≥ "下限"
線形スケールリング	線形スケールリングを有効にするかどうか指定します。このエントリは、外部タグにだけ設定できます。デフォルト設定は[無効化]です。 線形スケールリングの値は、数値またはテキスト形式で指定できます。許容値には、次のものがあります。 [無効化]の場合には"false"または"0" [有効化]の場合には"true"または"1"

リストエントリ	説明
PLC スケーリング上限値 PLC スケーリング下限値 HMI スケーリング上限値 HMI スケーリング下限値	この各上下限界値は、数値にだけ設定できます。
初期値	タグの開始値を指定します。デフォルト値には、次のものがあります。数値に対して 0、文字に対して空白、時刻と日付に対して実際値。
更新 ID	ジョブ番号は、ファンクションまたはジョブメールボックスの助けを受けて、タグの値を更新します。更新 ID は、HMI デバイスの中で一意であることが必要です。
コメント	ユーザー固有のタグコメント。最大 500 文字まで使用できます。
改行	改行(行頭復帰+改行)は、タグエントリを区切ります。

### タグインポートファイルのフォーマット

タグ用インポートファイルのフォーマットは、次のとおりです。

タグの"tag","Connection","DB 1 DBD 0","Real",,1,3,"1 min",20,10,2,1,1,100,10,10,1,15.5,33,  
コメント

たとえば、コンマは、リスト区切り子として使用できます。リストエントリが空の場合、リスト区切り子が 2 つ連続します。行でエントリの定義を終了したとき、行の終わりにリスト区切り子を入力する必要はありません。値が指定されていないリストエントリでは、デフォルト値が使用されます。

#### 注記

インポートファイルの一例が、WinCC flexible DVD の「CD\_3\Support\Tag Import」フォルダにあります。

### インポートファイルの編集

インポートファイルを、MS Excel またはテキストエディタで編集できます。

MS Excel を起動して、[ファイル]メニューから[開く]を選択します。[ファイルの種類]リストから、"テキストファイル(\*.prn、\*.txt、\*.csv)"のタイプを選択します。インポートファイルを、MS Excel でダブルクリックして開かないでください。そのデータ構造が壊れて、インポートできなくなります。

簡易テキストエディタでインポートファイルを開いて、データ構造が正しいか確認します。

### アドレス多重化を使用したタグの CSV インポート

アドレス多重化を使用してタグを CSV インポートするには、以下の規則に従ってください。

- まず参照タグをインポートし、次にアドレス多重化を使用したタグをインポートします。したがって、CSV ファイルデータは、参照タグで始まります。
- CSV エクスポートファイルには、タグが正しい順序で含まれています。
- CSV インポートするには、参照タグおよびアドレス多重化を使用したタグが、同じフォルダにある必要があります。

### 制限事項

小数を含むタグデータをエクスポートすると、Excel でエクスポートファイルを開いたときに値が正しく表示されないことがあります。エクスポートダイアログで、点"."を小数点として使用し、タグの小数点後に 1 から 12 の間の値があるときは、Excel では、この値が日付入力と解釈されます。このようなエラーを修正または回避するには、他の小数点記号を使用するか、小数点の後ろのタグ値にゼロを追加します。

例:

データタイプ"Float"のタグの上限を 20.5 に設定し、エクスポートダイアログで、小数点記号として点を選択しました。タグ値"20.5"が、タグデータにエクスポートされます。Excel は、CSV ファイルを開いたときに、この値を 5 月 20 日の日付入力と解釈します。

対策:

たとえば、コンマを小数点記号として設定します。この例では、タグ値を 20,50 に拡張することもできます。

## 画面の作成

### 5.1 基本

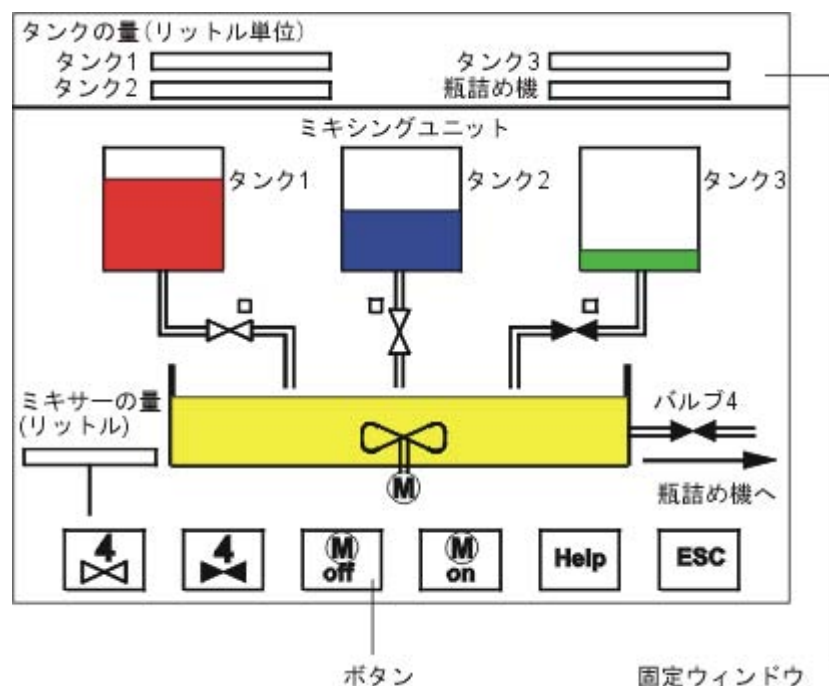
#### 5.1.1 画面の基礎

##### はじめに

WinCC flexible では、オペレータが機器やプラントを制御したりモニタするのに使う画面を作成します。独自の画面を作成するときには、プロセスのビジュアルイズやプラントのイメージ作成、およびプロセス値の定義に便利なオブジェクトテンプレートがあります。

##### 適用例

次の図で WinCC flexible で作成した画面を示します。この画面を使用して、オペレータはフルーツジュース製造システムのみキシングユニットを操作したりモニタしたりすることができます。フルーツジュースのベースは、種々のタンクからみキシングユニットへ供給されます。画面には、タンクやみキサーの充填レベルが示されます。また、画面にはバルブユニットやみキサーモーター用のコントロールエレメントも表示されます。



## 画面デザイン

プロセスを表示するために必要なオブジェクトを画面に挿入します。プロセスの必要条件に対応するオブジェクトをコンフィグレーションします。

画面は、スタティックエレメントとダイナミックエレメントで構成することができます。

- テキストやグラフィックオブジェクトといったスタティックエレメントは、ランタイム時にもそのステータスは変化しません。このミキシングユニットの例では、タンクラベルがスタティックエレメントになります。
- 一方、ダイナミックエレメントはプロセスに基づいてステータスが変わります。表示される現在のプロセス値は次のようになります。
  - PLC のメモリから
  - HMI デバイスのメモリから、英数字の組み合わせで表示されるトレンドとバー。  
HMI デバイスの入力フィールドもダイナミックオブジェクトと見なされます。このミキシングプラントの例では、タンクの充填レベル値もダイナミックオブジェクトです。

プロセス値とオペレータ入力は、タグを経由してコントローラと HMI デバイスの間で交換されます。

## 画面プロパティ

画面レイアウトは、作成する HMI デバイスの特性によって決まります。つまり、デバイスのユーザーインターフェースのレイアウトに対応します。設定されている HMI デバイスにファンクションキーがある場合、例として、画面ではこのようなファンクションキーが表示されます。画面解像度、フォントや色などその他のプロパティも、選択した HMI の特性に応じて決まります。

## ファンクションキー

ファンクションキーとは HMI デバイス上のキーです。WinCC flexible では 1 つまたは複数のファンクションを割り付けることができます。これらのファンクションは、オペレータが HMI デバイスの該当するキーを押した時点でトリガされます。

ファンクションキーには、グローバルファンクションまたはローカルファンクションを割り付けることができます。

- グローバルファンクションキーは、現在表示されている画面に関係なく同じアクションを常にトリガします。
- ローカル割り付けの行われているファンクションキーは、オペレータステーションで現在表示されている画面に応じて、異なるアクションをトリガします。この割り付けは、ファンクションキーを定義した画面にのみ適用されます。

## ナビゲーション;ナビゲーション

オペレータが Runtime で画面を呼び出せるようにするために、構成済みの画面をそれぞれオペレーティングプロセスに統合する必要があります。これらのファンクションの構成には次のようなさまざまなオプションがあります。

- "画面ナビゲーション"エディタを使用して、画面構造を定義し、画面ナビゲーションシステム全体を設定します。
- "画面"エディタを使用して、ほかの画面を呼び出すボタンやファンクションキーを作成します。

## 5.1.2 画面の HMI ベース依存性

### はじめに

HMI デバイスのファンクションによって、WinCC flexible におけるプロジェクトの表示とエディタの機能範囲が決定されます。

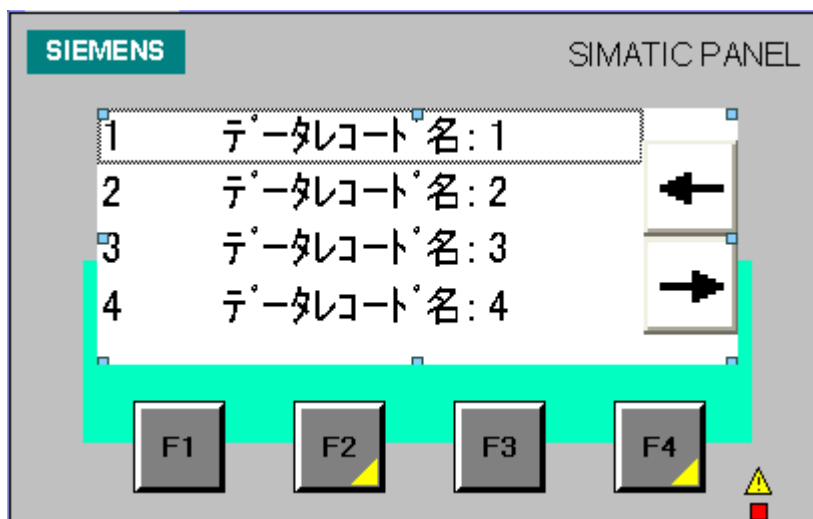
プロジェクトの作成時に、プロジェクトに対応した HMI デバイスを選択します。[プロジェクト]ウィンドウで、HMI デバイスのタイプを変更したり、新しいデバイスを追加したりすることができます。

以下の画面のプロパティは、選択した HMI デバイスのファンクションによって決定されます。

- レイアウト
- 画面の解像度
- カラー深度
- フォント
- 使用可能なオブジェクト

### デバイスレイアウト

設定における画面のデバイスレイアウトによって、HMI デバイスのイメージが形成されます。たとえば、画面のデバイスレイアウトには、HMI デバイス上で使用可能なファンクションキーがすべて表示されます。



### 画面の解像度

画面の解像度は、さまざまなオペレータパネルの表示特性によって決定します。画面の解像度は、"WinCC flexible RT"が PC 用に作成されていた場合にだけ影響を受けます。

## カラー深度

画面オブジェクトに色を割り付けることができます。色の範囲は、選択した HMI デバイス上でサポートされているカラー深度によって決定します。

## フォント

スタティックテキストまたはダイナミックテキストを含むすべての画面オブジェクトで、テキストの表示をカスタマイズすることができます。たとえば、画面上のテキストの優先順位を識別することができます。フォント、フォントスタイル、およびフォントサイズを選択して、アンダースコアなどの補助的効果を選択することができます。

選択した HMI デバイスによって、使用可能なフォントが決定します。選択したフォントによって、使用可能なフォントプロパティが決定します。



たとえば、テキストフォーマット、フォントスタイル(ボールド、イタリックなど)および効果(ストライクアウト、アンダースコア)は、画面オブジェクト上のすべてのテキストに適用されます。つまり、タイトルのすべての文字をボールドフォーマットで表示させることはできますが、文字や単語ごとにフォントスタイルを変えることはできません。

## 使用可能なオブジェクト

画面オブジェクトの中にはすべての HMI デバイス用に一括して設定できないものがあります。このような画面オブジェクトは、[ツールボックス]ウィンドウ内ではグレーで表示され、選択することができません。たとえば、TP 170 タッチパネルユニットの場合は、任意のボタンを設定することができません。

### 5.1.3 "画面"エディタ

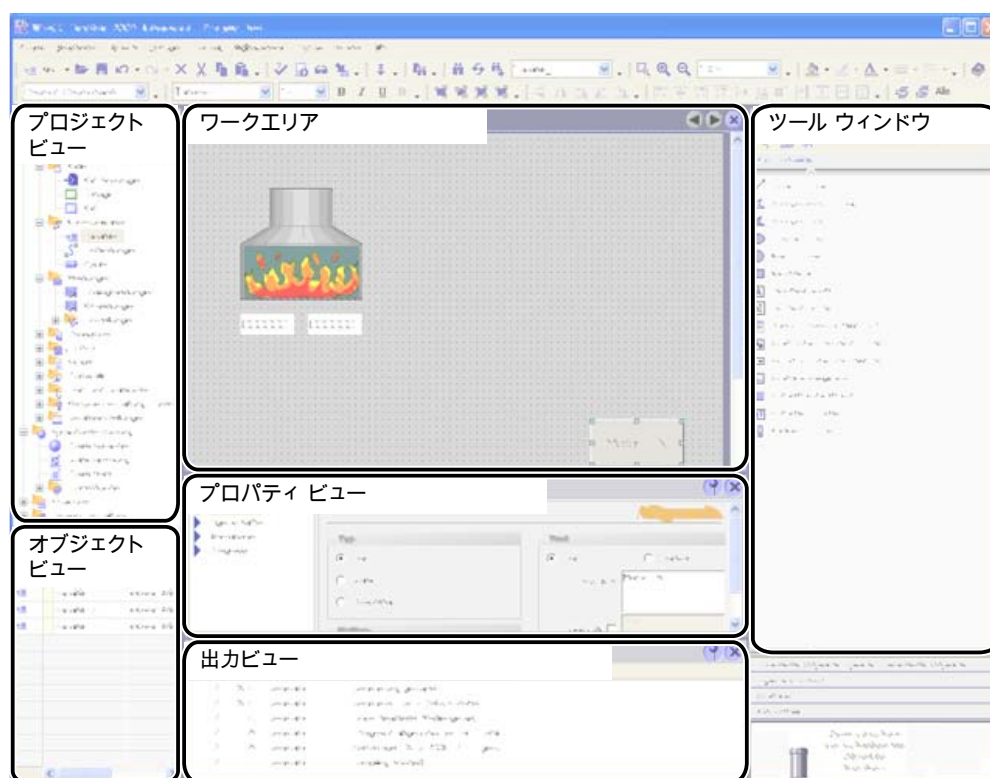
#### はじめに

"画面"エディタで画面をコンフィグレーションします。このエディタは、グラフィックプログラミングソフトウェアとプロセス可視化ツールを組み合わせたものです。"画面"エディタは、[プロジェクト]ウィンドウから開きます。

#### 開く;ヒラク

[プロジェクト]ウィンドウで、[画面]グループの[画面の追加]をダブルクリックします。作業エリアが開き、新規画面が1つ開きます。

#### レイアウト



#### メニューバー

メニューバーには、WinCC flexible の操作に必要なすべてのコマンドがあります。使用可能なショートカットキーはすべてメニューコマンドの隣に表示されます。

#### ツールバー

特定のツールバーを表示したり隠したりすることができます。

## 作業エリア

作業エリアでは、画面のコンフィグレーションを行います。

## ツールボックス

ツールボックスには、グラフィックオブジェクトやオペレーティングエレメントなど、画面に追加できる単純なオブジェクトと複雑なオブジェクトの選択肢が表示されています。さらに、ツールボックスは、オブジェクトテンプレートとフェイスプレートの集合を含むライブラリを提供しています。

## [プロパティ]ウィンドウ

[プロパティ]ウィンドウの内容は、作業エリアで現在選択されているオブジェクトによって決まります。

- 選択したオブジェクトのプロパティは、[プロパティ]ダイアログボックスで表示したり編集したりすることができます。
- アクティブな画面でオブジェクトを選択していない場合、この画面のプロパティが表示されるため、[プロパティ]ウィンドウでこのプロパティを編集できます。

### 5.1.4 手順;テジュン

#### 手順;テジュン

画面を作成するには、以下の初期処理を行っておく必要があります。

- プロセス表示の構成の概略を作成する、つまり画面構成や画面の数を定義します。  
例: 各プロセス区分を別個の画面に表示して、マスター画面で結合できます。
- 画面ナビゲーションコントロールの方針を決めます。
- テンプレートを適用します。

このテンプレートは、選択した HMI デバイス用の WinCC flexible に保存されており、すべてのプロジェクト画面に適用されます。このテンプレートでは、オブジェクトをローカルに定義してグローバルファンクションキーを割り付けることができます。HMI デバイスには、全画面に統合したいオブジェクトを 1 つの固定ウィンドウに配置できるものもあります。

- 画面を作成します。以下のオプションを使用して、効率よく画面を作成します。
  - "画面ナビゲーション"エディタで画面構造を作成します。
  - ライブラリの取り扱い
  - 銘板の取り扱い
  - レイヤーの取り扱い

## 5.2 ナビゲーションシステムの作成

### 5.2.1 ナビゲーションオプション

#### はじめに

複数の画面で構成された WinCC flexible プロジェクトの場合は、ランタイム時に以下の画面ナビゲーションオプションが提供されます。

- ナビゲーションボタンを使用するナビゲーション
- ファンクションキーを使用するナビゲーション
- ナビゲーションコントロールを使用するナビゲーション

WinCC flexible では、以下のプログラミングオプションがあります。

- プログラミングボタンまたはファンクションキーを使用する方法
- "画面ナビゲーション"エディタとナビゲーションコントロールを使用してグラフィック設定を実行する方法

---

#### 注記

画面のプロジェクトウィンドウで[表示]アニメーションを[非表示]に設定すると、この画面をランタイム時に呼び出せません。

---

## 5.2.2 画面ナビゲーションシステムのグラフィックプログラミング

### "画面ナビゲーション"エディタ

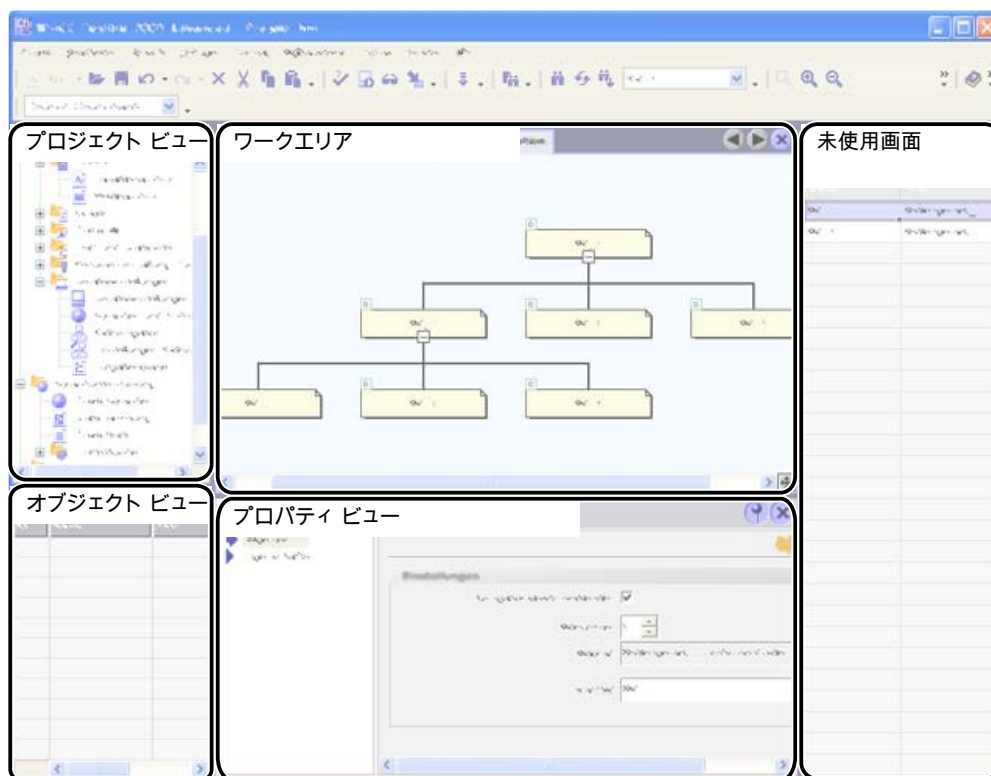
"画面ナビゲーション"エディタは、複数の画面間のナビゲーションをグラフィック設定するのに使用します。このエディタを使用することによって、プロジェクト画面を階層構造に整理することができます。オペレータはランタイム中にナビゲーションコントロールを使用して階層構造のさまざまな画面間のナビゲーションができます。たとえば、親の画面や隣接する画面に切り替えることができます。

これらの構造化した接続だけでなく、"画面ナビゲーション"エディタを使用すれば、所定の構造に関係なく、直接画面接続を生成することもできます。

### 開く

"画面ナビゲーション"エディタを開くには、プロジェクトウィンドウの[デバイス設定]で[画面ナビゲーション]をダブルクリックします。

### レイアウト



### メニューバー

メニューバーには、WinCC flexible の操作に必要なすべてのコマンドがあります。使用可能なショートカットキーはすべてメニューコマンドの隣に表示されます。

## 作業エリア

"画面ナビゲーション"エディタの表示は、画面構造を表しています。各種の画面が四角形で示されています。

画面の相互接続は、ランタイム中のナビゲーションオプションに対応します。各種の相互接続のタイプは色の付いたラインで識別されます。


- 黒のラインは、構造化された画面の相互接続を表します。
- 緑の矢印は、構造とは関係なくダイレクトな画面接続を表します。

## ショートカットメニュー

状況に応じたメニューには、"画面ナビゲーション"エディタ、および画面の作成、画面を開く、画面の削除、画面のコピー、または画面の名前変更をコンフィグレーションするのに使用できる各コマンドが表示されています。

## 表示レイアウトのカスタマイズ

"画面ナビゲーション"エディタの表示をカスタマイズするオプションは数種類あります。

- 表示を拡大または縮小すれば、"画面ナビゲーション"エディタの拡大セクションまたは縮小セクションを表示できます。
-  アイコンを使用してこのセクションを移動し、ウィンドウの他の部分を表示させることができます。
- 単一の画面、およびその画面のすべての子画面を表示することができます。
- すべての子画面を非表示または表示することができます。
- 横に並べて表示および縦に並べて表示を切り替えることができます。

## [未使用画面]

[未使用画面]ウィンドウには、ナビゲーションシステムに含まれていないすべてのプロジェクト画面が含まれています。このダイアログボックスからウィンドウ上に[未使用画面]をドラッグアンドドロップし、他の画面と相互接続させることができます。

## [プロパティ]ウィンドウ

画面の[プロパティ]ダイアログボックスを使用すると、ナビゲーションコントロールを有効化したり、画面番号を変更したり、ダイレクトな画面接続を作成することができます。

### 5.2.3 ナビゲーションコントロールの使用

#### ナビゲーションコントロールの使用

すべての画面のナビゲーションコントロールを開くことができます。ナビゲーションコントロールには、事前作成された画面ナビゲーションボタンが数多く含まれています。これらのボタンは、ランタイム中にプロジェクトのさらに詳細な画面を呼び出すのに用いられます。

#### ナビゲーションコントロールのカスタマイズ

"画面ナビゲーション"エディタで、ナビゲーションコントロールを適応させます。

- ナビゲーションコントロールを表示または非表示にすることができます。
- ナビゲーションコントロールとその中に含まれるコマンドボタンを設定することができます。

ターゲットが割り付けられていない場合、ボタンは無効になります。この場合、ランタイム中このボタンはラベルなしで表示されます。

[プロジェクトウィンドウ]に対して無効なナビゲーションバーで、画面ナビゲーションで使用する画面をコピーし、別の HMI に挿入するとき、コピーした画面は既存の画面ナビゲーションの HMI には表示されません。

---

#### 注記

##### HMI デバイスを変更する際のナビゲーションボタン

HMI デバイスを変更した後、プロジェクトを再コンパイルします。コンパイルによって、全画面が確実に現在の HMI デバイスに変換されます。

その後、最初の生成を行わないで再び HMI デバイスを変更した場合、ナビゲーションボタンが個別の画面で違った仕方で表示される可能性があります。

---

## 5.3 オブジェクトの取り扱い

### 5.3.1 オブジェクトの概要

#### はじめに

オブジェクトは、プロジェクトのプロセスグラフィックをデザインする際に使用するグラフィックエレメントです。

[ツールボックス]には、プロセス画面で頻繁に使用されるオブジェクトのさまざまなタイプが含まれています。

[ツールボックス]ウィンドウは、[表示]メニューの[ツールボックス]コマンドを使用して、フェードインおよびフェードさせることができます。また、[ツールボックス]ウィンドウは、画面上のどこにでも移動できます。

[ツールボックス]には、現在アクティブになっているエディタによって異なる、さまざまなオブジェクトグループが含まれています。"画面"エディタを開くと、ツールボックスに以下のオブジェクトグループのオブジェクトが提供されます。

- [単純なオブジェクト]

単純なオブジェクトとは、"ライン"や"円"などのグラフィックオブジェクト、および"I/Oフィールド"や"ボタン"などの標準オペレータコントロールエレメントです。

- [拡張オブジェクト]

これは、機能が強化されたオブジェクトです。拡張オブジェクトの目的の1つは、Sm@rtClient ウィンドウのように、プロジェクトにおける統合バーや Active X コントロールなどのプロセスをダイナミックに表示することです。

- [ユーザー固有のコントロール]

このオブジェクトグループでは、使用している PG / PC の Windows オペレーティングシステムに登録されている ActiveX コントロールをツールボックスに追加して、プロジェクトに統合することができます。

- [グラフィック]

たとえば、機械とプラントの構成要素、測定装置、コントロールエレメント、フラグ、および建屋のグラフィックオブジェクトが、ディレクトリツリー構造にテーマ別に表示されます。グラフィックファイルへのショートカットを作成することもできます。このフォルダーとネストされたフォルダにある外部グラフィックオブジェクトは、ツールボックスウィンドウに表示され、プロジェクトに統合されます。









- [ライブラリ]







[ライブラリ]には、パイプ、ポンプまたはデフォルトボタンのグラフィックなどのオブジェクトテンプレートが含まれています。再設定しなくても、プロジェクトにライブラリオブジェクトのマルチプルインスタンスを統合することができます。

WinCC flexible ソフトウェアパッケージには、ライブラリが含まれています。ユーザー定義のオブジェクトおよび銘板は、ユーザーライブラリに保存することもできます。












[銘板]は、事前に設定したオブジェクトグループを表します。これらのプロパティのいくつかは(すべてではない)、そのアプリケーションの該当する場所でコンフィギュレーション可能です。銘板は、中央ロケーションから編集できます。銘板を使用すると、コンフィギュレーションにかかる作業を軽減し、統一されたプロジェクトデザインを実現することができます。



## 単純なオブジェクト

シンボル	オブジェクト	注記
	[ライン]	終端が直線、円形または矢印の形のラインを選択できます。
	[多角折線]	[多角折線]はリンクされたパスで構成され、角の数は自由に設定できます。コーナーポイントは、その作成順に数えられます。また、コーナーポイントは、個々に修正、削除することができます。終端が直線、円形または矢印の多角折線を選択できます。[多角折線]は開いたオブジェクトです。このため、開始ポイントと終了ポイントが同一座標であっても、囲まれているエリアを塗りつぶすことはできません。
	[多角形]	[多角形]のコーナーポイントは、その作成順に数えられます。また、コーナーポイントは、個々に修正、削除することができます。多角形のエリアは、色またはパターンで塗りつぶすことができます。
	[楕円]	[楕円]は、色またはパターンで塗りつぶすことができます。
	[円]	[円]は、色またはパターンで塗りつぶすことができます。
	[四角形]	[四角形]の角は丸くできます。[四角形]は、色またはパターンで塗りつぶすことができます。
A	[テキストボックス]	1行または複数行のテキスト行を[テキストボックス]に入力した後、フォントとフォント色を定義できます。背面色またはパターンを[テキストボックス]に追加できます。
	[I/O ボックス]	I/O ボックスには、次のランタイムファンクションを指定できます。 <ul style="list-style-type: none"> <li>• タグへの値の出力</li> <li>• オペレータ入力の値; これらの入力値はタグに保存されます。</li> <li>• 入出力複合: オペレータは、ここでタグの出力値を編集して新しい値を設定できます。</li> </ul> [I/O ボックス]に表示されたタグ値の限界を定義します。 ランタイム中にオペレータ入力を非表示にする場合は、[入力を隠す]を設定します。
	[日付/時刻ボックス]	[日付/時間ボックス]には、次のランタイムファンクションを装備できます。 <ul style="list-style-type: none"> <li>• 日付および時刻の出力</li> <li>• 組み合わせ入力および出力; オペレータは、ここで出力値を編集して日付および時刻をリセットすることができます。</li> </ul> システム時間または対応するタグをソースとして使用し、日付と時刻を定義できます。 たとえば、拡張フォーマットでは 2003 年 12 月 31 日 火曜日、簡易フォーマットでは 12.31.2003 などのように、日付を出力できます。

シンボル	オブジェクト	注記
	[グラフィック IO フィールド]	<p>[グラフィック I/O ボックス]には、次のランタイムファンクションを装備できます。</p> <ul style="list-style-type: none"> <li>グラフィックリストエントリの出力</li> <li>入出力複合: [グラフィック IO フィールド]の内容を変更するために、オペレータはここでグラフィックリストからグラフィックを選択できます。</li> </ul> <p>出力フィールドとしての使用例 バルブのランタイムステータスを示すために、[グラフィック IO ボックス]は、閉じたバルブまたは開いたバルブのイメージを出力します。</p>
	[シンボル IO フィールド]	<p>[シンボル I/O ボックス]には、次のランタイムファンクションを装備できます。</p> <ul style="list-style-type: none"> <li>テキストリストエントリの出力</li> <li>入出力複合: [シンボリック IO ボックス]の内容を変更するために、オペレータはここでテキストリストからテキストを選択できます。</li> </ul> <p>複合 I/O ボックスとしての使用例 ランタイム中にモーターを制御するには、オペレータはテキストリストから[モーターOFF]または[モーターON]のテキストを選択します。このモーターは、開始または停止状態のどちらかで選択され、[シンボルの IO フィールド]には、モーターの現在のステータスが示されます(モーター OFF / モーター ON)。</p>
	[グラフィック表示]	<p>[グラフィック表示]では、外部のグラフィックプログラミングツールで作成されたグラフィックオブジェクトがすべて1つの画面に表示されます。グラフィックオブジェクトは、次のフォーマットで表示できます。 ".emf"、".wmf"、".dib"、".bmp"、".jpg"、".jpeg"、".gif"、".tif"</p> <p>たとえば、[グラフィック表示]では、他のグラフィックプログラミングツールのグラフィックオブジェクトを OLE(オブジェクトのリンクと埋め込み)オブジェクトとして統合することもできます。[グラフィック表示]の [プロパティ]ウィンドウで OLE を直接作成するときに使用したグラフィックプログラムで開き編集した OLE オブジェクト。</p>
	[ボタン]	オペレータは[ボタン]を使用して、プロセスを制御します。[ボタン]のファンクションまたはスクリプトを設定できます。
	[切り替え]	<p>ランタイム中に[スイッチ]を使用して、たとえば ON と OFF、あるいは押しているか押していないか、などの2つの状態を入力してビジュアルライズします。</p> <p>[スイッチ]のランタイムステータスを示すテキストまたはグラフィックを使用してラベル付けが可能です。</p>
	[バー]	[バー]は、スケールリングした棒グラフの形式でプロセス値を表しています。棒グラフを使用すれば、占有レベルのダイナミック値などを表示できます。

## 拡張オブジェクト

シンボル	オブジェクト	説明
	[スライダ]	[スライダ]を使用して、オペレータ入力および数値のモニタリングを行います。 <ul style="list-style-type: none"> <li>表示手段として使用する場合、スライダ位置はコントロールにより出力されるプロセス値を示します。</li> <li>この場合、オペレータはスライダ位置を変更して値を入力します。</li> </ul> スライダをカスタマイズして、縦方向にのみ操作できるようにすることができます。
	[時計]	HMI デバイスでは、デジタルまたはアナログフォーマットのどちらかでランタイム中に時計を表示することができます。
	[ステータスの強制]	Status/Control ファンクションにより、接続された SIMATIC S7 または SIMATIC S5 CPU の特定アドレスエリアに対する直接読み取り/書き込みアクセスが可能になります。
	[Sm@rtClient]ウィンドウ	オペレータは、[Sm@rtClient]ウィンドウを使用してリモートオペレータステーションをモニタおよび操作できます。
	[HTML ブラウザ]	オペレータは、[HTML ブラウザ]を使用して、HTML フォーマットでページを表示できます。
	[ユーザー]ウィンドウ	WinCC flexible では、パスワードを使用して画面オブジェクトへのアクセスを制御できます。 [ユーザー]ウィンドウで、管理者は Runtime で HMI デバイス上のユーザーを管理できます。[ユーザー]ウィンドウでは、管理者権限のないユーザーでもランタイム中に自分のパスワードを変更できます。
	[ゲージ]	[ゲージ]ダイアルで、ランタイム中に数値を表示できます。 [ゲージ]のレイアウトはコンフィグレーション可能です。たとえば、背面イメージやダイアルのレイアウトをカスタマイズできます。
	[トレンド]ウィンドウ	[トレンド]ウィンドウでは、PLC またはログから読取るプロセス値を表すトレンドのグループを表示できます。トレンド座標(つまりスケールリング、単位など)は、設定可能です。
	[レシピ]ウィンドウ	オペレータはランタイム中に[レシピ]ウィンドウを使用して、データレコードを表示、編集、管理します。
	[アラーム]ウィンドウ	[アラーム]ウィンドウでは、オペレータは、ランタイム中に、アラームバッファまたはアラームログで選択したアラームやアラームイベントを表示できます。
	[アラーム]ウィンドウ	[アラーム]ウィンドウでは、オペレータはランタイム中に、アラームバッファまたはアラームログで選択したアラームやアラームイベントを表示できます。 いつでも、テンプレートを編集して[アラーム]ウィンドウを設定できます。

シンボル	オブジェクト	説明
	[アラーム]インジケータ	"アラームインジケータ"は、まだ確認されていないアラームイベントをオペレータに通知します。 テンプレートを編集して、いつでも[アラーム]インジケータを作成できます。
	ヘルプインジケータ	ヘルプインジケータは、表示されている画面または画面上にあるオブジェクトについて利用できるヘルプテキストを示しています。ヘルプインジケータは HMI デバイス OP 73 と OP 73micro で使用できます。 テンプレートを編集して、いつでもヘルプインジケータを作成できます。

### 注記

ツールボックスオブジェクトの中には、使用できる機能に制限があるものと、全く使用できないものがあります。これは作成する HMI デバイスによります。オブジェクトで使用できないプロパティは無効なものとして表示され、選択できません。

## 5.3.2 オブジェクトの編集オプション

### はじめに

オブジェクトは、プロジェクトの画面をデザインする際に使用するグラフィックエレメントです。

オブジェクトを編集するには、以下のオプションを使用できます：

- オブジェクトの切り取り、コピー、挿入、および削除

実行するには、[編集]メニューで以下のコマンドを使用します。

- 切り取り
- コピー
- 挿入
- 削除

画面内のオブジェクトをコピーする際に、同じ名前のオブジェクトがすでに画面にある場合、オブジェクトの名前は変更されます。スクリプトで画面オブジェクトのプロパティにアクセスするためには、画面内のすべてのオブジェクトおよびテンプレートに一意の名前を付ける必要があります。

- 挿入するオブジェクトのデフォルトサイズを維持、あるいはオブジェクトのサイズをカスタマイズします。
- たとえばサイズなど、オブジェクトのプロパティの変更
- オブジェクトの位置付け
- 別のオブジェクトの前または後ろにあるオブジェクトの移動。
- オブジェクトの回転

- オブジェクトのミラーリング
- オブジェクトのデフォルトプロパティの変更
- オブジェクトのタブ順序の定義
- スタンプ: 同じタイプの複数のオブジェクトの挿入
- 複数オブジェクトの同時選択
- 複数のオブジェクトの位置変更とサイズ変更
- Windows オペレーティングシステムに登録されている ActiveX コントロールを[ユーザーコントロール]オブジェクトグループに追加したり、[ユーザーコントロール]オブジェクトグループから削除したりできます。
- 外部グラフィックを、[グラフィック画面]などでオブジェクトに割り当てることができません。

WinCC flexible プロジェクトのイメージブラウザに以前に保存したイメージだけを、表示できます。

イメージブラウザでのグラフィックの保存は以下の方法で行います。

- "グラフィック"オブジェクトグループから作業エリアへのドラッグアンドドロップ
- 次のフォーマットのグラフィックファイルとして: \*.bmp、\*.dib、\*.ico、\*.emf、\*.wmf、\*.gif、\*.tif、\*.jpeg、または\*.jpg。
- OLE オブジェクトとして

OLE オブジェクトを新規に作成するか、既存のグラフィックファイルを OLE オブジェクトとして保存します。OLE オブジェクトを保存するには、OLE 互換グラフィックプログラムを設定コンピュータにインストールする必要があります。

### 5.3.3 複数のオブジェクトの位置変更とサイズ変更

#### 可能な変更

オブジェクトを複数選択したら、次のようにこれらのオブジェクトを編集できます。

- マウスを使った移動
  - マークの付いたオブジェクトの絶対位置を変更するには、マウスポインタをオブジェクトの上に当てて、マウスボタンを押したまま複数選択を移動します。
  - 比率を保ったままスケーリングするには、マウスでのサイズ変更中、<Shift>キーを押したままにします。
- [整列]ツールバーの機能を使った移動
  - マークの付いたオブジェクトの位置を1つずつ変更します。
  - マークの付いたオブジェクトの高さと幅を合わせます。

### 5.3.4 外部グラフィック

#### はじめに

WinCC flexible では、外部グラフィックエディタで作成したイメージを使用できます。こうしたグラフィックを使用するには、これらを WinCC flexible プロジェクトのイメージブラウザに保存します。

イメージブラウザでのグラフィックの保存は以下の方法で行います。

- [グラフィック]オブジェクトグループから作業エリアにグラフィックオブジェクトをドラッグアンドドロップすると、これらのグラフィックオブジェクトは自動的にイメージブラウザに保存されます。グラフィック名には、たとえば Image\_1 のように、その作成順に番号が付けられます。<F2>キーを使用すれば、グラフィックの名前を変更できます。
- 次のフォーマットのグラフィックファイルとして：  
\*.bmp、\*.dib、\*.ico、\*.emf、\*.wmf、\*.gif、\*.tif、\*.jpeg、または\*.jpg。
- WinCC flexible に埋め込まれ、外部グラフィックエディタにリンクされた OLE オブジェクトとして保存されます。OLE リンクでは、外部グラフィックエディタは WinCC flexible から開くことができます。リンクされたオブジェクトはグラフィックエディタで編集できます。OLE ショートカットは、その外部のグラフィックエディタが設定コンピュータにインストールされていて、OLE がサポートされている場合にのみ機能します。

#### イメージブラウザからイメージを使用する

画面のイメージブラウザからイメージを使用するには、以下の方法で実行します。

- グラフィック表示
- イメージリスト
- ファンクションキーラベル

#### 透明なイメージ

WinCC flexible では、透明な背面のグラフィックも使用します。WinCC flexible グラフィックオブジェクトの透明な背面にイメージを挿入する場合、透明は、WinCC flexible グラフィックオブジェクトで定義された背面色で置換されます。選択した背面色はイメージと永続的にリンクされます。他の WinCC flexible グラフィックオブジェクトでそのイメージを使用する場合、それは、最初に設定されたグラフィックオブジェクトと同じ背面色で表示されます。イメージを異なる背面色で使用する場合、このイメージに違う名前を付けてイメージブラウザに追加する必要があります。画像の使用時に、追加の背面色を適切な WinCC flexible グラフィックオブジェクトに割り付けます。

## グラフィックの管理

WinCC flexible と共に、以下のようなグラフィックやシンボルの広範なコレクションもインストールされます。

- マシンおよびプラントのパーツ
- 計測器
- コントロールエレメント
- フラグ
- 建物

各グラフィックオブジェクトは、[グラフィック]オブジェクトグループの[WinCC flexible グラフィックディレクトリ]にテーマ別にファイルされています。WinCC flexible グラフィックフォルダへの参照に対して、削除、編集、名前の変更を実行できません。

さらに[グラフィック]オブジェクトグループを使用して外部グラフィックオブジェクトを管理することもできます。使用可能なフアクション：

- グラフィックフォルダへの参照を作成する  
このディレクトリと親ディレクトリに配置されている外部グラフィックは、ツールボックスウィンドウに表示されるため、プロジェクトに統合されます。
- フォルダリンク
  - 編集
  - 名前の変更
  - 更新
  - 削除
- プログラムを開いて、WinCC flexible で外部グラフィックを編集できます。

### 5.3.5 オブジェクトグループ

#### はじめに

グループとは、"グループ"ファンクションによって複数のオブジェクトを集めたものです。グループも、他のオブジェクトと同じように編集します。

#### 概要

WinCC flexible では、次の 3 つの方法で複数のオブジェクトを同時に編集できます。

- 複数選択
- オブジェクトグループの作成
- 銘板の作成

次の表は、3 つの方法の違いを示しています。

複数のオブジェクトを同時に編集する方法

	複数選択	オブジェクトグループの作成 (グループ)	銘板の作成 (エディタ"銘板")
サイズ変更	✓	✓	✓
グループのスケール	✓	✓	✓
位置の変更	✓	✓	✓
表示の変更	✓	-	特殊なケース
ライブラリでの保存	-	-	✓

#### 編集モード

グループのオブジェクトを個々に編集するには、ショートカットメニューで編集モードを変更します。このモードでは、個々のオブジェクトとそのプロパティにアクセスできます。

#### グループの拡張;グループノックアウト

グループにはさらにオブジェクトやグループを追加できます。すると、グループは新規のオブジェクトによって大きくなります。その後、グループを再び分離すると、このグループにグループとして追加したオブジェクトも含めて、すべてのオブジェクトはグループ化されていない状態になります。オブジェクトをグループ化した順序は考慮されません。

#### オブジェクトを囲む四角形

グループでは、グループ全体としてオブジェクトを囲んでいる 1 つの四角形だけが表示されます。ただし、複数選択の場合は、すべてのオブジェクトを囲んでいる四角形が表示されます。

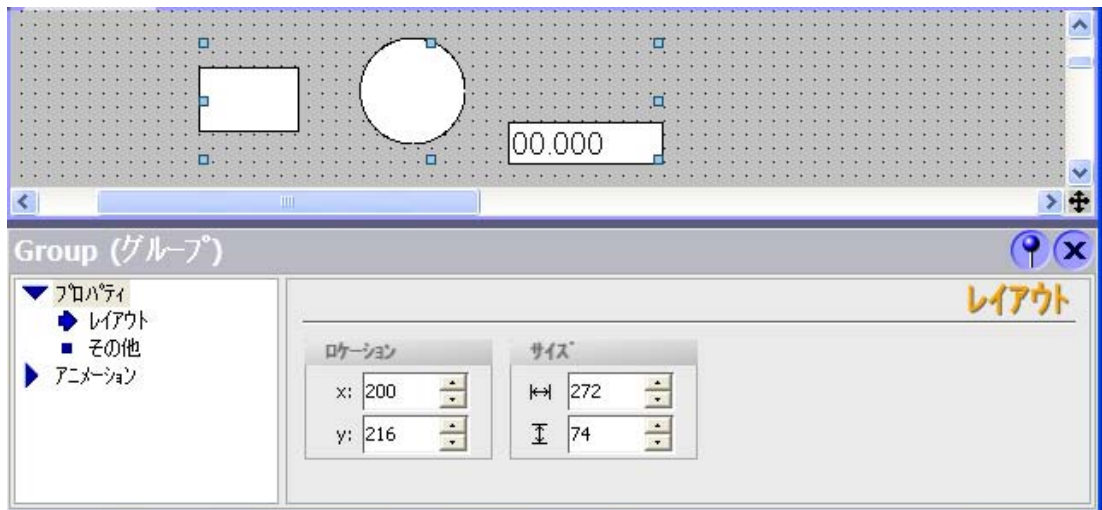
#### レイヤー

グループの全オブジェクトは、同じレイヤーに置かれます。

## 5.3.6 グループプロパティの定義

## はじめに

グループも、他のオブジェクトと同じように編集します。[プロパティ]ダイアログボックスには、選択したオブジェクトのプロパティがすべて表示されます。



共有グループプロパティに変更を加えると、このプロパティを割り付けた単一オブジェクトすべてに影響します。

グループを位置変更およびサイズ変更を行う場合、以下の規則を覚えておいてください。

- グループの位置を変更すると、グループオブジェクトの位置座標が自動的に適合します。グループに対する個々のオブジェクトの相対位置は影響を受けません。
- グループサイズの変更の割合に応じて、グループオブジェクトの高さと幅がシステムにより自動的に適用されます。

## 5.4 ダイナミックな更新ファンクションの割り付けのオプション

### はじめに

入力および出力用のすべてのオブジェクトは、ランタイム中はダイナミックな反応を示します。ダイナミックなプロパティをオブジェクトに割り付けることができます。それぞれのプロセス値に基づいてダイナミックに更新される、タンクの充填レベルのグラフィックなどが、この特徴を示した例です。オブジェクトのダイナミックな反応の別の例として、特定のファンクションを起動するボタンがあります。

### ダイナミックなオブジェクト;ダイナミックなオブジェクト

あらゆるグラフィックオブジェクトに、ダイナミックなプロパティを割り付けることができます。プログラミングのオプションには以下があります。

- オブジェクトの外観を変える。カラーまたは点滅プロパティ
- 画面オブジェクトをアニメーション化する。
- オブジェクトの表示または非表示。

コントロールエレメントには、以下のオプションも使用できます。

- オブジェクトに対するオペレータ制御の有効化またはロック。
- オブジェクトのオペレータ制御(たとえばクリック)が、ファンクションリストを実行するよう作成されているイベントを起動します。

### ダイナミックコントロールとオブジェクトプロパティ

ダイナミックコントロールは、オブジェクトプロパティのエレメントです。実際に使用できるダイナミックなコントロールファンクションとイベントは、選択されたオブジェクトにより異なります。オブジェクトをコピーすると、そのダイナミックなコントロールファンクションもコピーされます。

## 5.5 ファンクションキーの取り扱い

### はじめに

ファンクションキーは HMI デバイスの物理的なキーで、そのファンクションは設定することができます。ファンクションリストは、キーの[押す]イベントおよび[リリース]イベントで設定できます。

ファンクションキーには、グローバルファンクションまたはローカルファンクションを割り付けることができます。

### グローバル割り付け

グローバルファンクションキーは、現在表示されている画面に関係なく同じアクションを常にトリガします。

グローバルファンクションキーは、テンプレートで一度作成します。グローバル割り付けは、このテンプレートに基づいて選択した HMI デバイスのすべての画面に適用されます。

これらのグローバルキーは個々の画面に割り付ける必要がないので、グローバルファンクションキーを使用するとプログラミングでの手間が大幅に減少します。

### ローカル割り付け

画面内のローカルファンクションキーは、各画面で異なるアクションをトリガできます。この割り付けは、ファンクションキーを定義した画面にのみ適用されます。

ローカルファンクションキーを使用して、テンプレートのローカルファンクションキーとグローバルファンクションキーを上書きすることができます。

---

#### 注記

ローカルファンクションキーが表示されている画面の上にアラーム表示または[アラーム]ウィンドウが重なった場合、ランタイム中にファンクションキーは有効のままです。これは特に、画面の小さな HMI デバイスの場合に起こります(OP 77B など)。

---

### ホットキー割り付け






ボタンなどのホットキーを HMI デバイスに割り付けることができます。どのホットキーを使用できるかは HMI デバイスによって異なります。

### グラフィック

ファンクションキーを表示のすぐ近くに置くと、そこにグラフィックを割り付け、ファンクションキーのファンクションをはっきりさせることができます。

## 割り付けの表示

次の表は、ファンクションキーの割り付けを表示するシンボルを示しています。

ファンクションキー	説明
	割り付けなし
	グローバルに使用
	ローカルに使用
	ローカルに使用(ローカル割り付けはグローバル割り付けを上書きする)
	画面ナビゲーションによるボタンの割り付け

## 5.6 レイヤーの利点

### レイヤー

画面のオブジェクトのさまざまに異なる編集をアーカイブするには、レイヤーを使用します。1画面は32のレイヤーで構成されています。オブジェクトをレイヤーに割り付ける場合、それによって画面の深度を定義します。レイヤー0のオブジェクトは画面の背景に位置付けられ、レイヤー31のオブジェクトは前景に位置付けられます。

単一レイヤーのオブジェクトも階層的に割り付けられます。画面を作成する際に、最初に挿入したオブジェクトはレイヤー内の背面部に配置されます。その他の各オブジェクトは、1つ分手前に配置されます。オブジェクトはレイヤー内の手前または後に移動できます。

### レイヤー技術の原理

32のレイヤーのうち、必ず1つのレイヤーがアクティブになります。画面に追加する新規のオブジェクトは、常に動作中のレイヤーに割り付けられます。アクティブなレイヤーの数は、[レイヤー]ツールバーおよび画面の[プロパティ]ウィンドウに示されます。アクティブなレイヤーは、画面の[プロパティ]ウィンドウにカラーで強調表示されます。

画面を開くと、その画面の32個のレイヤーすべてが表示されます。画面の[プロパティ]ウィンドウでは、アクティブなレイヤーを除くすべてのレイヤーを非表示にできます。これにより、アクティブなレイヤーのオブジェクトを明確にして編集できます。

### 適用例

レイヤーは、たとえば次の場合に使用します。

- 編集中にオブジェクトのラベリングを非表示にする場合。
- 画面をさらにコンフィグレーションする際に、アラームウィンドウなどのオブジェクトを非表示にする場合。

## 5.7 オブジェクトライブラリ

### はじめに

ライブラリとは、画面オブジェクトのテンプレートを集めたものです。ライブラリオブジェクトは再構成する必要なく常に再使用可能なため、使用可能な画面オブジェクト蓄積が増えプログラミング効率が向上します。WinCC flexible ソフトウェアパッケージでは、たとえば[モータ]オブジェクトや[バルブ]オブジェクトなどを含む、広範囲なライブラリを提供しています。しかし、ユーザー独自のライブラリオブジェクトも定義できます。

ライブラリは、"画面"エディタの"ライブラリ"ツールボックスウィンドウで、管理されています。ツールボックスでは書込み禁止のライブラリがイタリックのフォントで示されます。書込み禁止のシステムライブラリは常にイタリックのフォントで表示されます。書込み禁止のライブラリは編集できません。

---

### 注記

ライブラリが複数のアプリケーションで開かれている場合は、ライブラリを[バージョンを付けて保存]機能を使って保存できません。

---

## プロジェクトライブラリ

各プロジェクトに対して1つのライブラリがあります。プロジェクトライブラリのオブジェクトは、プロジェクトデータと共に保存され、ライブラリが作成されたプロジェクトでのみ使用可能です。プロジェクトを別のコンピュータに移動する場合は、そこで作成されたプロジェクトライブラリも含まれます。プロジェクトライブラリは、オブジェクトが含まれていないと非表示になります。[ライブラリ]ウィンドウのショートカットメニューで、[プロジェクトライブラリの表示]コマンドを選択するか画面オブジェクトを[ライブラリ]ウィンドウにドラッグして、プロジェクトライブラリを表示します。

## グローバルライブラリ

プロジェクトライブラリのオブジェクトに加えて、プロジェクトの共有ライブラリのオブジェクトも組み込むことができます。グローバルライブラリは、プロジェクトデータから独立して、\*.lib という拡張子のついた別のファイルに保存されます。

プロジェクトでグローバルライブラリを使用すると、関連するプロジェクト内でこのライブラリへの参照を生成するだけです。プロジェクトを別のコンピュータに移動する場合、共有ライブラリは自動的に組み込まれません。プロジェクトと共有ライブラリ間の相互接続は、この移動アクションによって失われる場合があります。グローバルライブラリの名前を異なるプロジェクトや WinCC flexible 以外のアプリケーションで変更した場合も、相互接続が失われます。

プロジェクトは複数のグローバルライブラリにアクセスできます。グローバルライブラリは複数のプロジェクトで、並行して使用することができます。

ライブラリオブジェクトがプロジェクトによって変更されると、このライブラリは他のすべてのプロジェクトでこの修正済みの状態で開きます。

共有ライブラリには、WinCC flexible パッケージで提供されているライブラリもあります。

## カテゴリ

ライブラリオブジェクトを項目ごとにソートするために、ライブラリをカテゴリに分割するか、または複数のグローバルライブラリを作成することができます。たとえば、あるグローバルライブラリには、モータ制御を構成するために必要なオブジェクトがすべて含まれます。別のグローバルライブラリにはパルプ制御を構成するのに必要なオブジェクトがすべて含まれます。

## ライブラリオブジェクト

ライブラリには、たとえば画面、タグ、グラフィックオブジェクトまたはアラームなどの WinCC flexible オブジェクトがすべて含まれています。

ライブラリオブジェクトをプロジェクトで使用するには、そのオブジェクトと参照されているすべてのオブジェクトをプロジェクトにコピーします。コピーされたオブジェクトはライブラリとの相互接続を失っています。ライブラリ内での変更は、コピーされたライブラリオブジェクトには影響しません。

構成可能なオブジェクトグループのマルチプルインスタンスを使用したり、それらを集中的に編集したりするには、フェースプレートを作成する必要があります。

## 5.8 銘板の取り扱い

### 5.8.1 銘板の基礎

#### はじめに

フェイスプレートは、既存の画面オブジェクトからコンパイルするオブジェクトです。

フェイスプレートには次の利点があります：

- 一括変更
- 別のプロジェクトでの再利用
- エンジニアリング時間の削減

フェイスプレートは"フェイスプレート"エディタで作成と編集を行います。作成したフェイスプレートは"プロジェクトライブラリ"に追加され、他のオブジェクトと同じ方法で画面に挿入できます。

#### フェイスプレート設定

"フェイスプレート"エディタは、"画面"エディタと[フェイスプレート設定]ダイアログで構成されています。

#### "画面"エディタ

"画面"エディタでは、フェイスプレートに入れたいオブジェクトを配置します。オブジェクトを削除することもできますし、ツールボックスを使用して新しく追加することもできます。

#### "フェイスプレート設定"ダイアログ

"フェイスプレート設定"ダイアログには、次のエントリがあります：

- 全般

ここでは、フェイスプレートの名前を設定できます。フェイスプレートは"プロジェクトライブラリ"に、この名前が表示されます。

フェイスプレートのアスペクト比の動作を決定します。

フェイスプレートで使用する可能性のあるテキストのエクスポートやインポートには、対応するボタンを使用します。

- プロパティ

ここでは、フェイスプレートのプロパティを設定できます。このプロパティは他のオブジェクトとまったく同じように、以後の設定で設定できます。

フェイスプレートタグも作成できます。フェイスプレートタグは、フェイスプレート内だけで使用可能です。フェイスプレートタグは、フェイスプレートに含まれるオブジェクトやIOフィールドなどへ直接に相互接続されます。

- イベント  
ここでは、フェイスプレートのイベントを設定できます。このイベントは他のプロパティとまったく同じように、以後の設定で設定できます。
- スクリプト  
ここでは、フェイスプレートのスクリプトを設定します。スクリプトで、値を変更するなどのために、システム機能呼び出しや、新しい機能をプログラムできます。スクリプトはフェイスプレートでのみ使用可能です。

## 適用例

たとえば、フェイスプレートを"操作エレメント"と定義することができます。フェイスプレートは、たとえば以下のようなエレメントで構成されています。

- バー
- 名前のテキストフィールド
- 出力モード
- ボタン

フェイスプレートでは、ボタンの"クリック"イベントを、スクリプトに相互接続できます。このスクリプトは、2つのプロセスタグの平均値などを計算します。ダイナミックコントロールは、フェイスプレートに統合されています。フェイスプレートから作成したすべてのフェイスプレートインスタンスが、同じダイナミックコントロールを持つように、フェイスプレートを事前設定できます。

## フェイスプレートの使用

フェイスプレートを作成すると、オブジェクトとしてプロジェクトライブラリに表示されます。

フェイスプレートをプロセス画面に挿入可能で、[プロパティ]ウィンドウで関連する使用場所に設定できます。

## フェイスプレートインスタンス

プロセス画面にフェイスプレートを挿入すると、このフェイスプレートのインスタンスが作成されます。プロジェクトライブラリでフェイスプレートを編集する場合、ここで生成したフェイスプレートインスタンスは自動的に更新されます。

## フェイスプレートインスタンスの再利用

フェイスプレートに含まれるオブジェクトのオブジェクトプロパティに、直接設定したダイナミックコントロールは、個々のフェイスプレートインスタンスに含まれています。

別の方法として、フェイスプレートのインスタンスをあらかじめ設定しておき、それを再利用します。これにより、新規に生成された各インスタンスは、たとえば同じスクリプトを実行するか同じタグを使用します。フェイスプレートの最終設定インスタンスをライブラリ内にドラッグして、後ほど再使用できます。フェイスプレートが変更された場合でも、この事前設定したフェイスプレートインスタンスを引き続き使用できます。ただし、この結果、フェイスプレートのインターフェースが基本的に変更されないことが必要になります。

## セキュリティレベルの設定

事前設定したセキュリティレベルは、そのフェイスプレート内およびそのフェイスプレートインスタンスを使用するプロジェクトでのみ有効であるため、フェイスプレート内にセキュリティレベルを割り当てることはできません。セキュリティレベルを設定するには、フェイスプレート内のオブジェクトの[ユーザーオーソリゼーション]プロパティをインターフェースに接続します。そこで、ユーザーオーソリゼーションを、該当するフェイスプレートインスタンスへ割り当てます。

## 複数のプロジェクトでのフェイスプレートの再使用

WinCC flexible ではフェイスプレートをグローバルライブラリに追加することができます。この結果、他のプロジェクトでこのフェイスプレートを使用できます。共有ライブラリから画面にフェイスプレートを追加すると、システムはそのコピーを自動的にプロジェクトライブラリに保存します。プロジェクトライブラリのフェイスプレートを変更します。

### 5.8.2 ダイナミックコントロールオプション

#### はじめに

銘板のプロパティとイベントのダイナミックコントロール方法は 2 つあります：

- 銘板インスタンスで

銘板インスタンスでは、イベントまたはダイナミックプロパティは、使用場所に対して個別に設定できます。これを実行するには、まず銘板で各プロパティとイベントをセットアップしてください。

- 銘板に含まれるオブジェクトで

銘板に含まれるオブジェクトの、プロパティ、イベント、およびアニメーションは、直接設定できます。プロパティとアニメーションは銘板タグまたはダイナミックプロパティと相互接続できます。

さらに、銘板に含まれるオブジェクトのプロセス値を、ダイナミックプロパティとリンクできます。これを実行するには、まず銘板で各プロパティをセットアップしてください。こうすることで、銘板を事前に設定しておくことができます。このようにして生成された銘板インスタンスは、すべて同じダイナミックコントロールを有します。このダイナミックコントロールは、「銘板」エディタでのみ編集可能です。

## アラームシステムの作成

### 6.1 基本

#### 6.1.1 プロセスとシステムアラームの表示

##### はじめに

- ユーザー定義のアラーム  
HMI デバイス上で、プロセスの状態を表示したり PLC から受け取るプロセスデータの測定やレポートを行ったりするには、アラームを設定します。
- システムアラーム  
システムアラームは、HMI デバイスまたは PLC の特定のシステム状態を表示するように、これらのデバイスに事前定義されています。  
ユーザー定義のアラームやシステムアラームも、HMI デバイスまたは PLC によってトリガされ、HMI デバイス上で表示することができます。

##### アラームシステムのタスク

- HMI デバイスでの表示: プラントまたはプロセスで発生するイベントや状態の報告状態は、発生すると直ちに報告されます。
- レポート機能: プリンタにアラームイベントを出力します。
- ロギング: 引き続き編集や評価を行えるようにアラーム結果を保存します。

## 6.1.2 ユーザ一定義のアラーム

### 6.1.2.1 利用可能なアラーム手順

#### WinCC flexible でのアラーム方法

アラーム方法は、アラームをトリガする情報のタイプを識別するため、アラームのプロパティも識別します。

WinCC flexible は、以下のアラーム手順をサポートしています。

- ディスクリットアラーム手順  
PLC に特定のビットが設定されると、HMI デバイスによってアラームがトリガされます。WinCC flexible でこれを行うには、ディスクリットアラームを設定します。
- アナログアラーム手順  
特定の"タグ"が"限界値"を超えると、HMI デバイスによってアラームがトリガされます。WinCC flexible でこれを行うには、アナログアラームを設定します。
- アラーム番号手順  
PLC から HMI デバイスにアラーム番号(および関連付けられたアラームテキスト)が転送されます。このために、PLC のコンフィグレーションソフトウェアでさまざまなアラームを設定することができます。
  - SIMATIC STEP 7:  
ALARM\_S アラーム
  - SIMOTION SCOUT:  
ALARM\_S アラーム、および技術的アラーム

#### アラームの確認

非常に危険または人体に危害が及ぶ可能性のあるようなオペレーティング状態およびプロセス状態を示すアラームの場合、プラントオペレータがアラームを確認しなければならないように設定することができます。

権限を与えられたユーザーだけがアラームメッセージを編集できるようにする場合、該当するユーザーオーソリゼーションを、キーパッドでデバイスの[ACK]ボタンに割り付けます。

#### アラーム状態

ディスクリットアラームおよびアナログアラームには、以下のアラームステータスがあります。

- アラームをトリガする条件を満足したら、アラームステータスが"有効"になります。オペレータがこのアラームを確認すると、ステータスは"有効/確認済み"になります。
- アラームをトリガする条件が解消された場合、アラームステータスは"有効化/無効化"になります。無効化されたアラームをオペレータが確認すると、ステータスは"有効化/無効化/確認済み"になります。

アラームステータスの発生はそれぞれ HMI デバイス上で表示およびロギングしたり、印刷したりすることもできます。

### 6.1.2.2 アラームの確認

#### はじめに

非常に危険または人体に危害が及ぶ可能性のあるようなオペレーティング状態およびプロセス状態を示すディスクリートアラームおよびアナログアラームの場合、プラントオペレータがアラームを確認しなければならないように設定することができます。

#### アラーム確認のメカニズム

アラームは、HMI デバイス上のオペレータまたはコントロールプログラムによって確認できます。オペレータがアラームを確認したときに、タグ内にビットを設定することができます。

オペレータが確認する場合には、次のオプションが便利です。

- 確認キー<ACK>(特定の HMI デバイスでのみ使用可能)
- ファンクションキー、ソフトキー、または画面のボタン

さらに、ファンクションリストまたはスクリプトのシステムファンクションを介してアラームを確認することもできます。

---

#### 注記

権限を与えられたユーザーだけがアラームメッセージを編集できるようにする場合、該当するユーザーオーソリゼーションを、キーパッドでデバイスの[ACK]ボタンに割り付けます。

---

#### 確認を必要とするアラーム

アラームを確認する必要があるかどうかは、アラームクラスによって決まります。

アラームクラスでは、基本的に、HMI デバイスへのアラームの表示方法と確認動作を定義します。WinCC flexible には、事前定義されたアラームクラスとユーザー定義のアラームクラスを設定するオプションの両方があります。

#### PLC による確認

ディスクリートアラームは、PLC でタグ内に特定のビットを設定することで確認されます。

#### アラームの一括確認

アラームを設定する際、オペレータがアラームを個々に確認する必要があるのか、または同じアラームグループ内のアラームをまとめて確認できるのかを指定することができます。たとえば、アラームの原因が同じエラーである場合、アラームグループを使用すると便利です。

### 6.1.2.3 アラームクラス

#### アラームクラス

アラームクラスは、主に HMI デバイス上でのアラームの表示方法を決定します。アラームクラスは、アラームをグループ化してそれぞれ異なる方法で表示するときにも使用します。

WinCC flexible には、事前定義されたアラームクラスとユーザー定義のアラームクラスを設定するオプションの両方があります。

#### 使用可能なアラームクラス設定

各アラームクラスには、以下の設定を定義できます。

- 確認: このクラスのアラームは、確認をする必要があります。
- アラームを表示する際に各アラームステータスを識別するテキスト、色、および点滅モード
- このクラスのアラームに関連するすべてのイベントをロギングするアラームログ
- HMI デバイスにアラームを表示する際に、アラームクラスを示すためにアラーム番号の前に配置するテキスト
- このクラスのアラームに関連するイベントについての全メッセージを送信する電子メールアドレス

#### WinCC flexible の事前定義されたアラームクラス

- 非常に危険または人体に危害が及ぶ可能性のあるような動作状態およびプロセス状態を示すディスクリートアラームおよびアナログアラームの場合には、"エラー"。このクラスのアラームは、常に確認をする必要があります。
- 通常のオペレーティング状態、プロセス状態、およびプロセスシーケンスを示すディスクリートアラームとアナログアラームの場合には、"警告"。このクラスのアラームは確認を要求しません。
- HMI デバイスと PLC のオペレーティング状態についてオペレータに通知するシステムアラームの場合には、"システム"。このアラームクラスは、ユーザー定義のアラームには使用できません。
- SIMATIC S7 または SIMOTION PLC のステータスやイベントを示す、S7 診断メッセージの場合には、"診断イベント"。このクラスのアラームは確認を要求しません。

事前定義されたアラームクラスの場合には、ごく一部のプロパティしか変更できません。

### 6.1.3 システムアラーム

#### はじめに

システムアラームは、HMI デバイスと PLC のオペレーティング状態についてオペレータに通知するものです。システムアラームの内容は、たんなる注意から重大なエラーにまで及びます。

#### システムアラームのトリガ

HMI デバイスまたは PLC のいずれかにおいて、または 2 つのデバイス間での通信中に特定のシステムステータスまたはエラーが発生した場合に、アラームがトリガされます。

システムアラームは番号とアラームテキストで構成されます。アラームテキストには、アラームの原因をより詳しく示す内部システムタグを含めることもできます。システムアラームの場合、ごく一部のプロパティしか設定できません。

#### システムアラームのタイプ

システムアラームには次の 2 つのタイプがあります。

- HMI システムアラーム

HMI システムアラームは、特定の内部状態が発生した場合、または PLC との通信中にエラーが発生した場合に、HMI デバイスによってトリガされます。

- PLC によるシステムアラーム

このシステムアラームは PLC によって生成され、WinCC flexible では設定できません。

#### HMI デバイス上でのシステムアラームの表示

アラームシステムの基本設定で、HMI デバイスに表示するシステムアラームのタイプと、システムアラームをどのくらいの時間表示するかを指定することができます。

HMI デバイスにシステムアラームを表示するには、[アラームウィンドウ]と、[アラームウィンドウ]オブジェクトを使用します。

画面またはテンプレートにこれらのオブジェクトのいずれかを設定するたびに、"システム"アラームクラス設定を選択します。

#### デバイス固有のシステムアラーム

HMI デバイスのマニュアルには、起こり得るシステムアラームと、その原因、および対処方法のリストが記載されています。

HMI システムアラームについてオンラインサポートにお問い合わせいただく場合、アラーム番号とシステムアラームタグが必要です。

## 6.1.4 アラームの表示

### 6.1.4.1 HMI デバイスでのアラームの表示

#### HMI デバイスでのアラームの表示オプション

WinCC flexible には、HMI デバイスで以下のアラームの表示オプションがあります。

- [アラーム]画面

[アラーム]画面は、特定の画面に設定されます。設定されたサイズによっては、同時に複数のアラームを表示することができます。異なるアラームクラス用に別々の画面に複数の[アラーム]画面を設定することも可能です。

[アラーム]画面は、1つのアラーム行だけを含むように設定することができます。

- [アラーム]画面

アラームウィンドウは画面のテンプレートで設定されるため、プロジェクトのすべての画面の構成要素となります。設定されたサイズによっては、同時に複数のアラームを表示することができます。イベントによって、アラームウィンドウを閉じたり、再度開いたりするアクションをトリガすることができます。[アラーム]ウィンドウは、独自のレイヤーに保存されます。その実際上の理由は、これにより、設定中に特別にそのウィンドウを非表示にすることができるためです。

#### その他の信号: アラームインジケータ

アラームインジケータは、アラームが有効化されると画面に表示される、設定されたグラフィックシンボルです。アラームインジケータは画面のテンプレートで設定されるため、プロジェクトのすべての画面の構成要素となります。

アラームインジケータは次の2種類の状態のいずれかを表示できます。

- 点滅: 少なくとも1つの未確認アラームが保留中です。
- スタティック: アラームは確認されていますが、少なくともそのうちの1つがまだ回復していません。

ファンクションリストを使用すると、HMI デバイスの応答を設定することができます。

#### [アラーム]画面

以下の HMI デバイスは、たとえば確認によってアラームステータスが変化すると、アラームのタグ値を更新します。

- OP 73micro
- TP 177micro
- OP 73
- OP 77A
- TP 177A

その他の HMI デバイスのタグ値は変更されません。

保留中および未確認のアラームが[アラーム]ウィンドウに表示される場合、表示されたアラームは、以下の HMI デバイスでは、アラーム確認後に再ソートされません。

- OP 73micro
- TP 177micro
- OP 73
- OP 77A
- TP 177A

#### 6.1.4.2 アラーム表示のフィルタリング

##### はじめに

拡張された[アラーム]ウィンドウや[アラーム]ウィンドウでは、アラームのテキストによって表示をフィルタリングできます。フィルタ基準として、フィルタ変数の文字列や値を使用します。ランタイムでは、アラームテキストに設定された文字列を含むアラームや、フィルタ変数からの文字列だけが表示されます。フィルタは、ラインタイム時の表示だけに適用されます。すべてのアラームは、アラームバッファに保持されます。

##### フィルタリングのルール

固定のフィルタエントリを使用するフィルタリングでは、[アラーム]ウィンドウのプロパティの文字列を設定します。アラームテキストの文字列全体を含むアラームはすべて、ランタイム中に表示されます。

修正可能なフィルタリングのためのフィルタ基準のタグを使います。タグは、ランタイム中に I/O フィールドなどを通して、希望する文字列で取得します。アラームの表示は、タグの内容によってフィルタされます。

##### フィルタの動作

- フィルタのエントリが空の場合、保留されているアラームがすべて表示されます。
- フィルタタグは、恒久的に設定可能な文字列の上位にあります。もし、恒久的に割り当てられた文字列やフィルタタグが、設定中に割り当てられる場合、アラームはフィルタタグの内容によってフィルタされます。フィルタタグが空の場合、恒久的に割り当てられた文字列によって、フィルタリングします。
- フィルタリングでは、大文字・小文字を区別します。ワイルドカードや正規表現は、考慮されません。
- フィルタリングは、言語依存ではありません。言語が変わっても、設定された文字列は不変です。
- フィルタエントリの長さは、アラームテキストの最大長以下に制限されます。アラームテキストの最大長は、使用している HMI デバイスによって決まります。長さに関するより詳細な情報は、"システムの制限"の章をご覧ください

[アラーム]表示または[アラーム]ウィンドウのプロパティで、フィルタリングを設定します。

以下の[アラーム]表示に対しては、フィルタリングはできません：

- [簡易アラーム]ウィンドウ
- アラーム行;アラームギョウ
- 自動的に表示される[アラーム]ウィンドウ
- アラームログのアラームを表示するために設定された[アラーム]表示
- アラームログのアラームを表示するために設定された[アラーム]ウィンドウ

#### 6.1.4.3 アラームのロギングとレポート

##### アラームの評価と文書化

[アラーム]ウィンドウへのアラームイベントのリアルタイム表示に加えて、WinCC flexible には、アラームを評価および文書化するための以下のオプションがあります。

- アラームイベントを発生後直ちに印刷できます。
- アラームバッファからのアラームイベントをレポート形式で印刷できます。
- アラームイベントをアラームログにロギングできます。
- ログ済みのアラームイベントを HMI デバイスに表示したり、レポート形式で印刷したりすることができます。

##### アラームの即時印刷

プロジェクト全体のアラームの印刷は、アラームシステムの基本設定で有効化または無効化することができます。さらに、個々のアラームの印刷を有効にすることもできます。

##### アラームのロギング

アラームクラスは、アラームの割り付けをアラームログにコンフィグレーションするために使用されます。アラームログを各アラームクラス向けに指定することができます。このアラームクラスのアラームに関連するすべてのイベントが、指定したアラームログにロギングされます。

##### アラームのレポート

[印刷アラーム]オブジェクトのプロパティを使用して、レポートへのアラームの割り付けをコンフィグレーションします。データソース(アラームバッファまたはアラームログ)からレポートに出力できるだけでなく、アラームクラスに基づいてフィルタを行うこともできます。

## 6.1.4.4 アラーム編集のためのシステムファンクション

## システムファンクション

システムファンクションは、プログラミングの知識がなくてもランタイム時に数多くのタスクを実行できるように事前定義されたファンクションです。ファンクションリストまたはスクリプトの中でシステムファンクションを使うことができます。

この表は、アラームおよびディスプレイの操作を編集するのに使用可能なすべてのシステムファンクションを示しています。

システムファンクション	効果
EditAlarm	選択したすべてのアラームの[編集]イベントをトリガします。
ClearAlarmBuffer	HMI デバイスのアラームバッファからアラームを削除します。
ClearAlarmBufferProtoolLegacy	ClearAlarmBuffer と同じようなファンクションです。このシステムファンクションを保持して互換性を確保しておき、古い ProTool ナンバリングを使用します。
AlarmViewEditAlarm	任意のアラーム画面で選択されたすべてのアラームの[編集]イベントをトリガします。
AlarmViewAcknowledgeAlarm	所定の[アラーム]画面で選択されたアラームを確認します。
AlarmViewShowOperatorNotes	任意のアラーム画面で選択されたアラームの設定済みヘルプテキストを表示します。
AcknowledgeAlarm	選択されたすべてのアラームを確認します。
SetAlarmReportMode	プリンタのオンまたはオフ時のアラームの自動レポート機能を切り替えます。
ShowAlarmWindow	HMI デバイスの[アラーム]ウィンドウを表示または非表示にします。
ShowSystemAlarm	HMI デバイスのシステムアラームとして配信されたパラメータの値を表示します。

これらのシステムファンクションについての詳細は、[WinCC flexible での作業|参照|システムファンクション]を参照してください。

### アラーム表示用の各アラームとオブジェクトに対応したイベント

以下のイベントは、ランタイム時にアラームとなった場合、およびアラームのオブジェクトが表示された場合に発生します。ファンクションリストは、各イベント用に設定が可能です。

オブジェクト	設定可能なイベント
ディスクリートアラーム	動作、 終了、 確認、 編集
アナログアラーム	動作、 終了、 確認、 編集
[アラーム]画面	有効化 終了
アラームインジケータ	クリック、 点滅時にクリック

これらのイベントについての詳細は、[WinCC flexible での作業|参照|システムファンクション]を参照してください。

## 6.2 エlementと基本設定

### 6.2.1 アラームの構成要素とプロパティ

#### アラームのプロパティ

アラームは、常に、以下の要素から構成されています。

- アラームテキスト

アラームテキストには、アラームの説明が含まれています。該当する HMI デバイスでサポートされている文字フォーマットを使用すれば、一文字ごとにアラームテキストをフォーマットできます。

ヘルプテキストには、タグまたはテキストリストの現在値の出力フィールドを含めることができます。アラームバッファは、アラームステータスが変化したときの瞬間的な値を保持します。

- アラーム番号

アラーム番号は、アラームの参照に使用されます。以下のタイプのアラーム内では、各アラーム番号は一意になります。

- ディスクリートアラーム
- アナログアラーム
- HMI システムアラーム
- CPU 内の PLC からのアラーム

- アラームトリガ

- ディスクリートアラーム用タグ内のビット
- アナログアラーム用タグ用限界値

- アラームクラス

アラームのアラームクラスによって、そのアラームを確認する必要があるかどうかが決まります。アラームクラスを使用すれば、HMI デバイスでのアラームの表示方法を決定することもできます。また、対応するアラームのログを記録するかどうか、およびログの保存場所もアラームクラスによって決定されます。

---

#### 注記

SIMATIC STEP7 にプロジェクトを統合する場合は、WinCC flexible と STEP 7 では、合計で最大 7 つのアラームクラスを設定できます。

---

これらの構成要素は、アラームごとに自由に選択、入力することができます。

### オプションのアラームプロパティ

アラームの動作は、以下のプロパティによっても定義できます。

- アラームグループ

アラームがアラームグループに属している場合、同じグループ内の他のアラームとそのアラームを1回の操作で確認できます。

- 情報テキスト

ヘルプテキストには、アラームに関するその他の情報を含めることができます。ヘルプテキストは、オペレータが[ヘルプ]ボタンを押すと、オペレータデバイスの別のウィンドウに表示されます。

- 自動レポート機能

プロジェクト全体のアラームの自動レポート機能を有効化および無効化するオプションに加えて、個々のアラームに対してレポート機能を有効化することも可能です。

- PLC"確認書込みタグ"による確認"

ディスクリートアラームは、タグ内に特定のビットを設定することで、PLCプログラムによって確認されます。

- 確認の PLC"確認読取りタグ"への送信"

オペレータがディスクリートアラームを確認すると、特定のビットをタグ内に設定できます。

## 6.2.2 アラームを設定するためのエディタ

### 6.2.2.1 エディタの基本原理

#### アラームを設定するためのエディタ

WinCC flexible には、アラームをコンフィグレーションするための以下のテーブル形式エディタがあります。

- ディスクリットアラームの作成と変更のための"ディスクリットアラーム"エディタ
- アナログアラームの作成と変更のための"アナログアラーム"エディタ
- システムアラームのアラームテキストの変更のための"システムアラーム"エディタ
- アラームクラスの作成と変更のための"アラームクラス"エディタ
- アラームグループの作成と変更のための"アラームグループ"エディタ

#### エディタの取り扱い

すべてのエディタでは、以下の機能が使用可能です。

- 列表示の変更
- オブジェクトの削除とコピー
- 複数のテーブル行の自動入力
- ドラッグアンドドロップによるプロパティのコピー
- ドラッグアンドドロップによるオブジェクトウィンドウからのオブジェクトのコピー
- テーブルの内容のソート

#### 列表示の変更

以下のような列表示をコンフィグレーションします。

- 列の表示または非表示  
ショートカットメニューで列ヘッダーを選択します。  
このファンクションは、"アラームグループ"エディタではこのエディタが2つの列しかないため、使用できません。
- 列幅の変更  
列ヘッダーの右端を希望する幅までドラッグします。
- 列の順序の変更  
列を選択して列ヘッダーを希望する位置までドラッグします。  
このファンクションは、"アラームグループ"エディタでは使用できません。
- ソート  
列ヘッダーをクリックします。同じ列ヘッダーを再びクリックすると、ソート順を元に戻すことができます。  
対応する列ヘッダーが、矢印によりマークされます。矢印の方向は、ソート順序を示しています。

### オブジェクトの削除とコピー

1つ以上のオブジェクト全体の削除またはコピーは、行の左端にあるシンボルをクリックすることによって行います。これによって、テーブル行全体にマークを付けることができます。

### ドラッグによる複数のテーブル行の自動入力

WinCC flexible のテーブル形式エディタでは、1つの操作で複数のテーブル行の入力ができます。この機能は以下のタスクで使用できます。

- プロパティが似ている複数の新規オブジェクト(アラーム、アラームクラス、またはアラームグループ)の作成。以下のように実行します:
  - コピーするテーブル列が下になるようにテーブルをソートします。
  - コピーする列のはじめのElementを選択します。
  - 選択したテーブルElementの左下隅を下の空白のテーブルエリアまでドラッグします。
- 複数の既存のオブジェクトに対するプロパティの転送(たとえばトリガタグの変更)。以下のように実行します:
  - 関係のあるプロパティを持つテーブルElementを選択します。
  - 選択したテーブルElementの右下隅を、修正するテーブル行を越えて下方にドラッグします。

### ドラッグアンドドロップによるテーブル内でのプロパティのコピー

ドラッグアンドドロップ操作によって、個々のプロパティ(アラームテキストや色など)を1つのテーブルElementから別のテーブルElementにコピーできます。

### オブジェクトウィンドウからのドラッグアンドドロップ

ドラッグアンドドロップによって、タグなどのオブジェクトウィンドウからテーブルセルにオブジェクトをドラッグします。この機能は、テーブルセルでそのオブジェクトが使用できる場合に限って実行できます。

### 6.2.2.2 "ディスクリートアラーム"エディタ

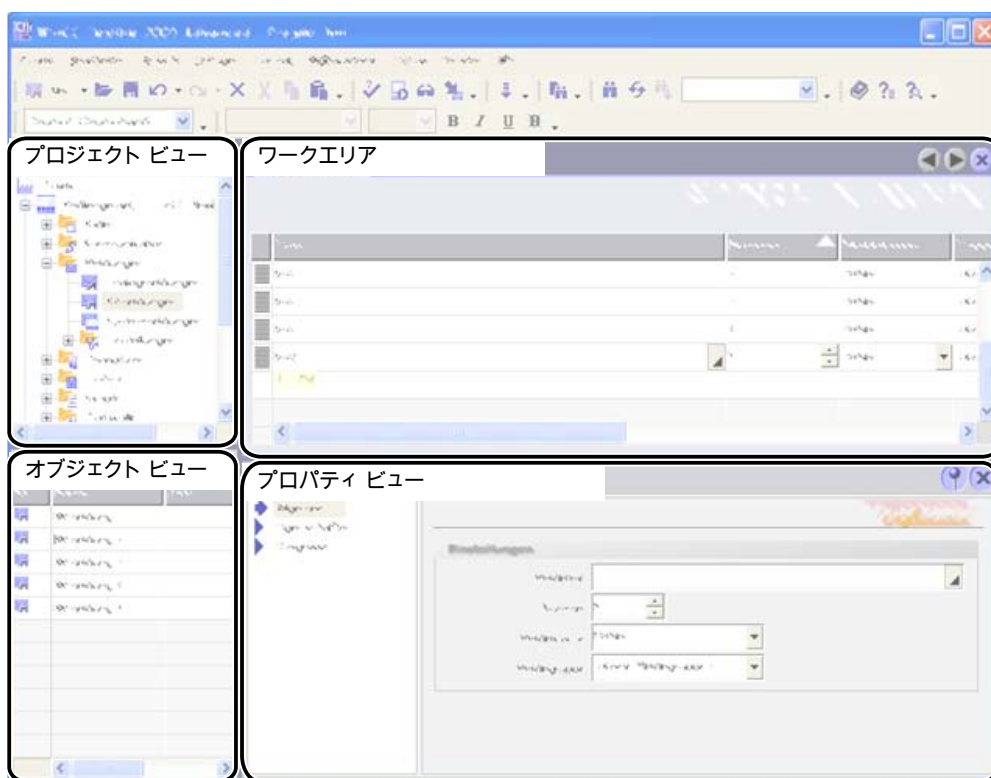
#### はじめに

"ディスクリートアラーム"テーブル形式エディタでは、ディスクリートアラームを作成し、そのプロパティを指定します。

#### 開く

[プロジェクト]ウィンドウで、[アラーム]グループ内の[ディスクリートアラーム]をダブルクリックします。

#### レイアウト



#### 作業エリア

作業エリアでは、設定済みのディスクリートアラームと関連する設定がすべてテーブル形式で表示されます。テーブルのセル内にあるディスクリートアラームのプロパティは、編集可能です。

#### [プロパティ]ウィンドウ

[プロパティ]ウィンドウでは、ディスクリートアラームを設定します。このウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。

### 6.2.2.3 "アナログアラーム"エディタ

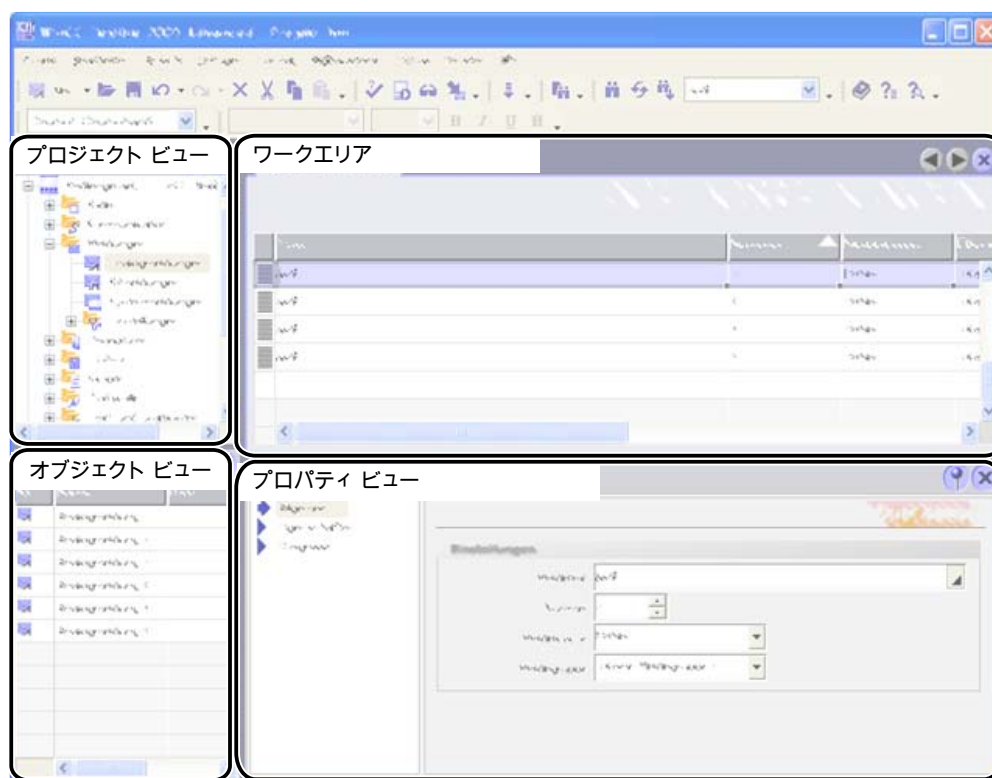
#### はじめに

"アナログアラーム"テーブル形式エディタでは、アナログアラームを作成し、そのプロパティを指定します。

#### 開く

[プロジェクト]ウィンドウで、[アラーム]グループ内の[アナログアラーム]をダブルクリックします。

#### レイアウト



#### 作業エリア

作業エリアでは、設定済みのアナログアラームと関連する設定がすべてテーブル形式で表示されます。テーブルのセル内にあるアナログアラームのプロパティは、編集可能です。

#### [プロパティ]ウィンドウ

[プロパティ]ウィンドウでアナログアラームをコンフィグレーションします。このウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。



## [プロパティ]ウィンドウ

システムアラームのアラームテキストは、[プロパティ]ウィンドウで修正します。アラーム番号とアラームクラスは、システムによって割り付けられます。

### 6.2.2.5 "アラームクラス"エディタ

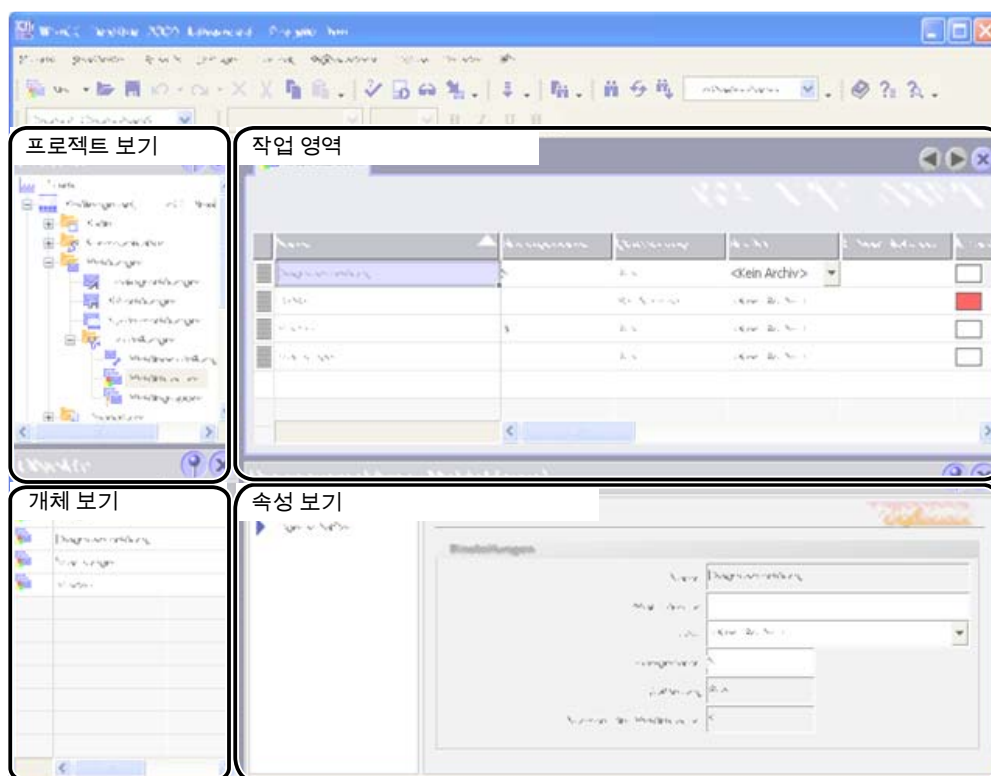
#### はじめに

"アラームクラス"テーブル形式エディタでは、アラームクラスを作成し、そのプロパティを指定します。

#### 開く

[プロジェクト]ウィンドウで、[アラーム|設定]グループ内の[アラームクラス]をダブルクリックします。

#### レイアウト



#### 作業エリア

作業エリアでは、設定済みのアラームクラスと関連する設定がすべてテーブル形式で表示されます。テーブルのセル内にあるディスクリットアラームのプロパティは、編集可能です。

## [プロパティ]ウィンドウ

[プロパティ]ウィンドウでは、アラームクラスの設定を行います。このウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。

### 6.2.2.6 "アラームグループ"エディタ

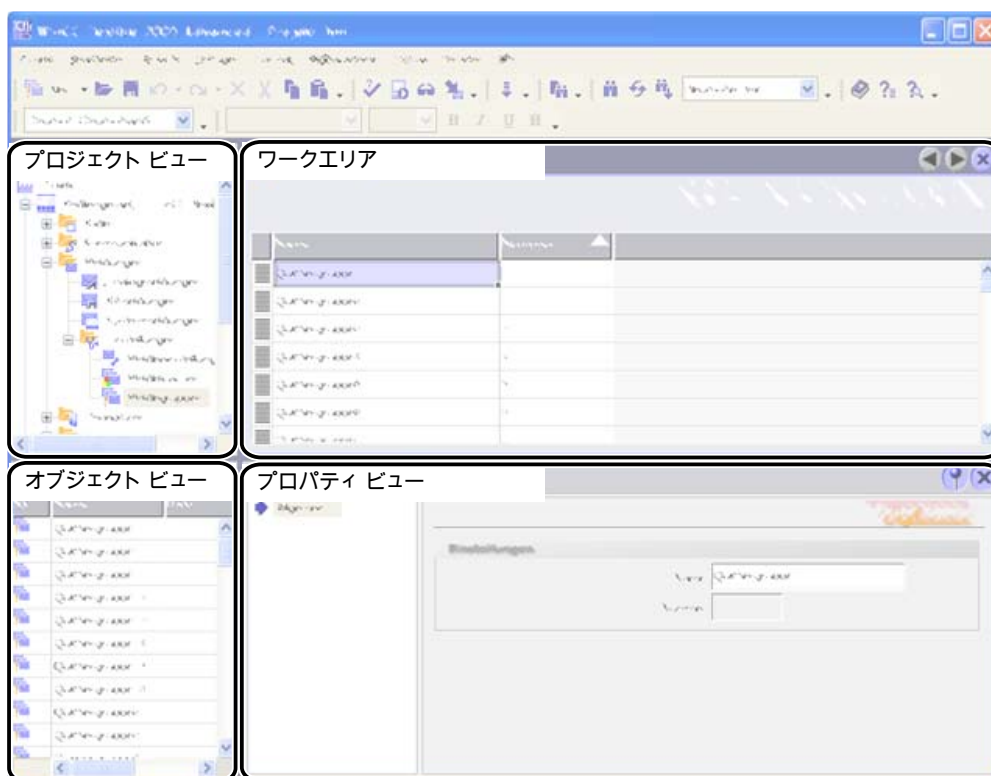
#### はじめに

"アラームグループ"テーブル形式エディタでは、アラームグループを作成し、そのプロパティを指定します。

#### 開く

[プロジェクト]ウィンドウで、[アラーム|設定]グループの[アラームグループ]をダブルクリックします。

#### レイアウト



#### 作業エリア

作業エリアでは、設定済みのアラームグループと関連する設定がすべてテーブル形式で表示されます。テーブルのセル内にあるアラームグループのプロパティは、編集可能です。

#### [プロパティ]ウィンドウ

アラームグループの名前は、[プロパティ]ウィンドウで修正します。アラームグループ番号は、システムによって割り付けられます。

### 6.2.2.7 アラームシステムの基本設定

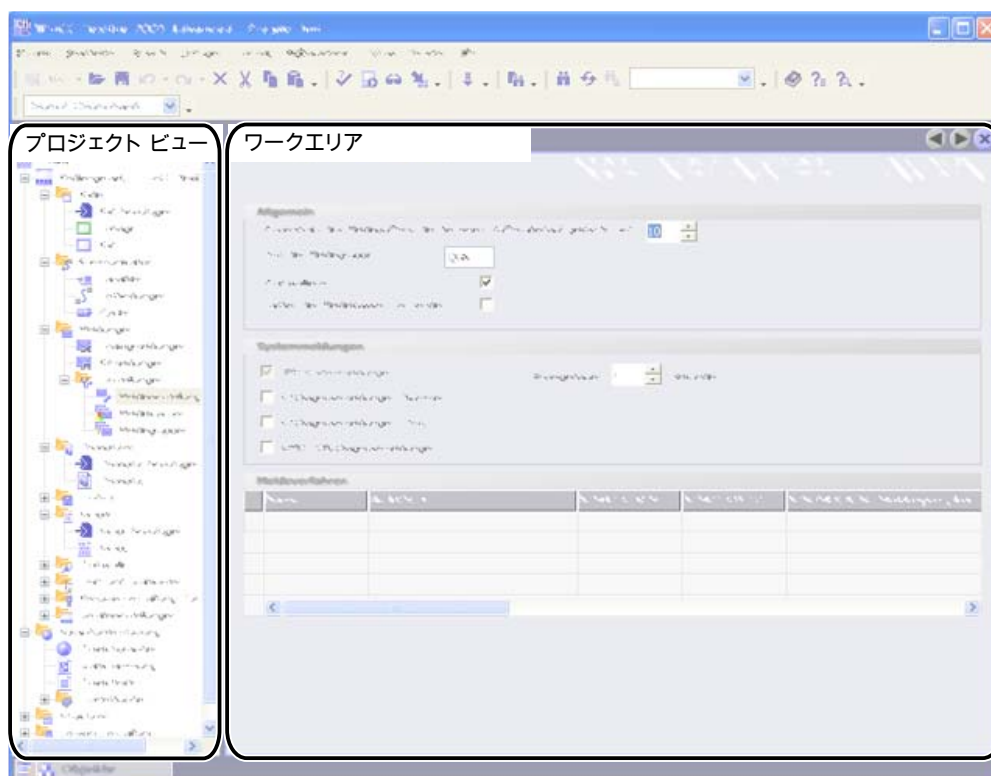
#### はじめに

WinCC flexible アラームシステムはデフォルト設定を使用して作動します。アラームシステムの動作を特定プラントの条件に合わせるには、これらのデフォルト設定を変更します。

#### 開く

[プロジェクト]ウィンドウで、[アラーム|設定]グループ内の[アラーム設定]をダブルクリックします。

#### レイアウト



#### 作業エリア

アラームシステムの設定は、作業エリアで定義します。選択した[システムアラーム]エリアで、たとえば HMI デバイスで表示するシステムアラームのタイプなどを選択します。統合操作では、[アラーム手順]エリアでさらに細かい設定を定義できます。

## 6.3 アラームの取り扱い

### 6.3.1 アラームのレポート機能

#### はじめに

WinCC flexible で、以下のアラームを出力できるレポートをコンフィグレーションします。

- アラームバッファの現在のアラーム
- アラームログからのアラーム

#### 必要条件

#### 手順

以下のように実行します：

1. ツールボックスオブジェクトから[印刷アラーム]をレポートに追加します。
2. プロパティを[プロパティ]ウィンドウに表示させるには、オブジェクトを選択します。
3. [プロパティ]ウィンドウから、コンフィグレーションするデータを作成します。
4. 選択したソース用に出力したいアラームクラスを指定します。
5. 出力用アラームのシーケンスを指定します。
6. ある間隔のアラームを出力するには、[表示開始]フィールドと[表示終了]フィールドについてそれぞれのタグを選択します。タグは、ランタイム中その期間に発生した最初または最後のアラームの日付と時刻を表示します。

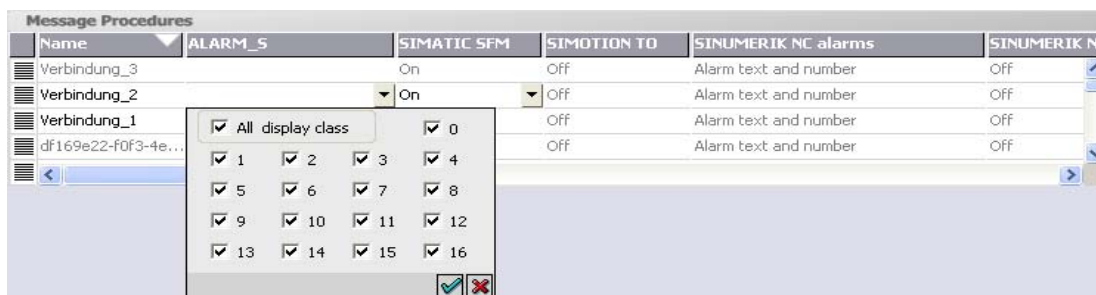
### 6.3.2 アラームナンバリング手順によるアラームの統合化

#### SIMATIC STEP 7 での設定

ALARM\_S および ALARM\_D は、アラームナンバリング手順です。アラーム番号は、STEP 7 の設定時に自動的に割り付けられます。これらの番号は、アラームメッセージを一意に割り付けるために使用されます。

STEP 7 でアラームを設定するとき、保存されたアラームと属性は STEP 7 設定データに保存されます。WinCC flexible は自動的に必要なデータをインポートし、それらを後で HMI デバイスに転送します。

表示クラスを使用して、ALARM\_S アラーム WinCC flexible の出力をフィルタリングします。プロジェクトウィンドウで[アラーム|設定]の順に選択した後、[アラーム設定]をダブルクリックします。[アラーム手順]エリアに既存の接続が表示されます。



必要な接続の行で[ALARM\_S 表示クラス]列にあるフィールドを選択し、選択ボタンを押して選択ダイアログボックスを開きます。必要な表示クラスを選択します。ボタンを押して選択ダイアログボックスを閉じます。

リンクの[SFM Alarms]列でシステムエラーを表示する必要があるか指定します。詳細については、STEP 7 のマニュアルを参照してください。

### ALARM\_S アラームの最大数

WinCC flexible における ALARM\_S アラームの最大数は 32767 です。実際には、構成可能なアラームの最大数は、HMI デバイスの利用可能なメモリーで制限されます。

### アラームクラスレイアウト

ALARM\_S と ALARM\_D の各アラームは、STEP 7 内の特定アラームクラスに割り付けられます。これらのアラームクラスの表示オプションを編集するには、WinCC flexible の[プロジェクト]ウィンドウで[アラーム]設定|アラームクラス]の順に選択します。ショートカットメニューを開き、[エディタを開く]コマンドを選択します。アラームクラス名に S7 という接頭語が付いているものが、アラームクラスです。

ALARM CLASSES							
Name	Acknowledgment	Log	I color	IO color	IA color	IOA color	
S7Alarm	On "incoming"	<No log>	Red	Orange	Yellow	Green	
S7NoAlarm	Off	<No log>	Magenta	Cyan			
S7OperationMessage	Off	<No log>					
S7OperatorInputRequest	Off	<No log>					
S7ProcessControlMaintainance	Off	<No log>					
S7ProcessControlSystemMessageOs	Off	<No log>					
S7ProcessControlSystemMessagePlc	Off	<No log>					
S7ProcessMessageAlarm	On "incoming"	<No log>					
S7ProcessMessageEvent	Off	<No log>					
S75statusMessage	On "incoming"	<No log>					
S7Tolerance	Off	<No log>					
S7Warning	Off	<No log>					

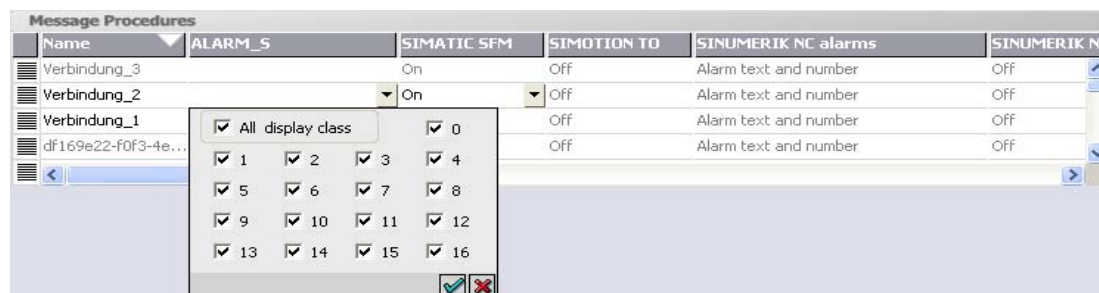
"アラームクラス"エディタを使うと、アラームクラスの表示オプションを設定できます。

## SIMOTION の Alarm\_S アラームの実装

Alarm\_S アラームも SIMOTION で使用できます。"アラーム設定"エディタを使用して、SIMOTION SCOUT で Alarm\_S アラームを設定します。

WinCC flexible は、STEP 7 の Alarm\_S アラームと同様に SIMOTION の Alarm\_S アラームを処理します。

表示クラスを使用して、ALARM\_S アラーム WinCC flexible の出力をフィルタリングします。プロジェクトウィンドウで[アラーム]設定の順に選択した後、[アラーム設定]をダブルクリックします。[アラーム手順]エリアに既存の接続が表示されます。



SIMOTION デバイスへの接続を含む[ALARM\_S 表示クラス]列のフィールドと行を選択します。選択ボタンを押して選択ダイアログボックスを開きます。必要な表示クラスを選択します。[OK]ボタンを押して選択ダイアログボックスを閉じます。

この接続の[アラーム]列へ行き、SIMOTION のプロセスアラームを表示するかどうかを定義します。詳細については、SIMOTION のドキュメントを参照してください。

アラームクラスの表示は、STEP 7 の Alarm\_S のアラームクラスと同様に設定されます。

## 6.4 アラームロギング;アラームロギング

### 6.4.1 アラームロギングの基本原理

#### はじめに

アラームは、プロジェクト内のプロセスの故障状態とオペレーティング状態を示します。通常、これらは PLC によってトリガされ、HMI デバイスの画面に表示されます。

WinCC flexible を使用すれば、アラームを記録し、プラントの稼動状態とエラー状態を文書化できます。

---

#### 注記

アーカイブは、すべてのデバイスでは使用できません。

---

#### 原理

ユーザーは、アラームのログ出力をコンフィグレーションすることができます。ログされるアラームは、アラームクラスを介してアラームログに割り付けられます。各メッセージは、特定のアラームクラスに属します。アラームクラスをコンフィグレーションする際に使用するアラームログを割り付けます。

これを実行する際には、ログに複数のアラームクラスのアラームを含めることができます。ログを作成する場合は、ログプロパティを指定し、ログ動作を選択します。ログは、以下のようなログ済みの各アラームのデータを記録します。

- アラームの日時
- アラームテキスト
- アラーム番号
- アラームステータス
- アラームクラス
- アラーム手順
- アラームテキストに含まれるタグの値
- コントローラ

---

#### 注記

アラームテキストとコントローラがロギングされるのは、ログのプロパティでコンフィグレーションされた場合だけです。

---

## 6.4.2 アラームロギング;アラームロギング

### はじめに

アラームをログ出力するには、アラームをアラームクラスにグループ化します。各アラームクラスは、別のログに記録することができます。HMI デバイスにより、構成設定を行う場合、ログタイプを選択することができます。ログを設定するときに、ログ動作を指定します。

### ログタイプ

WinCC flexible では、以下のログタイプから選択できます。

1. サイクリックログ
2. 一定期間のサイクリックログ
3. レベル依存のシステムアラームを持つログ
4. システムファンクションのレベル依存実行を伴うログ

アラームは、自動的にロギングすることも、オペレータによって管理することもできます。

### 記憶媒体

ログデータは、ファイルとデータベースのいずれかに保存できます。保存済みのデータは、他のプログラムで追加の処理(例: 分析目的)を受けます。

### ログ内容の表示

ログの内容は、HMI デバイスに表示します。これを行うには、[アラーム]ウィンドウを作成する必要があります。

### 6.4.3 "アラームログ"エディタ

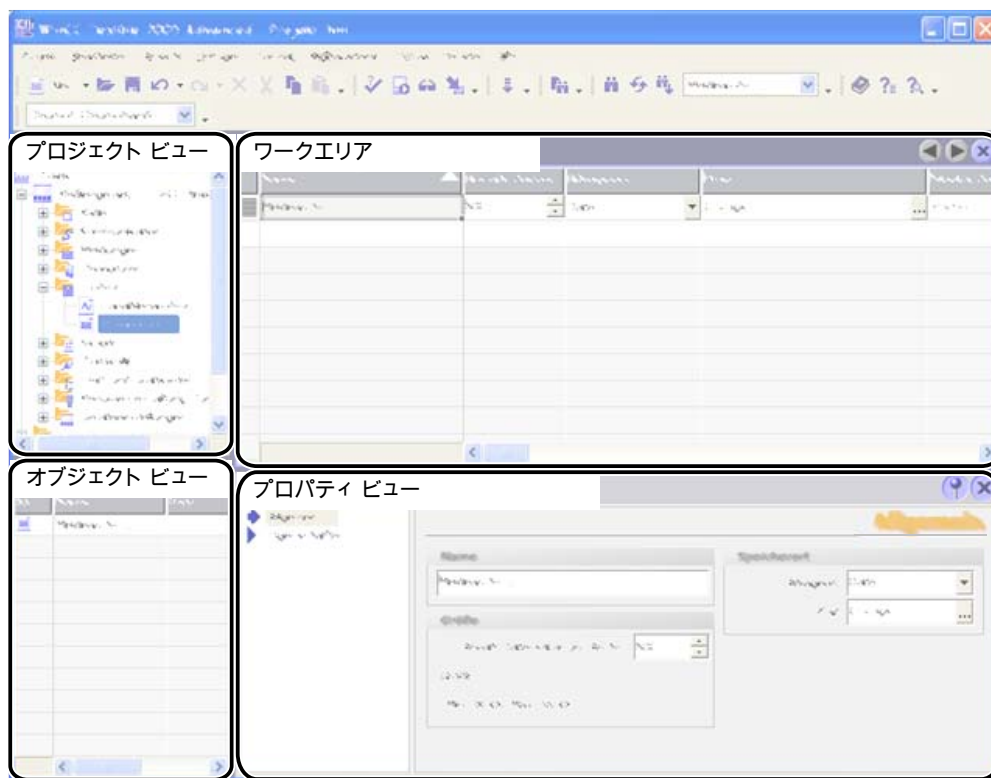
#### はじめに

"アラームログ"テーブルエディタでは、アラームクラスをロギングするためのアラームログを作成し、そのプロパティを定義します。

#### 開く

[ログ]グループのプロジェクトウィンドウで、[アラームログ]をダブルクリックして、"アラームログ"エディタを開きます。

#### 構造



#### 作業エリア

すべてのアラームログが、作業エリアのテーブルに表示されます。テーブルセル内のアラームログのプロパティを編集することができます。

## 6.4.4 アラームログの基本設定

### はじめに

アラームログのプロパティは、"アラームログ"エディタで指定します。

### [全般]プロパティ

アラームログについては、以下の全般プロパティを設定できます。

- 名前

アラームログの名前には、少なくとも 1 つの文字または数値が含まれている必要があります。

---

#### 注記

データソースの名前に使用できる文字は、記憶領域ロケーションにより異なります。

- \/\*?:"<>|などの文字は保存場所"ファイル - RDB"または"ファイル - CSV(ASCII)"に使用できません。 \/\*?:"<>|
- 記憶領域ロケーション"データベース"が使用された場合に使用できない文字: a-z A-Z 0-9 \_ @ # \$  
ただし、\_ @ # \$の各文字は、名前の先頭文字に使用できません。

- 保存先

アラームログは、ODBC データベース(PC 上のみ)または別のファイルに保存されます。したがって、保存先として[データベース]またはファイルを選択します。

HMI デバイスの構成によって、"パス"として以下を指定できます。

- PC のローカルハードディスク
- パネルのストレージカード
- 使用可能な場合は、ネットワークドライブ

保存先として ODBC データベースを選択している場合、データソースに次のような名前を指定できます。

- システムの割り付ける名前を使用する場合には、"システム定義のデータソース名"。

---

#### 注記

Windows VISTA では、オプション[システム定義のデータソース]のあるデータベースはサポートされません。

オプション[ユーザー定義のデータソース]のあるデータベースのみを使用できます。管理者として、これらのデータベースを作成して、PC にリリースしておく必要があります。

詳細については、使用するデータベースソフトウェアのマニュアルを参照してください。

ターゲットシステムで、Microsoft SQL サーバー専用のインスタンスも必要となります。このためには、たとえば無償で入手可能な SQL Server 2005 Express をダウンロードします。

Panel PC 477 ではこれを設定できません。

- "ユーザー定義のデータソース名"。データソースの名前を自分で割り付ける場合。

- サイズ

ログのサイズは、データレコードの数とエントリのおおよそのサイズから計算されます。エントリのサイズは、特にアラームテキストと関連したタグの値が同時にログされるかどうかによって異なります。

## アラームログのプロパティ

アラームログについては、以下の全般プロパティを設定できます。

- [スタートアップ動作]

セクション"有効化"で、ランタイム開始時のロギングの開始を指定できます。[ランタイム開始時のロギングの有効化]を有効にします。

他の方法でランタイム開始時の動作を制御することもできます。

- すでに有効になっているデータを新しいデータで上書きするには、[ログのリセット]を有効にします。
- ログ済みのデータを既存のログに追加するには、[ログの継続]を有効にします。

---

### 注記

システムファンクションは、ランタイム中にログの再起動を制御するために使用できません。

---

- [ロギング方法]

この欄で、ログが満杯になったときの処理を指定することができます。以下のオプションの1つを選択します。

- [サイクリックログ]  
ログが満杯になったとき、最も古いエントリが上書きされます。
- [一定期間のサイクリックログ]  
同じサイズの複数のログが作成され、1つのログが満杯になると別のログが使用されます。すべてのログが満杯になったとき、最も古いログが上書きされます。
- [システムアラームの表示]  
定義したレベルに達すると、システムアラームが表示されます。
- [トリガ]イベント  
ログが満杯になるとすぐに、[オーバーフロー]イベントがトリガされます。

- 設定

アラームがログされるたびに、アラームテキストとエラー箇所を保存するかどうかを定義します。アラームテキストは、現在のランタイム言語でログされます。

- コメント

ここに、ログに関する説明文を入力することができます。

## 6.4.5 アラームロギング;アラームロギング

### はじめに

ランタイム時にアラームをログに保存して、後で評価に使用することができます。アラームクラスのアラームをログするためには、以下を指定する必要があります。

- アラームの保存先となるログ
- 関連するアラームテキストおよびタグ値を保存するかどうか

### 原理

アラームロギングには次の手順が必要です。

- アラームログの作成および設定  
アラームログの作成時には、以下を定義する必要があります。
  - 名前、サイズ、保存先などの全般設定
  - ランタイム開始時の動作
  - ログが満杯になったときの動作
- アラームクラスでのアラームロギングの設定  
アラームログは、ランタイム時にアラームイベントが保存される各アラームクラスごとに指定することができます。
- ログ済みアラームのその後の処理  
ログ済みのアラームは、アラームウィンドウなどの使用中の WinCC flexible プロジェクト、または Excel などの他のアプリケーションで直接評価できます。

---

### 注記

アラームテキストのタグフィールドのシーケンスは言語によって異なります。Runtime 言語のシーケンスが、csv-log のアラームのロギングに使用されます。

1つの言語のタグフィールドのタグを変更すると、変更されたタグフィールドが他の言語すべてのアラームテキストの最後に表示されます。結果として、タグフィールドのシーケンスがログファイル内で思いがけず変更される場合があります。

---

## 6.4.6 ログ済みアラームの画面への表示

### はじめに

ランタイム中に、HMI デバイスの画面にログ済みアラームを表示することができます。このプロセス中に、ログデータベースからアラームクラスのアラームがダウンロードされ、[アラーム]ウィンドウに表示されます。

### 原理

HMI デバイスにログ済みアラームを表示するには、[アラーム]ウィンドウを作成する必要があります。[アラーム]ウィンドウをコンフィギュレーションするときは、表示するアラームのアラームクラスを指定します。

### 6.4.7 アラームが格納されている\*.csv ファイルの構造

#### はじめに

アラームログは CSV ファイルとして保存できます。CSV とは、コンマ区切り値(Comma Separated Value)の略です。この形式では、名前とエントリの値を含む列はセミコロンで区切られます。テーブルの各行は、キャリッジリターンで終了します。

#### \*.csv ファイルの例

この例はアラームがログされたファイルを示しています。

```
"Time_ms";"MsgProc";"StateAfter";"MsgClass";"MsgNumber";"Var1";"...;"
Var8";"TimeString";"MsgText";"PLC"37986550590,27;1;1;3;110001;"...;
";"30:06:99.13;15:51";"Change to operating mode
'online";37986550682,87;1;1;3;140010;"...;"30.06.99
13:12:59";"Connection established: PLC_1, Station 2, Rack 0,
Position 2";
```

#### \*.csv 形式のログファイルの構造

WinCC flexible ログファイルの各列には以下の値が入力されます。

パラメータ	説明
Time_ms	タイムスタンプを 10 進数値で指定します(変換については下を参照)
Msg_Proc	アラーム手順: 0 = 不明なアラーム手順 1 = システムアラーム 2 = アラームビット手順(オペレーティングアラーム) 3 = アラーム番号手順 ALARM_S 4 = 診断イベント 7 = アナログアラーム手順 100 = アラームビット手順(エラーのアラーム)
State after	アラームイベント: 0 = 受信済み/発信済み 1 = 受信済み 2 = 受信済み/確認済み/発信済み 3 = 受信済み/確認済み 6 = 受信済み/発信済み/確認済み
Msg_Class	アラームクラス 0 = アラームクラスなし 1 = "割り込み" 2 = "操作" 3 = "システム" 64 ... = ユーザーコンフィギュレーションアラームクラス
Msg Number	アラーム番号
Var1 ~ Var8	STRING によるアラームタグ値
時間文字列	STRING によるタイムスタンプ、つまり読取り可能な日付フォーマット
Msg text	読取り可能な STRING によるアラーム
PLC	アラームの位置の特定(関係のある PLC)

## タイムスタンプ 10 進数値の変換

この値を異なるプログラムで処理する必要がある場合、以下のように実行します。

1. Time\_ms を 1,000,000 で割ります。

例: 37986476928 : 1.000.000 = 37986,476928

2. 番号全体(37986)は、31.12.1899 から計算した日付です。

Excel では、タイムスタンプを日数に変換できます。これを実行するには、タイムスタンプを含むセルに、[日付]がグループ化するそれぞれのフォーマット形式を割り付けます。

結果: 37986 は、2003 年 12 月 31 日になります

3. (0,476928)のカンマの後の値は、時刻を表します。

– 値(0.476928)に 24 を掛けると、時間数(11.446272)が得られます。

– 余り(0.446272)に 60 を掛けると、分数(26.77632)が得られます。

– 余り(0.77632)に 60 を掛けると、秒数(46.5792)が得られます。

合計は、11:26:46.579 になります。

この変換は、例えば、Microsoft Excel でサポートされています。

## 6.4.8 ODBC ログデータベースへの直接アクセス

### はじめに

ログの保存先は、データベースまたはファイルが可能です。

データベースは、その"データソース名"(DSN)によってアドレスされます。Windows の[スタート]>[設定]>[コントロールパネル]>[ODBC データソース]を選択し、WinCC flexible で使用するデータベースを選択します。

ログデータを保存するには、構成設定を行うとき、ディレクトリ名の代わりに"データソース名"(DSN)を指定します。DSN を使用すると、データベースそれ自体とその保存先を参照していることとなります。

### 用途

ログデータの追加処理と評価に関して、データベースの機能範囲全体が使用可能です。

### 原理

ランタイムソフトウェアを収納しているコンピュータ上に、データベースに接続するデータソースを作成します。この後、WinCC flexible でログを作成するときに、上記の設定された DSN を指定します。

ODBC インターフェイスを使用して、MS SQL Server などの他のプログラムから、直接データベースにアクセスできます。

"StartProgram"システムファンクションを使用して、HMI デバイス上でプログラム呼び出しを作成することもできます。これは、ランタイムプログラムシーケンスを中断させません。



## 接続の操作

### 7.1 基本

#### 7.1.1 通信の基本

##### はじめに

2つの通信パートナー間でのデータの受け渡しは、通信として知られています。通信パートナーは、直接ケーブル接続またはネットワーク経由で接続できます。

##### 通信パートナー

通信パートナーとなることができるのは、ネットワーク上で他のノードとデータの通信および交換が可能なノードです。WinCC flexible の環境では、次のノードを通信パートナーとすることができます。

- 自動システムの中央モジュールおよび通信モジュール
- は PC での HMI デバイスおよび通信プロセスとすることができます。

通信パートナー間で転送されるデータは、さまざまな目的に使用できます。

- プロセス制御
- プロセスデータ獲得
- プロセス内の状態のレポート
- プロセスデータのログ記録

## 7.1.2 通信の原則

### はじめに

WinCC flexible はタグとエリアポイントを使って、HMI と PLC 間の通信を制御します。

### タグを使った通信

WinCC flexible では、タグは"タグ"エディタを使って集中的に管理されます。タグには、外部タグと内部タグがあります。外部タグは通信に使われ、PLC 上で定義済みのメモリロケーションのイメージを表します。HMI と PLC の両方には、このストレージロケーションへの読取りおよび書込みアクセス権があります。これらの読取りおよび書込み操作は、周期的な場合とイベントトリガの場合があります。

構成中に、特定の PLC アドレスを指すタグを作成します。HMI は定義済みのアドレスから値を読取り、それを表示します。オペレータは、関連 PLC アドレスに書き込まれる HMI デバイスに値を入力することもできます。

### エリアポイントを使用する通信

エリアポイントは特定のユーザーデータエリアのデータを交換するために使用されます。エリアポイントはパラメータフィールドです。WinCC flexible は、ランタイム時にこれらのパラメータフィールドから PLC のロケーションおよびデータエリアのサイズを受信します。通信中に、PLC と HMI デバイスは読取り操作および書込み操作のためにこれらのデータエリアに交互にアクセスします。データエリアに保存されたデータの評価に基づいて、PLC と HMI デバイスは定義されているアクションをトリガします。

WinCC flexible は次のエリアポイントを使用します：

- コントロール要求;コントロールヨウキユウ
- プロジェクト ID
- 画面番号
- データレコード;データレコード
- 日付/時刻
- 日付/時刻 PLC
- コーディネーション

さまざまなエリアポイントのうちのどれを使用できるかは、使用される HMI によって決まります。

### WinCC flexible とオートメーションシステム間の通信

WinCC flexible を使用する工業通信では、データはタグとエリアポイントを使って交換されます。データを取得するために、HMI は通信ドライバを使用するオートメーションシステムに要求メッセージを送信します。オートメーションシステム(AS)は、要求されたデータを応答フレームで HMI に返します。

## 通信ドライバ

通信ドライバは、自動システムと HMI デバイス間で接続を確立するソフトウェアコンポーネントです。そのため、通信ドライバによって、プロセス値とともに提供される WinCC flexible のタグを有効にできます。WinCC flexible は、さまざまな通信ドライバを持つさまざまなオートメーションシステムの相互接続に対応しています。

ユーザーは、各自固有の通信パートナーのインターフェース、プロファイルおよび転送速度を選択できます。

## HMI 間の通信

HMI 間の通信用に SIMATIC HMI HTTP プロトコルを使用できます。このプロトコルは "Sm@rtAccess" オプションの構成エレメントです。このプロトコルは、WinCC flexible ランタイムをインストールした PC および 270 シリーズのパネルで使用できます。詳細については、SIMATIC HMI HTTP プロトコル文書を参照してください。

## 均一でメーカーに依存しないインターフェース経由の通信

WinCC flexible は、OPC(OLE for Process Control)を使った均一でメーカーに依存しないソフトウェアインターフェースを提供します。このインターフェースにより、企業、オフィスおよび生産工場間での標準化されたデータ交換が可能になります。詳細については、OPC のドキュメントを参照してください。

## 7.2 エlementと基本設定

### 7.2.1 接続エディタ

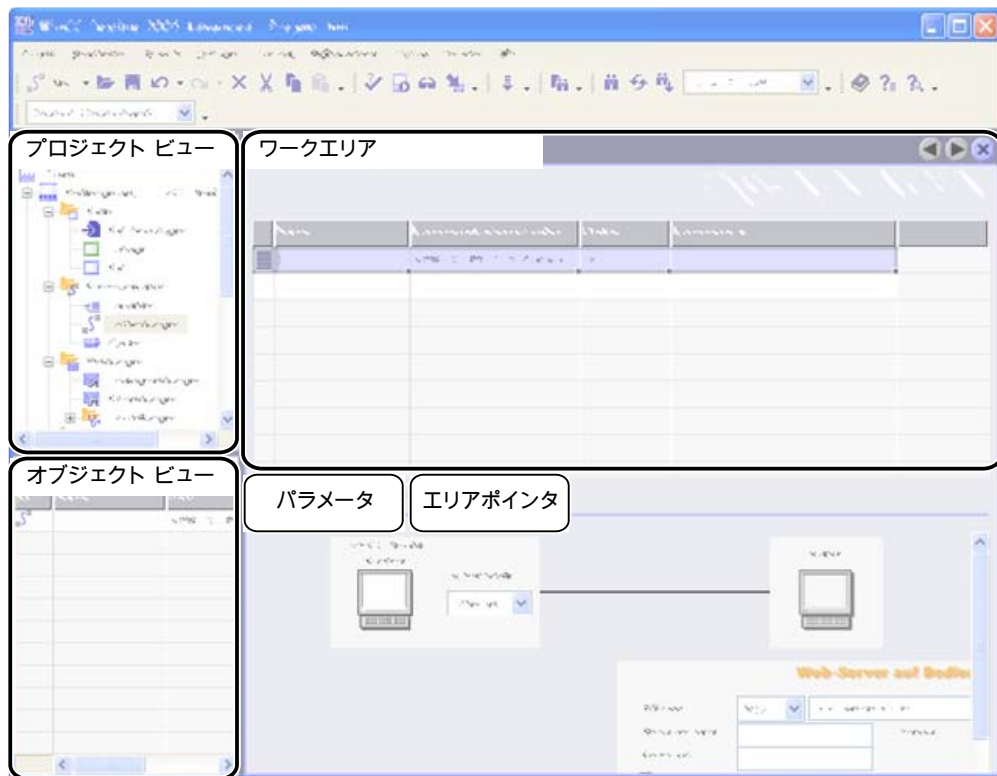
#### はじめに

"接続"エディタで接続を作成して、設定できます。

#### 開く

[プロジェクト]ウィンドウで[接続]を選択して、ショートカットメニューを開きます。このショートカットメニューで[新規接続]を選択します。新しい接続が作成され、作業エリア内で開かれます。

#### ストラクチャ




#### メニューバー

メニューバーには、WinCC flexible の操作に必要なすべてのコマンドがあります。使用可能なキーの組み合わせはメニューコマンドの隣に表示されます。

## ツールバー

このツールバーには、最も頻繁に使用するボタンが含まれています。

特定のツールバーを表示したり隠したりするには、[表示|ツールバー]を選択します。ツールバーのボタンは、そのツールバーの特定のボタンを表示したり隠したりするために使用できます。

## 作業エリア

すべての接続は、作業エリアにテーブル形式で表示されます。テーブルのセルから通信ドライバを選択して、関連する接続のプロパティを編集します。テーブルを列のエントリでソートするには、単純にその列ヘッダーをクリックします。

## [パラメータ]タブ

[パラメータ]タブを使用して、テーブルで選択した通信ドライバの設定を構成します。HMI、ネットワークおよびPLCの設定を選択します。

## [エリアポイント]タブ

[エリアポイント]タブを使用して、接続のエリアポイントを構成します。

## 7.2.2 接続のパラメータ

### はじめに

"接続"エディタの[パラメータ]タブを選択して、HMIと通信パートナー間の接続のプロパティを設定します。

### 構造体

通信パートナーは、[パラメータ]タブに図式的に表示されます。このタブには、[HMIデバイス]、[ネットワーク]および[PLC]エリアがあり、これらのエリアで使用される関連インターフェースのパラメータを宣言することができます。



システムはデフォルトのパラメータを設定します。パラメータを編集するときには常にネットワーク上で一貫性を保つようしてください。設定可能なパラメータの詳細については、サポートされているプロトコルの説明を参照してください。

### 7.2.3 接続のエリアポインタ

#### はじめに

"接続"エディタの[エリアポインタ]タブを使って、利用できるエリアポインタの使い方を設定できます。

#### 構造体

[エリアポインタ]タブには、エリアポインタの2つのテーブルが含まれます。[For all connections]テーブルには、プロジェクトで1回だけ作成され、1つの接続にだけ使用できるエリアポインタが含まれます。

[For each connection]テーブルには、利用できる接続ごとに別々に設定できるエリアポインタが含まれています。

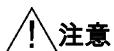
パラメータ		エリアポインタ					
すべての接続に対して							
接続	名前	アドレス	長さ	トリガモード	取得サイクル	コメント	
Connection_1	日付/時刻PLC	DB 1 DBW 0	6	サイクリック	1 min		
<未定義>	ユーザーバージョン		1	サイクリック	<未定義>		
<未定義>	画面番号		5	サイクリック	<未定義>		
各接続に対して							
有効化	名前	アドレス	長さ	トリガモード	取得サイクル	コメント	
オン	コーディネーション	DB 1 DBW 12	2	サイクリック	<未定義>		
オフ	デジタルボックス		5	サイクリック	<未定義>		
オフ	日付/時刻		6	サイクリック	<未定義>		
オフ	シフトメールボックス		4	サイクリック	<未定義>		

さまざまなエリアポインタのうちのどれを使用できるかは、使用される HMI デバイスによって決まります。エリアポインタとその設定の詳細については、サポートされているプロトコルの説明を参照してください。

## 7.3 接続とプロトコルの概要

### HMI の機能

HMI は、アラームおよびタグを読み、保存し、記録するために使用されます。さらに、HMI はプロセスに介入するために使用できます。



注意

#### Ethernet 通信

PROFINET IO、HTTP、Sm@rtAccess、Sm@rtService および OPC などの Ethernet での通信では、エンドユーザーがそのデータネットワークについての責任を負います。適正なデバイスの機能はいかなる状況においても保証できません。例えば、標的とした攻撃を受けると、デバイスに過負荷が発生することがあります。

### データ交換

操作機能とモニタ機能の前提条件として、HMI デバイスを PLC に接続します。HMI と PLC 間のデータ交換は、接続固有のプロトコルで制御されます。つまり、接続ごとに別のプロトコルが必要になります。

### 接続の選択規準

HMI と PLC の接続の選択基準の例は次のとおりです。

- PLC の種類
- PLC 内の CPU
- HMI デバイスタイプ
- PLC ごとの HMI デバイスの数
- 既存プラントの構造およびバスシステム
- その他の必要な構成エレメントの数量

## プロトコル;プロトコル

以下の PLC でプロトコルを使用できます。


PLC	プロトコル
SIMATIC S7	<ul style="list-style-type: none"> <li>• PPI</li> <li>• MPI <sup>1)</sup></li> <li>• PROFIBUS DP</li> <li>• TCP/IP (Ethernet)</li> </ul>
SIMATIC S5	<ul style="list-style-type: none"> <li>• AS 511</li> <li>• PROFIBUS DP</li> </ul>
SIMATIC 500/505	<ul style="list-style-type: none"> <li>• NITP</li> <li>• PROFIBUS DP</li> </ul>
SIMATIC HMI HTTP プロトコル	<ul style="list-style-type: none"> <li>• HTTP/HTTPS (Ethernet)</li> </ul>
SIMOTION	<ul style="list-style-type: none"> <li>• MPI</li> <li>• PROFIBUS DP</li> <li>• TCP/IP (Ethernet)</li> </ul>
OPC	<ul style="list-style-type: none"> <li>• DCOM</li> </ul>
Allen-Bradley	<p>PLC シリーズ SLC500、SLC501、SLC502、SLC503、SLC504、SLC505、MicroLogix および PLC5/11、PLC5/20、PLC5/30、PLC5/40、PLC5/60、PLC5/80</p> <ul style="list-style-type: none"> <li>• DF1<sup>2)</sup></li> <li>• DH+、KF2 モジュール経由 <sup>3)</sup></li> <li>• DH485、KF3 モジュール経由 <sup>4)</sup></li> <li>• DH485 <sup>4)</sup></li> </ul> <p>ControlLogix 5500 (1756-ENBT 使用)および CompactLogix 5300 (1769-L32E および 1769-L35E)PLC シリーズ</p> <ul style="list-style-type: none"> <li>• Ethernet</li> </ul>
GE Fanuc Automation	<p>SPS シリーズ 90-30、90-70 VersaMax Micro</p> <ul style="list-style-type: none"> <li>• SNP</li> </ul>
LG Industrial Systems (Lucky Goldstar) / IMO	<p>PLC シリーズ GLOFA GM(GM4、GM6 および GM7)/シリーズ G4、G6 および G7</p> <ul style="list-style-type: none"> <li>• 専用通信</li> </ul>
三菱電機	<p>PLC シリーズ MELSEC FX および MELSEC FX0</p> <ul style="list-style-type: none"> <li>• FX (Mitsubishi PG)</li> </ul>
三菱電機	<p>PLC シリーズ MELSEC FX0、FX1n、FX2n、AnA、AnN、AnS、AnU、QnA および QnAS</p> <ul style="list-style-type: none"> <li>• プロトコル 4</li> </ul>
OMRON	<p>PLC シリーズ SYSMAC C、SYSMAC CV、SYSMAC CS1、SYSMAC alpha、CJ および CP</p> <ul style="list-style-type: none"> <li>• Hostlink/Multilink (SYSMAC Way)</li> </ul>
Modicon (Schneider Automation)	<p>PLC シリーズ Modicon 984、TSX Quantum および TSX Compact</p> <ul style="list-style-type: none"> <li>• Modbus RTU</li> </ul> <p>SPS シリーズ Quantum、Momentum、Preimum、および Micro SPS シリーズ Compact および 984、イーサネットブリッジ経由</p> <ul style="list-style-type: none"> <li>• Modbus TCP/IP (Ethernet)</li> </ul>

PLC	プロトコル
Telemecanique	P47 411 付き PLC シリーズ TSX 7、P47/67/87/107 420 付き TSX 7、P47/67/87/107 425 付き TSX 7、特定 TSX 7 CPU 搭載 TSX SCM 21.6 モジュール、SCG 1161 モジュール付き TSX 17、TSX 37 (Micro)、TSX 57 (Premium) <ul style="list-style-type: none"> <li>• Uni-Telway</li> </ul>

- 1) S7-212 と接続している場合は不可
- 2) コントローラ SLC503、SLC504、SLC505、PLC5、MicroLogix に適用
- 3) コントローラ SLC504、PLC5 DF1 経由に適用
- 4) コントローラ SLC500 ~ SLC 505 および MicroLogix に適用

## 7.4 Ethernet 接続

### ポートアドレス

 <b>注意</b>
<b>Ethernet 通信</b>
PROFINET IO、HTTP、Sm@rtAccess、Sm@rtService および OPC などの Ethernet での通信では、エンドユーザーがそのデータネットワークについての責任を負います。すべての状況において、適正なデバイスの機能が保証されているわけではありません。例えば、意図的な攻撃を受けると、デバイスに過負荷が発生することがあります。

<b>通知</b>
ローカルおよびリモートポート設定が同一であることを確認します。

Ethernet 経由で接続する場合、たとえばファイアウォールやルーターの設定などのため、ポートアドレスが必要になります。

	ポートアドレス TCP/IP	ポートアドレス UDP
S7 通信	102	
HTTP (通信、転送)	80	
HTTPS (通信、転送)	443	
SmartServer (ブラウザコントロール経由)	80, 5800, 5900	
SmartServer (SmartViewer 経由)	5900	
SmartServer (Internet Explorer 経由)	5800, 59	
MiniWeb (ブラウザコントロール経由の HTML ページ)	80 HTTP 443 HTTPS	
MiniWeb (Internet Explorer 経由の HTML ページ)	80 HTTP 443 HTTPS	
MiniWeb (内蔵の Pocket Explorer を使用した HTML ページ)	80 HTTP 443 HTTPS	
WebServices (SOAP)	80 HTTP 443 HTTPS	
OPC-XML (OPC マスタとして CE)	80 HTTP 443 HTTPS	
OPC DCOM		
SendEmail	25	
転送(Ethernet、CE-Stub、PC Loader、PC 経由)	2308 代替案 50523	
ロギング(Ethernet 経由) CSV ファイル	139, 445	137, 138
ロギング(Ethernet 経由)データベース		

## 7.5 HMI の SNMP および MIB

### SNMP

SNMP (Simple Network Management Protocol)とは、ネットワークコンポーネントまたは HMI デバイスなどの端末デバイスの監視に使用するインターネット標準プロトコルです。SNMP は、TCP/IP プロトコルの一部で、クライアント/サーバーモデルにしたがって動作します。

SNMP は IETF (Internet Engineering Task Force)によって開発されました。HMI デバイスはバージョン 1(SNMPv1)およびコミュニティベースの SNMP バージョン 2 (SNMPv2c)をサポートしています。

HMI デバイスには SNMP エージェントが含まれています。SNMP エージェントによりデバイスの設定に関する情報を提供しています。この情報は、データ構造である MIB (Management Information Base)で管理されています。

### MIB

MIB は異なる SNMP タグで構成される、標準化されたデータ構造です。このデバイスでは、MIB II (RFC1213)を使用します。

HMI デバイス上の SNMP タグの読み込みおよび書き込みのため、"public"コミュニティがサポートされています。

デバイス名に関する注意点:

MIB II で使用されているデバイス名は PROFINET IO 名ではなく、ネットワークで使用されているデバイス名です。

デバイス名は、コントロールパネルの[通信]ダイアログに入力します。15 文字以内にする必要があります。SNMP 経由で MIB II データベースに 15 文字以上の名前を入力すると、そのデバイス名はデバイスのコントロールパネルで"WinCE"にリセットされます。

## レシピ管理システムの構造

### 8.1 基本

#### 8.1.1 レシピの基本原理

##### はじめに

レシピとは、グループになっているデータのコレクションです(マシンのパラメータ設定や生産データなど)。

例：

- 生産を別の製品型に変換するのに必要なマシンのパラメータ設定。
- 結果として別の最終製品の構成要素となる製品コンポーネント。

レシピでは、データ構造が固定されています。レシピの構造は、設定段階で定義されます。レシピにはレシピデータレコードが含まれています。各レシピデータレコードの値は異なりますが、構造は同じです。レシピは、HMI デバイスまたは外部記憶媒体に格納されます。たとえば、サーバーのデータベースに生産データが保存されている場合、ランタイム時に CSV ファイルとしてこの生産データをインポートできます。

レシピデータレコードは通常、HMI デバイスと PLC の間で 1 回のステップで完全に転送されます。

##### レシピの使用

レシピは以下の状況で使用できます。

- 手動生産  
必要なレシピデータを選択して、HMI デバイスで表示します。必要に応じてレシピデータを修正して、HMI デバイスに保存します。PLC にレシピデータを転送します。
- 自動生産  
コントロールプログラムが、PLC と HMI デバイス間のレシピの転送を開始します。HMI デバイスからの転送を自分で開始することもできます。その後、生産が自動的に実行されます。データの表示や修正は必須ではありません。
- ティーチンモード  
システム上で手動で最適化された生産データを最適化します(軸の位置や充てん量など)。このようにして計算した値は HMI デバイスに転送され、レシピデータレコードに保存されます。後日、保存したレシピデータを PLC に転送して戻すことができます。

## レシピの表示

レシピは、以下の方法で HMI デバイスで表示したり編集したりすることができます。

- プロセスピクチャ内の[レシピ]ウィンドウ
- [レシピ]画面

## レシピデータの入力と修正

データはそれぞれのレシピデータレコードに入力して、必要に応じて修正します。以下のオプションが使用できます。

- 設定中のデータ入力  
生産データがすでに存在している場合、レシピ設定中に、このデータを"レシピ"エディタに入力できます。
- ランタイム中のデータ入力  
生産データを頻繁に修正する必要がある場合、ランタイム時に直接入力を行うことができます。
  - データを直接 HMI デバイスで入力します。
  - マシン上で直接パラメータを設定します。その後、データを PLC から HMI デバイスに転送して、レシピに保存します。

### 8.1.2 レシピの使用例

レシピは、たとえば製造業や機械工学で使用されます。2つの例を使用して、WinCC flexible エンジニアリングシステムのレシピ機能で実行できる標準的なアプリケーションについて説明します。

- マシンパラメータ割り付け  
レシピ適用の一例として、製造業でのマシンパラメータの割り付けがあります。マシンは、各種サイズの木板を指定の寸法に切断し、ドリルで穴を開けます。ガイドレールやドリルは、板のサイズに合わせて新しい位置に移動させる必要があります。必要な位置データは、レシピ内にデータレコードとして保存されます。たとえば、板のサイズを新たに処理する場合、"ティーチイン"モードを使用してマシンパラメータを再度割り付けます。この新規位置データを PLC から HMI デバイスに直接転送して、新規データレコードとして保存します。
- ロット生産  
レシピ適用のもう一つの例として、食品加工業でのロット生産があります。フルーツジュースプラント内の充てん施設では、各種風味のジュース、ネクター、およびフルーツドリンクが生産されます。成分はすべて同じで、混合比率だけが異なります。各風味は、各レシピに対応します。各混合比率は、各データレコードに対応します。ボタンに触れると、混合比率に関する必要データをすべてマシンコントロールに転送できます。

### 8.1.3 レシピの構造

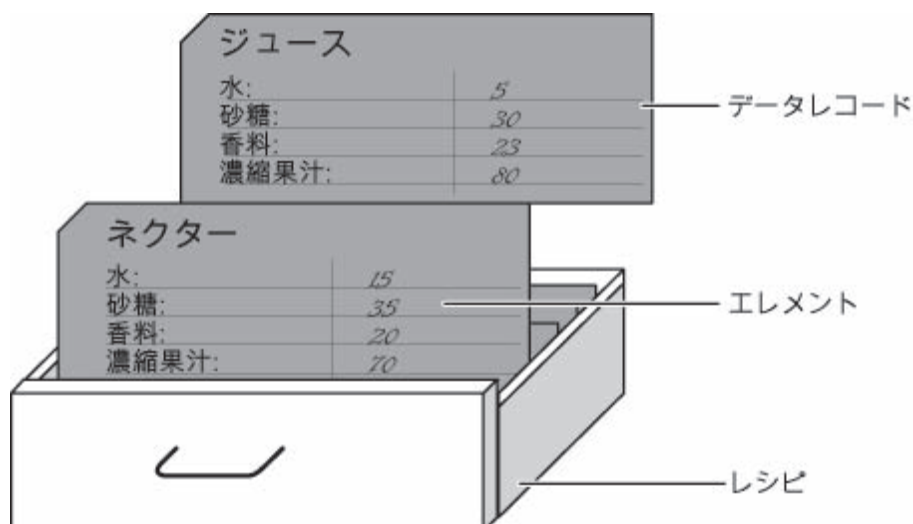
#### はじめに

フルーツジュースプラント内の充てん施設に関するレシピの基本構造について説明します。HMI デバイス内に複数の異なるレシピがあることがあります。レシピは、複数のインデックスカードを格納しているインデックスカードボックスと比較できます。インデックスカードボックスには、製品ファミリを製造するための複数の型が格納されています。それぞれの製造型についての完全なデータは、単一のインデックスカードにあります。

例:

ソフトドリンクの製造プラントでは、さまざまな風味を出すためにレシピが必要です。ドリンクの型には、フルーツジュースドリンク、ジュース、ネクターなどがあります。

#### レシピ



#### レシピデータレコード

各インデックスカードは、1つの製品型を製造するのに必要なレシピデータレコードに対応しています。

## レシピエントリ

引き出しにある各インデックスカードは、すぐに印刷できます。すべてのインデックスカードには、さまざまな成分のためのフィールドが含まれています。各フィールドは、各レシピエントリに対応します。したがって、レシピのすべてのレコードに同じエントリが含まれます。ただし、レコードの個々のエントリの値は異なります。

例:

すべてのドリンクには、同じ要素が含まれています：

- 水
- 濃縮果汁
- 砂糖
- フレーバ

ジュースドリンク、フルーツジュース、ネクターのレコードは、製造に使用される砂糖の量が異なります。

### 8.1.4 レシピの表示

#### はじめに

レシピは以下の方法で表示できます。

- [レシピ]ウィンドウ
- レシピ画面;レシピガメン

#### [レシピ]ウィンドウおよび[レシピ]画面での入力

[レシピ]画面または[レシピ]ウィンドウでレシピの値を変更することで、製造プロセスやマシンを変更することができます。

[レシピ]ウィンドウおよび[レシピ]画面は、レシピを使用する際に同じ機能を実行できます。この両者には、以下の点で違いがあります。

- 表示オプション
- 操作
- PLC と HMI デバイス間のデータ転送の可否

#### [レシピ]ウィンドウ

[レシピ]ウィンドウは、単純なレシピの表示に適しています。

[レシピ]ウィンドウは、レシピデータレコードの管理を目的とした、そのまま使用できる WinCC flexible 画面オブジェクトです。[レシピ]ウィンドウは、必ずプロセスピクチャの一部となります。[レシピ]ウィンドウには、表形式でレシピデータレコードが表示されます。表示や実行可能な操作は、個々のニーズに合わせて変更できます。

[レシピ]ウィンドウでプロジェクトのレシピを編集する場合、値はレシピデータレコードに保存されます。関連するオペレーティングエレメントを使用するまで、値は HMI デバイスと PLC 間で転送されません。

## [レシピ]画面

[レシピ]画面は個々のプラントの表示であり、以下によって構成されています。

- レシピ変数用の入力フィールド
- レシピを使用するためのコントロールオブジェクト("SaveDataRecord"など)

[レシピ]画面は、以下の状況に適しています。

- 大規模なレシピ
- 関連するプラントユニットのグラフィック表示へのレシピフィールドの割り付け
- 複数のプロセスピクチャへのレシピデータの分割

---

### 注記

設定によっては、レシピ変数の値は、以下のタイミングで PLC と[レシピ]画面間で転送されます。

- 修正直後
  - 関連するオペレーティングエレメントが使用されたとき
- 

## [レシピ]ウィンドウと[レシピ]画面間での同期化

[レシピ]ウィンドウと[レシピ]画面でレシピを編集する場合に、ランタイム中の[レシピ]ウィンドウの表示値と関連付けられているタグに保存されている値との間に差異が生じることがあります。これを防ぐために、レシピデータレコード値をレシピタグの値と同期化する必要があります。

保存されているまたは同期化されている値は、必ず完全なレシピデータレコードになります。

---

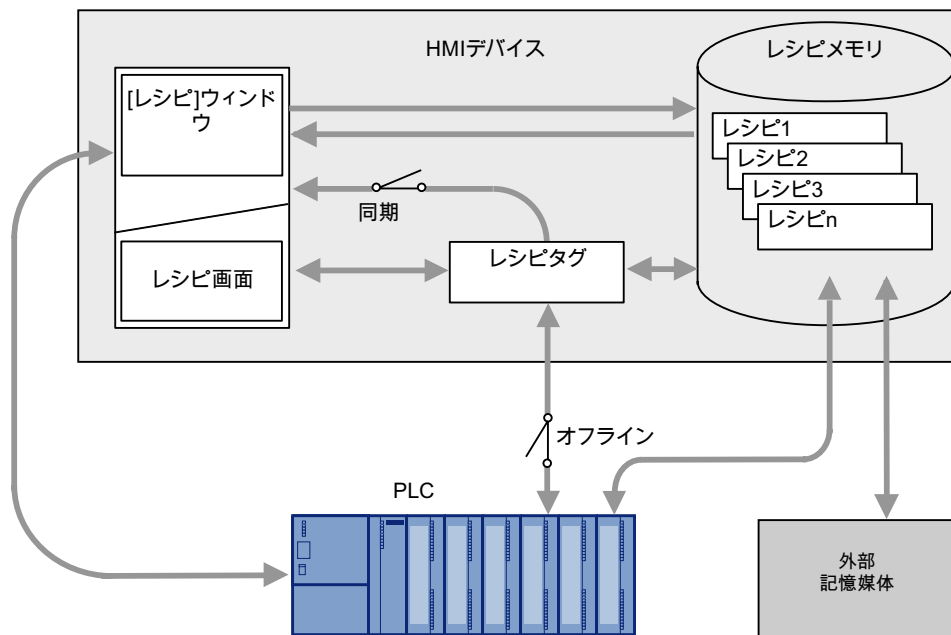
### 注記

レシピタグを同期化できるのは、詳細[レシピ]ウィンドウだけです。同期化は、このレシピに関する"タグの同期化"設定が有効になっている場合のみ、実行されます。

---

### 8.1.5 レシピデータレコードの転送

#### レシピにおけるデータフロー



#### コンポーネント間の相互作用

ランタイム中、以下のコンポーネント間で相互作用が行われます。

- [レシピ]ウィンドウと[レシピ]画面
 

HMI デバイスでは、レシピを[レシピ]ウィンドウまたは[レシピ]画面で表示して、編集します。

  - HMI デバイスの内部メモリにあるレシピデータレコードは、[レシピ]ウィンドウで表示して、編集します。
  - レシピタグ値は[レシピ]画面で表示して、編集します。

設定によっては、[レシピ]ウィンドウに表示される値は、レシピタグ値と同期化されます。
- HMI デバイスのレシピメモリ
 

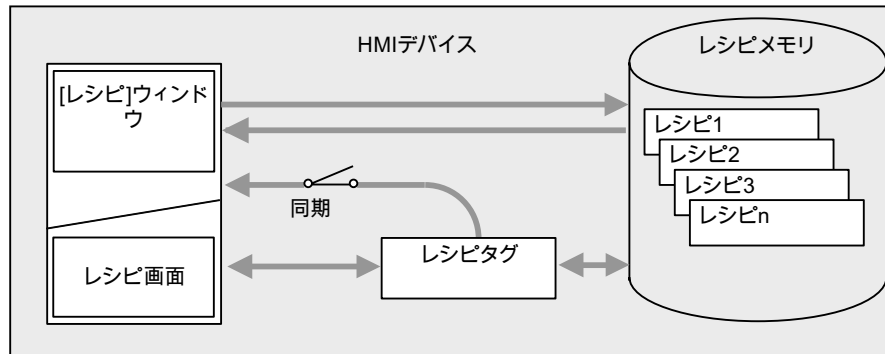
レシピは、HMI デバイスのレシピメモリにレシピデータレコード形式で保存されます。
- レシピタグ
 

レシピタグにはレシピデータが含まれます。[レシピ]画面でレシピを編集する時に、レシピ値がレシピタグに保存されます。 レシピタグ値が PLC と交換されるタイミングは、設定によって異なります。

#### 注記

レシピタグとレシピデータレコードは、両者に同じ値が保存されるように、同期化することができます。

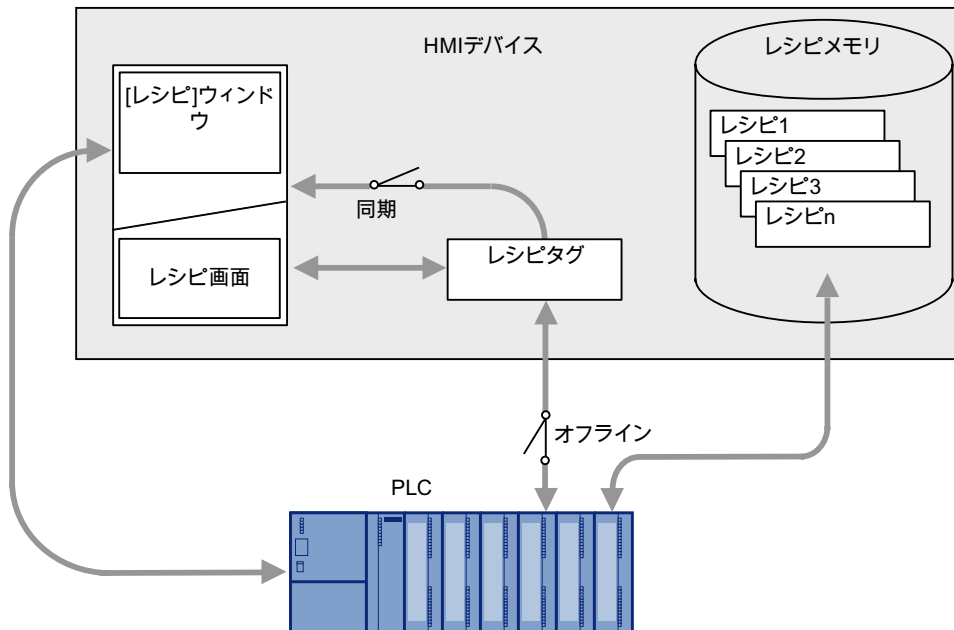
### レシピデータのロードと保存



完全なレシピデータレコードは、[レシピ]ウィンドウで HMI デバイス上のレシピメモリからロードしたり、レシピメモリに保存したりします。

レシピデータレコードの値は、[レシピ]画面でレシピメモリからレシピタグにロードされます。保存する際、レシピタグの値はレシピメモリ内のレシピデータレコードに保存されます。

### HMI デバイスと PLC の間のレシピ値のデータ転送



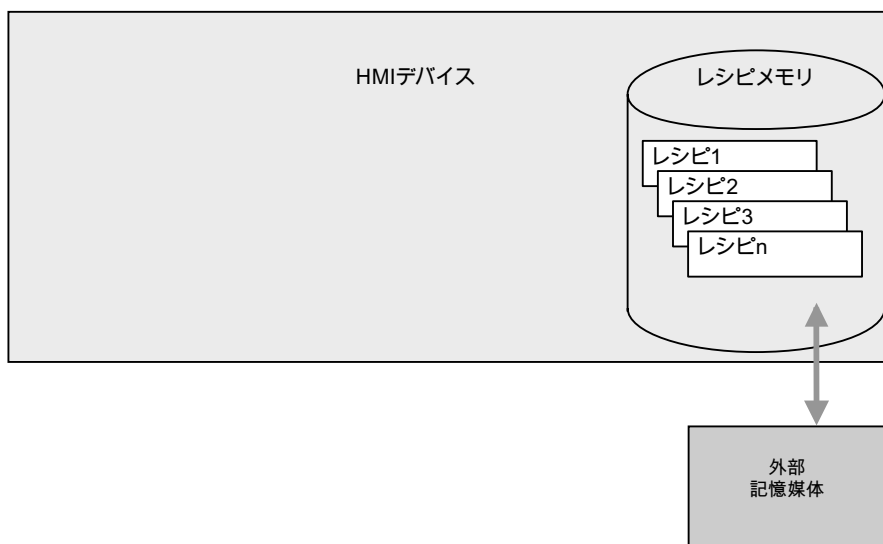
完全なレシピデータレコードは、[レシピ]ウィンドウと PLC 間で転送されます。

設定によっては、[レシピ]画面と PLC 間で以下の転送が可能です。

- PLC とレシピタグ間のレシピデータレコードの転送
- PLC とレシピタグ間の修正した個々の値の即時転送 これを行うには、レシピで以下の設定が必要になります。
  - [同期化タグ]が有効になっている。
  - [タグオフライン]が無効になっている。

レシピデータレコードは、HMI デバイスと PLC 間で直接転送することができます。このような状況では、HMI デバイス上での表示は必須ではありません。

### レシピデータレコードのエクスポートとインポート



レシピデータレコードは、HMI デバイスのレシピメモリからエクスポートされ、CSV ファイルで外部記憶媒体に保存されます。このレコードは記憶媒体からレシピメモリへ、再インポートできます。

HMI デバイスによっては、以下の外部記憶媒体が使用できます。

- メモリカード
- USB スティック
- ハードディスク

## 8.1.6 レシピのコンフィグレーション

### はじめに

レシピの設定は、使用目的によって異なります。

- [レシピ]ウィンドウでプロジェクトのレシピを編集する場合、値はレシピデータレコードにのみ保存されます。
- [レシピ]画面でプロジェクトのレシピを編集する場合、値はレシピタグに保存されます。

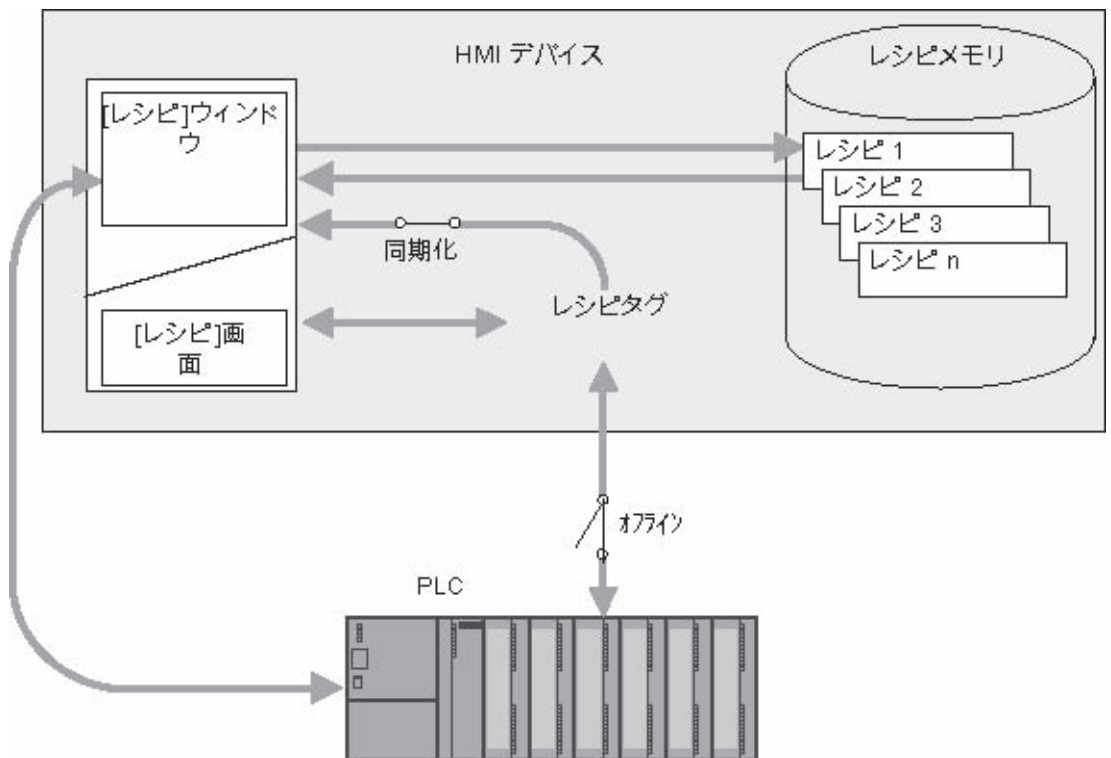
以下の設定が使用でき、これによってレシピデータレコード、レシピタグ、PLC のすべてでどのような相互作用が行われるのかを定義します。

#### [同期化タグ]無効

データレコードのデータは[レシピ]ウィンドウにのみ表示され、[レシピ]ウィンドウでのみ編集できます。[レシピ]ウィンドウ外でこれらの同じタグを使用しても、その値に影響を及ぼしません。

#### [同期化タグ]有効

[レシピ]ウィンドウと[レシピ]画面でレシピを編集する場合に、ランタイム中の[レシピ]ウィンドウの表示値と関連付けられているタグに保存されている値との間に差異が生じることがあります。これを防ぐために、レシピデータレコード値をレシピタグの値と同期化する必要があります。



---

**注記**

レシピタグを同期化できるのは、詳細[レシピ]ウィンドウだけです。

---

[レシピ]ウィンドウとそれに関連付けられたレシピタグの値は、自動的に同期化されません。レシピタグと[レシピ]ウィンドウは、"RecipeViewSynchronizeDataRecordWithTags"フアンクションによってオペレーティングエレメントを使用するまで同期化されません。

**[同期化タグ]および[タグオフライン]有効**

この設定では、変更されたレシピ値は、HMI デバイスの[レシピ]画面におけるレシピタグと PLC の間で即座には同期化されません。

値を同期化するためには、"SetDataRecordToPLC"フアンクションおよび "GetDataRecordFromPLC"フアンクションを持つオペレーティングエレメントが存在している必要があります。

レシピ値がコントローラで変更された場合、"GetDataRecordFromPLC"フアンクションを持つオペレーティングエレメントを使用すると、変更された値が[レシピ]画面に即座に表示されます。

**[同期化タグ]有効および[タグオフライン]無効**

この設定では、変更されたレシピ値は、HMI デバイスのレシピタグと PLC の間で即座に同期化されます。

[レシピ]画面でレシピ値を変更すると、PLC でこれらの変更が即座に適用され、プロセスに反映されます。

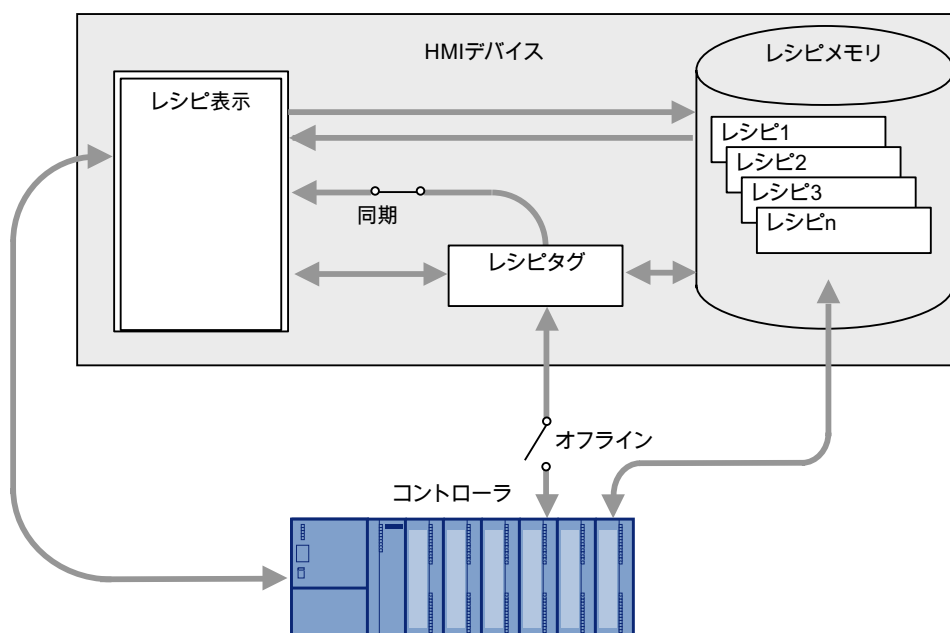
レシピ値が PLC で変更された場合、変更された値が、[レシピ]画面に即座に表示されます。

### 8.1.7 OP 77A および TP 177A 専用の機能

TP 177A と OP 77A HMI デバイスは、以下の点で他の HMI デバイスとは動作が異なります。

- 簡易[レシピ]ウィンドウのみがサポートされます。
- レシピのエクスポートやインポートはできません。外部記憶媒体は使用できません。

#### データフローの概要



#### コンポーネント間の相互作用

ランタイム中、以下のコンポーネント間で相互作用が行われます。

- [レシピ]ウィンドウ  
レシピは HMI デバイスの[レシピ]ウィンドウで表示して、編集します。  
HMI デバイスの内部メモリにあるレシピデータレコードは、[レシピ]ウィンドウで表示して、編集します。
- HMI デバイスのレシピメモリ  
レシピは、HMI デバイスのレシピメモリにレシピデータレコード形式で保存されます。
- レシピタグ  
レシピタグにはレシピデータが含まれます。

## OP 77A および TP 177A でレシピにアクセスできない

プロジェクトでレシピを変更し(たとえば、レシピエントリの削除または追加)、プロジェクトをレシピデータなしに HMI デバイスに転送すると、このレシピはデバイスで使用できなくなります。デバイスにレシピデータがすでに存在しても、アクセスできなくなります。

1. プロジェクト内のレシピに行われたすべての変更を元に戻します。
2. 再びプロジェクトを転送して、レシピデータをすべて除外します。

変更されたプロジェクトとレシピデータを HMI デバイスに転送することもできます。ただし、デバイスにあるレシピデータのすべてが失われ、復元できなくなります。

## OP 77A および TP 177A のレシピウィンドウの更新

レシピウィンドウで、更新の問題や、最新の値が表示されないという現象が発生することがあります。

レシピウィンドウでレシピデータレコードがプロセスされている間、ジョブ 69/70 を使ってレシピデータレコードの読み取り/書き込みをしないでください。

1. データレコードを編集する前に[レシピ]ウィンドウで押すことによって設定できるビットを設定します。
2. データレコードを編集した後で、[レシピ]ウィンドウでこのボタンを押してビットを再設定します。
3. このビットをコントロールプログラムで評価します。

これにより、コントロールジョブ 69 および 70 の実行が無効になります。

## レシピのタグ

OP 77A および TP 177A HMI デバイスで実装されているレシピタグは、ダイナミック I/O フィールドの設定や、I/O フィールドの表示などの、他の用途には使用できません。コンパイラはタグに複数のインスタンスがあればすべて報告します。

## レシピデータレコードの転送

"データレコードを上書きする"オプションが有効なときにプロジェクトが転送された場合、すべてのレシピデータレコードが HMI デバイスから削除され、転送されたレシピデータレコードで置換されます。

## [レシピ]ウィンドウ

- [レシピ]ウィンドウの空のフィールド  
簡易[レシピ]ウィンドウで長さ 1 のフィールドを指定して、レシピエレメントがこの長さを超えていると、ランタイムに空のフィールドが表示されます。
- 複数の[レシピ]ウィンドウを開く  
OP77A および TP177A で同じ画面内に複数の[レシピ]ウィンドウがあると、やり取りにはマイナスの影響があります。

このため、必ず 1 つの画面につき 1 つの[レシピ]ウィンドウを開くようにする必要があります。

## 8.1.8 PLC によるレシピデータレコードの同期化

### 概要

レシピデータレコードを HMI デバイスとコントローラ間で転送する場合、両方の通信ピアが互いの共通の通信エリアにアクセスします。

レシピデータレコードは、常に直接転送されます。タグの値は、設定されたアドレスに対して直接書き込みや読み取りが行われるため、クリップボードに一時保管されることはありません。

### データ転送タイプ

HMI デバイスと PLC 間でレシピデータレコードを転送する方法は 2 つあります。

- 非同期転送
- "データレコード"エリアポインタ"を介した同期転送

---

#### 注記

##### 同期転送

これを使用して、同期転送中にどちらかのコントロールプログラムによる偶発的なデータの上書きを防ぐことができます。

---

### 同期転送の必要条件

- [通信|接続]エディタで、必要な接続のために[データレコード]エリアポインタをセットアップする必要があります。
- HMI デバイスとデータレコードを同期転送する PLC が、"レシピ"エディタのレシピプロパティで指定されている必要があります。

### 同期転送

同期転送の場合は、PLC と HMI デバイスの両方で共有データコンパートメント内のステータスビットがセットされます。

同期データレコード転送は、たとえば次のような場合に便利な解決法となります。

- PLC が、レシピデータレコードを転送する"アクティブパートナー"になる場合。
- PLC が、レシピ番号とレシピ名だけでなく、レシピデータレコード番号と名前に関する情報を評価する場合。
- レシピデータレコードの転送は、以下の PLC ジョブによって開始されます。
  - "Set\_data\_record\_in\_PLC"
  - "Get\_data\_record\_from\_PLC"

## 8.2 エlementと基本設定

### 8.2.1 "レシピ"エディタ

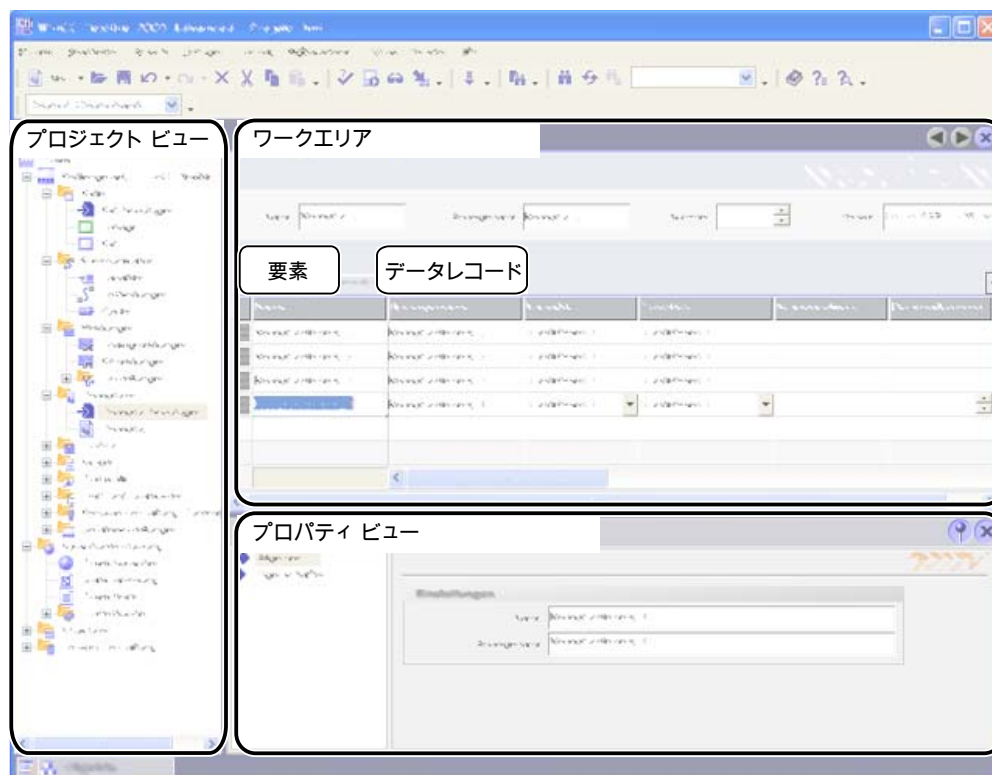
#### はじめに

レシピ、レシピエントリ、レシピデータレコードの作成、設定、および編集は、"レシピ"エディタで行います。さらに、"レシピ"エディタは、既存のレシピデータレコードへの値の入力にも使用することができます。

#### 開く

[プロジェクト]ウィンドウの[レシピ]グループで、[レシピの追加]または既存のレシピをダブルクリックします。

#### "レシピ"エディタの構造



#### 作業エリア

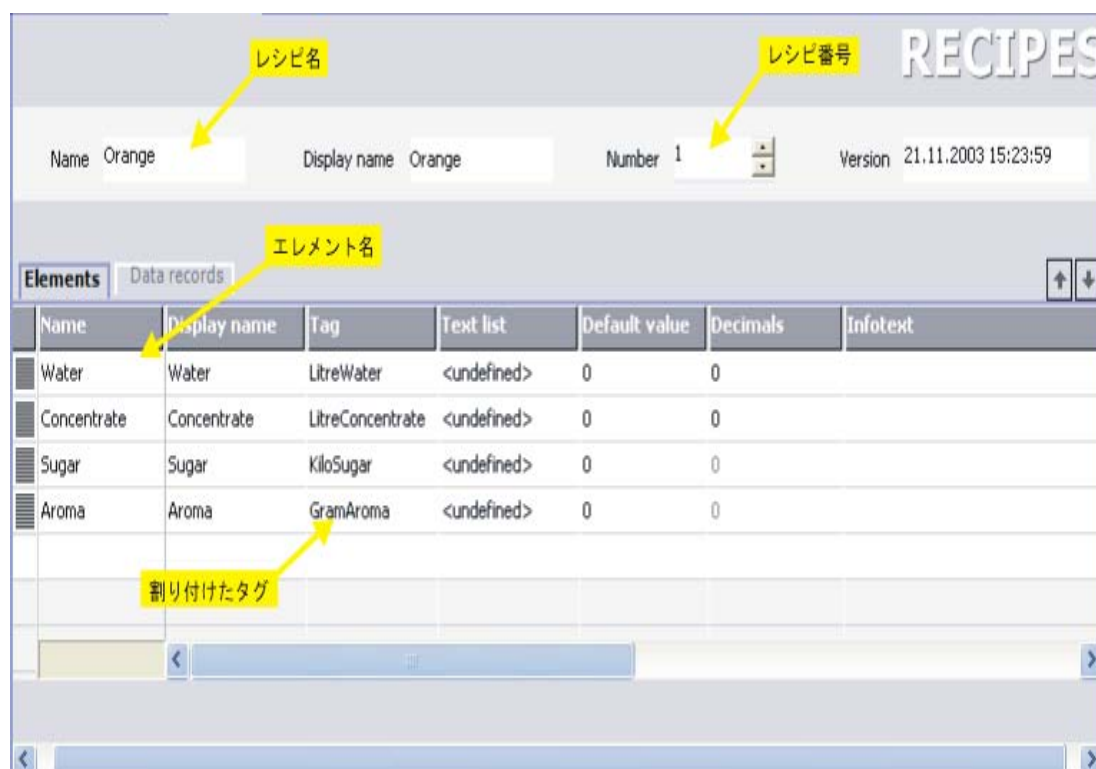
作業エリアは、レシピエントリやレシピデータレコードの作成および編集に使用します。レシピは、[エントリ]タブで定義します。[データレコード]タブを使用して、レシピデータレコードの値を定義します。

## [プロパティ]ウィンドウ

[プロパティ]ウィンドウでレシピをコンフィグレーションできます。レシピ設定の詳細については、「レシピ設定」を参照してください。

### 8.2.2 レシピエlement

#### [エlement]タブの構造



#### [エlement]タブのエlement

個々のレシピエントリについて、以下で簡単に説明します。

##### レシピ名

レシピは、レシピ名によって HMI デバイス内で一意に識別されます。

##### 表示名

レシピの表示名は、たとえばランタイム中に[レシピ]ウィンドウに表示されます。表示名は、複数の言語で設定できます。"FruitJuice\_Orange"のように、製品に直接関係する説明や意味を取り入れた名称を割り付けることもできます。

### レシピ番号

レシピは、レシピ番号によって HMI デバイス内で一意に識別されます。

### バージョン

バージョンによって、レシピに最新の変更が行われた日付と時間が識別されます。

### Element名

レシピエントリは、Element名によってレシピ内で一意に識別されます。"flavoring"といった、マシンまたは成分の軸となる説明や意味を取り入れた独自の名称を割り付けることができます。

### 割り付けタグ

各レシピエントリにはレシピタグが割り付けられ、ランタイム時には、ここにレシピデータレコードの値が保存されます。

### デフォルト値

デフォルト値は、レシピデータレコードを新規作成した時にデフォルトエントリとして使用されます。

### テキストリスト

テキストは、テキストリスト内の値または値の範囲に割り付けられます。たとえば、このテキストを出力フィールドに表示することができます。

レシピデータレコードでは、範囲で選択されるテキスト行のみが使用できます。

### 小数点

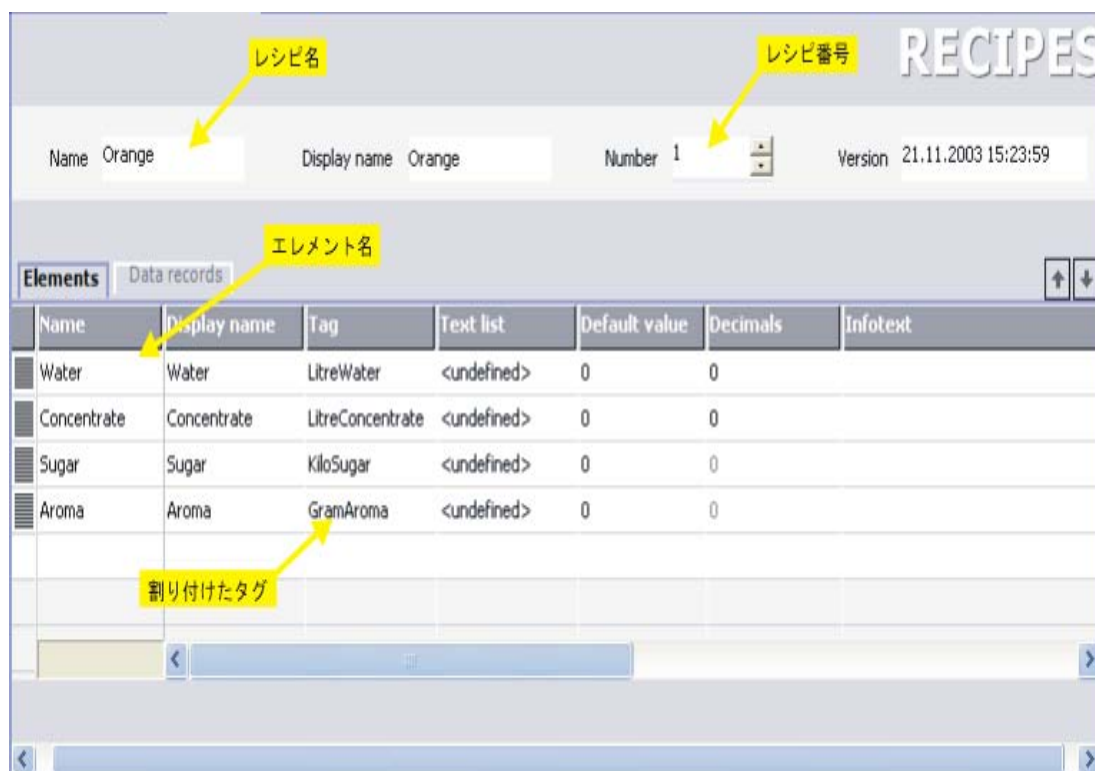
この数値は、ランタイム時にレシピデータレコードの値を小数点第何位まで表示するかを正確に定義します。

### 情報テキスト

レシピエントリに関するヘルプメッセージは[ヘルプテキスト]フィールドに入力できます。このメッセージは、ランタイム時にユーザーに対して表示されます。

## 8.2.3 レシピデータレコード

### [データレコード]タブの構造



### [データレコード]タブのエレメント

[データレコード]タブの個々のエントリについて、以下で説明します。

#### レシピデータレコード名

レシピデータレコードは、レシピデータレコード名によってレシピ内で一意に識別されます。

#### 表示名

レシピデータレコードの表示名は、たとえばランタイム時に[レシピ]ウィンドウに表示されます。表示名は、複数の言語で設定できます。製品番号のように、製品に直接関係する説明や意味を取り入れた名称を割り付けることもできます。

#### レシピデータレコード番号

レシピデータレコードは、レシピデータレコード番号によってレシピ内で一意に識別されます。

#### 入力値

設定時にレシピデータレコードに値を入力できます。プロジェクトがHMIデバイスに転送されると、レシピデータレコードも転送されます。HMIデバイスにある既存のデータレコードは、ユーザープロンプトが表示された後、転送の設定に従って上書きされます。

## コメント

レシピデータレコードに関するコメントを入力することができます。

## 8.2.4 レシピ設定

### はじめに

レシピのレシピ設定は[プロパティ]ウィンドウで入力します。

### レシピエントリおよびレシピデータレコードのプロパティ

[プロパティ]ウィンドウの表示は、"レシピ"エディタで行った選択によって異なります。[Element]タブまたは[データレコード]タブでレシピエントリまたはレシピデータレコードを編集する場合、[プロパティ]ウィンドウでもその内容を修正することができます。

例: レシピエントリのプロパティ



例: レシピデータレコードのプロパティ



## レシピのプロパティ

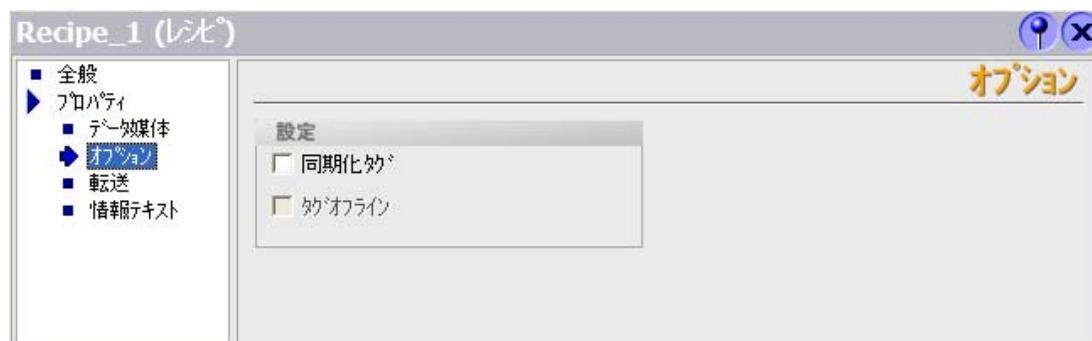
レシピ設定を修正するには、「レシピ」エディタで[レシピ名]または[レシピ番号]フィールドをクリックします。以下の設定は[プロパティ]ウィンドウで修正できます。

たとえば、[プロパティ]グループの[データ媒体]で、レシピデータレコードを格納しているファイルの保存先を定義します。選択機能は、使用しているオペレータパネルによって違ってきます。保存場所としてレシピメモリまたは HMI デバイスの外部記憶媒体を選択しますが、どちらを選択するかは HMI デバイスの設定によって異なります。オペレータパネルとして WinCC flexible Runtime を使用している場合、コンピュータのハードディスクドライブにファイルを保存します。パスを直接入力するか、ダイアログを使用してデータ媒体のディレクトリに移動します。



[プロパティ]グループでは、ランタイム時のレシピの動作を[設定]で設定します。[設定]で定義する内容は以下のとおりです。

- [レシピ]ウィンドウと[レシピ]画面間での同期化
- PLC と HMI デバイス間でのレシピタグの転送



[プロパティ]グループでは、PLC と HMI デバイス間でのレシピデータレコードの転送を同期で行うかどうかを[転送]で定義します。

### 注記

レシピデータレコードの転送を同期させる場合、選択した接続について[データレコード]エリアポイントをセットアップする必要があります。



## 8.3 ランタイムでのレシピの表示と編集

### 8.3.1 [レシピ]画面と[レシピ]ウィンドウ

HMI デバイスの[レシピ]ウィンドウまたは[レシピ]画面で、レシピを表示して、編集できます。

#### [レシピ]ウィンドウ

[レシピ]ウィンドウは、そのまま使用できる WinCC flexible の画面オブジェクトです。

[レシピ]ウィンドウは、以下のウィンドウで使用できます。

- 詳細[レシピ]ウィンドウとして
- 簡易[レシピ]ウィンドウとして

詳細[レシピ]ウィンドウのオペレーティングエレメントと、簡易[レシピ]ウィンドウで可能なオペレーションの両方を設定できます。

---

#### 注記

HMI デバイスでは、ディスプレイサイズが 6 インチ未満の簡易[レシピ]ウィンドウのみを使用することをお勧めします。

---

#### [レシピ]画面

[レシピ]画面はプロセスピクチャの一種です。レシピについての個々の入力画面によって構成されています。入力画面には IO フィールドなどの画面オブジェクトが含まれています。レシピデータレコードを保存する場合など、レシピファンクションはシステムファンクションによって実行されます。

---

#### 注記

##### [レシピ]画面

TP 170B 以上のモデルでは、レシピ画面を作成できます。

---

## 8.3.2 [レシピ]ウィンドウ

### [レシピ]ウィンドウ

[レシピ]ウィンドウは、レシピデータレコードの管理を目的とした、そのまま使用できる画面オブジェクトです。[レシピ]ウィンドウには、表形式でレシピデータレコードが表示されます。

[レシピ]ウィンドウは以下のように表示されます。

- 詳細[レシピ]ウィンドウとして
- 簡易[レシピ]ウィンドウとして

詳細[レシピ]ウィンドウのオペレーティングエレメントと、簡易[レシピ]ウィンドウで可能なオペレーションの両方を設定できます。

[レシピ]ウィンドウに表示または入力された値は、レシピデータレコードに保存されます。レシピデータレコードは、システムファンクションを介して PLC と交換されます。

### 詳細[レシピ]ウィンドウ

下図は、詳細[レシピ]ウィンドウの例を示しています。



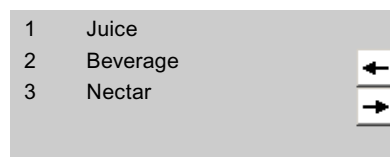
### 簡易[レシピ]ウィンドウ;カンイレシピウィンドウ

簡易[レシピ]ウィンドウは、3つのエリアで構成されています。

- レシピリスト
- データレコードリスト
- エレメントリスト

HMI デバイス上の簡易[レシピ]ウィンドウでは、各エリアが独立して表示されます。設定によっては、簡易[レシピ]ウィンドウはレシピリストから開始されます。

下図は、データレコードリストの例を示しています。



値の表示

**通知**

**バックグラウンドでのレシピデータレコードの変更**

次の場合に、レシピデータレコードの処理に適用されます。  
 対応するレシピデータレコードの値が制御ジョブによって変更されても、[レシピ]ウィンドウは自動的に更新されません。  
 [レシピ]ウィンドウを更新するには、個々のレシピデータレコードを再び選択します。

8.3.3 [レシピ]ウィンドウのコンフィグレーションオプション

はじめに

レシピ表示の動作と、レシピ表示の[プロパティ]ウィンドウに表示されるレシピを定義します。

一般設定

- レシピデータレコード値のみの表示  
 検査目的で、[レシピ]ウィンドウにレシピデータの表示のみを実行する場合、レシピデータレコードを編集できないように設定できます。[編集モードを有効にする]オプションを無効にします。



- 特定レシピの表示

[レシピ名]に定義したレシピのレシピデータレコードの表示のみが実行されます。  
Runtime にレシピ名を表示する場合は、[選択リストの表示]オプションを設定します。



- レシピまたはレシピデータレコードの番号または名前をタグに書き込みます(拡張レシピウィンドウのみ)。

[番号/名前用のタグ]で選択したレシピまたはレシピデータレコードの番号または名前がタグに保存されます。名前を保存する場合、文字列タイプのタグを指定する必要があります。たとえば、システムファンクションのパラメータとしてタグを転送できます。



## プロパティ

[プロパティ]ウィンドウで、オブジェクトの位置、形、スタイル、色、およびフォントタイプをカスタマイズします。以下の項目も定義できます。

- メニューおよびボタン

レシピウィンドウで使用できるメニューエントリとボタンを指定するには[ボタン]を選択します。[同期化タグ]ボタンは、詳細レシピウィンドウでのみ使用できます。

- [レシピ]ウィンドウのキャプション

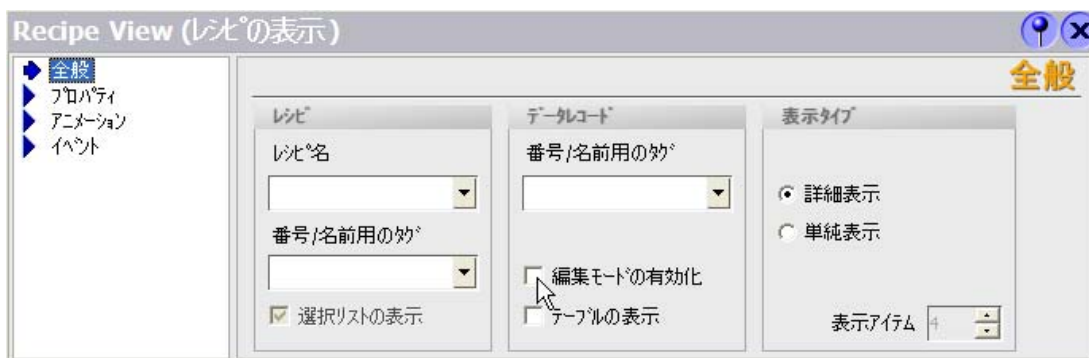
[キャプション]を選択して、詳細レシピウィンドウの列ヘッダーを指定します。

### 例: 選択リストとしてのレシピウィンドウの使用

以下のパラメータを設定すると、レシピ画面でのレシピおよびレシピデータレコード用の選択リストとして、レシピ表示を使用できます。

- [拡張ウィンドウ]: 有効
- レシピ用の[番号/名前用のタグ]: レシピ名用のタグ
- レシピデータレコード名の[番号/名前用のタグ]: レシピデータレコードの名前用のタグ
- [編集を許可]: 無効
- [テーブルの表示]: 無効
- [プロパティ]>[ボタン]: すべてのボタンが無効

プロセス画面には、レシピとレシピデータレコードを選択できる、2つの選択リストのみが表示されます。



### アニメーション

レシピウィンドウに、可視性、レイアウト、アニメーションなどのダイナミックプロパティを割り当てます。アニメーションを適切に使用します。

#### 注記

[簡易レシピウィンドウ]オブジェクトは、スクリプトを使ってダイナミックに操作することはできません。

たとえば、エンジニアリングシステムではオブジェクトの鮮明度を、[プロパティ]ウィンドウの[アニメーション]グループで、ダイナミックにコントロールできます。ボタンを設定しアニメーションを含めた後に、プロジェクトデータの整合性チェックを実行した場合などは、出力ウィンドウにエラーメッセージが出力されます。

### [レシピ]ウィンドウにおけるイベントの設定

ランタイム時に[レシピ]ウィンドウの選択や終了を行う場合、これは、処理されるファンクションリストをセットアップすることのできるイベントとなります。

#### 注記

該当するすべての[レシピ]ウィンドウのボタンが無効の場合、HMIデバイスの簡易[レシピ]ウィンドウのイベントは、ランタイムに使用するエンジニアリングシステムでのみ設定できます。

## 8.3.4 ランタイム中の[レシピ]ウィンドウの動作

### 画面切替え

[レシピ]ウィンドウから別の画面に切替えようとしたときにレシピデータの変更を保存していなかった場合は、レシピデータを保存するように要求されます。レシピ名、およびレシピデータレコードの名が表示され、まだ保存されていないレシピデータを示します。

ロードしたレシピデータを表示している[レシピ]ウィンドウを含むプロセス画面に切替えた場合は、レシピデータが自動的に更新されます。

### レシピデータレコードの作成、変更、コピーおよび削除

該当するレシピデータレコードがすでにある場合は、画面にシステムアラームが出力されます。

### ファンクションキーによる[レシピ]ウィンドウの操作

たとえば HMI デバイスにタッチ機能が付いていない場合、ファンクションキーを使用すれば[レシピ]ウィンドウを操作できます。"SaveDataRecord"などのファンクションを、HMI デバイスのファンクションキーに割り付けることができます。

### レシピデータのインポート後の表示

レシピデータのインポート中に[レシピ]ウィンドウを開くと、すでにインポートが完了したレシピデータだけが表示されます。[レシピ]ウィンドウは、データインポートごとに自動的に更新されるわけではありません。すべてのレシピデータを詳細表示するには、レシピデータのインポートが正常終了したことをシステムが知らせるまで、[レシピ]ウィンドウを開かないでください。または、インポート手順が正常終了した後に[レシピ]ウィンドウを更新します。

### レシピおよびレシピデータレコード用のタグの更新

現在のレシピデータレコードまたはその番号は、設定によっては、タグに保存することができます。タグは、以下の条件の下で更新されます。

- レシピデータレコードがロードされていること。
- レシピウィンドウの画面は、プロセス中は終了できません。

この操作には多少時間がかかります。

### 8.3.5 レシピ画面

#### はじめに

[レシピ]画面はプロセスピクチャの一種です。レシピについての個々の入力画面によって構成されています。入力画面には IO フィールドなどの画面オブジェクトが含まれています。レシピデータレコードを保存する場合など、レシピファンクションはシステムファンクションによって実行されます。

下図は、[レシピ]画面の例を示しています。

水	40	L	レシピ名	番号
濃縮果汁	70	L	オレンジ	1
砂糖	30	kg	データレコード名	番号
香料	30	L	ネクター	2

保存	PLCからのデータ
ロード	PLCへ

#### 注記

TP 170B 以上のモデルでは、レシピ画面を作成できます。

#### 原理

レシピ画面の設定時にカスタマイズすることができます。複数のプロセス画面にわたる大型レシピをトピックに従って開き、グラフィック画面オブジェクトなどの機能を使用して鮮明に表示することができます。

- トピックに応じた複数のプロセス画面へのレシピの分散
  - 複数のエントリのあるレシピデータを、複数のプロセス画像にわたって配布することができます。たとえば、各プラントセクションで、関連する入力画面の入ったプロセス画像をレシピデータレコード用に設定することができます。

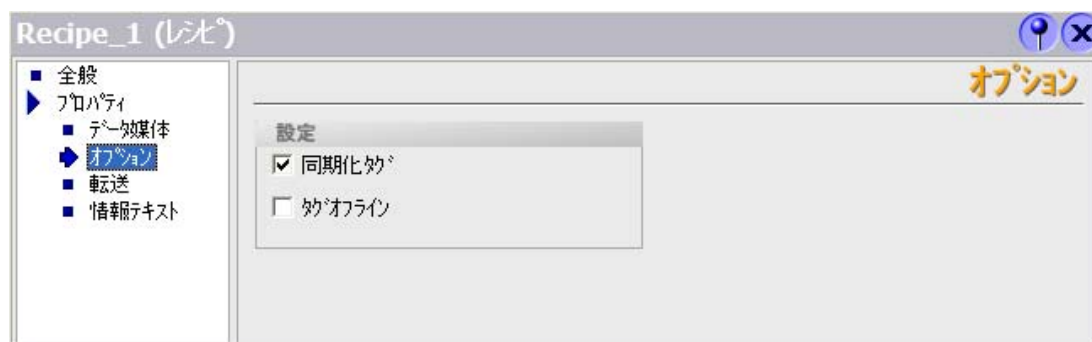
小型のディスプレイが装備された HMI デバイスでは、レシピを複数のプロセス画面に分割すると便利です。たとえば、ランタイム中にテーブルをスクロールしなくて済みます。

- 画像マシンシミュレーション
 

グラフィカルな画面オブジェクトを使って、使用中のマシンをプロセス画面でビジュアルにシミュレートすることができます。これにより、IO フィールドを軸やガイドレールなどのマシンエレメントのすぐ横に置いて、パラメータ設定をより鮮明に表示することができます。この機能を使用して、値とマシン間の直接的な参照を作成することができます。

## タグの同期化

[レシピ]ウィンドウ外の設定済み IO フィールドにレシピデータレコード値を入力できるようにするには、レシピプロパティの"タグの同期化"ファンクションを有効にする必要があります。WinCC flexible 2007 の設定は、以下の図のとおりです。



[レシピ]画面のタグと[レシピ]ウィンドウに表示されるレシピデータレコードの間でデータを同期化するには、タグを同期化する必要があります。タグを同期化できるのは、詳細[レシピ]ウィンドウだけです。

## タグオフライン

ランタイム中、入力した値を、接続した PLC にすぐに転送するには、[プロパティ]ウィンドウの[タグオフライン]を無効にする必要があります。

ランタイム中に、入力した値の直接転送を有効にしたり無効にしたりする場合、システムファンクション[レシピタグ設定]を設定します。

## システムファンクション

レシピ画面のオペレータ制御には、次のシステムファンクションを使用できます。

- ImportDataRecords
- ExportDataRecords
- LoadDataRecord
- SaveDataRecord
- SetDataRecordTagsToPLC
- GetDataRecordTagsFromPLC

レシピ画面で使用中、[レシピ]ウィンドウのオペレータ制御に次のシステムファンクションを使用できます。

- RecipeViewSaveDataRecord
- RecipeViewSaveAsDataRecord
- RecipeViewSynchronizeDataRecordWithTags
- RecipeViewDeleteDataRecord
- RecipeViewNewDataRecord
- RecipeViewGetDataRecordFromPLC
- RecipeViewRenameDataRecord ([単純なレシピ]ウィンドウのみ)
- RecipeViewShowOperatorNotes
- RecipeViewMenu ([単純なレシピ]ウィンドウのみ)
- RecipeViewOpen (単純な[レシピ]ウィンドウ専用)
- RecipeViewBack (単純な[レシピ]ウィンドウ専用)

レシピデータレコードとレシピのロード、保存、および転送に対応する各システムファンクションは、[レシピ]グループに配置されています。

### 8.3.6 [レシピ]ウィンドウのオペレータ入力

#### 適用

[レシピ]ウィンドウを使用して、データレコードの表示、編集、管理を行います。











#### 操作

コンフィグレーションによって、次のようなことができます。

- レシピデータレコードの作成、変更、コピーおよび削除
- 関連付けられたレシピタグによるレシピデータレコードの同期化
- レシピデータレコードの PLC からの読み取りまたは PLC への転送

## 操作エレメント

[レシピ]ウィンドウでは、次のオペレーティングエレメントを設定できます。

ボタン	ホットキー	機能
		設定済みのヘルプテキストが表示されます。
	<Ctrl+スペース>	新規レシピデータレコードを作成します。 開始値を設定すると、その値が入力フィールドに表示されます。
	<Ctrl+Enter>	レシピデータレコードの表示されている値を保存します。 プロジェクトで、保存先を事前に指定します。
	<Ctrl+*>	レシピデータレコードは、[レシピ]ウィンドウに関係なく、違う名前前で保存されます。名前を入力するダイアログボックスが開きます。
	<Ctrl+Del>	表示されたレシピデータレコードが削除されます。
	<Ctrl+=>	システムは、常に[更新レシピ]タグの値で[レシピ]ウィンドウの現在値を更新します。 [レシピ]ウィンドウに表示されている値が現在の[レシピ]タグの値より新しい場合、システムはこの値を[レシピ]タグに書込みます。 このファンクションは、事前にレシピプロパティの[同期化タグ]を有効にしていないと使用できません。
	<Ctrl+下矢印>	[レシピ]ウィンドウに表示されているレシピデータレコードの設定値が、PLC に転送されます。
	<Ctrl+上矢印>	PLC からのレシピ値が、[レシピ]ウィンドウに表示されます。

## [レシピ]ウィンドウのマウス制御またはタッチスクリーン制御

1. 使用するレシピを選択します。  
レシピのレコードが表示されます。
2. 編集するデータレコードをクリックします。
3. 実行したいファンクションが割り当てられたボタンを押します。

## [レシピ]ウィンドウでのキーボードの使用

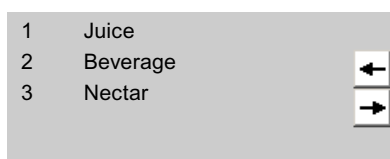
1. [レシピ]ウィンドウで、レシピが選択されるまで<Tab>キーを押します。
2. <Enter>キーを押します。  
レシピのドロップダウンリストボックスが開きます。
3. リストからレシピまたはレコードを選択します。上下左右の矢印カーソルキーを使用して、次または前のエントリに切り替えます。
4. 使用するオペレーティングエレメントが選択されるまで、<Tab>キーを押します。また、キーの組み合わせを使用して、[レシピ]ウィンドウを操作できます。

### 8.3.7 簡易[レシピ]ウィンドウのオペレータ入力

#### 表示モード

簡易[レシピ]ウィンドウは、3つのエリアで構成されています。

- レシピリスト
- データレコードリスト
- エlementリスト



簡易[レシピ]ウィンドウ - データレコードリストの例

HMI デバイス上の簡易[レシピ]ウィンドウでは、各エリアが独立して表示されます。ショートカットメニューを使用して、これらの表示エリアを個々に操作できます。

簡易[レシピ]ウィンドウは、必ずレシピリストから開始します。

#### 操作

設定によっては、以下の簡易[レシピ]ウィンドウが使用できます。

- レシピデータレコードの作成、変更、コピーおよび削除
- PLC からのレシピデータレコードの読み取りまたは PLC への転送


#### 簡易[レシピ]ウィンドウの使用

表示エリアとショートカットメニューとを切り替えて、簡易[レシピ]ウィンドウを操作します。


下の表は、表示エリアの操作を示しています。

ボタン	キー	機能
	<Enter>	下から 2 番目の表示エリア、つまりデータレコードリストまたはエlementリストが開きます。
←	<Esc>	前の表示エリアが開きます。
→	<右矢印>	表示エリアのショートカットメニューが開きます。
	<上矢印>/<下矢印>	前/次のエントリを選択します。
	<Page Up>/<Page Down>	表示を上/下のページに移動させます。
	<Home>/<End>	最初/最後のエントリを選択します。最初/最後のエントリが選択されます。

下表は、ショートカットメニューでの操作を示しています。

ボタン	キー	機能
	<Esc>	メニューが閉じます。 表示エリアが開きます。
	メニューコマンドの 番号を入力します	メニューコマンドが実行されます。

### 簡易[レシピ]ウィンドウのショートカットメニュー

各表示エリアで  ボタンを押すと、コマンドオプションの選択を呼び出すことができます。コマンドの選択には、カレントの表示エリアで使用可能なコマンドがリスト表示されます。番号が各コマンドに割り付けられます。この番号を入力すると、コマンドが実行されます。

- レシピリスト

メニューコマンド	機能
新規	選択されているレシピに、新規レシピデータレコードを作成します。 開始値を設定すると、入力フィールドに表示されます。
情報テキストの表示	簡易[レシピ]ウィンドウ用に設定された情報テキストが表示されます。
開く	選択されているレシピのレコードリストが開きます。

- データレコードリスト

メニューコマンド	機能
新規	新規レシピデータレコードを作成します。 開始値を設定すると、その値が入力フィールドに表示されます。
削除	表示されているレコードが削除されます。
名前を付けて保存	選択したデータレコードが、別の名前で作成されます。名前を入力するダイアログボックスが開きます。
名前の変更	選択されているレコードの名前が変更されます。ダイアログボックスが表示され、名前を入力することができます。
開く	選択したデータレコードのエレメントリストが開きます。
前へ	レシピリストが開きます。

- エレメントリスト

メニューコマンド	機能
保存	選択されているレコードの名前が変更されます。
PLC へ	選択されているレコードの表示値が、HMI デバイスから PLC に転送されます。
PLC から	PLC からのレシピ値が、HMI デバイスの[レシピ]ウィンドウに表示されます。
名前を付けて保存	新規の名前で、データレコードが保存されます。ダイアログボックスが表示され、名前を入力することができます。
	<ESC>キーを押して、データレコードリストを開きます。

以下のメニューコマンドは、TP 177A および OP 77A HMI デバイスでも設定できます。



- データレコードリスト

メニューコマンド	機能
PLC へ	選択したデータレコードの表示値が、HMI デバイスから PLC に転送されます。
PLC から	PLC からのレシピ値が、HMI デバイスの[レシピ]ウィンドウに表示されます。
ヘルプの表示	簡易[レシピ]ウィンドウの設定済みヘルプテキストを表示します。

- エレメントリスト

メニューコマンド	機能
ヘルプの表示	簡易[レシピ]ウィンドウの設定済みヘルプテキストを表示します。
名前の変更	選択したデータレコードの名前を変更します。ダイアログボックスが表示され、名前を入力することができます。
前へ	データレコードリストを開きます。

### 簡易[レシピ]ウィンドウのマウス制御またはタッチスクリーン制御

1. [レシピ]ウィンドウで目的のレシピを選択します。
2.  ボタンをクリックします。  
ショートカットメニューが開きます。
3. 目的のメニューコマンドを選択します。  
メニューコマンドが実行されます。
4. または、[レシピ]ウィンドウで目的のレシピを開きます。  
データレコードリストが表示されます。
5. 目的のデータレコードを開きます。または、 ボタンを使用してショートカットメニューを開き、メニューコマンドを選択します。  
メニューコマンドが実行されます。

### 簡易[レシピ]ウィンドウでのキーボードの使用

1. 簡易[レシピ]ウィンドウが選択されるまで、<Tab>キーを押します。
2. カーソルキーを使用して目的のレシピを選択します。
3. <右矢印>キーを押します。  
ショートカットメニューが開きます。
4. 目的のメニューコマンドが選択されるまで、<下矢印>キーを押します。
5. <Enter>キーを押してコマンドを確定します。
6. または、希望するメニューコマンドの番号を押します。  
メニューコマンドが実行されます。

## 8.3.8 レシピ構造の変更に対する応答

### はじめに

以下の場合に、異なるレシピ構造になる場合があります。

- 作動中の変更のイベント
- メーカーにより、マシン上で作業が行われた場合(retrofit)
- CSV ファイルをインポートした場合、CSV ファイルの構造がレシピの構造と異なる場合があります。

いずれにしても、作成したレシピデータレコードは使用できます。

<b>注意</b>
-----------

タグの名前を変更すると、その割り付けは失われます。
---------------------------

### 影響

構造の違いについては、以下のとおり対応します。

- 古いレシピデータレコードまたは CSV ファイルに追加の値が含まれている場合、これらの値は破棄されます。
- 古いレシピデータレコードまたは CSV ファイルに、間違ったデータタイプの値が含まれている場合、レシピデータレコードには、設定されているデフォルト値が使用されます。  
例: レシピデータレコードに、タンクの内容を示す値が格納されており、これらの値が浮動小数点数として入力されたとします。ただし、対応するレシピタグは整数値を要求します。この場合、システムは、転送した値を破棄し、設定したデフォルト値を使用します。
- 古いレシピデータレコードまたは CSV ファイルに含まれる値が少なすぎる場合、レシピデータレコードには、設定されているデフォルト値が再度使用されます。

## 8.4 シナリオ

### 8.4.1 シナリオ:ランタイム中のレシピデータレコードの入力

#### 目的

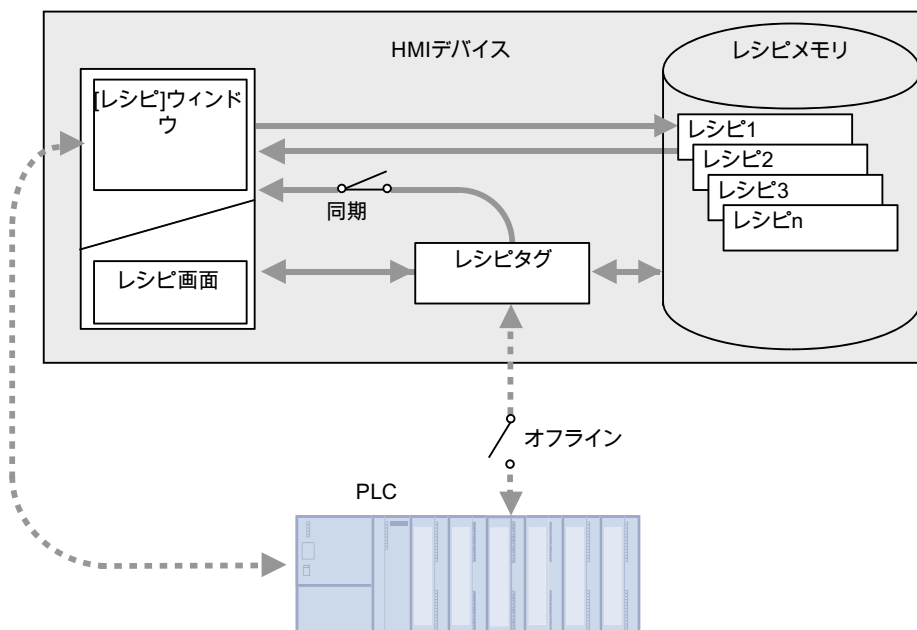
現在進行中のプロセスを妨げることなく、製造データを HMI デバイスに入力したい場合があります。このために、製造データは PLC に転送されないようになっています。

#### 必要条件

- レシピが作成されていること。レシピでは、以下の設定が使用されます。
  - [同期化タグ]が有効または無効。
  - [同期化タグ]が有効になっている場合、[タグオフライン]も有効になっている必要があります。

これにより、HMI デバイスと PLC 間でレシピタグが自動的に転送されるのを防ぎます。
- レシピ画面または[レシピ]ウィンドウの模倣が使用できます。
- レシピデータレコードを保存するオペレーティングエレメントがあります。

#### シーケンス



1. [レシピ]ウィンドウまたはレシピ画面に製品データを入力します。
2. 変更したレシピデータレコードを保存します。
3. またはレシピデータレコードを新しい名前で保存します。

レシピデータレコードは、HMI デバイスのレシピメモリに保存されます。

## PLC へのレシピデータレコードの転送

設定により、レシピデータを PLC に転送するオペレーティングエレメントが使用できる場合があります。

### 8.4.2 シナリオ:手動製造シーケンス

#### 目的

PLC に接続された読取りデバイスは、処理される加工品のバーコードを読取ります。レシピデータレコード名は、それぞれのバーコード名に対応しています。この結果、PLC は、HMI デバイスの記憶媒体から、必要なレシピデータレコードをロードできます。レシピデータレコードは検査のために画面に表示されます。

必要に応じて、転送された製造データをオンラインで修正できると便利な場合があります。

#### 必要条件

- レシピが作成されていること。レシピでは、以下の設定が使用されます。
  - [同期化タグ]が有効になっている。
  - [タグオフライン]が無効になっている。

---

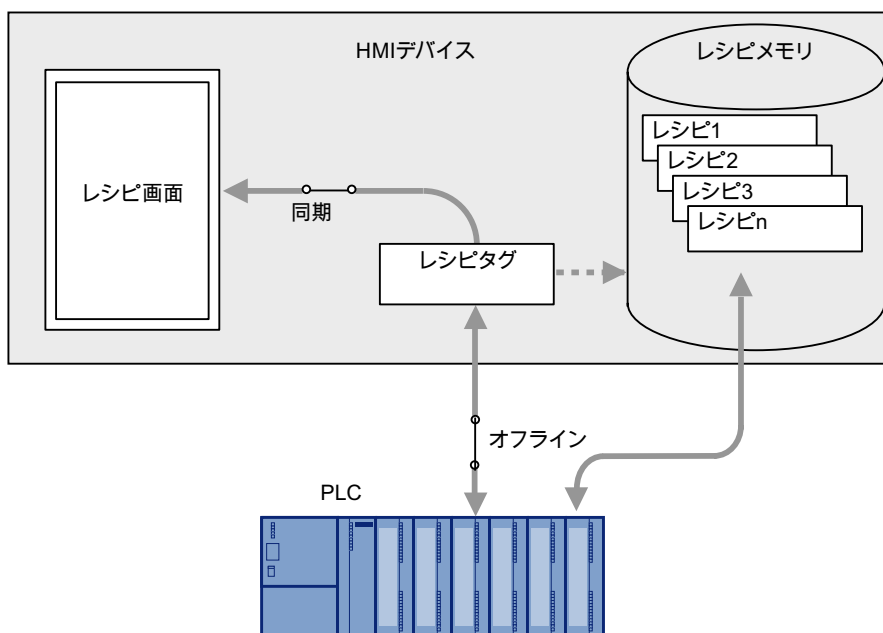
#### 注記

変更内容は PLC に直接転送されます。

---

- レシピ画面が使用できる。レシピ画面でレシピデータレコードを保存するオペレーティングエレメントが使用できる場合があります。

### シーケンス



#### [レシピ]画面使用時の動作

[レシピ]画面が使用されている場合、変更内容を直接転送することはできません。オペレーティングエレメントを使用してレシピデータレコードを PLC に転送する必要があります。

### 8.4.3 シナリオ:自動製造シーケンス

#### 目的

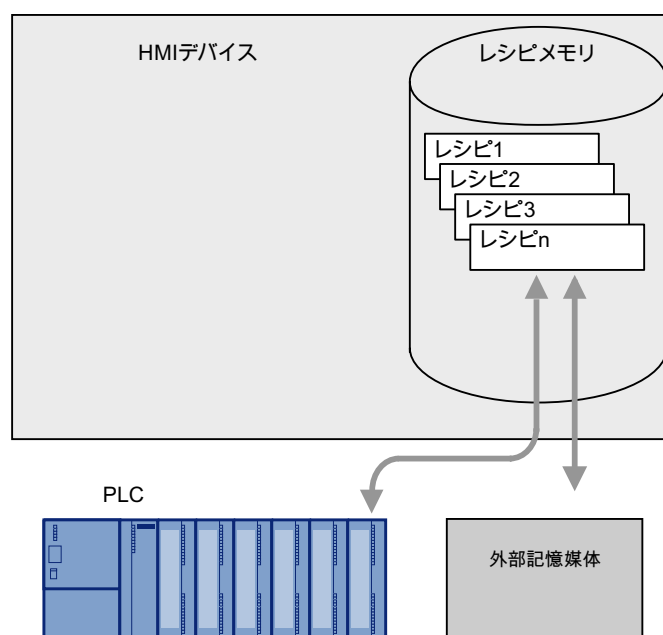
製造を自動的に実行します。製造データは、HMI デバイスのレシピメモリまたは外部記憶媒体から PLC に直接転送されます。画面表示は不要です。

#### 必要条件

- レシピが作成されていること。レシピでは、以下の設定が使用されます。
  - [同期化]は[転送]で有効にします。

製造データは PLC に転送されるため、データが誤って上書きされることを防ぐために、PLC と同期化することが必要です。

#### シーケンス



## 実装

以下の方法で、データの流れを管理できます。

- 管理プログラムにより、制御ジョブ、または必要に応じて WinCC flexible システムファンクションを使用して自動転送を管理します。

シーケンスは、メールボックスのステータス情報および使用されたファンクションからの戻り値により管理されています。

- 1つまたは複数のスクリプトによって、WinCC flexible システムファンクションでの自動転送を管理します。

シーケンスは、使用されたファンクションの戻り値を使って確認することができます。

使用可能なシステムファンクションを使って、自動製造シーケンスを実現することができます。

- "ImportDataRecords"

このファンクションにより、データレコードを\*.CSV ファイルから HMI デバイスのレシピメモリにロードします。

- "SetDataRecordToPLC"

このファンクションにより、データレコードを HMI デバイスのレシピメモリから PLC に転送します。

## タグのロギングと表示

### 9.1 基本

#### 9.1.1 データロギングの基本原理

##### はじめに

データロギングは、工業用の装置からのプロセスデータをキャプチャ、処理、およびログに記録するために使用されます。

次に、収集されたプロセスデータは、装置のオペレーティング状態に関する重要な業務情報、技術情報を抽出するために分析されます。

##### データロギングのアプリケーション

データロギングを使用して、故障を分析したり、プロセスの実行を文書化できます。データログを分析することによって、保守サイクルを最適化して製品の品質を向上させ、品質基準を確実に満たすようにするために必要な情報を抽出できます。

#### 9.1.2 WinCC flexible でのデータロギング

##### はじめに

データとは、プロセス中に収集され、接続された幾つかのオートメーションシステムのうちの、1つのシステムのメモリに保存される情報です。このデータは、温度、占有レベル、状態(例: モータオフ)などの装置の状態を反映します。プロセス変数を使って作業するには、WinCC flexible でタグを定義する必要があります。

WinCC flexible では、外部タグを使用して、プロセス値を収集し、接続されたオートメーションシステムのメモリロケーションにアクセスします。内部タグはどのプロセスにも接続されていないので、それぞれの HMI デバイスでのみ使用可能です。

原理

外部タグおよび内部タグの値は、データログに保存できます。各タグが保存されるログを個別に指定することができます。

データロギングは、サイクルとイベントによって制御されます。ロギングサイクルを使用すると、タグ値を確実に連続して取得したり、保存することができます。さらに、値の変化などのイベントによってもデータロギングをトリガすることができます。これらの設定は、各タグに個別に行うことができます。

ランタイム中には、ログされるタグ値は ODBC データベースまたはファイル内に取り込まれ、処理、保存されます。

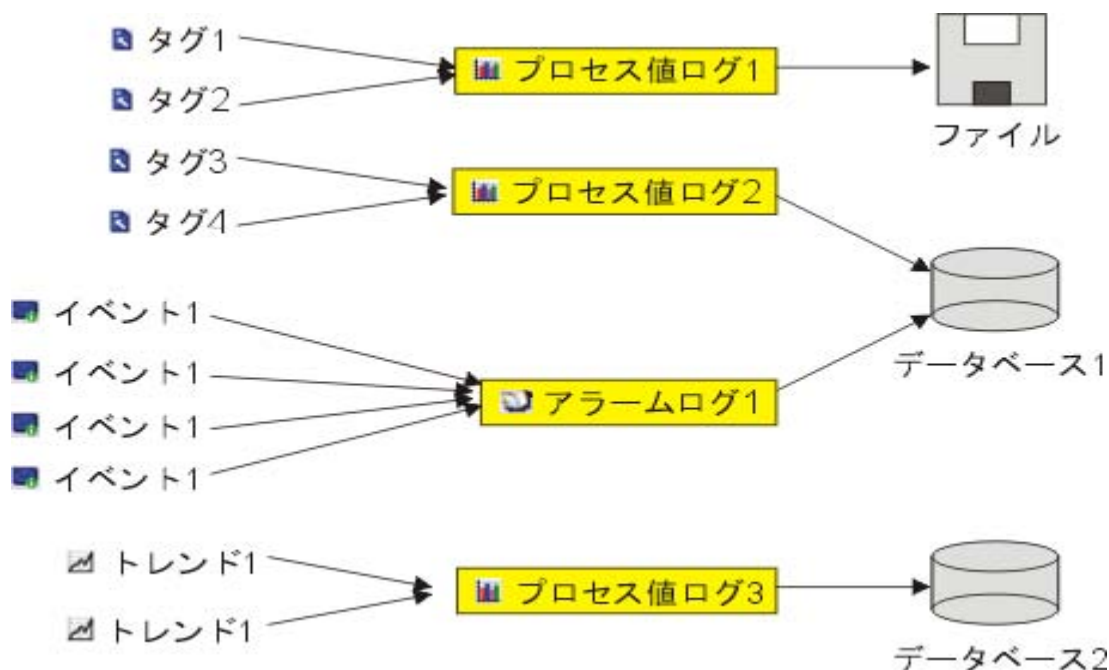
ログタイプ

WinCC flexible では、以下のログタイプから選択できます。

- サイクリックログ
- 一定期間のサイクリックログ
- いっぱいになったときにシステムアラームメッセージを送信するサークルログ
- いっぱいになったときにシステムファンクションを実行するサークルログ。

記憶媒体と保存場所

ログ済みのデータは、ODBC データベース(PC 上でのみ)かファイルに保存されます。



HMI デバイスのハードウェア設定によって、データは、ローカル(PC のハードディスク上またはパネルのストレージカードに)、またはネットワークドライブ(ある場合)にロギングされます。

以下の保存場所をのいずれか 1 つを選択します。

- データベース  
すべてのデータは ODBC データベースに保存されます。
- ファイル - CSV (ASCII)  
データは CSV ファイルに標準 ASCII フォーマットで保存されます。
- ファイル - RDB  
データはクイックアクセスできるようにリレーショナルデータベースに保存されます。  
保存済みのデータは、他のプログラムで追加の処理(例: 分析目的)を受けます。

## ログの内容の出力

ランタイム中に、ログ済みのタグ値をプロセス画面にトレンドとして出力することができます。

### 9.1.3 トレンド

#### はじめに

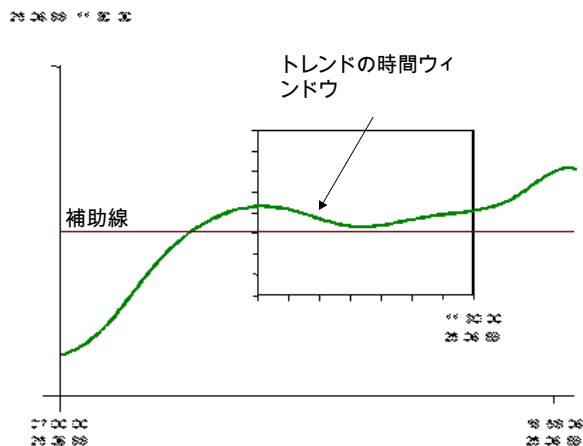
トレンドは、タグがランタイム中に取り値をグラフィックで表したものです。トレンドを表示するために、使用しているプロジェクトの画面の[トレンド]ウィンドウを作成します。

[トレンド]ウィンドウを作成するには、表示する値のトレンドタイプを指定します。

- ログタグのログ済み値の表示用
- リアルタイムパルスがトリガ済み:時刻トリガされた値の表示用
- リアルタイムビットがトリガ済み:イベントトリガされた値の表示用
- 履歴ビットトリガ:バッファリングデータの取得があるイベントトリガされた表示

#### ログされた値の表示

[トレンド]ウィンドウには、定義可能な時間間隔にログされた値が表示されます。ランタイム中に、オペレータはこの時間間隔をシフトして必要な情報(ログされたデータ)を表示することができます。



#### パルストリガされたトレンド

表示する値は、定義可能な時間パターンで個々に決定されます。パルストリガされるトレンドは、モーターの動作温度の変化など連続する過程を表現するのに適しています。

#### ビットトリガされるトレンド

表示する値は、[トレンド転送]タグで定義されたビットを設定して、イベントによってトリグンされます。読取り後にリセットされるビットは完了しています。ビットトリガされたトレンドは、プラスチック部品の製造における充填圧力など急速に変化する値を表示するのに適しています。

### バッファリングデータの取得があるビットトリガされたトレンド

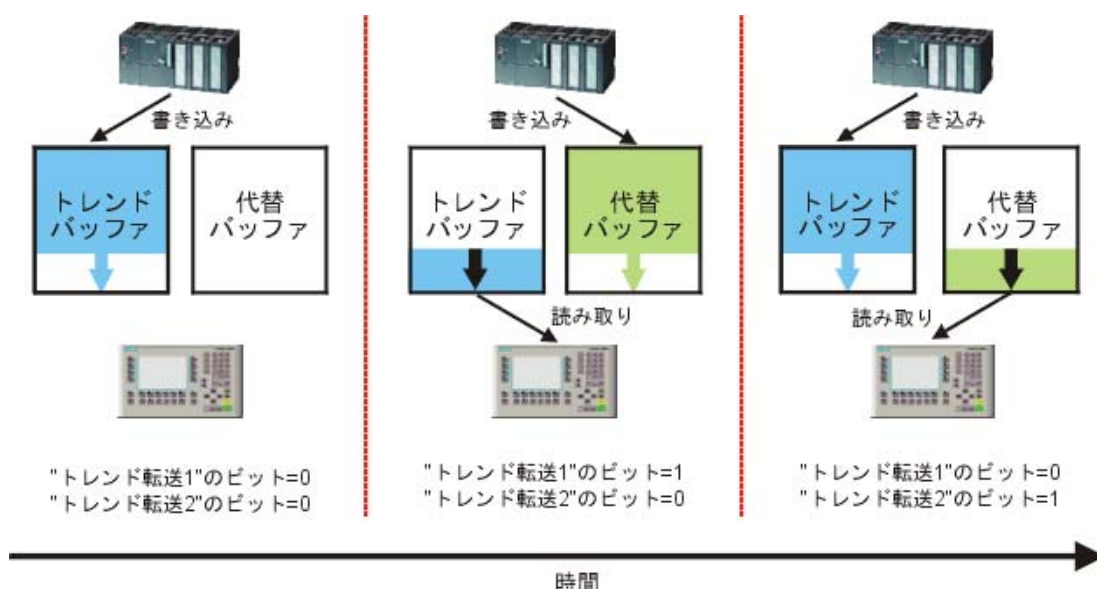
バッファリングデータの取得を設定すると、表示する値は PLC でバッファリングされ、ビットトリガされたトレンドでブロックとして読み取られます。このトレンドは、トレンドが個々の値よりは全体として意味を持つような、急速な変化を表示するのに適しています。

PLC でスイッチバッファを作成すれば、トレンドバッファが読み取られている間に新しい値を継続して書き込むことができます。スイッチバッファを使用すると、オペレータデバイスがトレンドの値を読み取っている間に PLC によって値が上書きされないようにできます。

trend buffer ファンクションと switch buffer ファンクション間のスイッチは以下のとおりです。

トレンドに割り付けられるビットが[トレンド転送 1]タグに設定されるたびに、すべての値がトレンドバッファから同時に読み取られ、オペレータデバイスにトレンドとして表示されます。[トレンド転送 1]にあるビットは、読取りが完了した後リセットされます。

オペレータデバイスがトレンドバッファからタグ値を読み取っている間に、PLC は新しいタグ値をスイッチバッファに書き込みます。トレンドに割り付けられるビットが[トレンド転送 2]タグに設定されるたびに、すべてのトレンドの値がスイッチバッファから読み取られ、オペレータデバイスに表示されます。オペレータデバイスがスイッチバッファを読み取っている間に、PLC は再びトレンドバッファに書き込みを行います。



## 9.2 エlementと基本設定

### 9.2.1 "データログ"エディタ

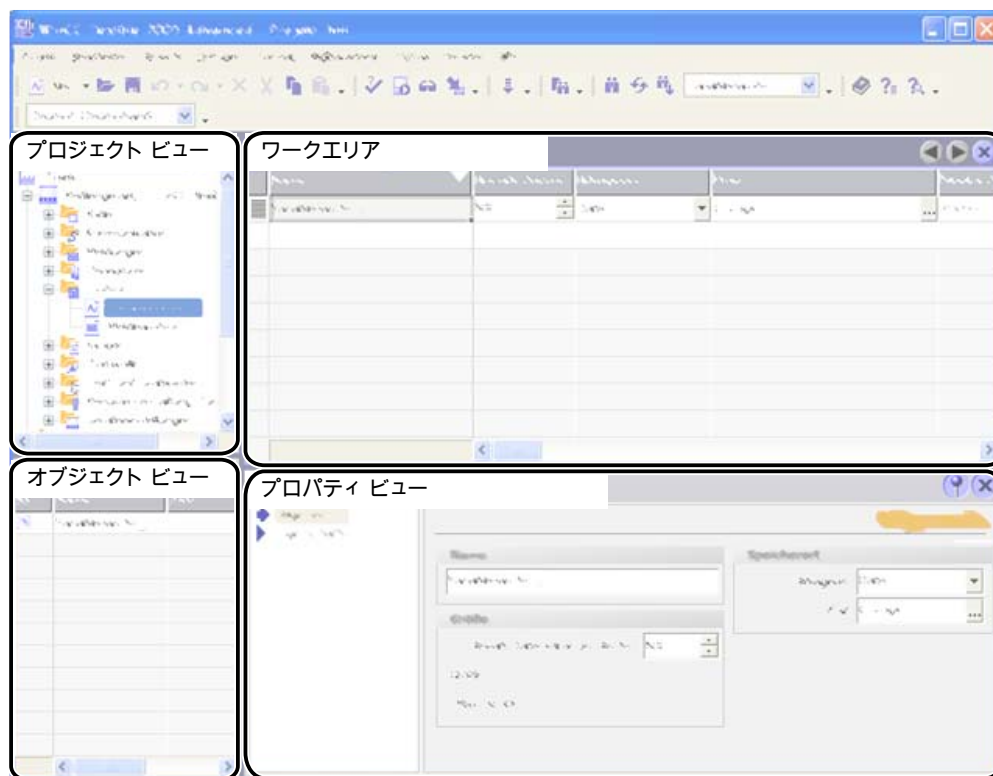
#### はじめに

プロセス値をログするには、プロセス値をログに割り付ける必要があります。テーブル形式の"データログ"エディタで、ログを計画したり、そのプロパティを指定することができます。

#### 開く

[ログ]グループのプロジェクトウィンドウで、[データログ]をダブルクリックして、"データログ"エディタを開きます。

#### 構造



"データログ"エディタ

#### 作業エリア

すべてのデータログが、作業エリアのテーブルに表示されます。テーブルセル内のデータログのプロパティを編集することができます。列のヘッダーをクリックして、列のエンタリによるテーブルのソートを実行することができます。

テーブルの列を表示または非表示にすることができます。これを行うには、列テーブルヘッダーのポップアップメニューでエンタリを有効または無効にします。

## [プロパティ]ウィンドウ

このダイアログボックスで、データログを設定します。[プロパティ]ウィンドウには、作業エリアのテーブルと同一の情報および設定が表示されています。

## 9.2.2 データログの基本設定

### はじめに

データログのプロパティは、"データログ"エディタまたはログの[プロパティ]ウィンドウで定義できます。

### [プロパティ]ウィンドウの構造

[プロパティ]ウィンドウの左側には、すべてのプロパティカテゴリを選択できるツリー構造があります。現在選択されているプロパティカテゴリの設定用フィールドは、[プロパティ]ウィンドウの右側に表示されます。

[プロパティ]ウィンドウで、データログの以下のプロパティを設定できます。

### [全般]プロパティ

- 名前  
データログにはどのような名前も付けられますが、名前には少なくとも1つの文字または数字を入れる必要があります。
- 保存先  
タグログは、ODBC データベース(PC 上のみ)または別のファイルに保存されます。該当する保存先として[データベース]またはファイルを選択します。  
HMI デバイスの設定によって、PC のローカルハードディスク、パネルのストレージカード、または、パスとしてネットワークドライブ(存在する場合)を選択することができます。  
保存先として ODBC データベースを選択している場合、データソースに次のような名前を指定できます。
  - システムの割り付ける名前を使用する場合には、"システム定義のデータソース名"。

---

#### 注記

Windows VISTA では、"システム定義のデータソース"オプションのあるデータベースはサポートされません。

[ユーザー定義のデータソース]オプションのあるデータベースしか使用できません。管理者は、これらのデータベースを作成して、コンピュータで有効にしておく必要があります。

詳細については、使用するデータベースソフトウェアのマニュアルを参照してください。

---

ターゲットシステムで、Microsoft SQL サーバー専用のインスタンスも必要となります。この目的から、無償の SQL Server 2005 Express をダウンロードすることがあります。

Panel PC 477 ではこれを設定できません。

- "ユーザー定義のデータソース名"。データソースの名前を自分で割り付ける場合。

- サイズ

ログのサイズは次のように算出されます。

アイテムの数×ロギングする各タグ値の長さ

[プロパティ]ウィンドウで、現在選択されている[データレコードの数]を使用したログの最小サイズと最大サイズが、[データレコードの数]入力フィールドに表示されます。ログの最大サイズは、HMI デバイスにある記憶領域によって制限されます。

## ログ動作の設定

- [スタートアップ動作]

セクション"有効化"で、ランタイム開始時にロギングの開始を指定できます。[ランタイム開始時にロギングの有効化]チェックボックスを有効にします。

他の方法でランタイム開始時の動作を制御することもできます。以前にログ済みのデータを新しいデータで上書きしたい場合は、[ログのリセット]を、新しいデータを既存のログに追加したい場合は、[既存のログにデータを追加]を有効にします。

---

### 注記

システムファンクションを使用して、ランタイム中にログの再起動を制御できます。

---

- [ロギング方法]

この欄で、ログが満杯になったときの処理を指定することができます。以下のオプションの1つを選択します。

- サークルログ: ログが満杯になったとき、最も古いエントリが上書きされます。
- セグメント化されたサークルログ: 同じサイズの複数のログが作成され、順番に1つずつ使っていきます。すべてのログが満杯になったとき、最も古いログが上書きされます。
- システムメッセージの表示時点: 定義された占有レベルに達したとき、システムメッセージが表示されます
- [トリガ]イベント: ログが満杯になるとすぐに、[オーバーフロー]イベントがトリガされます。

- [コメント]

ここに、ログに関する説明文を入力することができます。

## イベント

ここでは、ログのオーバーフローにより[オーバーフロー]イベントがトリガされるたびに処理されるファンクションリストを作成することができます。

## 9.3 タブ値の記録

### はじめに

ランタイム中にタグ値をログに保存し、後で評価できます。タグのロギング用に値が保存されるログを指定して、その頻度および特定の値範囲にあるタグ値のみを保存するのかどうかを指定する必要があります。

---

#### 注記

データロギングの主な目的は、外部タグの値をロギングすることです。ただし、内部タグの値もロギング可能です。

---

### 原理

データロギングにはいくつかのステップがあります。

- データログの作成および設定

データログの作成時には、以下を定義する必要があります。

- 名前、サイズ、保存先などの全般設定
- ランタイム開始時の動作
- ログが満杯になったときの動作

- タグのロギングの設定

すべてのタグにデータログを指定することができます。このログは、ランタイム中のタグの値を記録し、ログされた時間や値などその他の情報も記録します。

さらに、タグの値をいつ、どの程度の頻度でロギングするかを定義することができます。後者を実行するには、以下のオプションが必要です。

- "要求時"

タグ値は、LogTag システムファンクションを呼び出してログされます。

- "変化時"

オペレータデバイスでタグの値の変化が検出されるとすぐに、タグ値がログされます。

- "連続サイクリック"

タグ値は定期的な間隔でロギングされます。WinCC flexible で使用可能な標準サイクルに加えて、標準サイクルに基づいてユーザー独自のサイクルを追加することができます。

さらに、ロギングの対象を許容範囲内または許容範囲外の値に制限することができます。この方法では、後で行う別の分析用に、異なるログへ個別にタグ値を配信できます。

タグ"要求時"をロギングする場合、タグ"連続サイクリック"または"変化時"がロギングされるシーケンスログにこのタグをロギングすることはできません。要求時のロギングがめったに起こらない場合、シーケンスログは、たとえば周期的にログされる値によって完成されるなどして、次のシーケンスログがセットアップされます。そのあと要求時にロギングされたタグへのアクセスがあると、このタグは現在のシーケンスログがアクセスされるランタイム中に表示できなくなります。対処法として、めったにログされないタグについては別のタグログをセットアップします。

- ログ済みタグ値のそれ以外の処理

ログ済みプロセスタグ値は、たとえば[トレンド]ウィンドウなど WinCC flexible プロジェクトで、または Excel など別のアプリケーションを使用して直接評価できます。

## 9.4 ログ済みデータの出力;ログスミデータノシュツリョク

### 9.4.1 タグ値の画面への出力

#### はじめに

ランタイム中に、タグ値をトレンドの形でオペレータデバイスの画面に出力できます。データは現在のプロセスから PLC により要求されるか、ログデータベースからロードされません。

#### 表示された値

画面の[トレンド]ウィンドウを作成して、タグ値をオペレータデバイスで表示できるようにする必要があります。[トレンド]ウィンドウを作成するときに、表示するタグ値を指定します。

- PLC からの現在値

トレンドは、PLC からの個々の値(リアルタイム表示)、または PLC からの読取りと読取りの間にバッファに保存されるすべての値(間隔を開けた表示)を使用して継続できます。

読取り時間はビットを設定するか、サイクルを使用して制御できます。

- ログ済みタグ値

ランタイム中に、[トレンド]ウィンドウにはデータログからのタグ値が表示されます。[トレンド]ウィンドウには、特定ウィンドウのログ済み値が時間に応じて表示されます。ランタイム中に、オペレータはウィンドウの内容を時間に応じて移動し、必要な情報(ログ済みデータ)を表示することができます。

## 9.4.2 タグ値が格納されている\*.csv ファイルの構造

### はじめに

\*.csv (カンマで区切られた値)ファイル形式では、テーブルの列(エントリ名とエントリ値)はセミコロンによって区切られています。テーブルの各行は、キャリッジリターンで終了します。

### \*.csv ファイルの例

この例は、ログ済みタグ値が格納されているファイルを示しています。

```
"VarName";"TimeString";"VarValue";"Validity";"Time_ms"
"Var_107";"01.04.98 11:02:52";66.00;1;35886460322.81
"Var_107";"01.04.98 11:02:55";60.00;1;35886460358.73
"Var_107";"01.04.98 11:02:57 AM";59.00;1;35886460381.22
```

### \*.csv 形式のログファイルの構造

WinCC flexible ログファイルの各列には以下の値が入力されます。

パラメータ	説明
VarName	WinCC flexible タグの名前
時間文字列	STRING としてのタイムスタンプ(読取り可能なデータフォーマットなど)
VarValue	タグ値
有効性	有効性: 1 = 値は有効です 0 = エラーが発生しました(割り込みプロセス接続など)
Time_ms	タイムスタンプを 10 進数値で指定します(変換については下を参照)。 トレンドにタグ値を表示する必要があるのみ。

### タイムスタンプ 10 進数値の変換

この値を異なるプログラムで処理する必要がある場合、以下のように実行します。

1. Time\_ms を 1,000,000 で割ります。

例: : 36343476928:1 000 000 = 36343,476928

2. 番号全体(36344)は、31.12.1899 から計算した日付です。

例: 36343 は、1999 年 7 月 2 日になります。

この結果、Excel で、[日付]グループの適当なフォーマットをセル(タイムスタンプを含んでいます)に割り付けることによって、タイムスタンプ値を日数に変換することができます。

結果: 37986 は、2003 年 12 月 31 日になります

3. (0,476928)のカンマの後の値は、時刻を表します。

– 値(0,476928)に 24 を掛けると、時間数(11,446272)が得られます。

– 余り(0,446272)に 60 を掛けると、分数(26,77632)が得られます。

– 余り(0,77632)に 60 を掛けると、秒数(46,5792)が得られます。

合計は、11:26:46.579 になります。

この変換は、例えば、Microsoft Excel でサポートされています。

### 9.4.3 ODBC ログデータベースへの直接アクセス

#### はじめに

ログの保存先は、データベースまたはファイルが可能です。

データベースは、その"データソース名"(DSN)によってアドレスされます。Windows の[スタート]>[設定]>[コントロールパネル]>[ODBC データソース]を選択し、WinCC flexible で使用するデータベースを選択します。

ログデータを保存するには、構成設定を行うとき、ディレクトリ名の代わりに"データソース名"(DSN)を指定します。DSN を使用すると、データベースそれ自体とその保存先を参照していることになります。

#### 用途

ログデータの追加処理と評価に関して、データベースの機能範囲全体が使用可能です。

#### 原理

ランタイムソフトウェアを収納しているコンピュータ上に、データベースに接続するデータソースを作成します。この後、WinCC flexible でログを作成するときに、上記の設定された DSN を指定します。

ODBC インターフェースを使用して、MS SQL Server などの他のプログラムから、直接データベースにアクセスできます。

"StartProgram"システムファンクションを使用して、HMI デバイス上でプログラム呼び出しを作成することもできます。これは、ランタイムプログラムシーケンスを中断させません。

## レポートの取り扱い

### 10.1 レポートシステムの原理

#### はじめに

WinCC flexible では、レポートを使用してプロセスデータや完了した生産サイクルを文書化することができます。メッセージやレシピデータをレポートして、シフトレポートの作成、バッチデータの出力、または受領試験用の製造プロセスの文書化を行うことができます。

#### 概要

グラフィックエディタでレポートファイルを編集することができます。このエディタでは、レポートのレイアウトを設定して出力データを定義します。レポートファイルには、データ出力用にさまざまなオブジェクトを追加することができます。ツールボックスオブジェクトの中には、使用できる機能に制限があるものと、まったく使用できないものがあります。これは作成する HMI デバイスによります。ツールボックスで使用できないオブジェクトはグレー表示され、選択できません。

さまざまなタイプのデータをレポートするために、個別のレポートファイルを作成することができます。レポートファイルごとに、個別の出力のトリガを設定することができます。データの出力を特定の時間にトリガするか、決められた頻度でトリガするか、または別のイベントによりトリガするかを選択することができます。

これらの機能はモジュラー構造になっており、必要条件に応じてレポートを明示的に設定することができます。

#### アプリケーション事例集

交代勤務の最後に、完了した生産分のバッチデータおよびエラーイベントを含む交代勤務レポートを作成します。

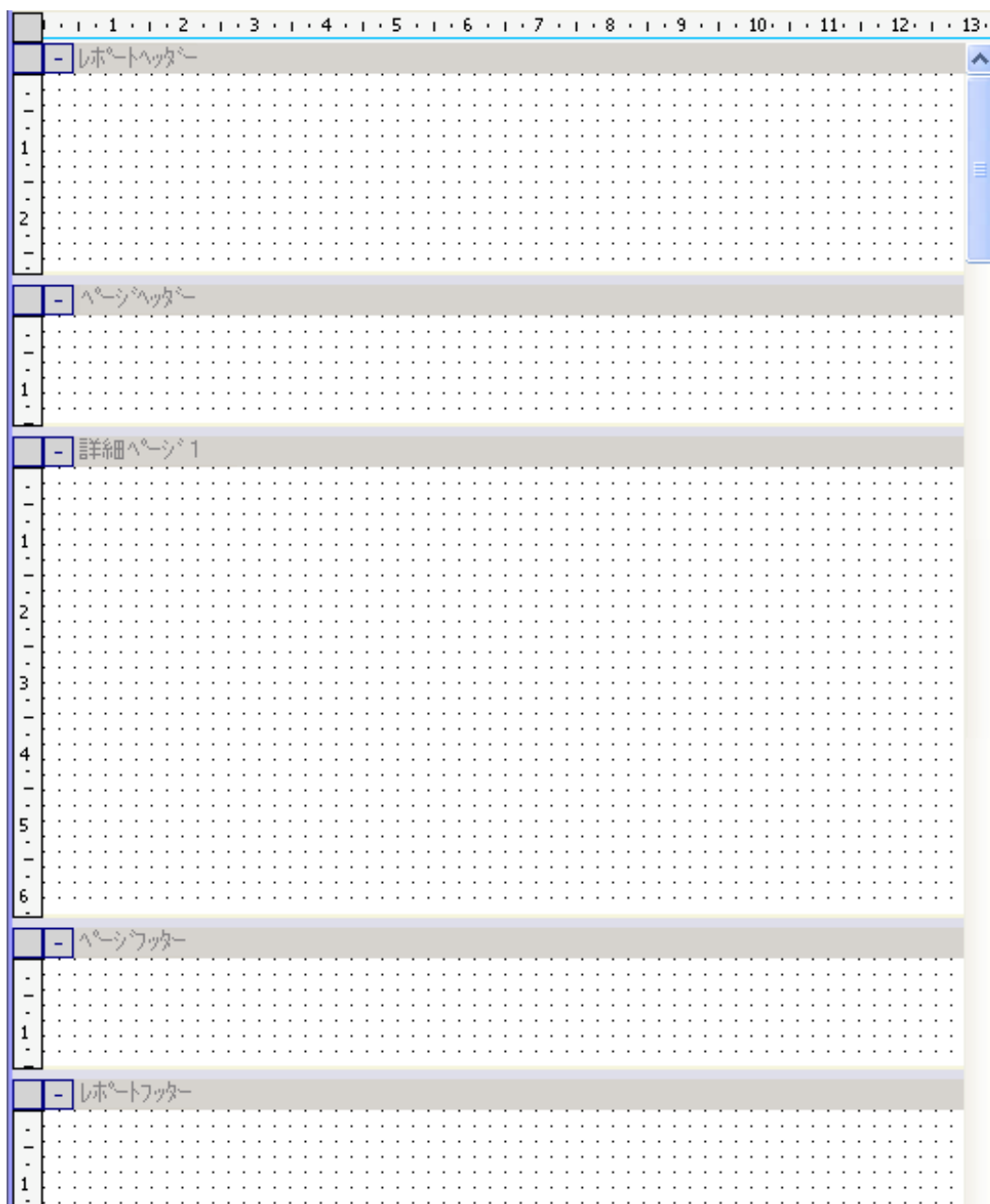
バッチ生産分の記録された生産データを出力するレポートを作成します。

特定のクラスまたはタイプのメッセージを出力するレポートを作成します。

## 10.2 レポートのストラクチャ

### レポートのストラクチャ

WinCC flexible 内のレポートの基本構造はすべて同じです。レポートは、次図のように異なるセクションに分割されています。



プロトコル

個々のセクションは、さまざまなデータ出力のために使用され、一般オブジェクトと固有のレポートオブジェクトを含むことができます。

- レポートヘッダー  
レポートヘッダーはレポートのカバーシートとして機能します。レポートヘッダーは、プロジェクトタイトルと全般情報をプロジェクトに対して出力するために使用されます。レポートヘッダーには、ページヘッダーとページフッターは出力されません。レポートヘッダーは、レポートの始めに一度だけ出力されます。
- レポートフッター  
レポートフッターは、レポートの最終ページとして使用されます。レポートフッターは、レポートのまとめ、またはレポートの最後に必要なその他の情報の出力に使用されます。レポートフッターは、ページヘッダーとページフッターなしで出力されます。レポートフッターは、レポートの終わりに一度だけ出力されます。
- ページヘッダー  
ページヘッダーは、レポートの全ページに出力されます。ページヘッダーは、日時、タイトル、その他の全般情報などの出力に使用されます。
- ページフッター  
ページフッターは、レポートの全ページに出力されます。ページフッターは、ページ番号、総ページ数、その他の全般情報などの出力に使用されます。
- 詳細ページ  
ランタイムデータが[詳細ページ]エリアに出力されます。ランタイムデータの出力用オブジェクトが[詳細ページ]エリアに挿入されます。データの出力時に、データ量によってページ区切りが自動的に追加されます。種々の出力オブジェクトの構成を適切に分割する目的で、レポートにいくつかのページを挿入することができます。  
レポートの作成については、「レポートの作成」章で説明しています。

## 10.3 エlementと基本設定

### 10.3.1 エディタ"レポート"

#### はじめに

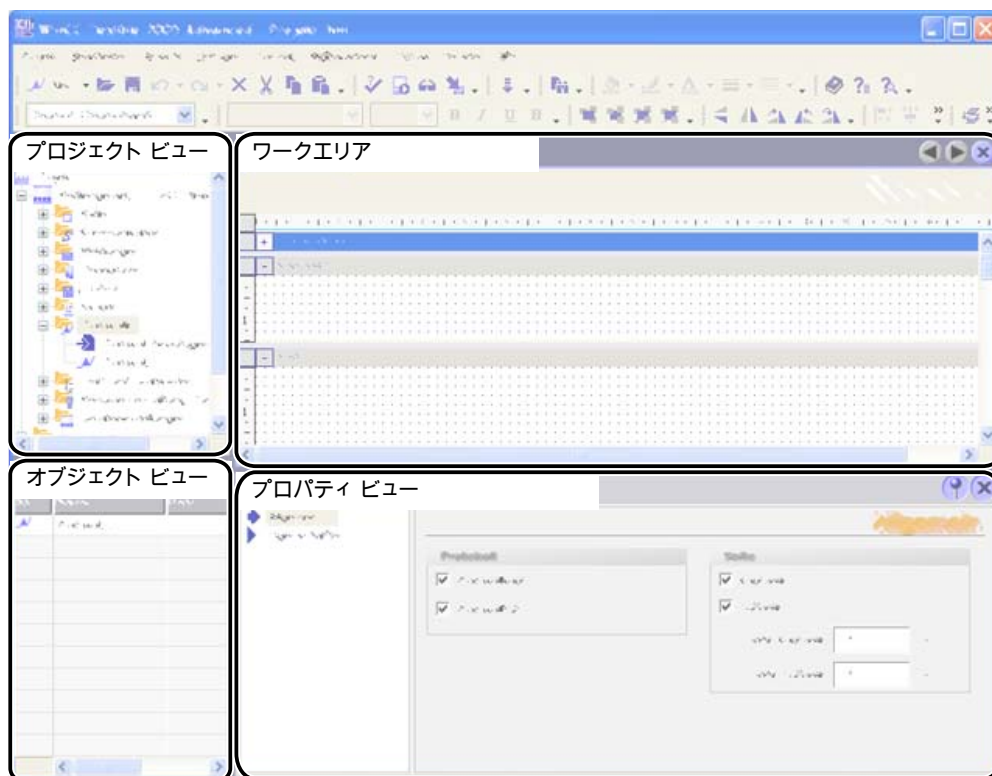
レポートエディタではレポートの作成と編集を行うことができます。

#### 開く

[プロジェクト]ウィンドウで[レポート]エントリを選択し、ポップアップメニューを開きます。ポップアップメニューの[レポートの新規作成]コマンドを選択します。新しいレポートが作成され、作業エリアで開かれます。

既存のレポートを開くには、[オブジェクト]ウィンドウで開きたいレポートをダブルクリックします。選択したレポートが開かれます。

## レイアウト




## メニューバー

メニューバーには、WinCC flexible の操作に必要なすべてのコマンドがあります。使用可能なショートカットキーはすべてメニューコマンドの隣に表示されます。

## ツールバー

ツールバーには、最も頻繁に使用するボタンが含まれます。

[表示|ツールバー]の順にメニューを選択すれば、使用可能なツールバーを表示したり、非表示にしたりできます。ツールバーの  ボタンは、そのツールバーの個々のボタンを表示したり隠したりするために使用します。

## 作業エリア

作業エリアでは、レポートのコンフィグレーションを行います。

## ツールボックス

ツールボックスから、レポートのコンフィグレーションに必要なオブジェクトにアクセスできます。オブジェクトは、ドラッグアンドドロップによってレポートに挿入できます。

## [プロパティ]ウィンドウ

オブジェクトを選択している場合、[プロパティ]ウィンドウで、その選択したオブジェクトのプロパティを編集できます。

オブジェクトを選択していない場合、[プロパティ]ウィンドウで、レポートの有効エリアのプロパティを編集できます。

### 10.3.2 [ツールボックス]ウィンドウの使用法

#### はじめに

[ツールボックス]ウィンドウには、[単純なオブジェクト]のレポートに挿入できるオブジェクトのグループと[レポートオブジェクト]グループの選択肢があります。

ワークエリアでレポートウィンドウが有効になっている場合、ツールボックスはレポートで使用できるオブジェクトだけを表示します。ツールボックスオブジェクトの中には、使用できる機能に制限があるものと、全く使用できないものがあります。これは作成する HMI デバイスによります。ツールボックスで使用できないオブジェクトはグレー表示され、選択できません。

#### デフォルトのプロパティの変更

[ツールボックス]ウィンドウでは、さまざまなオブジェクトタイプに対してデフォルトのプロパティが事前設定されています。オブジェクトを[ツールボックス]ウィンドウからレポートに挿入すると、デフォルトのプロパティがオブジェクトによって置き換えられます。

プロジェクトの必要条件に合うよう、オブジェクトタイプのデフォルトのプロパティをカスタマイズすることができます。オブジェクトタイプのデフォルトのプロパティを変更する場合、既に挿入済みのオブジェクトのプロパティは変更されません。したがって、デフォルトのプロパティを調整してからオブジェクトを挿入することを推奨します。

オブジェクトのデフォルトのプロパティは、オペレーティングシステムにログオンしたときのユーザー名と対になっています。

デフォルトのプロパティを変更するには、[ツールボックス]ウィンドウにあるオブジェクトのポップアップメニューを開きます。コマンド[デフォルトプロパティの編集]を選択します。[プロパティ]ダイアログボックスが表示されます。プロジェクトの必要に応じてオブジェクトのデフォルトのプロパティを調整します。

#### [ツールボックス]ウィンドウの表示方法

[表示]ツールボックスウィンドウ]の順にメニューを選択すれば、ツールボックスを表示したり、非表示にしたりできます。

## 10.4 レポートの取り扱い

### 10.4.1 レポートの作成

#### はじめに

レポートを作成する場合には、個々のセクションと内容を指定します。以下のセクションの内容をコンフィグレーションします。

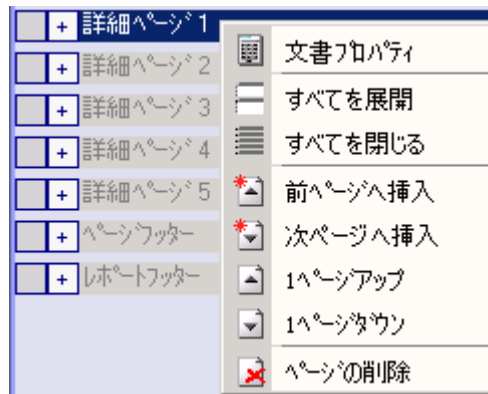
- レポートヘッダー
- ページヘッダー
- ページ
- ページフッター
- レポートフッター

#### コンフィグレーションの概要

[ツールボックス]ウィンドウのオブジェクトを使うと、レポートをデザインしたり、出力データをコンフィグレーションしたりできます。オブジェクトの中には、レポートで使用される場合に、スクリーンエディタの同類のオブジェクトに比べて機能の範囲が制限されているものもあります。たとえば、IO フィールドは、出力フィールドとしてしか機能しません。

レポートエディタでレポートを作成するとき、レポートは WYSIWYG として表示されます。[アラームの印刷]および[レシピの印刷]のようなデータ出力用のダイナミックオブジェクトは例外です。ダイナミックオブジェクトのサイズは既存のデータ量に依存するため、オブジェクトをコンフィグレーションしたときの高さは出力フォーマットとは関係ありません。改ページは、データ量によって、ページに連続的に挿入されます。レポートの各ページにこれらのオブジェクトを1つだけ挿入できます。これらのダイナミックオブジェクトのいずれかの下で、同じ側に位置するオブジェクトは出力されません。[印刷アラーム]と[レシピの印刷]オブジェクトは、レポート用にコンフィグレーションされた幅で自動的に挿入されます。出力の幅は、コンフィグレーションされたレポートの幅に従います。

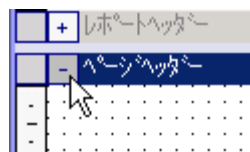
新規のレポートは、常に1ページだけです。このページは、出力用のページを表します。必要に応じて、このレポートに追加のページを挿入できます。ページを追加するには、既存のページのタイトルバーの上にカーソルを移動し、マウスの右ボタンをクリックしてポップアップメニューを開きます。[前ページに挿入]および[次ページに挿入]コマンドを使用して、既存のページの前後に新規ページを挿入します。ページには、連続番号が割り付けられます。1レポートにつき最大10ページが許可されています。11ページ以上作成すると、過剰なページの連続番号はカギ括弧(不等号)で囲まれます(例: ページ<11>)。余分なページは、出力されません。ページのポップアップメニューで[ページの削除]コマンドを使用して、選択したページを削除します。



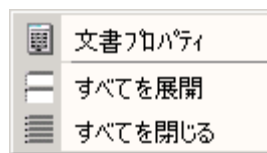
ページのポップアップメニュー

作成したページの順序は後で変更することができます。ページの順番を変更するには、目的のページの上にカーソルを移動し、マウスの右ボタンをクリックしてポップアップメニューを開きます。ポップアップメニューで対応する[1 ページアップ]または[1 ページダウン]コマンドを選択します。このページは選択したコマンドに従って移動します。ページの連番は、維持されます。たとえば、ページ 4 が、対応するコマンドによって"1 ページアップ"されると、ページ 4 がページ 3 と交換されます。

作業エリアの外観を把握するため、個々のレポートセクションを閉じることができます。最小化または最大化するために、そのセクション指定の前にあるノードをクリックします。



すべてのエリアを同時に表示または非表示にすることもできます。これを行うには、レポートエリアのタイトルバーの上にカーソルを移動し、マウスの右ボタンをクリックしてポップアップメニューを開きます。ポップアップメニューで対応する[すべて表示]または[すべて非表示]コマンドを選択します。



## 10.4.2 レポートプロパティの調整

### はじめに

レポートプロパティで、レポート用の出力オプションとフォーマットオプションを編集します。以下のプロパティグループを使用できます。

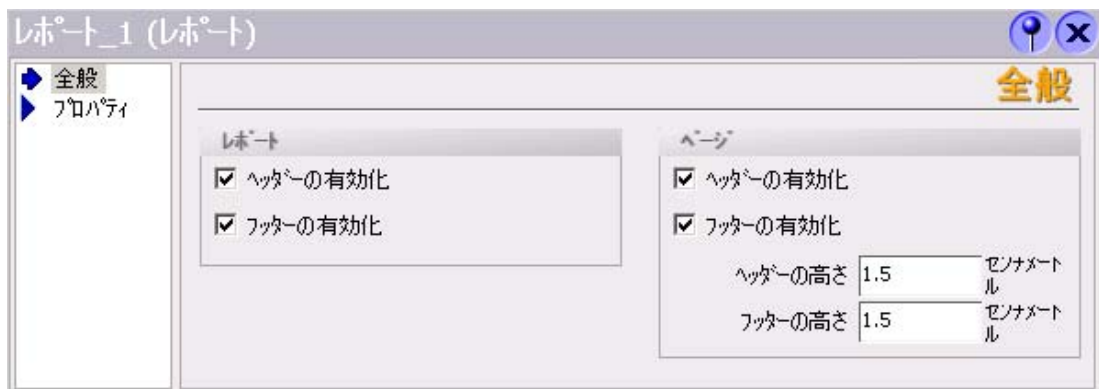
- 一般
- プロパティ/表示

### 必要条件

- プロパティを変更するレポートを開く必要があります。
- [プロパティ]ウィンドウを開く必要があります。

### レポートプロパティの編集

これを行うには、たとえば、[ページヘッダー]のタイトルバーなどのレポートエリアのタイトルバーにカーソルを移動します。右ボタンをクリックしてポップアップメニューを開き、[文書プロパティ]コマンドを選択します。レポートのプロパティは[プロパティ]ウィンドウに表示されます。



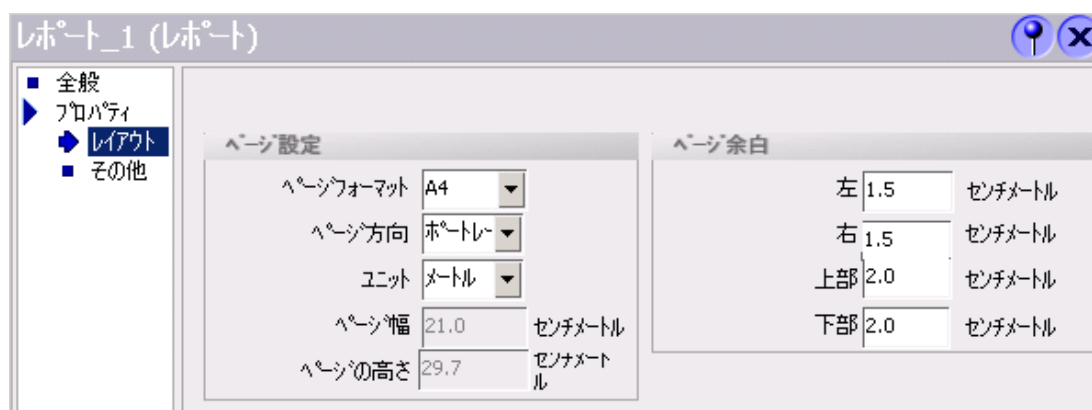
[プロパティ]ウィンドウの[全般]プロパティグループを有効にします。

[レポート]エリアで、レポートヘッダーおよびレポートフッターの出力を有効または無効にします。

[ページ]エリアのページヘッダーおよびページフッターの出力を有効または無効にします。

レポートエリアの出力を無効にした場合、このエリアには"(X)"が付いてタイトルバーに表示されます。

[プロパティ]ウィンドウの[プロパティ/表示]プロパティグループを有効にします。



[ページ]フィールドで出力用のページフォーマットを選択します。

または、"ユーザー定義"フォーマットを選択します。その後、[幅]と[高さ]フィールドで独自フォーマット用の値を入力することができます。

[ページ方向]フィールドで"縦方向"または"横方向"を選択します。

[ユニット]フィールドでページサイズとページ余白を設定するための測定単位を選択します。

[ページ余白]エリアのフィールドを使ってページ余白のサイズを設定します。ページ余白の設定は、プリンタのページ余白の設定より小さくすることはできません。

### 10.4.3 レポート作成用のオブジェクト








#### はじめに






オブジェクトはプロジェクトレポートをレイアウトするためのグラフィックエレメント、または出力データのためのダイナミックエレメントとすることができます。オブジェクトの制限は、構成されている HMI デバイスによって異なります。オブジェクトの説明にある情報にご注意ください。

オブジェクトは、[ツールボックス]ウィンドウの[単純なオブジェクト]オブジェクトグループに用意されています。

レポートの作成に特殊レポートオブジェクトも使用できます。特殊レポートオブジェクトは、[レポートオブジェクト]オブジェクトグループにあります。

### 単純なオブジェクト

シンボル	オブジェクト	説明
	ライン	[ライン]は開いたオブジェクトです。ラインの長さや傾斜は、オブジェクトを囲む四角形の幅と高さで定義されます。ラインの端には、矢印やドットなどが表示できます。
	多角折線	[多角折線]は開いたオブジェクトです。開始ポイントと終了ポイントの座標が同じ場合でも表面を塗りつぶすことはできません。多角折線は頂点をいくつでも持つことができます。各頂点は作成された順番に番号が付けられ、個々に変更したり、削除することもできます。多角折線のラインの終端は、矢印またはドットなどで表されます。
	多角形	[多角形]は、色や柄で塗りつぶすことのできる閉じたオブジェクトです。1つの多角形は、任意の数の頂点を持つことができます。各頂点は作成された順番に番号が付けられ、個々に変更したり、削除することもできます。
	楕円	[楕円]は、色やパターンで塗りつぶすことのできる閉じたオブジェクトです。水平方向または垂直方向に整列するように、楕円の幅と高さをカスタマイズできます。
	円	[円]は、色やパターンで塗りつぶすことのできる閉じたオブジェクトです。円の半径は自由に調整することができます。
	四角形	[四角形]は、色やパターンで塗りつぶすことのできる囲まれたオブジェクトです。四角形はその高さや幅を自由に調整して水平方向または垂直方向にそろえることができます。丸みのある四角形の角は、必要に応じて丸みをつけることができます。
	テキストフィールド	スタティックテキスト用フィールドは、色やパターンで塗りつぶすことのできる囲まれたオブジェクトです。スタティックテキストはどんなサイズのテキストフィールドにも入力できます。すべての設定された言語に対して、単一または複数の行テキストを入力できます。

シンボル	オブジェクト	説明
	[I/O フィールド]	IO フィールドは、レポートでは出力フィールドとしてしか使用できません。[I/O フィールド]を使って、各値を以下のデータフォーマットで出力することが可能です。2進、日付、日付-時刻、10進数、16進数、文字列、および時刻。
	[日付-時刻フィールド]	日付と時刻は、[日付-時刻]フィールドを使ってレポートに出力されます。システム時間を出力するか、または[日付-時刻]フィールドに対応する値を提供する WinCC flexible タグに接続することができます。
	[グラフィック IO フィールド]	グラフィック IO フィールドは、レポートでは出力フィールドとしてしか使用できません。フィールドは、グラフィックリストからグラフィックを選択するために使用されます。たとえば、これを使ってタグの状態をグラフィックで表示することができます。 例: 0 や 1 の値の代わりに、閉じた値や開いた値にそれぞれ 1 つのグラフィックを出力できます。
	[シンボル IO フィールド]	ドロップダウンリストは、レポートでは出力フィールドとしてしか使用できません。テキストリストからテキストを選択するのにこのフィールドを使用します。たとえば、これを使ってタグの状態をテキスト形式で表示することができます。 例: 0 や 1 の値の代わりに、モータの状態に対して[モータオフ]と[モータオン]を出力できます。
	[グラフィック表示]	グラフィックオブジェクトでは、ほかのプログラムを使ってレポート内に作成したグラフィックを挿入することもできます。グラフィックまたはイメージを以下のフォーマットで挿入することができます: "*.emf"、 "*.wmf"、 "*.dib" および "*.bmp" レポートにおけるグラフィックオブジェクトのサイズとプロパティを定義します。

#### 注記




ツールボックスオブジェクトの中には、使用できる機能に制限があるものと、まったく使用できないものがあります。これは作成する HMI デバイスによります。ツールボックスで使用できないオブジェクトはグレー表示され、選択できません。

## 10.4.4 レポートオブジェクトの使用

### はじめに

ツールバーの[レポートオブジェクト]セクションにある特殊オブジェクトはレポートに使用できます。これらのオブジェクトは、レポート専用です。

### オブジェクトの概要

シンボル	オブジェクト	簡単な説明
	ページ番号	レポートにページ番号が出力されます。このオブジェクトは、レポート内のページフッターなどに一度挿入するだけで済みます。
	印刷レシピ	レシピデータがレポートに出力されます。
	印刷アラーム	レポートにアラームが出力されます。

## 10.5 アラームのレポート機能

### 10.5.1 アラームのレポート機能

#### はじめに

WinCC flexible で、以下のアラームを出力できるレポートをコンフィグレーションします。

- アラームバッファの現在のアラーム
- アラームログからのアラーム

#### 必要条件

#### 手順

以下のように実行します:

1. ツールボックスオブジェクトから[印刷アラーム]をレポートに追加します。
2. プロパティを[プロパティ]ウィンドウに表示させるには、オブジェクトを選択します。
3. [プロパティ]ウィンドウから、コンフィグレーションするデータを作成します。
4. 選択したソース用に出力したいアラームクラスを指定します。
5. 出力用アラームのシーケンスを指定します。
6. ある間隔のアラームを出力するには、[表示開始]フィールドと[表示終了]フィールドについてそれぞれのタグを選択します。タグは、ランタイム中その期間に発生した最初または最後のアラームの日付と時刻を表示します。

## 10.5.2 アラームプロトコルに応じた出力パラメータの処理

### はじめに

[プロパティ]ウィンドウでアラームプロトコルに応じて出力パラメータを処理します。プロパティを表示するには、レポートに[アラームの印刷]オブジェクトを挿入する必要があります。[プロパティ]ウィンドウを開く必要があります。

### [印刷アラーム]の出力パラメータ

作業エリアで[印刷アラーム]オブジェクトを選択します。[プロパティ]ウィンドウにオブジェクトのプロパティが表示されます。データソースを選択し、[全般]カテゴリで出力用データの選択とレイアウトを設定します。



アラームの出力には、以下のエントリが使用できます。

属性	機能	必要条件
[アラームのソース]	アラームのソースを選択するために使用します。以下のオプションを選択することができます。 <ul style="list-style-type: none"> <li>アラームイベント</li> <li>アラームログ</li> </ul>	
[ソート]	出力用のシーケンスを指定するために使用します。以下のオプションを選択することができます。 <ul style="list-style-type: none"> <li>最も古いメッセージが最初</li> <li>最も新しいメッセージが最初</li> </ul>	
[エントリごとのライン]	アラームごとに使用可能な行数を指定します。必要な行数は、ログ、使用フォントおよびプリンタの用紙フォーマットの選択された列数や幅によって異なります。	
[ページヘッダーの表示]	各列のヘッダーをテーブルに出力するかどうかを指定するために使用します。	
[アラームログ]	出力のアラームログを選択するために使用します。	アラームログが"アラーム用ソース"の中でアラームソースとして指定されていること。

属性	機能	必要条件
[アラームクラス]	出力用アラームクラスの選択に使用されます。以下のオプションを選択することができます。 <ul style="list-style-type: none"> <li>• [アラーム]</li> <li>• [アラームイベント]</li> <li>• [システムアラーム]</li> <li>• [S7 診断イベント]</li> </ul>	
[表示開始]	特定期間のアラームを出力するための、最初のアラームを選択するために使用します。選択リストを介して、フィールドをタグに接続します。入力フィールドを介するなどして、ランタイムのタグに開始値を指定します。	タグは[日付-時刻]タイプであること。
[表示終了]	特定期間のアラームを出力するための、最後のアラームを選択するために使用します。選択リストを介して、フィールドをタグに接続します。入力フィールドを介するなどして、ランタイムのタグに終了値を指定します。	タグは[日付-時刻]タイプであること。

[プロパティ]カテゴリの[外観]サブカテゴリを選択します。ここで、前面色、背面色、スタイル、使用するフォントの設定をコンフィグレーションします。



[プロパティ]カテゴリの[表示]サブカテゴリを選択します。[印刷アラーム]オブジェクトの位置とサイズを設定します。レポートへの出力用の列を[表示エレメント]エリアで選択します。

以下の列を出力できます。

- [アラーム番号]
- [時間]
- [アラームステータス]
- [アラームテキスト]
- [日付]
- [アラームクラス]
- アラームグループ;アラームグループ
- [診断可能]

- [PLC]



#### 注記

レポートの中で設定された[印刷アラーム]オブジェクトの高さは、出力には影響しません。レポートの出力中に、膨大なデータが発生することがあるため、発生するすべてのデータが出力されるように[印刷アラーム]オブジェクトはダイナミックに拡張されます。ページの長さを超えている場合、自動改ページが実行されます。

## 10.6 レシピのレポート機能

### 10.6.1 レシピのレポート機能

#### はじめに

WinCC flexible の中でレシピレコードのレポートを作成します。

#### レシピの出力データ

レシピレポートを作成するため、[ツールボックス]ウィンドウからレポートに[レシピの印刷]オブジェクトを挿入します。プロパティを[プロパティ]ウィンドウに表示させるには、オブジェクトを選択します。[プロパティ]ウィンドウでレポート用の選択データをコンフィグレーションします。

[プロパティ]ウィンドウの[全般]カテゴリで選択データを指定します。エクスポート用のレシピレコードを選択します。

以下の選択が可能です。

- レシピの全レコード
- レシピのレコード範囲
- 複数またはすべてのレシピの全レコード
- 複数またはすべてのレシピのレコード範囲

複数のレシピを選択するときは、連続するレシピの特定範囲しか選択できません。システムでは、レシピの数を指針として使用します。同じ挙動がデータレコードの範囲にも適用されます。

#### 出力のためのフォーマット

[プロパティ]ウィンドウにおいて、[プロパティ|表示]カテゴリで、データを 1 行ずつ出力するかまたはテーブル形式で出力するかのどちらかを指定します。同じカテゴリで、出力用の[表示エレメント]エリアでレコードエレメントを選択します。

以下のレコードエレメントを出力できます。

- [レシピ番号]
- [レシピ名]
- [データレコード番号]
- [データレコード名]
- [タグ名]
- [タグタイプ]
- [エレメント]

レポートは、時間制御またはイベント駆動で出力されます。

## 10.6.2 レシピレポート用の出力パラメータの編集

### はじめに

レシピレポートの出力パラメータは、[プロパティ]ウィンドウで編集されます。プロパティを表示するには、[レシピの印刷]オブジェクトをレポートに挿入する必要があります。[プロパティ]ウィンドウを開く必要があります。

### [レシピの印刷]の出力パラメータ

作業エリアで[レシピの印刷]オブジェクトを選択します。[プロパティ]ウィンドウにオブジェクトのプロパティが表示されます。レポートへの出力用レシピを[全般]カテゴリで選択します。

レシピ選択には、以下のエントリを使用できます。

属性	機能	必要条件
[レシピ選択]	レシピを出力するための選択基準を選ぶために使用します。以下のオプションを選択することができます。 [すべて] [名前] [番号]	
[レシピ名]	名前でレシピを選択するために使用します。 すでにコンフィグレーションを終えているレシピがある場合は、そのレシピの名前を入力するか、オブジェクトリストから選択します。	[名前]オプションは、[レシピ選択]選択フィールドで選ぶ必要があります。
[最初のレシピ]	複数のレシピのレコードを出力するための開始レシピ番号を選ぶために使用します。固定の開始番号をフィールドに入力するか、選択リストを使ってフィールドをタグに接続します。タグには、ランタイム時にダイナミックに開始値を与えることが可能です。	[番号]オプションは、[レシピ選択]選択フィールドで選ぶ必要があります。

属性	機能	必要条件
[最後のレシピ]	複数のレシピのレコードを出力するための終了レシピ番号を選ぶために使用されます。固定の終了番号をフィールドに入力するか、選択リストを使ってフィールドをタグに接続します。タグには、ランタイム時にダイナミックに終了値を与えることが可能です。	[番号]オプションは、[レシピ選択]選択フィールドで選ぶ必要があります。
[データレコードの選択]	レシピのレコードを出力するための選択基準を選ぶために使用されます。以下のオプションを選択することができます。 [すべて] [名前] [番号]	
[データレコード名]	名前でレコードを選択するために使用します。	[名前]オプションは、[データレコード選択]選択フィールドで選ぶ必要があります。
[最初のデータレコード]	レシピのレコード範囲を出力するための開始レコード番号を選ぶために使用します。固定の開始番号をフィールドに入力するか、選択リストを使ってフィールドをタグに接続します。タグには、ランタイム時にダイナミックに開始値を与えることが可能です。	[番号]オプションは、[データレコード選択]選択フィールドで選ぶ必要があります。
[最後のレコード]	レシピのレコード範囲を出力するための終了レコード番号を選ぶために使用します。固定の終了番号をフィールドに入力するか、選択リストを使ってフィールドをタグに接続します。タグには、ランタイム時にダイナミックに終了値を与えることが可能です。	[番号]オプションは、[データレコード選択]選択フィールドで選ぶ必要があります。

[プロパティ]カテゴリの[外観]サブカテゴリを選択します。ここで、前面色、背面色、スタイル、使用するフォントの設定をコンフィグレーションします。



[プロパティ]カテゴリの[表示]サブカテゴリを選択します。[位置とサイズ]エリアで[レシピの印刷]オブジェクトの位置とサイズを設定します。または、マウスを使用して、作業エリアで[レシピの印刷]オブジェクトのサイズと位置を変更できます。

[設定]エリアでテーブルまたは 1 行ずつか、どちらかの出力形式を選択します。テーブル形式で出力する場合は、[列の幅]フィールドで列の幅用の数字と文字を指定します。設定した幅はテーブルのすべての列に影響します。

レポートへの出力用のレコードエレメントを[表示エレメント]エリアで選択します。



#### 注記

レポートで設定した[レシピの印刷]オブジェクトの高さは、出力には影響しません。レポートの出力中に、膨大なデータが発生することがあるため、発生するすべてのデータが出力されるように[レシピの印刷]オブジェクトはダイナミックに拡張されます。ページの長さを超えている場合、自動改ページが実行されます。

## 10.7 レポートの出力

### はじめに

WinCC flexible では、レポートの出力に関して以下のオプションを提供します。

時間により制御される出力。たとえば、

- 繰り返しがなく、時間で制御される出力
- 一定間隔で繰り返される出力

イベントにより制御される出力。たとえば、

- タグの値の変更によるもの
- WinCC flexible 画面内の作成されたボタンの実行によるもの
- ログのオーバーフロー
- WinCC flexible のスクリプトによるもの

### 出力のコンフィグレーション

時間で制御される出力は、スケジューラによって作成されます。スケジューラによって使用可能になるシステムイベントで、レポート出力をさらに細かく制御することもできます。

イベントで制御されるオブジェクトの出力は、タグ、WinCC flexible 画面のボタン、またはログで直接作成されます。

---

#### 注記

Windows ベースの HMI デバイスでは、出力はデフォルトのプリンタに送信されます。

Windows-CE ベースの HMI デバイスの場合、プリンタは HMI デバイスの[コントロールパネル]で設定されます。ネットワークプリンタは、プリンタ名によってアクセス可能でなければなりません。つまり、プリンタは DNS サーバ経由でネットワークに接続する必要があります。Windows-CE ベースの HMI デバイスでは、IP アドレスによってネットワークプリンタのアドレスを指定することはできません。

---

## ユーザー管理

### 11.1 ユーザー管理のアプリケーションの分野

#### 原理

アクセス保護によって、ランタイム中のデータおよびファンクションへのアクセスのルールが決まります。これにより、アプリケーションを未許可のアクセスから保護します。プロジェクトを作成するときに、特定のユーザーグループに対して安全関連の操作はすでに制限しています。特定のアクセス権、つまりオーソリゼーションを与えられる該当するユーザーおよびユーザーグループをセットアップします。安全関連のオブジェクトを操作するために必要なすべてのオーソリゼーションを設定します。たとえば、オペレータには特定の操作エレメントに対するアクセスのみを許可します。たとえば、コミッシングエンジニアにはランタイム中に無制限のアクセスを許可します。

#### 定義

管理ユーザー、ユーザーグループ、オーソリゼーションは、WinCC flexible のユーザー管理で一元的に管理します。ユーザーとユーザーグループを、プロジェクトデータを含めてオペレータステーションに転送します。オペレータステーションの[ユーザー]ウィンドウを使ってユーザーとパスワードを管理します。

#### 適用例

セットアップパラメータにアクセスする必要があるフィールドサービス技術者用の"サービス"オーソリゼーションを設定します。オーソリゼーションを"サービス技術者"ユーザーグループに割り付けます。これで、このグループのすべてのメンバーに保護されたセットアップパラメータへのアクセスが許可されます。

#### 注意

アクセス保護では、誤操作に対する保護は実行されせん。適切なトレーニングを受けた担当者だけがシステムやマシンを設計、コミッシング、操作、維持する権限を与えられるようにするためには、すべて自分で判断する必要があります。

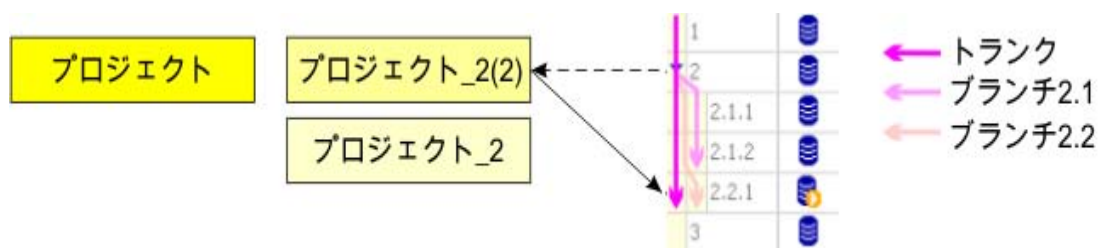
作業ルーチンの定義とその遵守のモニタに関するアクセス保護は不適切です。

## 11.2 ユーザー管理の構造

### はじめに

製造エンジニアリングのプロジェクトの場合、プラントオペレータとしては、装置メーカーの環境とエンドユーザーの環境とを区別する必要があります。装置メーカーが、たとえば、フォアマン氏というユーザーにプロジェクトへの特定のアクセスを許可します。しかしながら、エンドユーザー環境にフォアマンという名前のユーザーが存在していない、という状況が発生します。

たとえば、任意のプラントの各種プロジェクトを、プロセスエンジニアリング内のプロジェクトの 1 つに統合する場合、同じような問題が発生します。プロジェクトを統合するには、コミショニング中に個々のプロジェクトのデータに無制限にアクセスできることが必要になります。



このため、ユーザー管理では、オーソリゼーションをユーザーに直接割り付けるのではなく、ユーザーグループに割り付けます。たとえば、"マスター"という名前のユーザーを"コミショニングエンジニア"ユーザーグループに割り付けて、オーソリゼーションを付与します。各オーソリゼーションは各ユーザーに個々に割り付けられるのではなく、あくまでユーザーグループに割り付けられます。

環境が異なる、たとえば顧客側では、ユーザーが変更されます。ただし、プロジェクトのオーソリゼーションとユーザーグループは変更されません。ユーザーだけが、"コミショニングエンジニア"などのユーザーグループに再度割り付けられます。

ユーザー管理では、ユーザーの管理とオーソリゼーションの設定が区別されます。この結果、確実に、柔軟性の高いアクセス保護を実行できます。

## 11.3 エlementと基本設定

### 11.3.1 "ユーザー"ユーザー管理

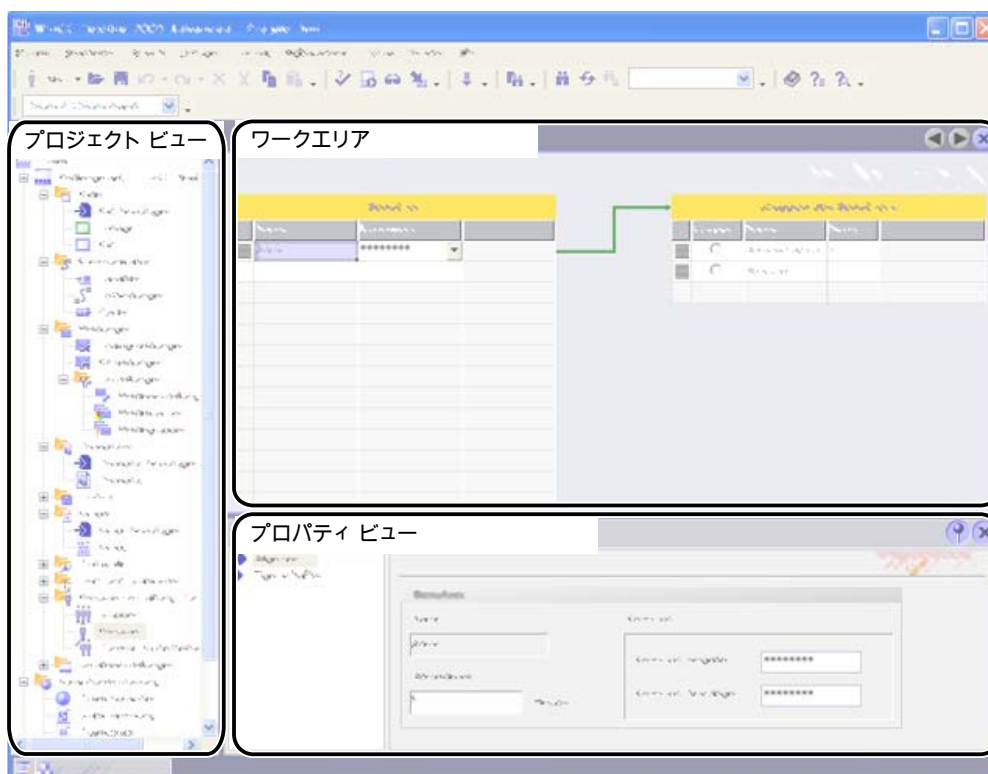
#### はじめに

ユーザー管理では、ランタイム時にデータとファンクションへのアクセスをコントロールするために、ユーザーとユーザーグループを管理します。ユーザー管理は、ユーザーの管理とユーザーグループの管理に分かれます。このセクションでは、ユーザーの管理について説明します。

#### 開く

[ユーザー]をダブルクリックして、[プロジェクト]ウィンドウでユーザーの管理を開きます。

#### ストラクチャ



#### 作業エリア

[ユーザー]作業エリアは、割り付けられる既存のユーザーとユーザーグループを表示します。

#### 注記

ユーザーは、1つのユーザーグループにしか割り付けることができません。

## [プロパティ]ウィンドウ

ユーザーが選択されている状態で、パスワードおよびユーザーが自動的にログオフされる時刻を[全般]グループ内で編集します。

### 11.3.2 "グループ"ユーザー管理

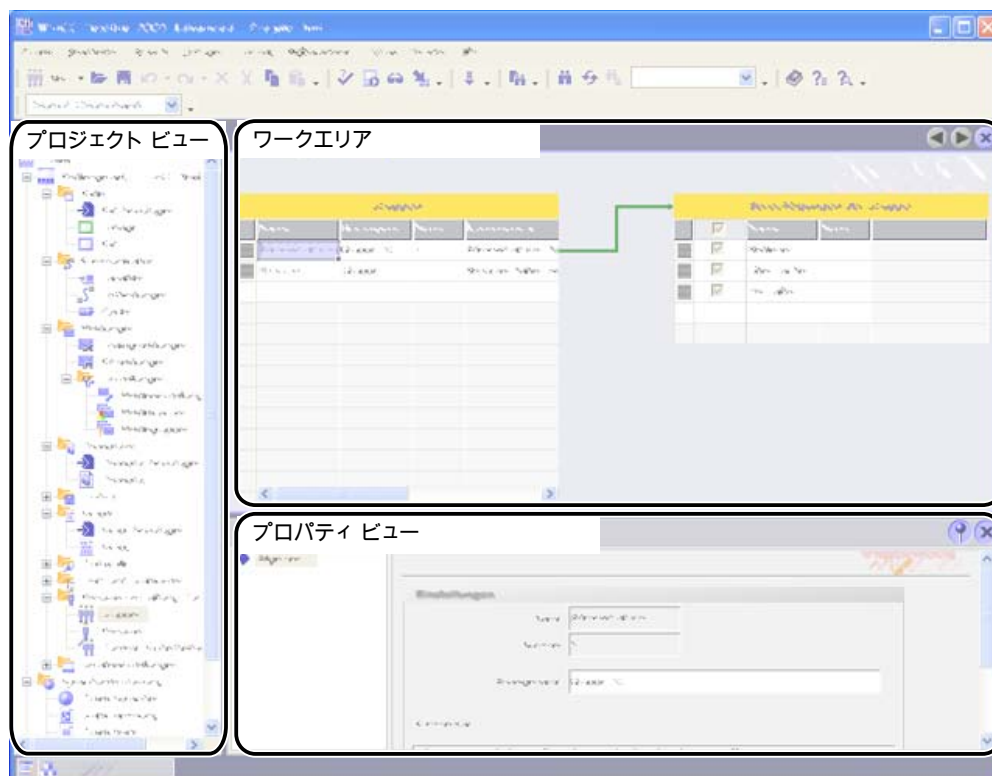
#### はじめに

ユーザー管理では、ランタイム時にデータとファンクションへのアクセスをコントロールするために、ユーザーとユーザーグループを管理します。ユーザー管理は、ユーザーの管理とユーザーグループの管理に分かれます。このセクションでは、ユーザーグループの管理について説明します。

#### 開く

[グループ]をダブルクリックして、[プロジェクト]ウィンドウでユーザーグループの管理を開きます。

#### ストラクチャ



## 作業エリア

[グループ]作業エリアは、既存のユーザーグループとそれらのオーソリゼーションを表示します。

## [プロパティ]ウィンドウ

ユーザーグループまたはオーソリゼーションが選択されている場合、[全般]グループでユーザーの指定およびコメントを編集することができます。

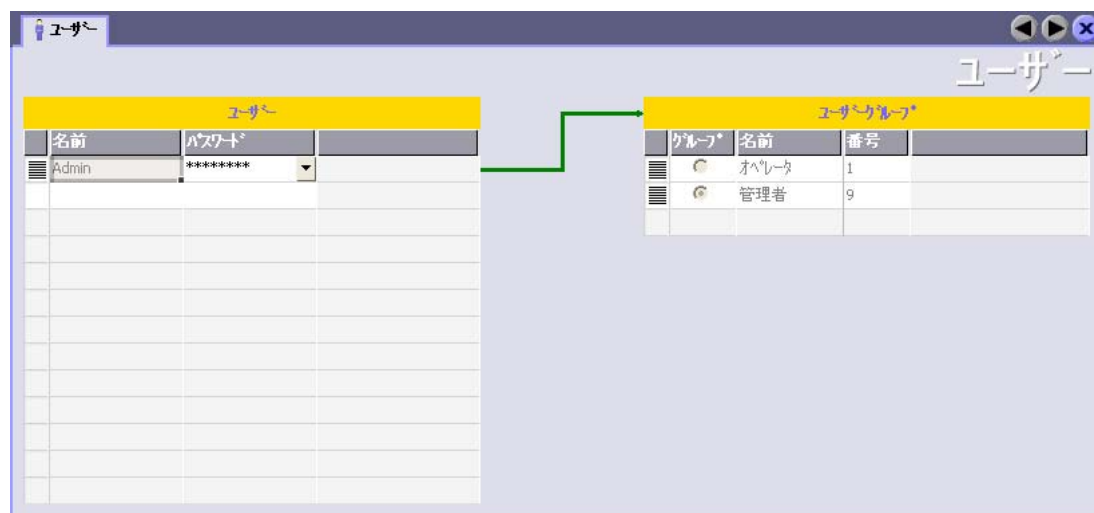
### 11.3.3 ユーザー作業エリア

#### はじめに

[ユーザー]作業エリアには、ユーザーとユーザーグループとそれらのオーソリゼーションのテーブルが表示されます。ユーザーを管理して、それらをユーザーグループに割り付けます。

#### 原理

作業エリアは、[ユーザー]と[ユーザーグループ]テーブルで構成されます。



[ユーザー]テーブルは、既存のユーザーを表示します。このテーブルでユーザーを選択する場合、[ユーザーグループ]テーブルには、ユーザーが割り付けられているユーザーグループが表示されます。

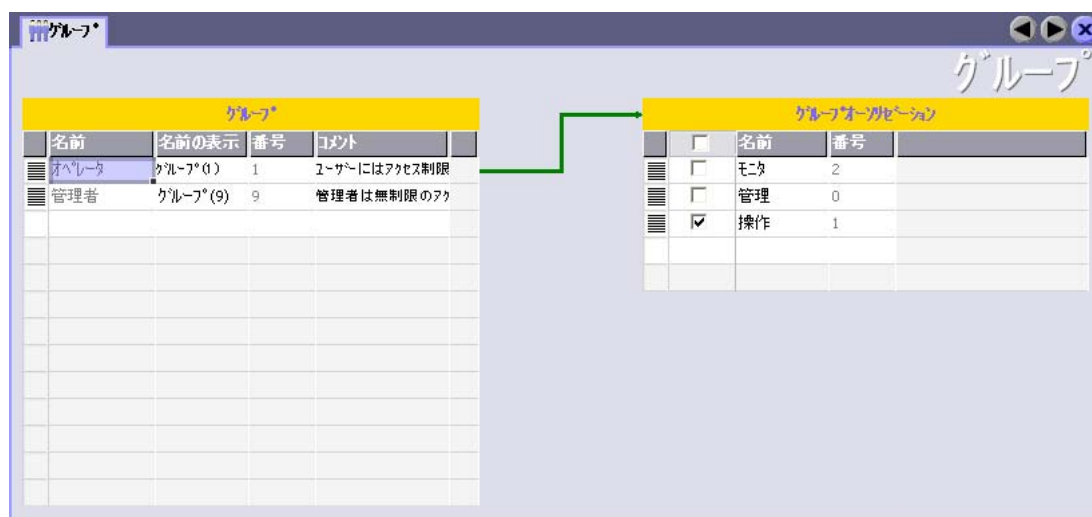
### 11.3.4 ユーザーグループ作業エリア

#### はじめに

[グループ]作業エリアには、ユーザーグループとそれらのオーソリゼーションのテーブルが表示されます。ユーザーグループを管理して、それらにオーソリゼーションを割り付けます。

#### 原理

作業エリアは[グループ]と[グループオーソリゼーション]テーブルで構成されます。



[グループ]テーブルには、既存のユーザーグループが表示されます。このテーブルでユーザーグループを選択すると、[グループオーソリゼーション]テーブルには、ユーザーグループに割り付けられているオーソリゼーションが表示されます。

ユーザーグループとそのオーソリゼーションの数は、[ユーザー管理]で割り付けられます。ユーザー指名とコメント記述は管理者が割り付けます。

### 11.3.5 ランタイムセキュリティの設定

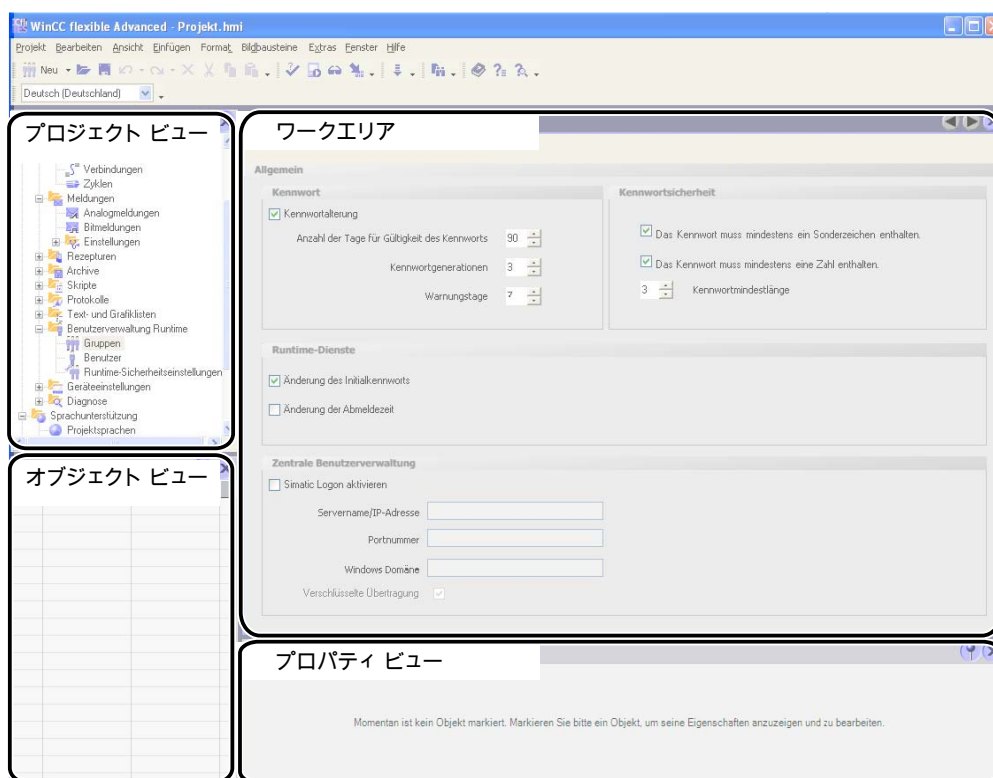
#### はじめに

"ランタイムセキュリティ設定"エディタを使用すれば、ランタイム中にユーザーパスワードのセキュリティを設定できます。

#### 開く

[プロジェクト]ウィンドウで、[ランタイムユーザー管理]グループ内の[ランタイムセキュリティ設定]をダブルクリックします。

#### 構造



#### 作業エリア

作業エリアでグループパスワードの期限を定義します。パスワードでの特定の文字や数字の使用を許可または却下したり、ユーザーがログインパスワードを入力するだけでよいようにします。

SIMATIC ログオンを使用してユーザー認証することで、一元的なユーザー管理の対応と非対応を決定できます。

## ユーザー管理のエクスポートとインポート

ユーザー管理がエクスポートされる場合、そのエクスポートファイルには、ユーザーの現在の有効パスワードだけが格納されます。以前のパスワードは保存されません。

このため、HMI デバイスへのユーザー管理のインポート後には、ランタイムセキュリティ設定に関係なく、次のルールが適用されます。

- 初めてシステムにログオンしたときに、ユーザーは、管理者が割り付けたパスワードを保持できます。
- 新規パスワードは、以前のパスワードと比較されません。これは、以前のパスワードは使用できないからです。
- パスワードの満了期間は、ユーザー管理をインポートしたときから始まります。

"ExportImportUserAdministration"システムファンクションを使用したユーザー管理のインポートは、監査トレイルに記録されます。

管理者が、電子署名またはコメント付きユーザー管理のインポートを確認する必要がある場合、"NotifyUserAction"システムファンクションを追加設定します。

"ExportImportUserAdministration"システムファンクションを使用したユーザー管理のエクスポートは、監査トレイルに記録されます。

---

### 注記

#### ユーザーデータのエクスポートのシミュレーション

シミュレーション時に、ユーザーデータのエクスポートもテストできます。シミュレーションの完了後、PC のパスワードリストを使用して、エクスポートファイルを削除します。これによって、権限を与えられていない人がパスワードにアクセスできないことを確認します。

---

## 11.4 ユーザー管理の使用

### 11.4.1 SIMATIC ログオンを使用した一元的ユーザー管理

#### はじめに

SIMATIC ログオンを有効にして、複数のアプリケーションまたは複数の HMI デバイスのユーザーおよびユーザーグループを一元的に管理します。

#### 原理

SIMATIC Logon は、システム全体でユーザー管理を行うためのツールです。外部のアプリケーションや HMI デバイスのユーザーは、SIMATIC Logon アプリケーションを使用して一元的に管理されます。

WinCC flexible のローカルユーザー管理と同じように、ユーザーグループと対応するオーソリゼーションを設定します。サーバー上と WinCC flexible でまったく同じ名前をユーザーグループに割り付けます。この同じ名前に基づいて、ユーザーは、Runtime の各ユーザーグループのオーソリゼーションに割り付けられます。

ドメインサーバーでのユーザーの作成 Runtime にログオンしたユーザーのデータは、サーバーでダイナミックに統一されます。サーバーで認識されたユーザーは、各ユーザーグループのオーソリゼーションに割り付けられます。アプリケーションまたは HMI デバイスは、SIMATIC Logon を使用してログオン要求やパスワード変更を処理するため、サーバーに転送します。

---

#### 注記

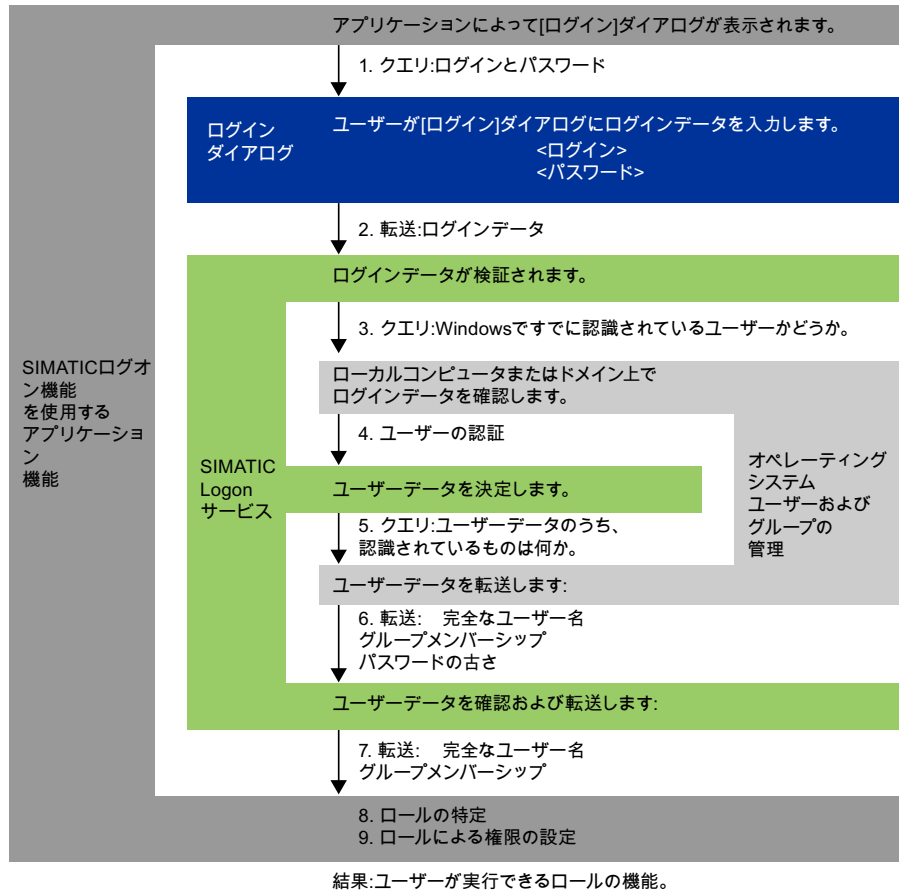
SIMATIC Logon は、ライセンスを必要とする製品です。SIMATIC Logon の詳細については、<<http://support.automation.siemens.com>>にアクセスしてください。

検索フィールドに ID である"22657587"を入力して検索を開始します。『SIMATIC ログオン - 電子署名』マニュアルは、ダウンロードして入手できます。

---

### SIMATIC Logon Service を経由したログオンプロセス

次の図は、ユーザーが Runtime にログオンした際に自動的に実行されるプロセスを示しています。



## 11.4.2 ランタイム時のユーザー

### 原理

エンジニアリングシステムでは、ユーザーとユーザーグループを作成して、オーソリゼーションを割り当てます。オーソリゼーションが指定されたオブジェクトを設定します。HMI デバイスへの転送後、オーソリゼーションによりコンフィグレーションしたオブジェクトはすべて、Runtime での未許可のアクセスから保護されます。

### [ユーザー]ウィンドウ;[ユーザー]ウィンドウ

エンジニアリングシステムで[ユーザー]ウィンドウを設定すると、HMI デバイスへの転送後に[ユーザー]ウィンドウでユーザーを管理できます。

#### 注意

[ユーザー]ウィンドウ内の変更は、ランタイム中は直ちに有効になります。ランタイム中の変更は、エンジニアリングシステムでは更新されません。ユーザーとユーザーグループがエンジニアリングシステムから HMI デバイスへの転送されたとき、ユーザーへのプロンプトの後、転送の設定に基づいて[ユーザー]ウィンドウのすべての変更が上書きされます。

HMI デバイスによっては、[ユーザー]ウィンドウをサポートしていないものもあります。[ログオン]および[ログオフ]のファンクションをサポートするのは HMI デバイスだけです。[管理者]ユーザーだけがログオンおよびログオフします。"管理者"を割り付けられるのは、ユーザーグループ"管理者"だけです。

### ユーザーデータのエクスポートとインポート

システムファンクションを使用して、HMI デバイスの既存のユーザーとパスワードを、異なるオペレータパネルにエクスポートおよびインポートします。この結果、各種 HMI デバイスのユーザー管理を確実に同じステータスにしておくことができます。

#### 注記

WinCC flexible 2004 を使用してユーザーデータをエクスポートすると、その後、このファイルを WinCC flexible 2005 にインポートできます。

WinCC flexible 2005 を使用してユーザーデータをエクスポートすると、その後、このファイルを WinCC flexible 2004 にインポートできません。

### 11.4.3 [ユーザー]ウィンドウ

#### 目的

ランタイムでユーザーの管理も行うことができるように、エンジニアリングシステムで [ユーザー]ウィンドウを設定します。

#### はじめに

ユーザーとユーザーグループは、エンジニアリングシステムで作成され、HMI デバイスに転送されます。"管理"オーソリゼーションを所持しているユーザーは、ユーザー全員を管理するために、[ユーザー]ウィンドウに無制限にアクセスできます。ほかのユーザーは、ユーザー自身を管理するために、自分の[ユーザー]ウィンドウだけにアクセスできます。

#### 注意

[ユーザー]ウィンドウ内の変更は、ランタイム中は直ちに有効になります。ランタイム中の変更は、エンジニアリングシステムでは更新されません。

ユーザーとユーザーグループがエンジニアリングシステムから HMI デバイスへ転送されると、[ユーザー]ウィンドウの変更がすべて上書きされます。

#### レイアウト

[ユーザー]ウィンドウでは各行に、ユーザー、そのパスワード、そのユーザーが所属するユーザーグループ、ログオフ時間が表示されます。

ユーザー	パスワード	グループ	ログオフ時間

#### 注記

ユーザーがログオンしていない場合、[ユーザー]ウィンドウは空になります。[ユーザー]ウィンドウをクリックすると、ログオンダイアログボックスが開きます。個々のフィールドの内容は、ログオン後に表示されます。

## 管理者の[ユーザー]ウィンドウ

ユーザー	パスワード	グループ化	ログオフ時間
Admin	*****	グループ (9)	5
Operator	*****	グループ (9)	5
PLC User	*****	グループ (1)	5

管理者がログオンすると、[ユーザー]ウィンドウにユーザーが全員表示されます。管理者は、ユーザー名およびパスワードを変更します。管理者は、ユーザーを新たに作成して、既存のユーザーグループに割り付けることができます。

## ユーザーの[ユーザー]ウィンドウ

ユーザー	パスワード	グループ	ログオフ時間
Miller	*****	Programmer	5

ログオンした人が管理者でない場合、[ユーザー]ウィンドウには、ログオンしたユーザーだけが表示されます。ユーザーは、自身のパスワードを変更できます。名前を変更するには、ユーザーが、管理者としてログオンする必要があります。

## 注記

SIMATIC ログオンユーザーがログインすると、ユーザー名に加えてドメインが表示されます。

#### 11.4.4 アクセスセキュリティ

##### はじめに

未許可のアクセスからオブジェクトを保護するために、オブジェクトにオーソリゼーションを設定します。このオーソリゼーションを所持しているログオンユーザーはすべて、このオブジェクトにアクセスできます。ユーザーがオブジェクトを操作するオーソリゼーションを持たない場合、ログオンダイアログが自動的に表示されます。

---

##### 注記

"ユーザー管理"下ではシステムファンクションを複数使用できます。たとえばコントロールシステムでユーザー、パスワード、およびユーザーグループを編集できます。

---

## システムファンクションとランタイムスクリプト

### 12.1 基本

#### 12.1.1 システムファンクションとランタイムスクリプト

##### はじめに

WinCC flexible では、共通のコンフィグレーションタスクに対応する所定のシステムファンクションが用意されています。システムファンクションを使って、ランタイム中にいくつかのタスクを実行することができます。そのためのプログラミングのスキルは必要ありません。

より複雑な問題の解決にはランタイムスクリプトを使用することができます。ランタイムスクリプトには、ランタイム中のプロジェクトデータの部分にアクセスできるプログラミングインターフェースが備わっています。ランタイムスクリプトは、Visual Basic (VB)と Visual Basic Script (VBS)の知識があるプロジェクトプランナを使用対象としています。

##### システムファンクションの使用

システムファンクションは、オペレーティングエレメントへのファンクション割り付けをサポートします。

- PCL にビットを設定します。
- タグの値を変更します。
- ロギングの開始

システムファンクションは、ファンクションリストとスクリプトで設定できます。

---

**注記**

内部タグの[値の変更]イベントでファンクションまたはスクリプトをコンフィグレーションする場合、任意の製品グループでは実行できません。

例:

ファンクション IncreaseValue(V,+1)は内部タグの値の変更イベント[V]でコンフィグレーションされます。

バージョン 1:

同じファンクション"IncreaseValue(V,+1)"はキーとしてコンフィグレーションされます。キーを押すと、タグの値 V が変更されます。

[V]の値の変更でファンクションが実行されると、常に相互に隣接しているファンクションによって HMI デバイスが操作不能になります。そのため、ファンクションの実行ができなくなっています。

バージョン 2

入力フィールドに入力すると、タグの値[V]が変更されます。この場合、"IncreaseValue(V,+1)"ファンクションが正確に一度実行されます。

---

## ランタイムスクリプトの使用

ランタイムスクリプトは OP270/TP270 から入手できるため、WinCC flexible Standard から入手できます。プログラミング言語として VBScript がサポートされています。ランタイムスクリプティングを使用すれば、コンフィグレーションを実現する際の柔軟性が高くなります。たとえば、ランタイム中に次の追加機能が必要になる場合、Runtime スクリプティングによってスクリプトを作成します。

- 値の変換

スクリプトを使用すれば、各種測定単位間で値を変換できます。たとえば、温度が挙げられます。

- 製造シーケンスの自動化

スクリプトを使用して生産データを PLC に転送すれば、製造シーケンスをコントロールできます。ステータスは戻り値でチェックすることができ、必要に応じて適切な指標を使用することもできます。

## スクリプト

自分専用の VB スクリプトコードをスクリプトに保存できます。プロジェクトでは、スクリプトをシステムファンクションと同じように使用できます。スクリプトでは、プロジェクトのタグ、および WinCC flexible のランタイムオブジェクトモデルにアクセスできます。さらに、このスクリプトでは、標準 VBS ファンクションをすべて使用できます。スクリプトで、ほかのスクリプトとシステムファンクションを呼び出すことができます。

## システムファンクションとスクリプトの実行

ボタン上でマウスをクリックするなど、設定したイベントが開始されると、ランタイム中にシステムファンクションとスクリプトが実行されます。

## 反復レベル;ハンプクレベル

スクリプトの反復レベルは、HMI デバイスのスタックサイズにより限定されます。ランタイム中、反復が無制限に発生すると、システムエラーメッセージが表示されます。このため、各スクリプト内の反復の回数を制限してください。

### 12.1.2 システムファンクション

#### はじめに

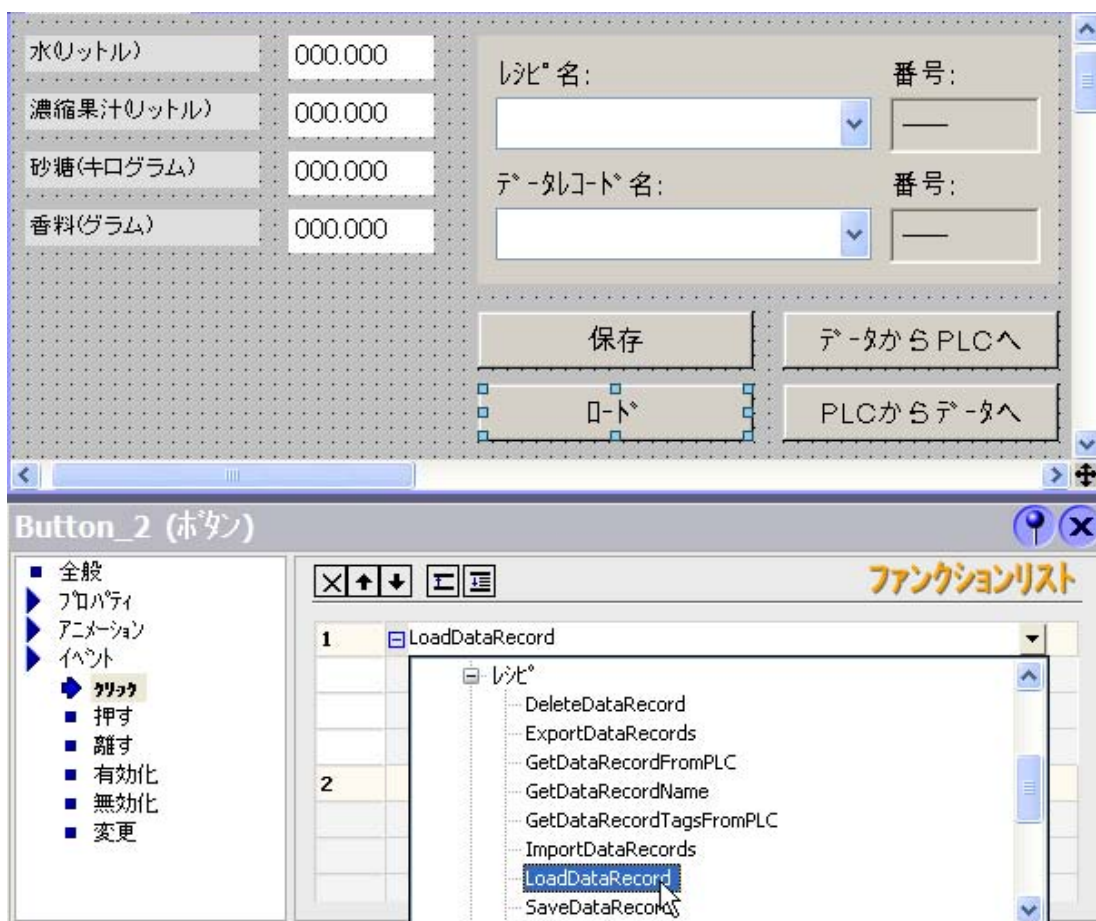
システムファンクションとは、事前定義ファンクションであり、システムファンクションを使用すれば、プログラミングの知識がなくても、ランタイム中に数多くのタスクを実行できます。たとえば、次のタスクを実装できます。

- 計算。たとえば一定のまたは変動する量だけ変数の値を増加する場合は挙げられます。
- ログ機能。たとえば、プロセス値ログの開始。
- 設定。たとえば、PLC の変更、PLC 内でのビットの設定。
- アラーム。たとえば、異なるユーザーのログオン後。

## アプリケーション

ファンクションリストまたはスクリプトの中でシステムファンクションを使うことができます。システムファンクションは事前定義ファンクションであるため、変更することはできません。

ファンクションリストのコンフィグレーション時、カテゴリでソートされた選択リストからシステムファンクションを選択します。



スクリプトでシステムファンクションを使用する場合、選択リストからシステムファンクションを選択できます。<Ctrl+Space>キーを使用すれば、スクリプトで選択リストを呼び出すことができます。

## 言語依存性

システムファンクションの名前は、設定したプロジェクト言語によって違ってきます。このため、プロジェクトプランナは直ちに機能を認識できます。

例外: スクリプトの中でシステムファンクションを呼び出す場合には、システムファンクションの英語名を使用してください。システムファンクションの英語名については、システムファンクションのリファレンスを参照してください。

## 使用可能性

WinCC flexible で作成できるのは、選択した HMI デバイスでサポートされている機能だけです。したがって、ファンクションリストで作成できるのは、選択したオペレーティングユニットでサポートされているシステムファンクションだけです。1つのプロジェクトを複数のオペレーティングユニットで使用する場合、オペレーティングユニットのどれか1つでもサポートしていないシステムファンクションがあると、そのシステムファンクションには色でマークされます。

### 12.1.3 システムファンクションの使用

#### はじめに

ファンクションリストは、コンフィグレーションされたイベントがランタイム中に発生したときに処理されます。オペレータは、たとえば、HMI デバイス上でファンクションキーを押せば、イベントをトリガできます。またイベントは、たとえば、プロセス値が限界値を下回った場合などにシステムによってトリガされることがあります。

#### アプリケーション

イベントに反応できるオブジェクトすべてにシステムファンクションをコンフィグレーションできます。ファンクションリストでもスクリプトでもシステムファンクションを直接使用できます。この結果、ファンクションリストやスクリプトの進路を制御できます。

- ファンクションリスト;ファンクションリスト

システムファンクションは、ファンクションリストで、最初のシステムファンクションから最後のシステムファンクションに向かって逐次処理されます。待ち時間を避けるため、比較的長いランタイムをもつシステムファンクション(たとえばファイル操作など)は、同時に処理されます。たとえば、直前のシステムファンクションがまだ完了していないうちに次のシステムファンクションを実行できます。

ファンクションリストのコンフィグレーションの例は以下のとおりです:現在表示中の HMI デバイスの動作モードの変更

- スクリプト

スクリプトでは、コード内の順序および必要条件と関連付けてシステムファンクションを使用できます。この結果、特定のシステム状態に対応してスクリプトを実行できます。さらに、たとえば、システムファンクションの戻り値を評価できます。戻り値に応じて、テストファンクションを実行して、たとえば、スクリプトの進路に順次に影響を及ぼすことができます。

## 12.1.4 スクリプト

### はじめに

スクリプトでは VB スクリプトコードをプログラムします。完成したスクリプトは、システムファンクションと同じようにプロジェクトで使用できます。スクリプトを作成する場合、スクリプトのタイプを決定し、転送パラメータを定義します。タイプ"Function"のスクリプトには、戻り値が含まれます。"Sub"タイプのスクリプトはプロシージャとも呼ばれ、戻り値はありません。

### スクリプトのプロパティ

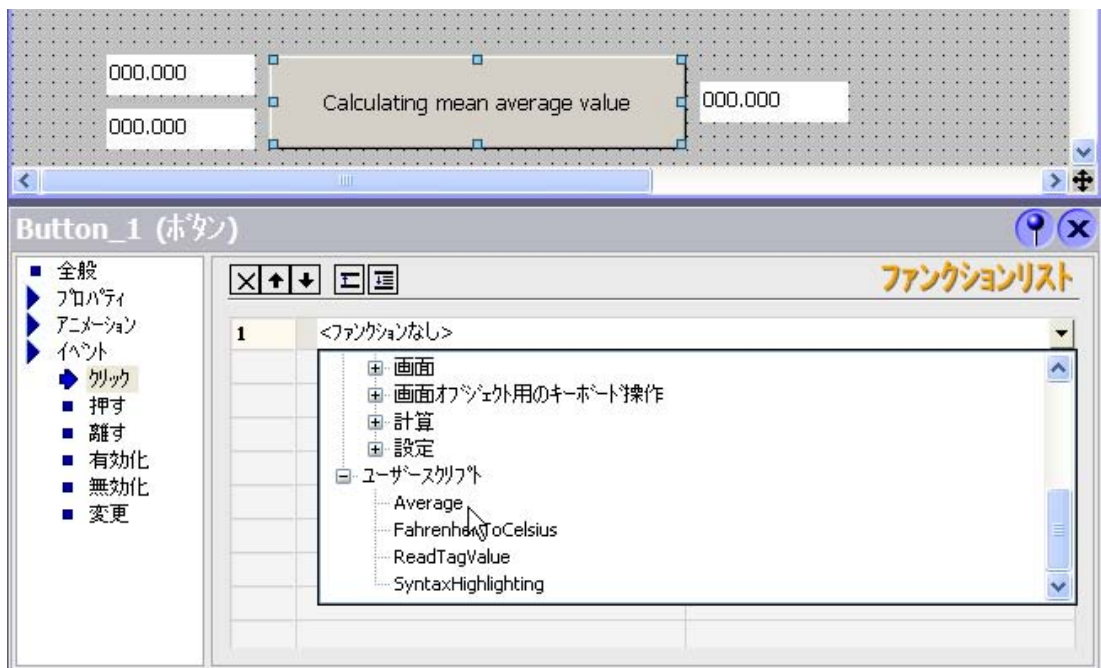
スクリプトの中で、ほかのスクリプトとシステムファンクションを呼び出すことができます。ランタイムオブジェクトモデルを使用すれば、WinCC flexible のランタイムオブジェクトにアクセスできます。システムファンクションを呼び出す場合には、システムファンクションの英語名を使用してください。スクリプトでは、Microsoft VBScript の言語をすべて使用できます。ユーザーの操作に対応したファンクションとメソッドが実行されます。たとえば、"MsgBox"が挙げられます。

設定されたオペレーティングユニットで使用できないシステムファンクションをスクリプト内で使用した場合、警告メッセージが表示されます。さらに、スクリプト内のそれぞれのシステムファンクションには、青い波線のアンダーラインが引かれます。

### スクリプトのオーガニゼーション

スクリプトはプロジェクトデータベースに保存されます。使用可能なスクリプトは、[プロジェクト]ウィンドウの[スクリプト]の下にリストされます。

ファンクションリストでスクリプトを使用する場合、[スクリプト]下の選択リストにスクリプトが表示されます。



## 12.1.5 スクリプトの使用

### 原理

プログラミング言語の制御エレメントを使用すると、スクリプトの柔軟性が高くなります。ランタイム中にスクリプトを使用して、たとえば、任意のプロジェクトで個々のソリューションを実行します。

- 高度なファンクションリストの作成

スクリプトでシステムファンクションとほかのスクリプトを呼び出して、スクリプトをファンクションリストと同じように使用します。

条件に応じてスクリプトでシステムファンクションとスクリプトを実行したり、システムファンクションとスクリプトを繰り返し実行したりできます。その後、ファンクションリストにこのスクリプトを追加します。

- 新規ファンクションのプログラミング

スクリプトは、プロジェクト全体で使用できます。スクリプトは、システムファンクションを使用するのと同じように使用します。これらのスクリプトに、発信パラメータと戻り値を定義できます。たとえば、スクリプトを使用して値を変換します。

## 12.2 ファンクションリストの使用

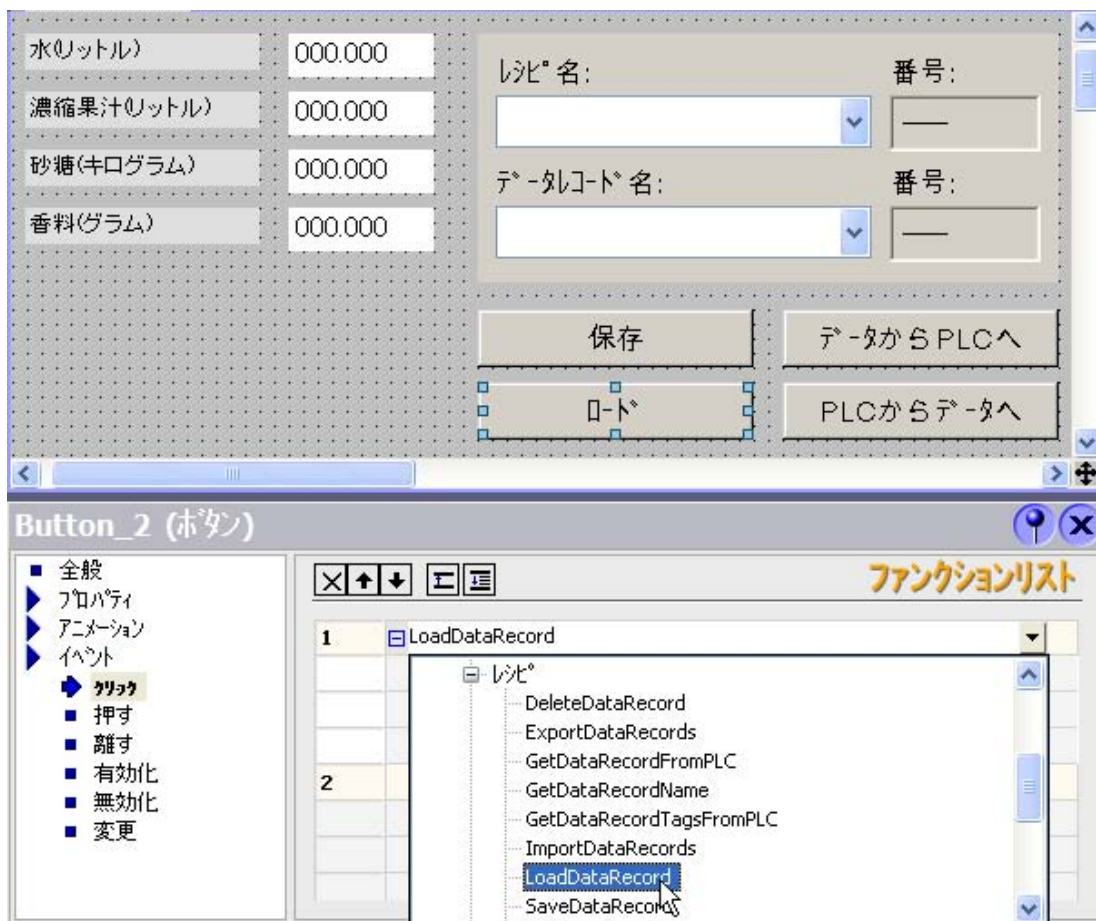
### 12.2.1 ファンクションリストの基本原則

#### はじめに

設定したイベントが発生すると、ファンクションリストによりシステムファンクションとスクリプトを複数実行できます。

#### 原理

ファンクションリストは、画面オブジェクトやタグなどのオブジェクトのイベントを設定します。使用可能なイベントは、選択したオブジェクトと HMI デバイスによって異なります。



イベントが発生するのは、プロジェクトがランタイムモードになっている場合に限ります。  
イベントに含まれる内容:

- タグの値変更
- 配列の値の変更 = 配列エレメントの値の変更
- ボタンを押す
- ランタイムの有効化

イベントごとにファンクションリストを正確に作成できます。

---

#### 注記

ファンクションリストの中で指定できるシステムファンクションの種類は、選択した HMI デバイスによって違ってきます。

---

## 12.2.2 ファンクションリストのプロパティ

### HMI デバイス依存性

各種 HMI デバイスにプロジェクトを 1 つ使用できます。プロジェクトの HMI デバイスを変更する場合、選択した HMI デバイスでサポートされていないシステムファンクションとスクリプトはすべて黄色でマークされます。サポートされていないシステムファンクションは、ランタイムでも実行されません。

### ステータス情報

コンフィグレーション中、プロジェクトデータはバックグラウンドでテストされます。各ファンクションリストで、ステータス情報は、各システムファンクションとスクリプトのステータスを返します。

ステータス情報には次の意味があります。

- オレンジ色:ランタイム中にファンクションリストが実行されません。これは、少なくとも 1 つのシステムファンクションまたはスクリプトにパラメータが完全には与えられていなかったためです。
- 黄色:ランタイム中にファンクションリストは実行されます。ただし、このファンクションリストに指定されたシステムファンクションまたはスクリプトの少なくとも 1 つが、HMI デバイスでサポートされていません。たとえばデバイスタイプを変更した場合に、この状態が発生します。

### システムファンクションとスクリプトの実行

ファンクションリスト内のシステムファンクションとスクリプトは、一番上から一番下に向かって逐次処理されます。待ち時間を避けるため、比較的長いランタイムをもつシステムファンクション(たとえばファイル操作など)は、同時に処理されます。たとえば、直前のシステムファンクションがまだ完了していないうちに次のシステムファンクションを実行できます。

逐次手順および条件付き手順のプログラムを避けるには、ループ、条件付きステートメント、および取り消し要求を含むスクリプトを使用します。

## 12.3 エlementと基本設定

### 12.3.1 "スクリプト"エディタ

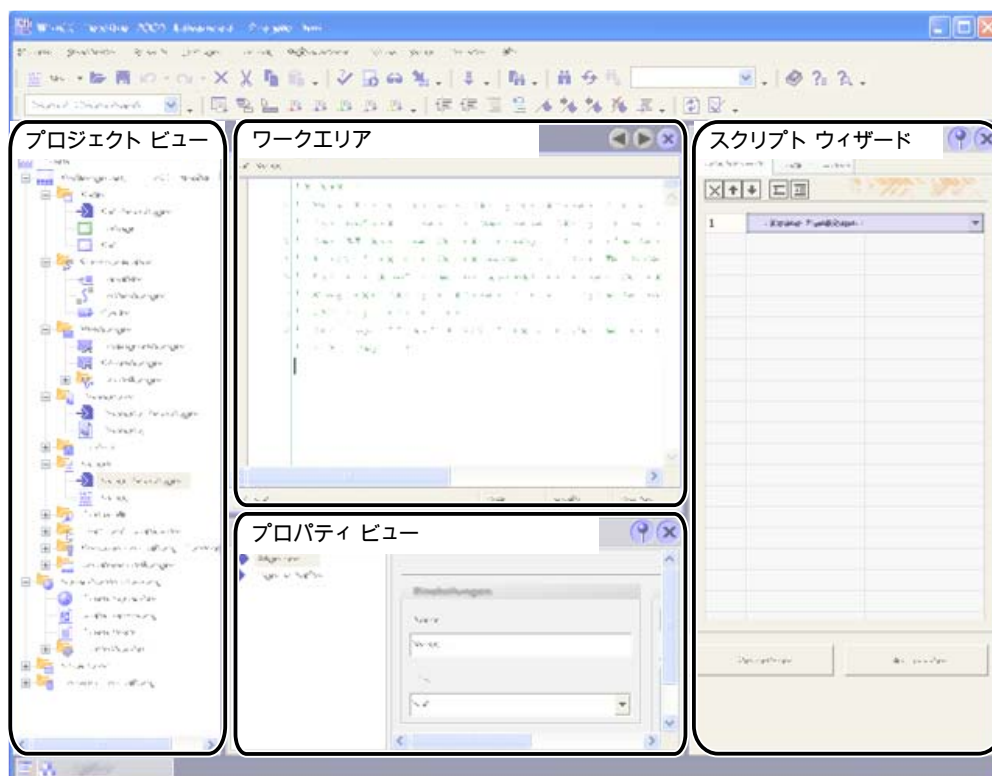
#### はじめに

"スクリプト"エディタでスクリプトの作成および編集を行います。

#### 開く

スクリプトエディタは、新規のスクリプトが作成されたとき、または既存のスクリプトが開いたときに自動的に開きます。

#### レイアウト



#### メニューバー

メニューバーには、WinCC flexible の操作に必要なすべてのコマンドがあります。使用可能なショートカットキーはすべてメニューコマンドの隣に表示されます。

## [スクリプト]ツールバー

[スクリプト]ツールバーには、オブジェクトとタグを同期化するコマンドやスクリプトの構文をチェックするコマンドがあります。

## [詳細編集]ツールバー

以下のアクションを行うためのコマンドは、[詳細編集]ツールバーにあります。

- ブックマークの取り扱い
- コードのインデントとインデント解除
- コードのコメント追加
- 特定のコード行へのジャンプ

## [IntelliSense]ツールバー

[IntelliSense]ツールバーには、選択リスト(例:オブジェクトモデルの全オブジェクト、使用可能なシステムファンクション、VBS 定数など)の表示コマンドがあります。

## 作業エリア

作業エリアでは、スクリプトの作成および編集を行います。スクリプトの作成は、構文の強調表示および IntelliSense によってサポートされます。

## [プロパティ]ウィンドウ

[プロパティ]ウィンドウでは、スクリプトのコンフィグレーションを行います。スクリプトがプロシージャであるのか、またはファンクションであるのかを定義します。さらに、スクリプトのパラメータを宣言することができます。

## [スクリプトウィザード]

[スクリプトウィザード]では、ファンクションリスト内と同様に、割り付けられたパラメータでシステムファンクションおよびスクリプトをセットアップすることが可能です。ファイル化されたシステムファンクションおよびスクリプトを、[スクリプトウィザード]から現在のスクリプトに移動することも可能です。この方法を使うと、パラメータ割り付けが1回で済みます。

システムファンクションまたはスクリプトが1つのイベントに対して既にコンフィグレーションされている場合は、それらを[スクリプトウィザード]にコピーアンドペーストで移動することができます。スクリプトで許容されているシステムファンクションだけが、[スクリプトウィザード]でファイル化できます。スクリプトで使用することのできないシステムファンクションをコピーアンドペーストで移動する場合、そのシステムファンクションにはマークが付きます。

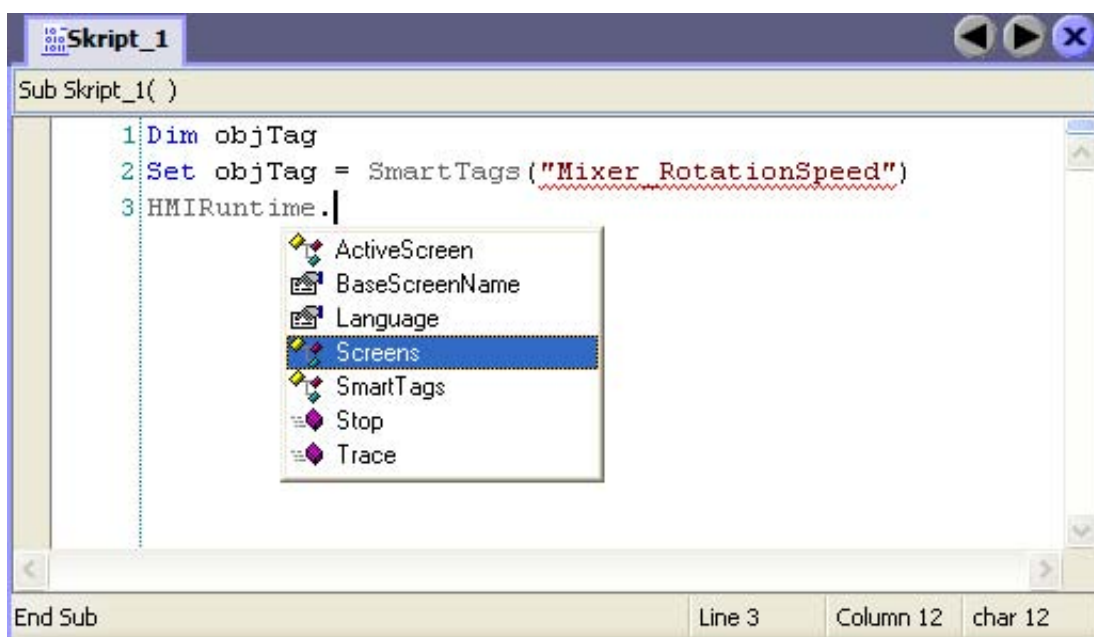
## 12.3.2 [スクリプト]エディタのプロパティ

## はじめに

[スクリプト]エディタは、IntelliSense、強調表示された構文およびその他の機能でプログラムをサポートします。たとえば、タグの参照はドラッグアンドドロップで作成することができます。

## IntelliSense

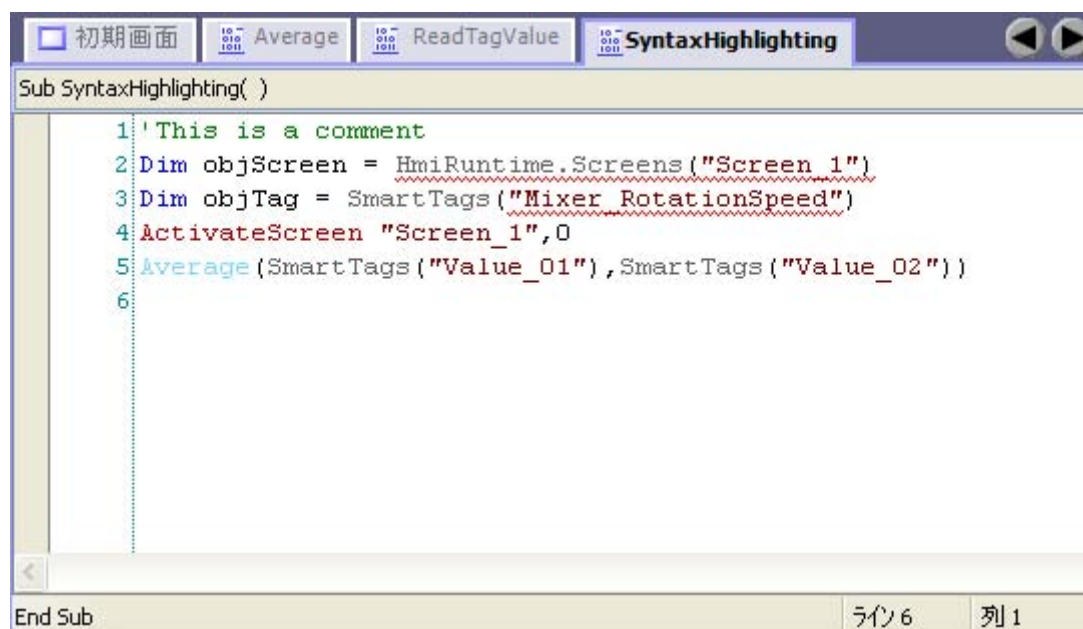
VBS オブジェクトモデルのオブジェクト、メソッドまたはプロパティを利用する場合は、IntelliSense でサポートされます。



指定されたオブジェクトが持つメソッドおよびプロパティは、選択リストから選ぶことができます。

## 構文の強調表示

[スクリプト]エディタでは、キーワードが異なる色で強調されます。[スクリプト]エディタで認識されるオブジェクトは太字で表示されます。未定義の単語には、赤色の波線の下線が引かれます:



```

Sub SyntaxHighlighting( )
1 'This is a comment
2 Dim objScreen = HmiRuntime.Screens("Screen 1")
3 Dim objTag = SmartTags("Mixer_RotationSpeed")
4 ActivateScreen "Screen_1",0
5 Average(SmartTags("Value_01"),SmartTags("Value_02"))
6
End Sub

```

最も重要なキーワードを表す既定の色を、表に記載します。

色	意味	例
青	キーワード(VBS)	Dim, If, Then
灰色	キーワード(オブジェクトモデル)	HmiRuntime
紺	スクリプト	FahrenheitToCelsius
茶	システムファンクション	IncreaseValue
赤	タグ	Tag_1
緑	コメント	'これはコメントです'

## オブジェクトの同期化

スクリプトを開くと、オブジェクト(タグなど)のインスタンスは自動的にコンフィギュレーションデータと同期が取られます。[タグ]エディタでタグの名称を変更した場合などは、この変更内容はスクリプトにも反映されます。名称の変更を行い、関連するスクリプトが開いている場合、名称が変更されたオブジェクトに青色の波線で下線が引かれます。これにより、[スクリプト]エディタで手動で同期を取ることも可能です。

### オブジェクトリスト

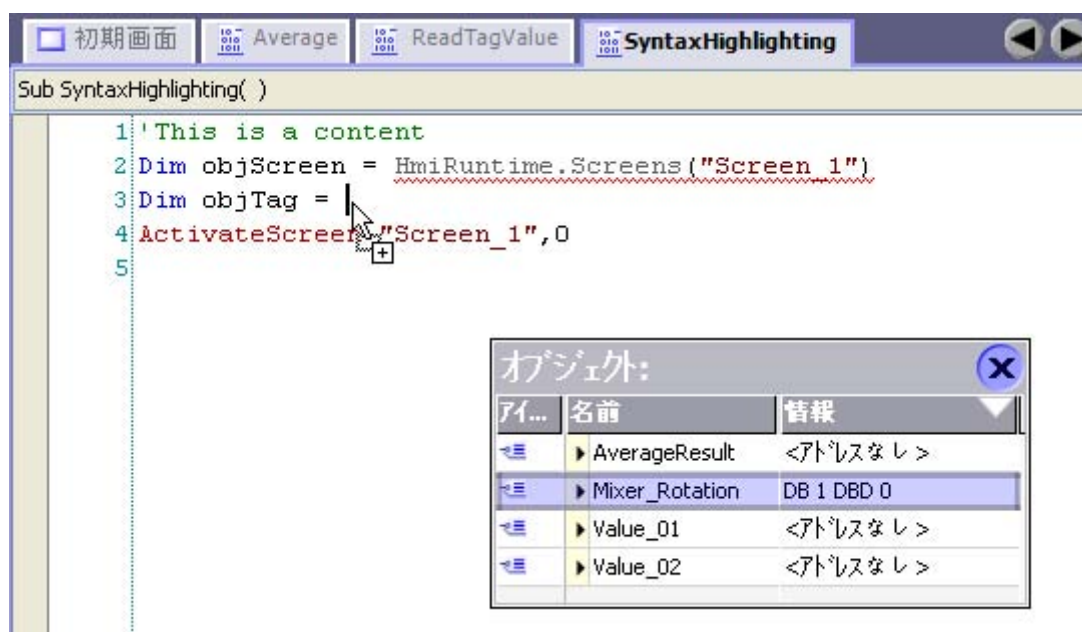
<Alt+右矢印>キーを押すと、オブジェクトリストを呼び出すことができます。オブジェクトリストには、コード文脈に依存する使用可能なオブジェクトがすべて表示されます。たとえばパラメータを割り付けるとき、またはリストを使用しているときなどにオブジェクトリストを呼び出せます。

例:[画面リスト]を使用して既存のプロセス画面を参照するには、[スクリプト]エディタに "HmiRuntime.Screens"と入力し、<Alt+右矢印>キーでオブジェクトリストを呼び出します。プロジェクト内の既存のプロセス画面は、すべてオブジェクトリストでリストされています。

目的のプロセス画面を選択し、<Return>キーで選択を確定します。

### ドラッグアンドドロップ

スクリプト中でタグが必要な場合は、[オブジェクト]ウィンドウから引用することができます。

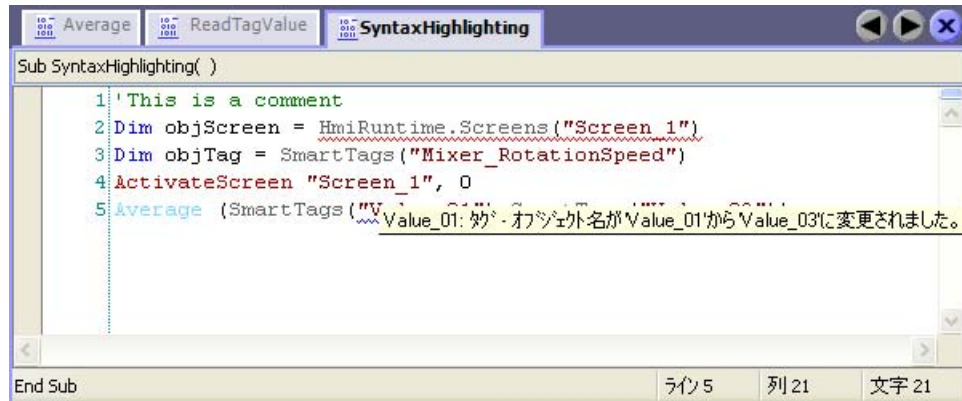


## ヘルプファンクション

プログラミング中、メソッドおよびシステムファンクションの必須パラメータに関する簡単な説明が自動的に表示されます。さらに、[スクリプト]エディタでは下記のヘルプファンクションを使用することができます:

- ツールヒント

認識できない、または誤ったキーワードには波線で下線が引かれます。キーワード上にマウスを移動すると、ツールヒントが表示されます。



認識できるキーワードについては、ツールヒントにキーワードのタイプが表示されます。

- パラメータ情報

[パラメータ情報]では、システムファンクションまたは VBS の標準ファンクションの構文およびパラメータに関する情報が提供されます。

- 文脈依存のヘルプ

状況対応のヘルプは、システムファンクション、VBScript 言語要素、オブジェクトなどに関する情報を提供します。

オブジェクト、メソッドまたはプロパティについての情報が必要な場合は、対象となるキーワード上にマウスポインタを移動し<F1>キーを押します。これで、オンラインヘルプの対応する参照説明が表示されます。

### 12.3.3 基本設定

#### はじめに

[スクリプト]エディタの基本設定は、個々に調整することができます。[ツール]メニューの[設定]コマンドを選択します。[スクリプト]グループの基本設定を変更します。

#### エディタ設定の変更

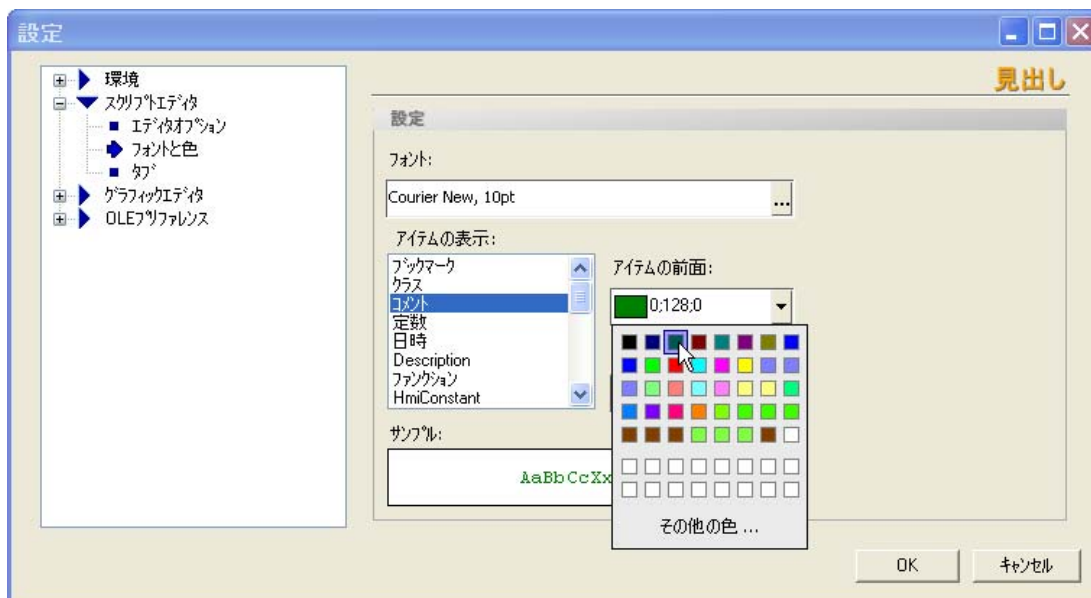
スクリプトのエディタ設定は、変更することができます。たとえば、スクリプトの情報テキストを非表示にすることができます。

エディタ設定を変更するには、[スクリプト]グループの[エディタ設定]のエントリをクリックします。

### 構文の強調表示の変更

それぞれのテキストエレメントに対する既定の色を変更することができます。[スクリプト]グループの[フォントと色]エントリをクリックします。

[テキストエレメント]選択リストから、目的のテキストエレメントを選択します。次に、目的の色を選択するか、新規の色を定義します。



### コードフォーマットの変更

コードフォーマットは以下の手順で変更することができます:

- フォントのタイプとサイズを変更するには、[スクリプト]グループの[フォントと色]のエントリをクリックします。
- インデントとタブストップを変更するには、[スクリプト]グループの[インデント]のエントリをクリックします。

### HMI デバイスに転送する際の設定

次のランタイム用デバイス設定を使用すると、スクリプトの転送後に HMI デバイスのメモリ容量が節約されます。

- 名前情報の転送
- スクリプトコメントの表示

オブジェクト名ではなく番号が転送されます。この設定は、スクリプトのテストでエラーがない場合に限り使用してください。デバッガでスクリプトをテストする際には、オブジェクト名およびコメントが表示され、コードがより分かりやすくなります。

## 12.4 スクリプトの作成

### 12.4.1 タグへのアクセス

#### はじめに

プロジェクト内に設定した外部と内部のタグにスクリプトでアクセスすることができます。タグの値をランタイム中に読み出したり、変更したりすることができます。

さらに、スクリプト内にカウンタやバッファ記憶領域としてローカルタグを設定することができます。

スクリプトは Runtime メモリから、外部タグの値を取り出します。Runtime の開始時に、現在値が PLC から読み取られ、Runtime メモリに書き込まれます。次に、タグ値が更新されてサイクルタイムを設定します。スクリプトは、前のスキャンサイクルチェックポイントで PLC から読み取られたタグ値に、まずアクセスします。

#### プロジェクトタグ

プロジェクト内のタグ名が VBS の命名規則に対応している場合は、スクリプトから直接タグを使用することができます。

```
'VBS_Example_03  
If BeltDriveOilTemperature > 100 Then [instruction]
```

プロジェクト内のタグ名が VBS の命名規則に対応していない場合は、SmartTags リストを使ってタグを参照する必要があります。以下の例には、VBS の命名規則では許可されていない&記号がタグ名に含まれています。

```
'VBS_Example_04  
SmartTags("Test&Trial")= 2005
```

VBS の命名規則については、情報システム内の VBS 用のヘルプを参照してください。

#### ローカルタグ

ローカルタグは、Dim ステートメントを使ってスクリプト内で定義することができます。ローカルタグはスクリプト内部でだけ使用することができます。そのため、"タグ"エディタには表示されません。

たとえば、スクリプトでは、For ステートメント用のカウンタとしてローカルタグを使用します。

```
'VBS_Example_05  
Dim intCount  
For intCount = 1 To 10[Instruction]Next
```

---

#### 注記

"For"のタグが必要な場合、ローカルタグを使用する必要があります。"For"内では、プロジェクトタグを使用できません。

---

## スクリプトのタグへのアクセス

スクリプトにおいて、スクリプトでの実行時間中にのみ名前がコンパイルされるタグを使用する場合、このタグは[サイクリック継続]取得モードで設定します。

このタグを[使用時サイクリック]取得モードで設定する場合、IO フィールドなどの別のインスタンスでもこのタグが使用される画面に対してのみ、このスクリプトが呼び出されることを確認してください。

---

### 注記

スクリプトタグはスクリプトのみによって認識されます。スクリプト以外でスクリプトタグの値を使用するには、内部または外部タグに相互接続します。

スクリプトタグには未定義値を設定できます。WinCC flexible は、内部または外部タグへの未定義値の割り付けをサポートしません。

---

## 12.4.2 スクリプト内でのスクリプトとシステムファンクションの呼び出し

### 原理

システムファンクションおよびその他のスクリプトをスクリプト内に呼び出すことができます。

戻り値のないシステムファンクション("Sub")またはスクリプトは、以下のように呼び出します。

<ファンクション名> [パラメータ 1], [パラメータ 2], [...]

戻り値のあるシステムファンクション("Function")またはスクリプトは、式への割り付けによって呼び出します。

<式> = <ファンクション名> (パラメータ 1, パラメータ 2, ... [パラメータ N])

戻り値を評価したくない場合は、戻り値のないシステムファンクションまたはスクリプトと同様の呼び出しを使用します。

## システムファンクションの呼び出しにおける特性

システムファンクションおよびスクリプトは、[スクリプトウィザード]からスクリプトに挿入することができます。システムファンクションは、[スクリプトウィザード]で現在コンフィグレーションされている言語で表示されます。

スクリプト内でシステムファンクションを呼び出すときは、必ずシステムファンクションの英語名を使用します。

```
SetValue Tag1, 64
```

システムファンクションの英語名は、[構文]の下にあるシステムファンクションリファレンスにあります。設定されているプロジェクト言語は考慮されません。

システムファンクションのパラメータ配信には以下のルールが適用されます。

- 定数

定数をパラメータとして使用する場合、パラメータタイプが、次の3種類のデータタイプのどれかに対応している必要があります。整数、倍精度実数、または文字列。使用可能な定数は、パラメータ割り付け時に選択リストに表示されます。定数には通常のVBS表記規則が適用されます。

- タグ

スペルに関係なく、タグは必ず"参照による呼び出し"として配信されます。配信されるタグがVBS名の表記規則に対応していれば、このタグ名はキーワードのSmartTagsなしで配信できます。

```
SetValue Tag1, 64
```

または

```
SetValue SmartTags("Tag1"), 64
```

- プロセス画面、接続、ログなどオブジェクトへの参照

オブジェクト参照は、引用符で囲まれたパラメータとして配信されます。

```
ActivateScreen "MainScreen", 0
```

## スクリプトの呼び出しにおける特殊性

スクリプトを呼び出すときは、「参照による呼び出し」として配信されます。たとえば、タグをパラメータとして渡す場合、スクリプトでの値の割り付けは直接タグの値に影響します。

## スクリプトにおけるオペレータデバイスの依存性

スクリプトのコードは選択したオペレータデバイスによって異なります。選択したオペレータデバイスによってサポートされないスクリプトでシステムファンクションを使用する場合、エラーメッセージは出力ウィンドウで受信されます。

### 12.4.3 オブジェクトへのアクセス

#### はじめに

添付プロパティおよびメソッドを使用したランタイムオブジェクトモデルのオブジェクトは、スクリプト内で使用可能です。

オブジェクトプロパティは、ランタイム中の読取りおよび変更が可能です。

#### オブジェクトの参照

スクリプト内で、添付リストを使用してオブジェクトを参照します。オブジェクトを識別するには、オブジェクト名またはリスト内の位置番号を使用します。

[MainScreen]画面の最初のオブジェクトは、以下のステートメントで参照します。

```
'VBS_Example_01
Dim objObject
'Change to Screen "MainScreen"
HMIRuntime.BaseScreenName = "MainScreen"
Set objObject = HMIRuntime.Screens(1).ScreenItems(1)
```

名前とオブジェクトプロパティによって参照されるオブジェクトは、以下のステートメントによって変更します。これを実行するためには、オブジェクトはスクリーン内にこの名前を設定されている必要があります。

```
'VBS_Example_02
Dim objCircle
HMIRuntime.BaseScreenName = "MainScreen"
Set objCircle = HMIRuntime.Screens(1).ScreenItems("Circle_01")
objCircle.BackColor = vbGreen
```

## 12.4.4 タグとオブジェクトの同期化

### はじめに

WinCC flexible でオブジェクト名を変更すると、この変更はプロジェクト全体に影響を及ぼします。こうした変更は、スクリプトでは「同期化」とみなされます。

### 適用例

タグエディタ内で、スクリプトで使用する[オイル温度]タグを定義します。設定中、タグエディタでこの[OilTemperatureMotor1]タグの名前を変更します。

- 名前の変更中にスクリプトを開いている場合。

スクリプトの中で、以前のタグ名は青色の波線で下線が引かれます。タグ名にマウスポインタを合わせると、ツールチップが表示されます。[同期]ボタンをクリックすると、スクリプトでそのタグの名前が変更されます。

```
Sub SyntaxHighlighting( )  
1 'This is a comment  
2 Dim objScreen = HmiRuntime.Screens("Screen_1")  
3 Dim objTag = SmartTags("Mixer_RotationSpeed")  
4 ActivateScreen "Screen_1", 0  
5 Average (SmartTags ("Value_01: 効力 - オブジェクト名が Value_01 から Value_03 に変更されました。"  
End Sub
```

- 名前の変更中にスクリプトを閉じている場合。

スクリプトを再度開いたときに、タグが自動的に同期します。

## 12.4.5 スクリプトの保存

### 原理

プロジェクトを保存するとスクリプトが保存されます。たとえば、別のコンピュータでスクリプトを使用する場合、このスクリプトからテキストファイルにコードをコピーするか、コピー操作と貼り付け操作を使用して、別のプロジェクトに転送できます。

## 12.5 デバッグ;デバッグ

### 12.5.1 スクリプトのデバッグ

#### はじめに

デバッグにより、ランタイム時の論理プログラムエラーのスクリプトをテストすることができます。たとえば、タグに正しい値が渡ったか、および取り消し用語が正しく実現されているかをテストすることができます。

作成したスクリプトをデバッグするには、Microsoft Office XP に装備されている "Microsoft Script Debugger" または "Microsoft Script Editor" のどちらか一方だけを使用します。

スクリプトに VBScript ファンクションが含まれている場合、"VBScript for Windows" と "VBScript for Windows CE" を区別する必要があります。例えば CreateObject ファンクションなど、ファンクションが異なるものもあります。

デバッガは、Windows 向け VBScript 構文をチェックします。スクリプトに Windows CE 向けファンクションが含まれている場合、システムは対応するアラームを出力します。

#### エラータイプ

デバッグ時、次のエラータイプが区別されます。

- ランタイムエラー

ランタイムエラーが発生するのは、不正な命令または間違っただ命令を実行しようとした場合です。たとえば、タグが定義されていない場合が挙げられます。

ランタイムエラーを防ぐために、スクリプトで "On Error Resume Next" 命令を使用できます。この設定を行うと、ランタイムエラーが検出されたときに、スクリプトにより次のステートメントが実行されます。次の行で、エラーコードとエラーオブジェクトをチェックできます。スクリプトでランタイムエラーの処理を停止するには、"On Error Goto 0" 命令を使用します。エラー処理の詳細については、情報システムの Microsoft VBS ヘルプを参照してください。

- 論理エラー

論理エラーが発生するのは、期待しているイベントが発生しない場合です。たとえば、条件のチェックが間違っている場合が挙げられます。論理エラーを解決するには、機能しないスクリプト部分を識別するためにステップごとにスクリプトを調べます。

## 12.5.2 デバッガの統合

### WinCC flexible スクリプトデバッガのインストール

WinCC flexible でスクリプト内のエラーを検索するために、スクリプトデバッガをインストールしておく必要があります。

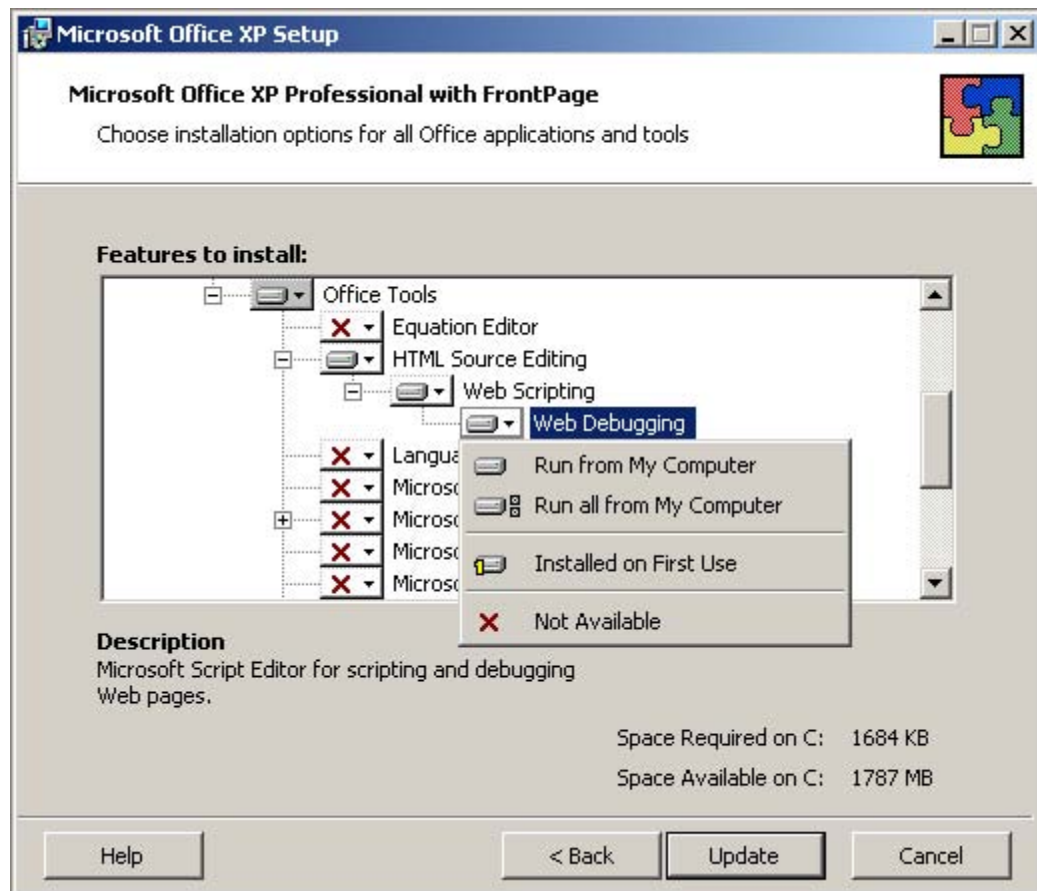
次のスクリプトデバッガをテストしてリリースしておきます。

- Office XP の Microsoft Script Editor (Windows Vista OS には使用不可)
- Microsoft Script Debugger

インストールされているスクリプトデバッガは、スクリプト内でのランタイムエラーの発生時に自動的に起動させたり、[Start runtime system with script debugger]コマンドを使って手動で起動することができます。

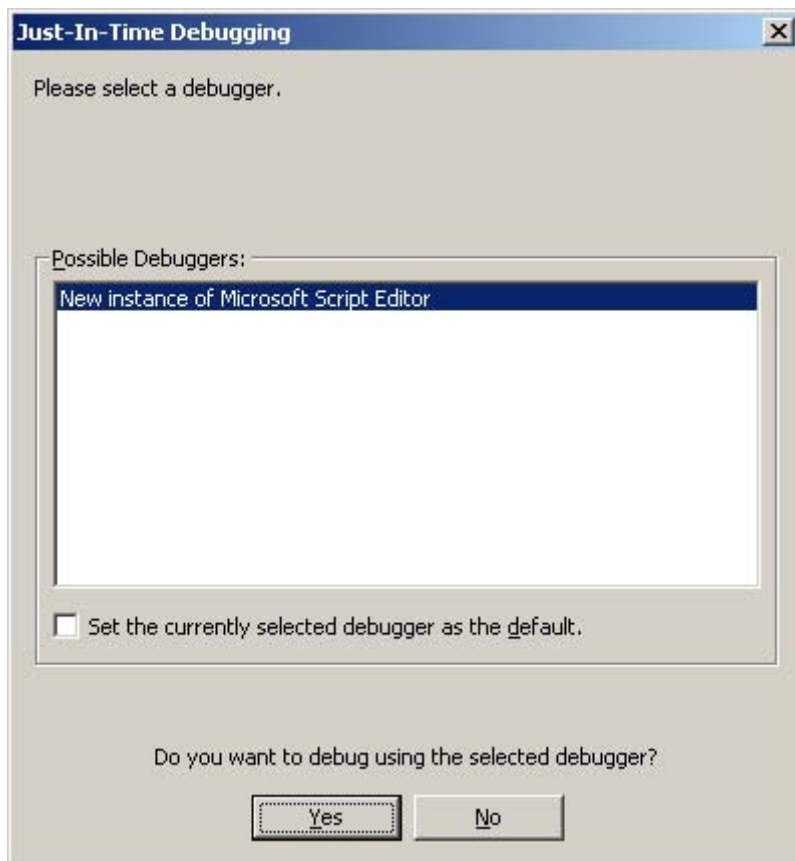
### Microsoft Script Editor

Microsoft Office XP のコンポーネントである"Microsoft Script Editor"にはスクリプトデバッガが含まれています。Microsoft Office のセットアップがデフォルト設定になっている場合、"Microsoft Script Editor"コンポーネントは("Installed on First Use")に設定されていたこととなります。これらのコンポーネントを明示的にインストールしたい場合には、Microsoft Office のセットアップで指定します。コンポーネントの選択で[ウェブデバッグ]をクリックし、オプション[このコンピュータから起動]を選択します。

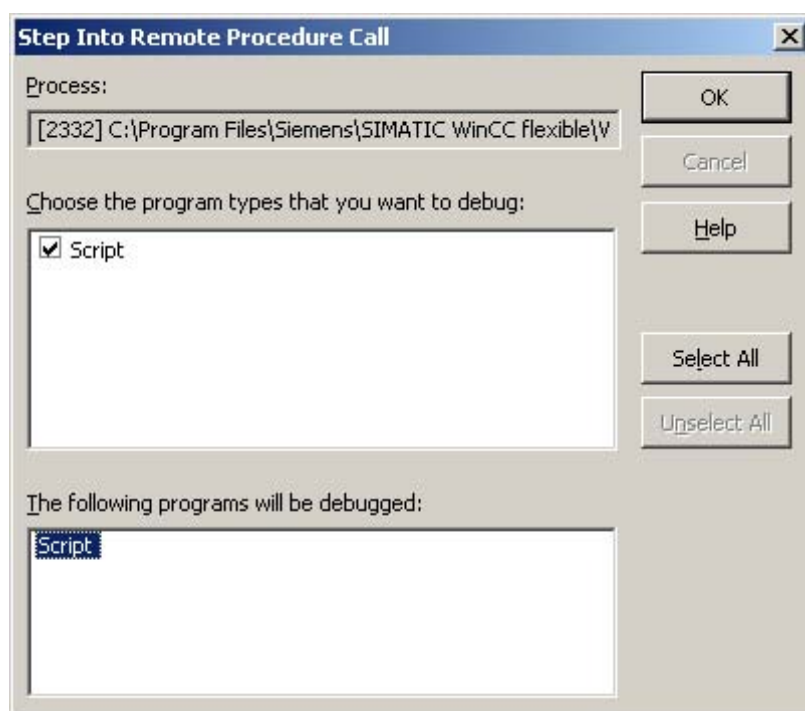


コマンド"Start runtime system with script debugger"によって WinCC flexible でプロジェクトが有効になっている場合、最初のスクリプトの呼び出しで、使用可能なスクリプトデバッガのリストが表示されます。

"Microsoft Visual Interdev"あるいは"Microsoft Visual Studio .NET"のようなその他のインストールされたスクリプトデバッガがリストに含まれていることがあります。 "Microsoft Script Editor"を選択し、[はい]をクリックして確定します。



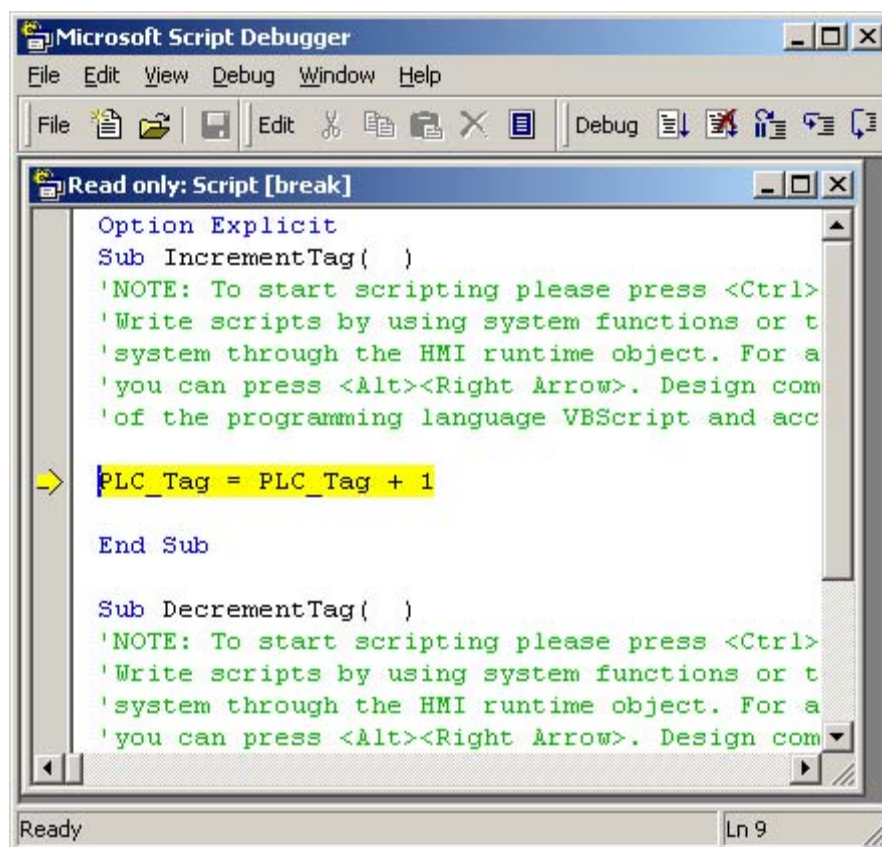
[リモートプロシージャへのステップ呼び出し]ダイアログで、[スクリプト]プログラムを有効にし、[OK]をクリックして確定します。



"Microsoft Script Editor"が起動します。実行は最初のスクリプトの最初の行で一時停止します。

## Microsoft Script Debugger

スクリプトデバッガがない場合、"Microsoft Script Debugger" (scd10en.exe)を Microsoft から無償でダウンロードすることができます。インストールした後は、WinCC flexible 内で自動的に起動することができます。



---

### 注記

すでに他のスクリプトデバッグシステムがコンピュータ上で有効になっている場合、"Microsoft Script Debugger"はサポートされません。

---

### 注記

#### Windows Vista オペレーティングシステム

WinCC flexible で使用するには、管理者としてログオンし、Microsoft Script Debugger をインストールします。

---

## ランタイムが起動した状態ではスクリプトデバッガは起動不能

Microsoft Script Debugger をインストールして、"Start Runtime with Script Debugger"コマンドがデバッガを動作させない場合、Windows レジストリで次の設定を行い、Microsoft Script Debugger をデフォルトの Just-In-Time (JIT) Debugger に設定します。

- "HKEY\_CLASSES\_ROOT\CLSID\{834128A2-51F4-11D0-8F20-00805F2CD064} (Default)="ScriptDebugSvc Class"  
"AppID"="{A87F84D0-7A74-11D0-B216-080000185165}"
- [HKEY\_CLASSES\_ROOT\CLSID\{834128A2-51F4-11D0-8F20-00805F2CD064}\LocalServer32] (Default)="c:\Program Files\Microsoft Script Debugger\mscrdbg.exe"
- [HKEY\_CLASSES\_ROOT\CLSID\{834128A2-51F4-11D0-8F20-00805F2CD064}\ProgID] (Default)="ScriptDebugSvc.ScriptDebugSvc.1"
- [HKEY\_CLASSES\_ROOT\CLSID\{834128A2-51F4-11D0-8F2000805F2CD064}\VersionIndependentProgID] (Default)="ScriptDebugSvc.ScriptDebugSvc"

"LocalServer32"ファイルのパスが、Microsoft Script Debugger のインストールフォルダを示す必要があります。Microsoft Script Debugger が異なるフォルダにインストールされている場合、それに応じてパス名を編集する必要があります。

## 12.6 ランタイム中のファンクションのランタイム動作

### 12.6.1 ランタイム中のファンクションリストの実行

#### 原理

ランタイム中、ファンクションリストは一番上から一番下に向かって実行されます。同期実行と非同期実行が区別されるため、実行中に待機期間は発生しません。この区別は、システムファンクションの各種ランタイムを評価することによりシステムが実行します。スクリプトは、ランタイムから独立して常に同期的に処理されます。システムファンクションがエラーステータスを返すと、ファンクションリストの実行がキャンセルされます。

#### 同期実行

同期実行中に、ファンクションリスト内のシステムファンクションは逐次実行されます。直前のシステムファンクションが終了してはじめて、次のシステムファンクションを実行できます。

#### 非同期実行

保存操作および読み取り操作などのファイル操作を実行するシステムファンクションは、たとえばタグ値を設定するシステムファンクションよりも長いランタイムがかかります。

したがってランタイムの長いシステムファンクションは非同期形式で実行されます。システムファンクションが記憶媒体にレジピレコードなどを書き込んでいるうちに、次のシステムファンクションがすでに実行されます。システムファンクションを並列実行するので、HMI デバイスでの待機期間が回避されます。

## 12.6.2 ランタイム中のスクリプトの処理

### 原理

ランタイム中は一度に1つのスクリプトのみ実行可能です。編集を待つスクリプトが複数ある場合、これらのスクリプトはキューに並べられて順次完了していきます。

---

#### 注記

したがって、スクリプトが非同期でトリガされた場合でも、スクリプトのループによって、キューにある他のスクリプトの実行は阻止されます。

---

WinCC flexible は、最大で8つまでのスクリプトの深さのネストをサポートします。このネストの深さはチェックされないので注意してください。

---

#### 注記

「Runtime-Stop」イベントに対応したスクリプトを設定する場合、スクリプト内で使用できるのは、「Runtime-Stop」システムファンクションの参照で、設定可能オブジェクトとして指定されているファンクションだけです。

ランタイムの終了がこのスクリプトの実行によって妨げられないことを確認します。

---

---

#### 注記

##### スクリプトの設定

設定中に、同時に有効にするスクリプトの数が多くなりすぎないように注意してください。また、システムを100%の負荷で長時間作動させないでください。

スクリプトは、操作や値の表示の妨げにならないように低い優先度で処理されます。このため、負荷が高いシステムでは、実行するスクリプトはキューに入れられます。キューに入れられるリストの最大サイズは、デバイスによって異なります。

- OP 270,TP 270,MP 270,OP 277,TP277 : 50 エントリ
- MP 277,MP 370,MP 377,PC-Runtime: 200 エントリ

画面変更時などに、事前にマーク付けが可能な数を超えるスクリプトが1度に有効にされた場合、超過した呼出しは破棄され、システムアラームが表示されます。

スクリプトには、フェースプレートで使用されるものも含まれます。設定されたスクリプトは、すべてのフェースプレートインスタンスで、独立したオブジェクトとして再利用され、エントリがキューリストに入れられます。このように使用すると、キューリストの最大容量にすぐ達します。

---

### 12.6.3 値の受渡しと戻し

#### 値の転送

スクリプトを呼び出すときは、パラメータは「値による呼び出し」の原理にしたがって受け渡されます。パラメータなどの変数を受け渡す場合、変数の値はスクリプトの実行時にスクリプトに受け渡されます。

スクリプトとシステムファンクションを呼び出すときは、パラメータは「参照による呼び出し」の原理にしたがって受け渡されます。

例:

[クリック]イベントに、スクリプトをコンフィグレーションします。システムファンクション"SetValue(Y,X)"はスクリプトで呼び出されます。システムファンクション"SetValue(Y、X)"は、値"5"を変数"IndexTag"に割り付けます。SetValue IndexTag, 5

ランタイムでボタンをクリックすると、スクリプトが実行されて値 5 が変数"IndexTag"に割り付けられます。

#### 戻り値

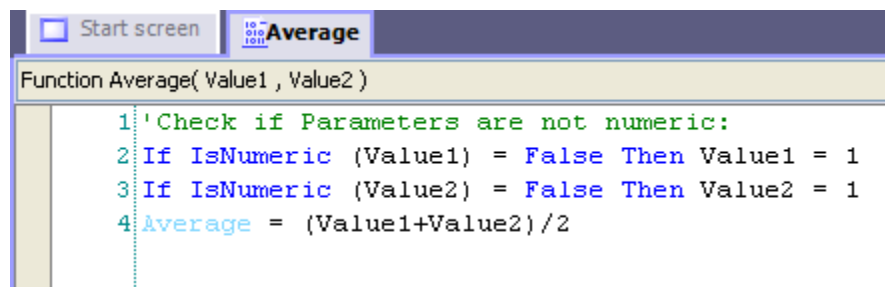
戻り値は、計算結果を返すことができます。たとえば、2つの数の平均値などが挙げられます。ただし、戻り値は、命令が適切に実行されたかどうかに関する情報も指定できます。

したがって、「削除」などのファイル操作を実行するシステムファンクションも値を返します。

#### 注記

システムファンクションの戻り値に割り付けることができるのは、外部変数または内部変数だけです。

スクリプトが値を返すようにするには、スクリプトに"Function"タイプを選択する必要があります。スクリプトで、スクリプトの名前に戻り値を割り付けます。



```
Function Average( Value1 , Value2 )  
1 'Check if Parameters are not numeric:  
2 If IsNumeric (Value1) = False Then Value1 = 1  
3 If IsNumeric (Value2) = False Then Value2 = 1  
4 Average = (Value1+Value2)/2
```

たとえば、2つの数の平均値を生成するには、Average ファンクションを呼び出して、処理する値を変数に受け渡します。

```
AverageValue = Average (4, 6)
```

その後、出力フィールドにこの平均値を出力できます。

## 12.6.4 VBS でのランタイム時のオブジェクトプロパティの変更

### はじめに

VBS を使用すればランタイム中にスクリーンオブジェクトとタグのオブジェクトプロパティにアクセスできます。VBS を使用してオブジェクトプロパティ値を変更しても、プロジェクトデータは影響を受けません。

### オブジェクトプロパティの変更

ランタイム中に VBS を使用して画面エレメントのオブジェクトプロパティを変更すると、画面が有効になっている期間だけこの変更が有効になります。画面を変更するか、画面を再ロードすると直ちに、コンフィグレーションしたオブジェクトプロパティが表示されます。

### 言語切り替え

ランタイム中に言語を変更すると、コンフィグレーションデータから外国語ラベルがロードされます。VBS を使用してテキストを変更した場合、このテキストは上書きされます。

## 12.6.5 スクリプト内での HMI デバイス依存システムファンクション

### 原理

設定されたオペレーティングユニットで使用できないシステムファンクションをスクリプト内で使用した場合、警告メッセージが表示されます。さらに、スクリプト内のそれぞれのシステムファンクションには、青い波線のアンダーラインが引かれます。

## マルチ言語プロジェクトの構造

### 13.1 複数言語を扱う

#### WinCC flexible での複数言語コンフィグレーション

WinCC flexible を使って、複数言語でプロジェクトを構築することができます。複数言語でプロジェクトを作成する理由はさまざまです。

- 複数の国でプロジェクトを使用すると仮定します。プロジェクトは複数言語で作成されます。そして HMI デバイスを調整するときに、オペレータが各地域で話す言語だけが HMI デバイ스에 転送されます。
- プラントにいるさまざまなオペレータに複数の言語を提供すると仮定します。サービス担当者はオペレータと同じ言語を話さないで、プロジェクトを複数言語で作成します。例:HMI デバイスが使用される場所は中国ですが、サービス担当者が理解するのは英語のみです。

#### プロジェクトテキストを翻訳する

WinCC flexible を使って、"画面"エディタや"プロジェクトテキスト"エディタなどの個々のエディタに、複数の言語でプロジェクトテキストを直接入力できます。さらに、WinCC flexible には、コンフィグレーションのエクスポートおよびインポートのためのオプションが翻訳用に用意されています。このオプションは、多量のテキストが含まれるプロジェクトを構築し、それを翻訳する場合に特に役立ちます。

#### WinCC flexible での言語管理および翻訳

プロジェクトウィンドウの以下の領域が WinCC flexible での言語管理とテキスト翻訳に使用されます。

領域	簡単な説明
プロジェクト言語	プロジェクト言語、言語の編集、および基準言語の管理。
言語とフォント	HMI デバイスで使われるランタイム言語およびフォントの管理。
プロジェクトテキスト	すべてのプロジェクト言語で構築されたテキストの中央管理。
グラフィックブラウザ	グラフィックおよびグラフィックの言語依存変換形の管理。
辞書	システム辞書およびユーザー辞書の管理。

## 13.2 WinCC flexible 専門用語

### WinCC flexible の言語原理

WinCC flexible では、複数の言語機能がさまざまな言語レベルで実現されています。

### ユーザーインターフェース言語とプロジェクト言語

WinCC flexible では、2つの言語レベルが区別されています。

- ユーザーインターフェース言語

コンフィグレーション時に、テキストが WinCC flexible メニューとダイアログボックスにユーザーインターフェース言語で表示されます。WinCC flexible のインストール時に、使用するユーザーインターフェース言語を選択します。[オプション|設定]の順にメニューコマンドを使用すれば、ユーザーインターフェース言語を変更できます。

- プロジェクト言語

プロジェクト言語は、プロジェクトを複数の言語で作成するのに使用します。

2つの言語レベルはお互いに完全に独立しています。例えば、ドイツ語のユーザーインターフェースを使って英語のプロジェクトをいつでも作成でき、その逆も同じです。

### プロジェクト言語

WinCC flexible では以下のプロジェクト言語がリリースされています。

- 中国語(PRC)
- 中国語(台湾)
- デンマーク語
- ドイツ語
- 英語
- フィンランド語
- フラマン語
- フランス語
- ギリシャ語
- イタリア語
- 韓国語
- ノルウェー語
- ポーランド語
- ポルトガル語
- ロシア語
- スウェーデン語
- スペイン語
- チェコ語

- トルゴ語
- ハンガリー語
- 日本語

通常、Windows で使用できる言語ならどの言語でも設定できます。ただし、言語の中には設定に使用する場合に制約が適用されるものもあります。たとえば、次のものがあります。

- HMI では、ヘブライ語やアラビア語など右から左に向かって読む言語はサポートされていません。
- 言語固有のフォントは使用できません。
- WinCC flexible に保存されている編集不可能なテキストは、英語で表示されます。

以下の言語は、プロジェクト言語内で区別されます。

- 基準言語

基準言語は、プロジェクトを最初に構成するのに使う言語です。

コンフィグレーション時に、プロジェクト言語のどれか 1 つを基準言語として選択します。基準言語は、翻訳用のテンプレートとして使用します。プロジェクトのテキストはすべて、まず基準言語で作成されてから翻訳されます。テキストを翻訳しながら同時に基準言語で表示することができます。

- 編集言語

テキストの翻訳を編集言語で作成します。

一度、基準言語でプロジェクトを作成すれば、それ以外のプロジェクト言語にテキストを翻訳することができます。そのために、プロジェクト言語のどれか 1 つを編集言語として選択し、その言語に合わせてテキストを編集します。編集言語はいつでも変更することができます。

---

#### 注記

プロジェクト言語を切り替える場合は、キーボードのキーへの割り付けも変更します。言語のなかには(スペイン語など)、オペレーティングシステムが原因で、キーボード割り付けの切り替えができない場合があります。この場合、キーボード割り付けは、英語に切り替わります。

---

- ランタイム言語

ランタイム言語とは、HMI デバイスに転送されるプロジェクト言語のことです。プロジェクト要件に合わせて、どのプロジェクト言語を HMI デバイスに転送するかを決めます。

オペレータがランタイム時に言語間を切り替えられるよう、適切なオペレータ制御工レメントを提供する必要があります。

## 13.3 言語設定

### 13.3.1 オペレーティングシステムの言語設定

#### はじめに

コンフィグレーションコンピュータのオペレーティングシステムの設定は、次の領域にある WinCC flexible の言語管理に影響を与えます。

- プロジェクト言語の選択
- 日付、時刻、通貨、および数値の地域フォーマット
- ASCII 文字の表示

#### オペレーティングシステムの言語設定

言語は、オペレーティングシステムにインストールされていない場合、プロジェクト言語として使用することはできません。

- Windows XP での設定

[地域と言語のオプション]ダイアログは、[スタート|設定|コントロールパネル|日付、時刻、言語および地域のオプション]と選択して表示される同名のコントロールパネルアイコンを使って表示できます。言語の選択やインストールは、後から[言語]タブで行うこともできます。

IME (Input Method Editor)は、Windows でアジア言語のテキストを設定するのに使用します。IME がないと、アジア言語のテキストを表示することはできますが、編集はできません。IME の詳細については、Windows のマニュアルを参照してください。

アラームテキストなどの言語に依存するプロジェクトテキストをシミュレータにアジア言語で表示する場合は、オペレーティングシステムをその言語に切り換える必要があります。

#### 日付、時刻、通貨、および数値の地域フォーマット

WinCC flexible では、[日付 - 時間]フィールドで、選択したプロジェクト言語およびランタイム言語の日付と時刻の固定フォーマットが指定されます。

選択した編集言語で日付、時刻および数値を正しく表示させるためには、その言語をコントロールパネルの[地域のオプション]に設定する必要があります。

#### ASCII 文字の表示

テキスト出力フィールドでは、128 種類の ASCII 文字以外の表示は、設定言語、および使用中のオペレーティングシステムによって違ってきます。

異なるコンピュータで同じ特殊文字を表示するには、各コンピュータで、同じオペレーティングシステムと国設定を使用する必要があります。

### 13.3.2 アジア言語に対応したオペレーティングシステム設定

#### 欧米のオペレーティングシステムでの設定

アジア言語を入力する場合、オペレーティングシステムで、この言語のサポートを有効にする必要があります。この操作をするには、[コントロールパネル]を開き、[地域と言語のオプション]を選択します。[言語]タブの[東アジア言語用ファイルのインストール]チェックボックスを有効にします。その後、[テキストサービスと入力言語]下の[詳細]をクリックします。[テキストサービスと入力言語]ダイアログが開きます。[設定]タブの[インストールされているサービス]下に、必要なデフォルト入力言語を追加します。

欧米のオペレーティングシステムでアジア文字を入力するには、IME (Input Method Editor) もインストールする必要があります。インストールするには、Windows XP の[コントロールパネル]で、[地域と言語のオプション]>[言語]>[詳細]の順に開きます。[テキストサービスと入力言語]の[インストールされているサービス]下に、必要なデフォルト入力言語を追加します。

設定時にアジア言語を入力するには、IME (Input Method Editor)でアジア言語入力方式に切り替えます。

#### アジア言語オペレーティングシステムでの設定

アジア言語オペレーティングシステムで設定している場合、たとえばオブジェクト名などに ASCII 文字を入力するには、英語デフォルト入力言語に切り替える必要があります。オペレーティングシステムの基本インストールには英語デフォルト入力言語が含まれているので、入力ローケルを追加インストールする必要はありません。

### 13.3.3 "プロジェクト言語"エディタ

#### はじめに

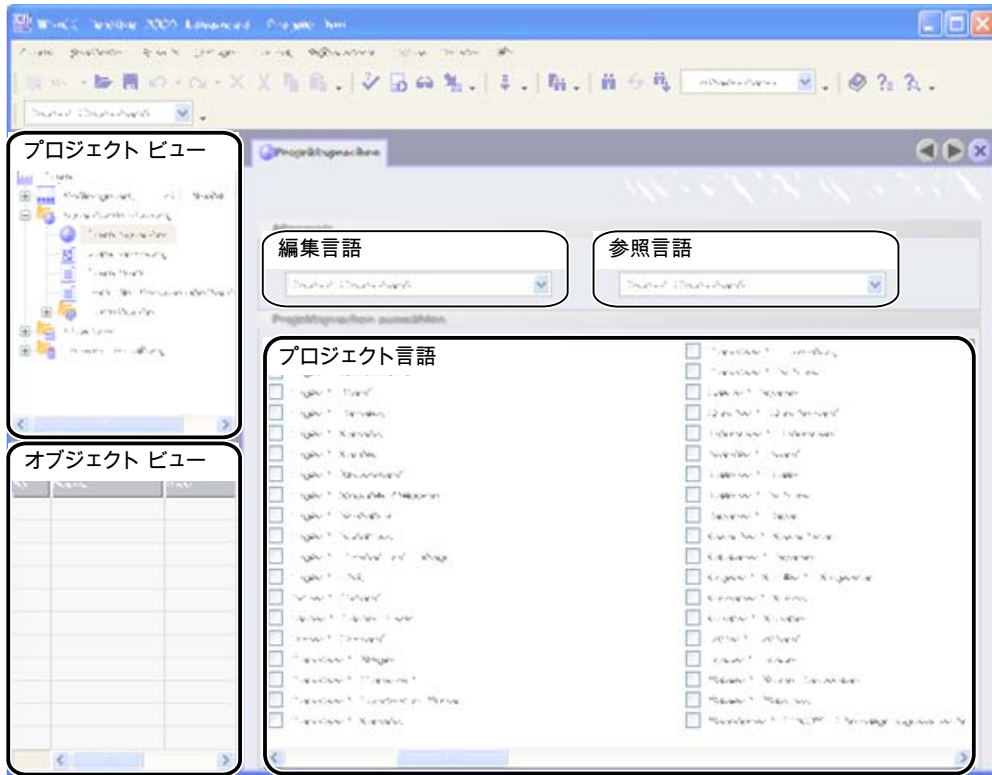
"プロジェクト言語"エディタでプロジェクトを作成するときの言語を選択します。

- プロジェクトの作成に使用するプロジェクト言語
- 最初にプロジェクトを設定する参照言語。
- テキストを変換する編集言語。

#### 開く

"プロジェクト言語"エディタを開くには、[プロジェクト]ウィンドウの[ローカライズ]グループに表示されている[プロジェクト言語]をダブルクリックします。

ストラクチャ



プロジェクト言語

ここでは、プロジェクトを作成する際のプロジェクト言語を有効にします。

参照言語

ここでは、プロジェクト言語の参照言語を選択します。表示される言語は、使用言語リストで有効にした言語だけです。

編集言語

ここでは、プロジェクト言語の中から編集言語を選択します。表示される言語は、使用言語リストでプロジェクト言語として有効にした言語だけです。

## 13.4 複数の言語でのプロジェクトの作成

### 13.4.1 複数の言語でのプロジェクトの作成

#### 翻訳方法

まず、1つの言語のプロジェクトだけを作成し、それをテストします。

その後、必要なその他の言語をすべて設定します。以下のオプションが使用できます。

- 個々のオブジェクトの作成に使用するエディタで直接プロジェクトテキストを翻訳する
- "プロジェクトテキスト"エディタでプロジェクトテキストを集中化して翻訳する
- プロジェクトテキストをエクスポートして別のプログラムで翻訳し、翻訳済みのテキストを WinCC flexible にインポートする

翻訳の速度を上げ、翻訳済みテキストの一貫性を保つために辞書を使用する

#### 推奨ワークフロー

1. 自分が使い慣れている言語を"プロジェクト言語"エディタの基準言語として設定します。コンフィグレーションの開始時に、基準言語が編集言語に対応していなければなりません。
2. プロジェクトをこの基準言語で作成します。基準言語のテキストは、翻訳用のソース言語として使用されます。
3. "プロジェクト言語"エディタで、他のプロジェクト言語のどれか1つを編集言語として設定します。
4. すべてのプロジェクトテキストをこの言語に翻訳します。個々のエディタで直接、翻訳することもできるし、集中化された"プロジェクトテキスト"エディタで翻訳してすべてのオブジェクトテキストとその使用ポイントを表示することもできます。  
また、テキストを\*.csv ファイルにエクスポートして翻訳を行い、そのあと翻訳済みのテキストをプロジェクトにもう一度インポートすることもできます。
5. 必要に応じて、プロジェクト内のグラフィックを修正し、編集言語またはその国固有の要素を反映させます。
6. その他すべてのプロジェクト言語に対して、ステップ3から5を繰り返します。

#### 結果

これで、プロジェクトをコンパイルして HMI デバイスに転送することができます。HMI デバイスで使用するランタイム言語を、デバイスの転送設定で指定します。

## 13.4.2 エンジニアリングシステムのアジアおよび東洋言語に特有な特徴

### はじめに

アジア言語の設定時、またはアジア言語には、特有な特徴を遵守する必要があります。これらの特有な特徴は、複雑な文字を使用する他の言語でも遵守しなければなりません。

### 設定の基本原則

プロジェクトの機能をすべて保証するには、WinCC flexible での設定時に、エレメントの一部に複雑な文字を指定しないでください。次の各エレメントの場合、制約が適用されます。

- オブジェクト名
- アラームテキスト

WinCC flexible のオブジェクト名は、一意の名前であるため、言語の切り替え時に翻訳されません。オブジェクト名は機能上で使用され処理されるため、いくつかの制約を受けます。オブジェクト名には、特殊文字、ウムラウト記号、または複雑な文字を指定できません。たとえば、影響を受けるオブジェクト名には、プロジェクト名、タグ名、画面名などがあります。

アラームテキストで各アラームを記録する場合、アジアランタイム言語は使用できません。アジアランタイム言語を使用すると、たとえアラームテキスト自体を英語などで書き込んでも、アラームテキストを記録できません。制約はロギングだけに適用されます。ランタイム中に表示や出力は相変わらずできます。ロシア語などの 1 バイト言語は、この制約の影響を受けません。

Sm@rtAccess および Sm@rtService を使用している場合、HMI デバイスで認識される文字以外使用できません。

### ユーザー管理

アジア言語などの複雑な文字は、ユーザー名とパスワードに使用できません。

### プロジェクト文書

プロジェクト文書に、[設定]ダイアログで各アジアまたは東洋言語フォントを選択すれば、プリントアウトの外観を最適化できます。

### STEP 7 への統合

STEP 7 に統合されているアジア言語プロジェクトは、WinCC flexible を使用して開始する必要があります。統合されたアジア言語プロジェクトを STEP7 を使用して開始すると、エラーメッセージおよび表示エラーが表示されます。

### その国特有の文字の表示

"WinCC flexible"文字セットに基づくテキストに含まれるその国特有の文字により、ランタイムに OP 73micro、TP 177micro、OP 73、OP 77A、および TP 177A HMI へ出力されるときに、明瞭さに欠ける場合があります。これらの HMI では、文字セット"WinCC flexible High"を使用する必要があります。WinCC flexible がこの文字セットを提供しない場合は、OS インストールフォルダの"Fonts"フォルダで"WinCC flexible High"文字セットを見つけてから、エントリをダブルクリックします。

## プロジェクト名とオブジェクト名

STEP 7 オブジェクトのプロジェクト名とオブジェクト名だけに使用できるのはアジア文字だけです。

### 13.4.3 エディタでプロジェクトテキストを変換する

#### はじめに

原則として、複数言語のプロジェクトを作成している場合、すべてテキストは精通している言語で最初に構成されます。この言語を基準言語として変換に使用します。

"プロジェクト言語"エディタでプロジェクト言語を定義します。プロジェクト言語から、基準言語とそれを変換する各編集言語を選択します。

#### 言語依存オブジェクトを用いるエディタ

以下のエディタには言語依存オブジェクトが含まれます。

- 画面
- プロトコル
- アナログアラーム
- ディスクリートアラーム
- システムアラーム
- レシピ
- テキストリスト
- グラフィックリスト

#### WinCC flexible エディタで編集言語を切り替える

[ローカライズ]ツールバーを使って、編集言語を切り替えられます。編集言語はすべてのエディタに適用されます。

#### 参照テキスト

原則として、複数言語のプロジェクトを作成している場合、すべてテキストは周知の言語で最初に構成されます。この言語は基準言語として保存されます。

次に別の言語でテキストを入力するために編集言語を切り替えると、すべてのテキストフィールドが空になります。

WinCC flexible には便利な参照テキスト機能が用意されていて、変換のためのテンプレートに使えます。ダイアログおよびエディタには、基準言語のテキストが入っている[リファレンステキスト]ウィンドウを表示できます。

#### 13.4.4 "プロジェクトテキスト"エディタ

##### "プロジェクトテキスト"エディタ

"プロジェクトテキスト"エディタで、プロジェクトのすべてのテキストにアクセスできます。

例:

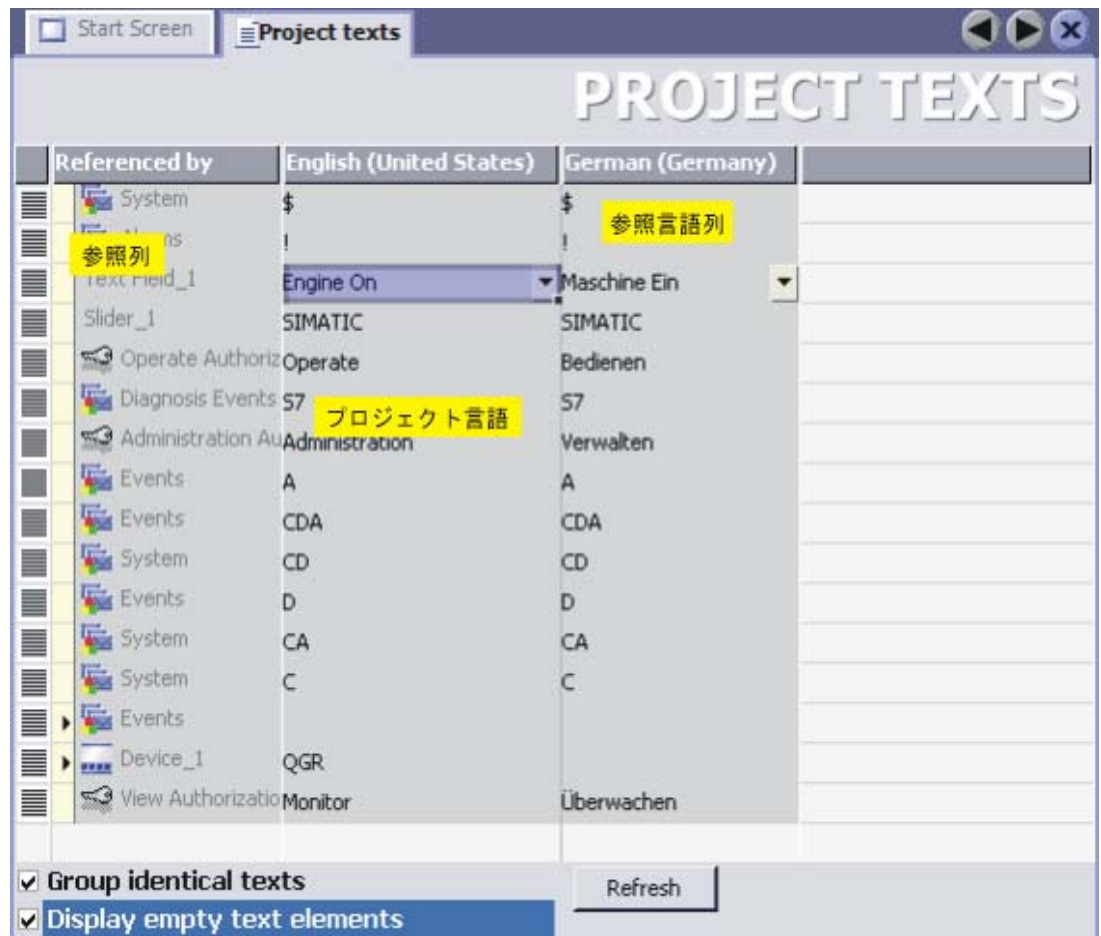
- 画面のテキスト
- アラーム
- コメント
- ヘルプテキスト
- レシピのテキスト

テキストの中央表示

##### "プロジェクトテキスト"エディタを開く

"プロジェクトテキスト"エディタを開くには、[プロジェクト]ウィンドウの[ローカライズ]グループにある[プロジェクトテキスト]をダブルクリックします。

## "プロジェクトテキスト"エディタの構造



## "プロジェクトテキスト"エディタのエレメント

"プロジェクトテキスト"エディタでは、設定したプロジェクト言語ごとに列が1つずつ作成されます。作成されたオブジェクトのテキストが、テーブル内の1つの行にすべての言語で表示されます。行は以下のような列に分類されます。

- 参照言語列  
最初の列には、テキストが参照言語で表示されます。
- 参照列  
[参照元]列は、元のテキストがあるエディタを示しています。
- プロジェクト言語  
テキストが既に翻訳済みであれば、残りの列にテキストがその他のプロジェクト言語で表示されます。

## 翻訳方法

作成されたテキストを、以下の方法で翻訳できます。

- "プロジェクトテキスト"エディタでのテキストの内部翻訳

この方法は、ほとんど翻訳する箇所のないテキストに適しています。

- エクスポートおよびインポート機能を使ったテキストの外部翻訳

この方法は、翻訳する量が多いテキスト、またはプロジェクト言語が多いときに適しています。

"プロジェクトテキスト"エディタは、その他のエディタにリンクされています。ここで導入されたテキストは自動的に、その他のエディタでも必要に応じて利用可能になります。"プロジェクトテキスト"エディタから直接、翻訳対象のオブジェクトの使用ポイントにジャンプすることもできます。

### 13.4.5 トランスレータでテキストを変換する

#### はじめに

複数言語のテキストが多量にある場合は、プロジェクトテキストの外部変換が便利です。エクスポート機能を使って、プロジェクトテキストを外部トランスレータに転送できます。そのあとインポート機能を使って、翻訳済みのテキストを使用するプロジェクトに再統合できます。

#### エクスポートとインポートの適用範囲

- すべてのプロジェクトテキストのエクスポートとインポート

すべてのテキストを翻訳目的で転送する場合(新規のプロジェクトの場合など)は、すべてのテキストをプロジェクトから\*.csv ファイルに翻訳用にエクスポートします。その後で翻訳テキストをインポートします。翻訳済みのテキストは、プロジェクト内で使用する修正ポイントへ自動的に割り付けられます。

この間に WinCC flexible でプロジェクトテキストに変更を行った場合は、修正済みテキストはインポート時に上書きされません。

- 特定エディタでのテキストのエクスポートとインポート

WinCC flexible では、テキストのエクスポートとインポートを特定のエディタに限定できます。

- 部分的に翻訳されたプロジェクトの新規テキストのエクスポートとインポート

すでに翻訳済みのプロジェクトに新規テキストを挿入した場合は、まだ翻訳していないテキストを選択してエクスポートできます。これによって、翻訳の手間を必要最小限にします。

#### 必要条件

"プロジェクトテキスト"エディタが開いていること。

## 手順

1. ターゲット言語の列ヘッダーをクリックします。  
    エントリはアルファベット順でソートされます。空のセルは列の先頭に表示されます。
2. 空のセルを選択します。  
    <Shift>キーを使えば、複数のセルを一度に選択できます。
3. [オプション]メニューの[テキスト|エクスポート]コマンドを選択します。
4. テキストのエクスポート先となる\*.csv ファイルの名前とパスを入力します。
5. ソース言語を選択します。
6. ターゲット言語を選択します。
7. [エクスポートファイルにテキストインスタンスを入力]チェックボックスを有効にします。
8. [OK]をクリックして入力を確認します。

## 結果

選択した列のすべてのテキストが、ソースおよびターゲット言語の両方でエクスポートされます。

---

### 注記

コンピュータに MS Office パッケージがインストールされている場合、テキストを XLS ファイルにエクスポートするオプションがあります。

---

## 13.5 ディレクトリの取り扱い

### 13.5.1 ディレクトリの取り扱い

#### システム辞書とユーザー辞書

WinCC flexible でプロジェクトを翻訳するときは、さまざまな辞書が支援します。

- システム辞書

WinCC flexible 内のシステム辞書には、よく使用されるプロセス自動化用語と対応する翻訳が含まれています。システム辞書は表示できますが、修正することはできません。

ソース文書の著作権はすべて下記の組織に属します。"Landesinstitut für Erziehung und Unterricht (LEU)", Rotebühlstraße 131, 70197 Stuttgart, Germany, Tel. +49 711 6642-235, Fax +49 711 6642-203

www.schule-bw.de/unterricht/faecher/englisch/tech\_english/tech\_woerterb

この規定に反する指定がない限り、また他の当事者の権利に影響を及ぼさない限り、これらの文書の全体または部分的な配布は、電子的または印刷された形のどちらでも、出典(Landesbildungs-Server Baden-Württemberg)と URL を明記した状態で行うものとします。

LEU の事前の書面による許可なく、商業目的で文書を配布することは、厳重に禁止します。

- ユーザー辞書

プロジェクトテキストで繰り返し現れる用語の翻訳は、ユーザー辞書に登録できます。ユーザーディレクトリに項目を直接入力することも、エディタのプロジェクトテキストを使用することもできます。

WinCC flexible では、ユーザー辞書を複数使用できます。これらの辞書は、物理的に 1 つのファイル内で管理されます。さらに、これらの辞書を新規プロジェクト内に統合できます。

#### Auto translate ファンクション

"プロジェクトテキスト"エディタの"自動翻訳"ファンクションを有効化すると、すべての辞書で翻訳対象の用語が検索されます。検索された用語は、"プロジェクトテキスト"エディタに翻訳候補として入力されます。ユーザーは、この翻訳候補を受け入れたり、修正したりすることができます。翻訳するテキストにタグやテキストリストが含まれていると、自動翻訳が失敗します。

#### ユーザー辞書のアンインストール

アンインストールすると、ユーザー辞書も削除されます。これらの辞書を保存して使用するには、辞書のバックアップコピーを作成する必要があります。

1. "C:\Documents and Settings\All Users\Application Data\Siemens AG\WinCC flexible"フォルダ内で"UserDictionary.dct"ファイルを見つけます。
2. "UserDictionary.dct"および"UserDictionary\_log.LDF"ファイルをコピーします。

---

#### 注記

Windows Vista では、以下の保存場所を使用します。

"C: \ProgramData\Siemens AG\WinCC flexible"

---

## 13.5.2 "システム辞書"エディタ

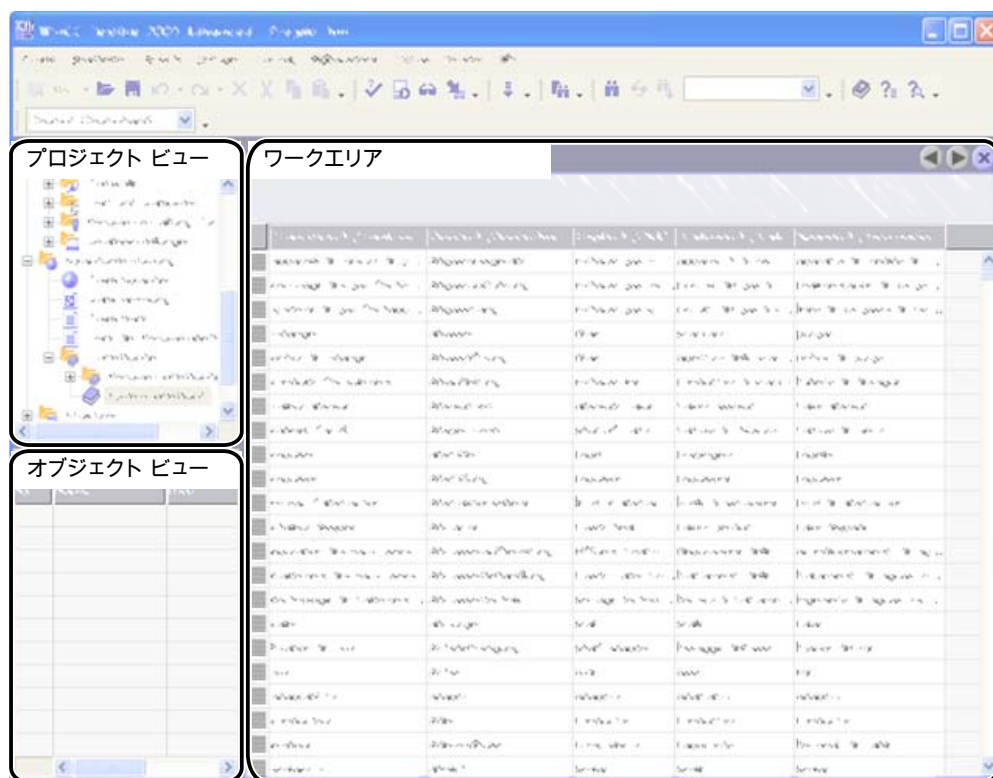
### はじめに

システム辞書の用語は"システム辞書"エディタで管理されます。このエディタでは、システム辞書を見たりソートすることができますが、変更することはできません。

### "システム辞書"エディタを開く

"システム辞書"エディタを開くには、[プロジェクト]ウィンドウの[ローカライズ]グループにある[辞書|システム辞書]をダブルクリックします。

### "システム辞書"エディタの構造



### 作業エリア

作業エリアのテーブル内に言語が表示されます。各言語ごとに個別に列が作成されます。テーブルの各行には、プロセスオートメーション用語とその訳語が入っています。

特定の用語の訳をすばやく検索するために、列内のエントリ別にテーブルをアルファベット順にソートすることができます。これを実行するには、該当する列のヘッダーをクリックします。

### 13.5.3 "ユーザー辞書"エディタ

#### はじめに

"ユーザー辞書"エディタでユーザー辞書の用語を表示したり、編集したりすることができます。

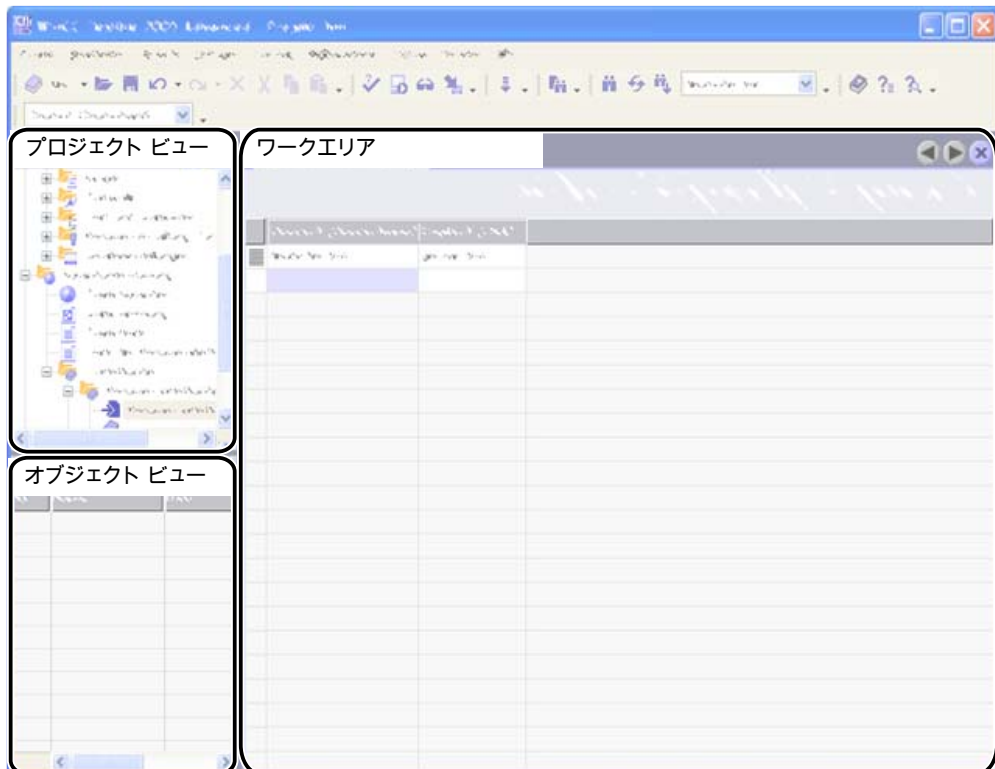
#### "ユーザー辞書"エディタを開く

ユーザー辞書は、[プロジェクト]ウィンドウの[ローカライズ]辞書]ユーザー辞書]グループに表示されます。"ユーザー辞書"エディタを開くには、ユーザー辞書の名前をダブルクリックします。

#### 注記

WinCC flexible の複数のインスタンスが開いている場合、1つのインスタンスのみが書き込みアクセスでき、残りのすべてのインスタンスは読み取りアクセスのみとなります。

#### "ユーザー辞書"エディタの構造



## 作業エリア

ユーザー辞書の内容は、作業エリアにテーブル形式で表示されます。ユーザー辞書を作成する時に、辞書に含める言語を選択します。エディタ内のテーブルには、言語ごとの列が含まれています。テーブルの各行には、ユーザー辞書に登録しておいた参照言語による用語とその訳語が記載されています。

## 13.6 言語依存グラフィックの使用

### 13.6.1 言語依存グラフィックの使用

#### 言語に依存している画面バージョン

"グラフィック"エディタを使って、プロジェクトにグラフィックをインポートし、その言語に依存しているバージョンを管理します。これらのグラフィックは、"画面"エディタでプロジェクトのプロセス画面にリンクされます。プロジェクトを複数の言語で作成する場合、以下の理由により各プロジェクト言語ごとに異なったグラフィックが必要になります。

- グラフィックにテキストが含まれている。
- 文化的要素がグラフィック内で大きな役割を果たす。

どちらの場合も、グラフィックの言語依存バージョンを作成する必要があります。

#### 基本的な処理

1. まず、"画面"エディタで1つの言語に対応するプロセス画面をすべて作成します。
2. グラフィックプログラムで、プロジェクト言語ごとのグラフィックのバージョンを作成します。
3. "グラフィック"エディタで、言語依存グラフィックをプロジェクトにインポートします。

#### 結果

"画面"エディタに、現在の編集言語設定に対応した各グラフィックのバージョンが表示されます。ランタイム中に、現在のランタイム言語設定に対応した各グラフィックのバージョンが表示されます。

## 13.6.2 "グラフィック"エディタ

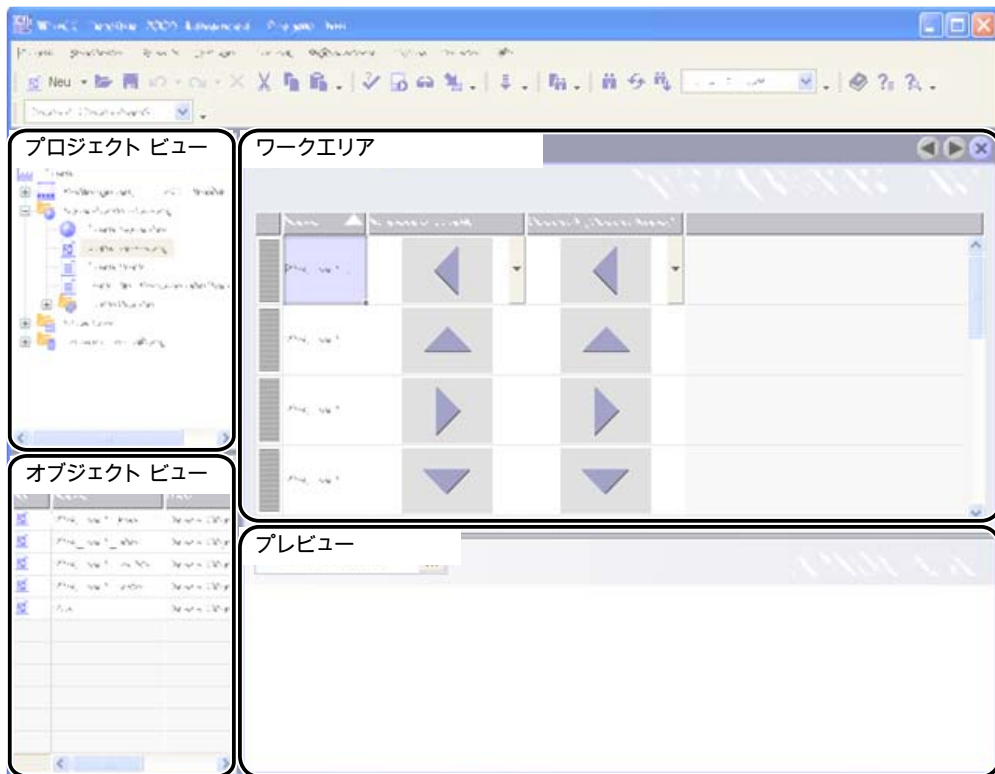
### はじめに

"グラフィック"エディタでは、複数の言語バージョンで設定済みのグラフィックオブジェクトを管理します。

### "グラフィック"エディタを開く

"グラフィック"エディタを開くには、[プロジェクト]ウィンドウの[ローカライズ]グループにある"グラフィック"エディタエントリをダブルクリックします。

### "グラフィック"エディタの構造



### 作業エリア

設定されたすべてのグラフィックオブジェクトがテーブルに表示されます。プロジェクト言語ごとに個別にテーブル列が作成され、その列には各言語に対応したグラフィックのバージョンが入ります。

さらに、プロジェクト言語に対応した言語固有のグラフィックが存在しない場合はいつでも、表示するグラフィックごとにデフォルトグラフィックを指定することができます。

### プレビュー

ここでは、さまざまな HMI デバイス上のグラフィック表示をプレビューできます。

## 13.7 ランタイム中の言語

### 13.7.1 ランタイム中の言語

#### 複数のランタイム言語の使用

特定の HMI デバイス上でランタイム言語として使用するプロジェクト言語を決めることができます。HMI デバイス上で同時に使用できる言語の数は、デバイスによって異なります。オペレータがランタイム時に言語間を切り替えられるようにするには、対応するオペレータ制御エレメントを設定する必要があります。

ランタイムが開始されると、最新の言語設定に従ってプロジェクトが表示されます。初めてランタイムが開始される場合は、「言語設定の順序」の最も小さい番号の言語が表示されます。

#### 設定時のランタイム言語の設定

「言語とフォント」エディタで指定できるもの:

- 各 HMI デバイスのランタイム言語として使用できるプロジェクト言語
- 言語の切り替え順序

#### アジア言語のテキストフィールド長

アジア言語を含む複数言語のプロジェクトを扱うときには、テキストフィールドに適切な長さを確保するようにしてください。フォントやフォントサイズによっては、フィールドコンテンツが一部見えなくなる場合があります。

1. [プロパティ]ウィンドウで[プロパティ]>[表示モード]を開きます。
2. [自動調整]チェックボックスを無効にします。
3. ランタイム中に表示が適切かどうかを確認します。

## 13.7.2 言語切り替えのコンフィグレーション

### はじめに

HMI デバイスで複数のランタイム言語を使用できる場合、言語切り替えを設定する必要があります。オペレータが各種ランタイム言語間を切り替えることができるようにするには、この設定が必要です。

### 言語切り替えの方法

以下のような言語切り替え方法を設定できます。

- ダイレクトな言語選択

各言語を個別のボタンで設定します。この場合、ランタイム言語ごとにボタンを1つ作成します。

- 言語切り替え

オペレータが1つのボタンだけで言語を切り替えます。

使用する方法に関わらず、ボタン名は使用するそれぞれの言語に翻訳する必要があります。また、現在の言語設定を表示する出力フィールドもコンフィグレーションできます。

## 13.7.3 ランタイム中のアジアおよび東洋言語に特有な特徴

### はじめに

アジア言語に対応して設定している場合、ランタイム中の操作には特有な特徴がいくつかあります。

### アジア言語文字セットのメモリ必要条件

アジア言語を使用している場合、当然、メモリ必要条件は大きくなります。このため、コンパイル時に発生するエラーメッセージにはすべて注意してください。

### 東洋およびアジア言語文字(非 ASCII)の入力

PC ベースの HMI デバイスでは、東洋およびアジア言語文字は入力できません。

### アジア言語文字の解釈

Sm@rtAccess および Sm@rtService を使用している場合、HMI デバイスで認識される文字以外使用できません。アジア言語文字を使用できるようにするには、エンジニアリングシステムでアジア言語文字を設定する必要があります。文字を追加設定すると、HMI デバイスのメモリ量を追加する必要があります。HMI デバイスで使用可能なメモリのサイズを調べてください。

## 設定可能な文字セット

270 シリーズ HMI デバイス、および MP 370 では、デフォルトのヨーロッパおよびアジア言語文字セットと一緒に、設定可能なヨーロッパ言語文字セットだけ使用できます。さらに、アジア言語の設定可能な文字セットは現在使用できません。

## アジア言語文字セットのフォントサイズ

アジア言語用に作成されたプロジェクトのテキストを Runtime で表示するには、10 ポイント以上のフォントサイズを使用してください。これより小さいフォントサイズを使用すると、アジア言語が読めなくなります。この規則は、"言語とフォント"エディタで使用されるデフォルトフォントについても有効です。



## プロジェクト文書

### 14.1 基本

#### 14.1.1 プロジェクト文書

##### はじめに

プロジェクト文書は、WinCC flexible プロジェクトの設定データ、たとえば使用されるタグやそのパラメータが指定されたテーブルを印刷するのに役立ちます。

##### アプリケーション

設定データをプロジェクトレポートに出力することができます。以下の項目についてプロジェクトレポートを出力することができます。

- 完全な WinCC flexible プロジェクト
- WinCC flexible の構成エレメント
- 単一オブジェクトまたは複数のオブジェクト

出力データの選択は、選択されたオブジェクトまたは構成エレメントにより異なります。データの構成は選択された出力フォーマットが"コンパクト"か"全体"かにより異なり、システムによってプロジェクトレポートが生成されるときに実行されます。

WinCC flexible の複数あるいはすべての構成エレメントの設定データを出力する場合、構成エレメントごとに別々の章が出力されます。可能なデータ量を考慮して、各 WinCC flexible の画面に対して別々の章が出力されます。

出力を開始する前に、プレビューでプロジェクトレポートを開くことができます。このプレビューを使用すれば、プロジェクトレポートを確認できます。

## 出力媒体

以下に対してプロジェクトレポートをオプションで出力することができます。

- プリンタ
- ファイル
- 画面

## 注記

プロジェクトレポートが生成され、次の必要条件が満たされた場合、WinCC flexible で使用するすべての画面は個別フォルダに保存されます。

- プロジェクトプレビューを使用して[終了]で終了した場合。
- プロジェクトレポートをファイルに出力した場合。

画面が".emf"の拡張子付きファイルに保存された場合。

## 14.1.2 レイアウトのストラクチャ

## はじめに

プロジェクト文書のレイアウトは、カバーシート、および作成データがダイナミックに入力される正式なコンテンツシートで構成されます。出力データが複数のページに及ぶ場合は、システムが自動的に改ページを挿入します。レイアウトは各種エリアに分割されます。ページエリアにはレイアウト全体が表示されます。印刷の余白はこのエリアで指定できます。印刷可能エリアはヘッダー、フッター、およびページ本体で構成されます。



ヘッダーとフッターはプロジェクトレポートのすべてのページに出力されます。ヘッダーとフッターはカバーシートには出力されません。

## カバーシートのレイアウト

プロジェクトに関する一般情報はカバーシートに出力することができます。カバーシートは事前定義されたフィールドを含み、このフィールドヘダイアログボックスによって対応する情報を入力できます。カバーシートには以下の情報を出力することができます。

- プロジェクト
- 会社名
- 部署
- 作成者
- 会社ロゴ
- プロジェクトロゴ

## コンテンツページのレイアウト

作成データはコンテンツページに出力されます。以下のエレメントはプロジェクトレポートに出力されます。

ライン	目次
タイトル	プロジェクトレポート用に選択した構成要素の名称
ラベル;ラベル	指定したオブジェクト、属性、及び出力する WinCC flexible 画面の指定。
配列	オブジェクトの作成した属性値の出力。

テーブルに記載された行はプロジェクトレポートに含まれるすべてのオブジェクトで繰り返されます。2つの形式で出力が可能です。

[簡易]形式では5列のテーブルに出力されます。[簡易]形式ではオブジェクトの最も重要な5つの属性が出力されます。出力する5つの属性は、システムで事前設定されます。これらの属性の選択は修正できません。

[完全]形式ではデータは2つの列に出力されます。[完全]形式ではオブジェクトのすべての属性がレポートに出力されます。

出力形式は[プロジェクト文書の印刷]ダイアログボックスで選択します。[コンテンツ]タブの[文書のプロパティ]エリアで[簡易]または[完全]形式を選択します。


## 14.2 レイアウトの使用

### 14.2.1 レイアウトの使用

#### はじめに

[プロジェクト文書の印刷]ダイアログボックスはレイアウトの編集に使用されます。このダイアログボックスは、新しいレイアウトの作成および既存のレイアウトの複製や削除に使用されます。

#### 概要







WinCC flexible では、プロジェクトレポートのためのベースとして既製のレイアウトを使用することができます。  コマンドボタンを使用して新規レイアウトを作成する場合、設定済みの既製のレイアウトが常に使用されます。WinCC flexible は既成のレイアウトから "標準レイアウト" を作成します。レイアウトは "Print Selection" ファンクションを使って、プロジェクトレポートの出力に使用されます。

プロジェクトレポート用のレイアウトは、WinCC flexible に集中して保存されており、すべてのユーザーがすべてのプロジェクトで使用することができます。作成者、会社名、プロジェクト名、ヘッダー、フッター、表示、および出力設定など、WinCC flexible プロジェクトのレイアウトの共通プロパティを設定します。このテンプレートを何度か複製して、それぞれの出力用の異なるコンフィグレーションデータを指定することができます。たとえば、WinCC flexible の各コンポーネント用に個別のプロジェクトレポートを作成します。

予め定義されたスタイルは、プロジェクトレポートをレイアウトするために使用できます。スタイルは、必要に応じて修正できます。スタイルは、個々のレイアウトとともに保存されるわけではありません。したがって、スタイルの変更はすべての既存のレイアウトに影響を及ぼします。

## レイアウト編集のコマンド


以下のコマンドを使用して、[プロジェクト文書の印刷]ダイアログボックスでレイアウトを編集することができます。

ボタン	ポップアップメニューコマンド	ホットキー
	新規作成	<CTRL+SHIFT+N>
	複製	<CTRL+SHIFT+D>
	削除	<DEL>
	名前の変更	F2
	印刷	<CTRL+SHIFT+P>
	プレビュー	<CTRL+SHIFT+V>
	エクスポート	<CTRL+SHIFT+E>

### 14.2.2 プロジェクト文書用のレイアウトの編集

#### はじめに

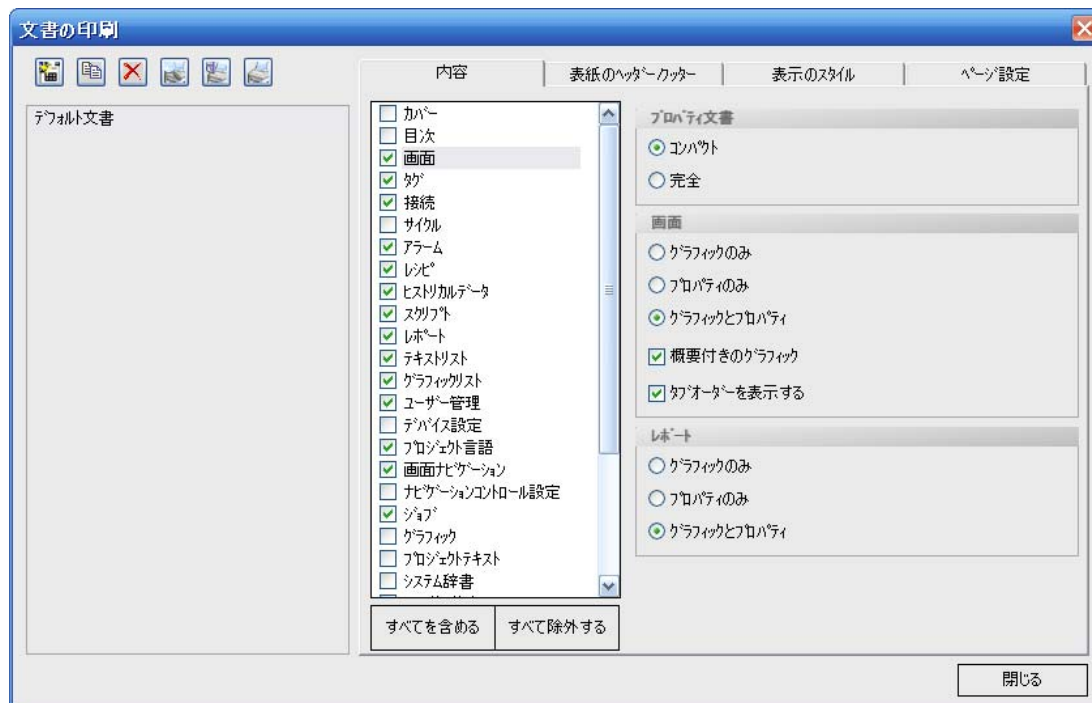
レイアウトは、[プロジェクト文書の印刷]ダイアログボックスを使って開きます。[プロジェクト文書の印刷]ダイアログボックスには次のようにしてアクセスします。

- [プロジェクト|プロジェクト文書の印刷...]メニューコマンド
- [印刷]ツールバーの  コマンドボタン

マウスを使って、編集するレイアウトのプロパティを開いてレイアウトを選択します。

### 編集可能なもの

選択したレイアウトのプロパティが、[プロジェクト文書の印刷]ダイアログボックスに表示されます。



次の表に、使用可能なカテゴリおよび編集できるものを示します。

タブ	編集可能なもの
"内容"	出力のデータの選択に使用されます。
"カバーシート"	カバーシートとヘッダーとフッターの内容の入力に使用されます。
"カバーページのスタイル"	スタイルの設定に使用されます。
"ページ設定"	用紙フォーマット、用紙方向、マージン、およびヘッダー、フッターの高さの設定に使用されます。

## 14.3 プロジェクトレポートの作成

### 14.3.1 プロジェクトレポート用のデータの選択

#### はじめに

データの選択は使用されているレイアウトで選択した出力フォーマットに依存します。システムによりそれぞれ指定された5つの属性が、各オブジェクトに対するプロジェクトレポートに"全体"出力フォーマットで出力されます。WinCC flexible コンポーネントの作成したすべてのオブジェクトの属性が、[簡易]あるいは[完全]出力フォーマットで出力されます。

#### 概要

[プロジェクト文書の印刷]ダイアログボックスの[内容]タブで、出力する WinCC flexible 構成要素を選択します [文書のプロパティ]エリアで、[簡易]または[完全]出力フォーマットを選択します。WinCC flexible 構成要素の[画面]および[レポート]では、データの出力を制限できます。以下のオプションが使用できます。

- "グラフィックのみ"
- "プロパティのみ"
- "グラフィックとプロパティ"

[画面] エリアに、さらに選択オプションがあります。

- "概要グラフィックス" は、画面に[常駐] ウィンドウを出力します。
- "タブ順序の表示" は、画面にタブ順序を出力します。

### 14.3.2 選択したオブジェクトのデータの出力


#### はじめに

WinCC flexible では、個々のオブジェクトの設定データを出力できます。複数の選択したオブジェクトに対して出力を実行することもできます。

#### 概要

[プロジェクト]ウィンドウまたは[オブジェクト]ウィンドウで、データ出力のためのオブジェクトを選択します。

選択したオブジェクトの設定データの出力は、常に"標準レイアウト"で実行される必要があります。目的の出力オプションは、このレイアウトで設定される必要があります。個々のオブジェクトデータの文書化に対して、別のレイアウトを選択することは有効です。

選択したオブジェクトの設定データは、プレビューで開きます。プリンタへの出力は、プレビューから開始できます。  ボタンを使用すると、データを他で利用するためにクリップボードにコピーできます。

### 14.3.3 プロジェクト文書用オブジェクトの選択

#### はじめに

WinCC flexible には、WinCC flexible の構成エレメントの個々のオブジェクトまたは複数のオブジェクトの設定データを出力するためのさまざまなオプションがあります。次の手順を使用して、出力を開始します。

- メインメニュー
- ツールバー
- 選択したオブジェクトのショートカットメニュー


#### オブジェクトの選択

[オブジェクト]ウィンドウを有効にし、[表示]>[オブジェクト]メニューコマンドを使用して [プロジェクト]ウィンドウで必要な WinCC flexible 構成エレメントを選択します。WinCC flexible の構成エレメントの既存のオブジェクトが、[オブジェクト]ウィンドウに表示されます。マウスを使用して、データを[オブジェクト]ウィンドウに出力したい1つまたは複数のオブジェクトを選択します。

WinCC flexible の構成エレメントのノードは、[プロジェクト]ウィンドウで開くこともできます。WinCC flexible の構成エレメントの既存のオブジェクトが表示されます。マウスを使用して1つまたは複数のオブジェクトを選択します。

#### データの出力

WinCC flexible には、データ出力用オプションが複数装備されています。オブジェクト選択の後、以下の手順で出力します。

-  ボタン
- 選択のショートカットメニューの[印刷の選択]コマンド
- メニューバーの[プロジェクト|印刷の選択]コマンド

設定データが"標準レイアウト"に挿入され、プレビューウィンドウで開きます。

## モバイルワイヤレス

### 15.1 基本原理

#### 15.1.1 Mobile Panel Wireless のアプリケーションのフィールド

##### オートメーションの WLAN

現在および将来の情報技術のモバイル通信の傾向は、オートメーションエンジニアリングの革新的なワイヤレスネットワークの使用へと移りつつあります。WLAN 経由のワイヤレス HMI システムにより、使いやすく柔軟性のある方法で管理でき、リアルタイムで事実上場所を問わずにマシン上またはプラントからモニタすることができます。

- これは稼働およびメンテナンスのコスト面および時間面での非常に大きな利点です。
- また、トラブルシューティングや障害の削減などにおける効率が向上します。
- モバイルオペレータコントロールおよびモニタのための最高レベルの柔軟性

##### 業界内のアプリケーション

ワイヤレスネットワークは、現在以下の業界内のアプリケーションで使用されています。

- 生産およびプロセスのオートメーション
- 自動車
- マシンおよびプラントの建設
- 食品、飲料、およびタバコ
- 倉庫業および物流
- コンベヤシステム

##### 機能

Mobile Panels Wireless と有線の Mobile Panel の違いはわずかです。ただし、Mobile Panel へのアクセスポイントは、多くのアプリケーションで異なります。この顧客の要求には、Mobile Panels Wireless により最適に応じています。

## アプリケーション事例集

Mobile Panel Wireless を使用すると、制限なく、安全工学を用いても用いなくてもプラントを移動することができます。状況に応じて、移動マシン部品またはプラント全体を制御およびモニタできます。これは、Mobile Panel Wireless により必要な詳細表示ができることで可能になりました。プラントの構造はますます複雑になっていますが、ボタンを追加した大型のディスプレイを作成できます。大きなボタンのついた大型のディスプレイにより、手袋をはめた状態であっても確実に、簡単にタッチコントロールできます。

## ソリューション

シーメンスにより統合されたソリューションにより、ワイヤレスネットワークを素早く、確実に計画、設定、シミュレーション、および拡張できます。Mobile Panels Wireless は標準化された WLAN 技術を基本としており、特にオートメーションに対する要求に応えるようになっています。

- 制限のない流用性と柔軟性
- 革新的で継続使用できるネットワークアーキテクチャ
- リアルタイムでの確実に決定的なワイヤレス転送
- ワイヤレス転送における最大の安定感とセキュリティ
- 十分な帯域幅と信号強度のモニタリング
- 固定 HMI デバイスおよび有線 Mobile Panel と互換性のあるファンクション
- 既存のシステムと全面的に統合可能
- アクセスコントロール、認証、および暗号化により保護された情報
- 安全性ファンクションによるフェイルセーフおよび PROFIsafe 経由の通信
- 厳しい業界状況に対応する堅牢な構造

## 15.1.2 Mobile Panel Wireless の動作

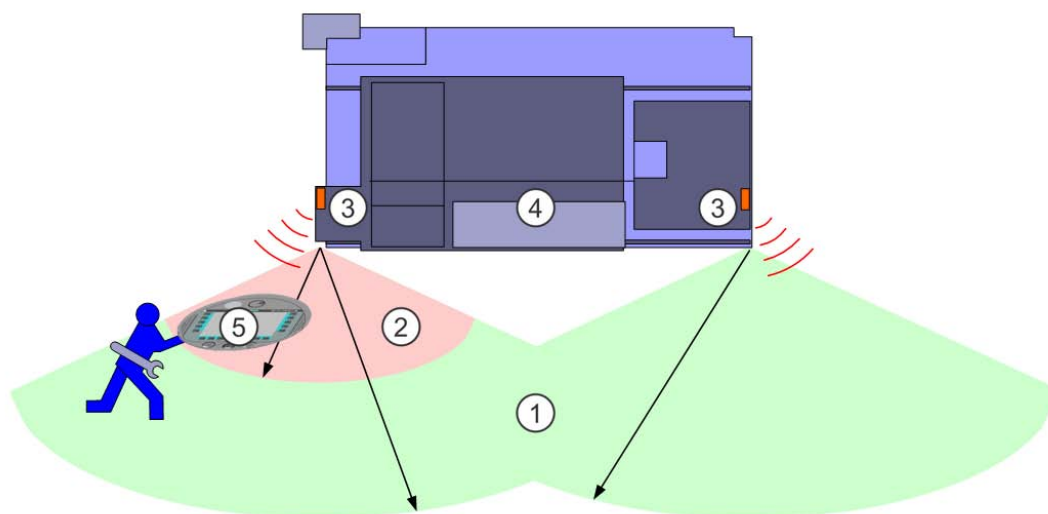
### 概要

プラントの特定の空間的部分である、ゾーンや有効範囲はワイヤレスネットワークを介したオペレータ制御やモニタにおいて特に重要です。ゾーンまたは有効範囲は、以下によって決定します。

- プラントまたはマシンの特定の部分。
- ワイヤレスネットワークの計画全体。
- 関連のコンポーネントの構成およびパラメータの設定。
- プラントでのトランスポンダの分布

これについて基本的に理解するため、下記の相互作用について以下のセクションに説明が記載してあります。

- HMI デバイス
- ゾーン
- 有効範囲
- トランスポンダ



### WLAN エリア

WLAN エリアとは、HMI デバイスが PLC と通信できる十分な信号強度のある、ワイヤレスでつながったネットワークのエリアです。Mobile Panel Wireless は WLAN エリアでのオペレータ制御およびプラントのモニタに使用されます。

## ゾーン

ゾーン(1)は Mobile Panel (5)により登録されたマシンまたはプラントのすぐ上流の空間的範囲です。これにより、HMI デバイスは特にゾーンが入力されたときに応答できます。たとえば、ゾーンを設定することにより画面が適切なプロセス表示に変更されるようにできます。

有線の Mobile Panel では、接続ケーブルの長さにより接続ボックスのゾーンが定義されていました。Mobile Panel Wireless では、マシン上の 1 つまたは複数のトランスポンダからの距離により WLAN エリア内のゾーンを定義しています。

## 有効範囲

安全性関連のオペレータ入力は、マシンまたはプラント(4)上部の WLAN エリアの制限された部分でのみ可能です。これは、有効範囲(2)として知られています。WLAN エリア全体で起こる緊急停止はこのルールの例外です。その他のすべての安全性関連のオペレータ入力では、下記の要件を満たす必要があります。

- HMI デバイス(5)で、Mobile Panel 277F IWLAN などのフェイルセーフオペレーションをサポートしている。
- HMI デバイスが PROFIsafe 通信に組み込まれている。
- HMI デバイスが有効範囲にログインしている。

HMI デバイスが有効範囲内でのみログインすることができます。また、他の HMI デバイスが同じ有効範囲にログインしていないことも必要です。

有効範囲の概念により、Mobile Panel のマシンへの必要な空間的な割り当てがワイヤレスの障害なく確実に行われます。これにより、安定性、フェイルセーフティ、および有効範囲内の全体の安全性の概念となるアクセスコントロールを強化しています。

### 通知

詳細については、「フェイルセーフオペレーション」に関するファンクションマニュアルを参照してください。未承認のアクセスに対するフェイルセーフオペレーションの Mobile Panel Wireless を含むプロジェクトを保護するため、常に管理者権限を割り当てるか、暗号化を設定する必要があります。

## トランスポンダによりゾーンおよび有効範囲を形成

ゾーン(1)および有効範囲(2)はマシン(4)またはプラントへのトランスポンダ(3)の空間的な割り当てにより形成されています。

### 通知

ログインおよびマシンオペレータ入力が明確であるようにするには、1 つの有効範囲が他の有効範囲と重なっておらず、ゾーンも別のゾーンと重なっていないようにする必要があります。そのためには、各トランスポンダに 1 つの有効範囲とゾーンのみが割り当てられている必要があります。

有効範囲とゾーンには、まったく異なるファンクションがあります。

- 有効範囲およびゾーンを個別に設定します。
- 有効範囲とゾーンは重複できます。

特種なエリアである有効範囲およびゾーンは 1 つまたは複数のトランスポンダからの最大間隔で定義されています。トランスポンダは、HMI デバイスがトランスポンダからの距離を計測している間に、その ID を円錐形のパターンで送信します。

設定されたゾーンおよび有効範囲は HMI デバイ스에格納されます。少なくともエリア内の 1 つのトランスポンダで以下の条件が満たされている場合、HMI デバイスは確実にエリア内にあります。

- HMI デバイスがトランスポンダのラジオコーン内にあり、トランスポンダ ID を受信できる。
- HMI デバイスが、エリアの設定した間隔以下のトランスポンダからの間隔を計測する。

#### 例:

- Transponder1 および Transponder2 の両方から 8 メートルの一定の距離が "MixingPlant" ゾーンに設定されている。これにより、ラジオコーンの開く角度と共に、画面変更のゾーンの外枠の制限を定義しています。
- 画面の "MixingPlant\_1" 画面への変更は、"MixingPlant" ゾーンに入力して設定されています。
- Transponder1 から 5 メートルの一定の距離が "MixingAxisControl" 有効範囲に設定されています。ラジオコーンの開く角度とともに、これにより安全性関連のオペレータ入力のできる有効範囲の外枠の制限を定義しています。

HMI デバイスが以下の間隔を計測:

- Transponder1 から 7 メートルの間隔。
- Transponder2 から 8 メートルの間隔。

#### 結果:

- HMI デバイスは "MixingPlant" ゾーンにあります。 "MixingPlant\_1" 画面が表示されています。
- HMI デバイスは "MixingAxisControl" 有効範囲の外側にあります。緊急停止以外は、HMI デバイスでは安全性関連のオペレータ入力はできません。



## 作業エリア

[ゾーン]作業エリアには、設定されたゾーンとそこに割り当てられたトランスポンダが表示されます。

## [プロパティ]ウィンドウ

ゾーンを選択すると、[全般]カテゴリのゾーンの名前、表示名および制限を編集できます。ゾーンには[エントリ時]ゾーンおよび[終了時]ゾーンがあります。イベントにシステムフアンクション ActivateScreen を割り付けます。

1つのゾーンに割り当てられるのは1つのトランスポンダのみで、それによりゾーンが重ならないようにします。

## 15.2.2 ゾーン作業エリア

### はじめに

[ゾーン]作業エリアには、ゾーンおよびそこに割り当てられたトランスポンダがテーブル形式で表示されます。トランスポンダのリストを作成して、特定のトランスポンダをゾーンに割り当てます。ゾーンの制限については、トランスポンダからの最大間隔で定義されています。

### 原理

作業エリアは[ゾーン]テーブルと[トランスポンダ]テーブルで構成されています。

ゾーン				
名前	表示名	ID	限界	コメント
MixingPlant	MixingPlant	1	8 m	
FillingPlant	FillingPlant	2	7 m	

トランスポンダ			
名前	ID	ゾーン	有効範囲
トランスポンダ_1	1	MixingPlant	MixingAxisControl
トランスポンダ_2	2	MixingPlant	
トランスポンダ_3	3	FillingPlant	
トランスポンダ_4	4		FillingAxisControl

ゾーンを[ゾーン]テーブルで選択すると、[トランスポンダ]テーブルには以下が表示されます。

- 有効なトランスポンダ: トランスポンダは選択したゾーンに割り当てられています。
- 無効なトランスポンダ: トランスポンダはゾーンに割り当てられていません。
- 使用できないトランスポンダ: トランスポンダは別のゾーンにすでに割り当てられています。割り当て直すには、該当するゾーンに切り替え、トランスポンダを無効化します。

ゾーン ID およびトランスポンダ ID は、最初に自動的に割り当てられます。ただし、これらは後で変更することができます。

以下の事項に注意してください。

- 最大で 254 のゾーンを設定できます。
- ゾーン ID は一意で、1~254 の範囲内である必要があります。
- トランスポンダをゾーンに割り当てずに、最初にトランスポンダを設定できます。
- ゾーン、有効な範囲、または両方に割り当てられるトランスポンダは 255 個のみです。
- トランスポンダ ID は一意で、1~65534 の範囲内である必要があります。

トランスポンダ ID はトランスポンダに設定されています。

### 15.2.3 有効な範囲

#### はじめに

次のセクションでは、Mobile Panel 277F IWLAN などのフェイルセーフオペレーションに対応している Mobile Panels Wireless だけを説明します。これらの HMI デバイスに限り"有効範囲"作業エリアが表示されます。

安全性関連のオペレーションを制御するため、有効な範囲を設定します。有効な範囲は 1 つまたは複数のトランスポンダからの最大間隔で定義されています。

通知
詳細については、「フェイルセーフオペレーション」に関するファンクションマニュアルを参照してください。

#### 開く

[デバイス設定]の[有効範囲]をダブルクリックして、作業エリアを[プロジェクト]ウィンドウで開きます。



## 15.2.4 [有効範囲]作業エリア

## はじめに

[有効範囲]作業エリアには、有効な範囲とそのトランスポンダがテーブル形式で表示されます。トランスポンダのリストを作成して、特定のトランスポンダを有効な範囲に割り当てます。有効な範囲の制限については、トランスポンダからの最大間隔で定義されています。

## 原理

作業エリアは[有効範囲]テーブルと[トランスポンダ]テーブルで構成されています。

有効範囲				
名前	表示名	ID	限界	コメント
MixingAxisControl	MixingAxisControl	1	5 m	
FillingAxisControl	FillingAxisControl	2	8 m	

トランスポンダ			
名前	ID	有効範囲	ゾーン
トランスポンダ_1	1	MixingAxisControl	MixingPlant
トランスポンダ_2	2		MixingPlant
トランスポンダ_3	3		FillingPlant
トランスポンダ_4	4	FillingAxisControl	

HMI デバイスはローカルデータからチェックサムを計算して、設定した有効な範囲およびトランスポンダが確実にマシン上でローカルに一致するようにします。計算されたチェックサムが[有効範囲]エディタに保存されているチェックサムとローカルで一致する場合は、プロジェクトは HMI デバイスからのみ起動できます。

## 通知

詳細については、「フェイルセーフオペレーション」に関するファンクションマニュアルを参照してください。

有効な範囲を[有効範囲]テーブルで選択すると、[トランスポンダ]テーブルには以下が表示されます。

- 有効なトランスポンダ: トランスポンダは選択した有効な範囲に割り当てられています。
- 無効なトランスポンダ: トランスポンダは有効な範囲に割り当てられていません。
- 使用できないトランスポンダ: トランスポンダは別の有効な範囲にすでに割り当てられています。割り当て直すには、該当する有効な範囲に切り替え、トランスポンダを無効化します。
- 有効な範囲のほか、トランスポンダに割り当てられたゾーンが表示されます。

有効な範囲 ID およびトランスポンダ ID は、最初に自動的に割り当てられます。ただし、これらは後で変更することができます。

以下の事項に注意してください。

- 最大で 127 の有効な範囲を設定できます。
- 有効な範囲 ID は一意で、1～127 の範囲内である必要があります。
- 有効な範囲の表示名は、その ID とは異なる名前である必要があります。
- トランスポンダを有効な範囲に割り当てずに、最初にトランスポンダを設定できます。

- ゾーン、有効な範囲、または両方に割り当てられるトランスポンダは 255 個のみです。
  - トランスポンダ ID は一意で、1~65534 の範囲内である必要があります。
- トランスポンダ ID はトランスポンダに設定されています。

<b>通知</b>
-----------

詳細については、「フェイルセーフオペレーション」に関するファンクションマニュアルを参照してください。
--

### ランタイム時の有効な範囲

ランタイム時に有効な範囲にログオンする場合、有効な範囲の表示名のみがランタイム時の言語でオペレータに表示されます。オペレータは有効な範囲 ID をプラントで読み込み、その ID を HMI デバイスに入力します。これにより、使用しているマシンが適切なマシンであることを確認します。ログイン完了後に確認ボタンが有効になります。

## 15.3 有効な範囲での作業

### 有効性

次の章では、Mobile Panel 277 IWLAN などのフェイルセーフオペレーションに対応している Mobile Panels Wireless だけを説明します。フェイルセーフオペレーションに対応していない Mobile Panels Wireless の設定例は「ゾーンを使用した作業」の章を参照してください。

### はじめに

以下の設定ガイドで、Mobile Panel Wireless のフェイルセーフオペレーションの有効な範囲の設定に必要な手順について説明しています。

<b>通知</b>
-----------

詳細については、「フェイルセーフオペレーション」に関するファンクションマニュアルを参照してください。
--

## 手順の概要

1. 有効範囲の設定:  
"MixingAxisControl"有効な範囲を 5 メートルの距離で"Transponder1"の円錐形エリアとして設定します。
2. 有効範囲へのログオン設定:  
ランタイム時にオペレータが有効範囲にログオンできるように[有効範囲の指定]オブジェクトを設定します。
3. 追加の有効範囲オブジェクトの設定:  
有効範囲内での位置や信号強度の表示用に追加オブジェクトを設定します。
4. 転送およびデバイスのパラメータを設定:
  - PROFIsafe 通信
  - WLAN ネットワーク
  - 電源管理
  - 転送モード
5. データチャネルの設定
6. ネットワーク操作の設定
7. トランスポンダの設定
8. 有効な範囲の実行
9. HMI デバイスの電源投入とテスト
10. 手動転送の開始
11. プラントの有効範囲の確認:  
有効範囲とそれらのトランスポンダを確認します。
12. チェックサムの決定
13. チェックサムでプロジェクトの再転送:  
プロジェクトで決定したチェックサムを入力して、プロジェクトを再度転送します。
14. 有効な範囲のテスト

---

### 注記

項目 5～10 の詳細については、HMI デバイスの取扱説明書を参照してください。

項目 11～14 の詳細については、「Mobile Panel 277F IWLAN のフェイルセーフオペレーション」のファンクション マニュアルを参照してください。

---

## ジョブの計画;ジョブノケイカク

### 16.1 スケジューラのアプリケーションフィールド

#### 定義

スケジューラの中で、システムファンクションまたはスクリプトをイベントにリンクします。たとえば、システムファンクションメール送信を[ランタイムの停止]イベントとリンクして、操作の終了時に特定の受信者に必ず電子メールを送信するようにします。

このため、次の場合のためのタスクが存在します。このイベントが発生すると、リンクしたファンクションが呼び出されます。ランタイムが終了すると、電子メールが送信されます。

#### 適用例

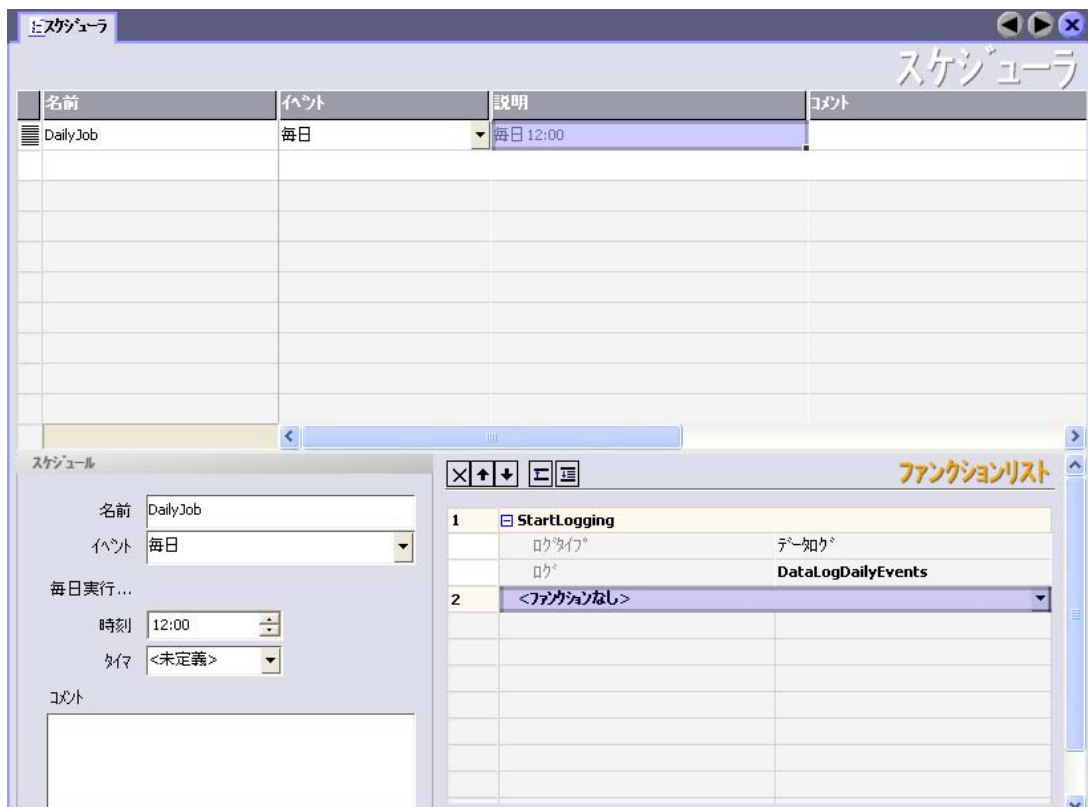
スケジューラを使用すれば、イベント制御ジョブを自動的に実行できます。たとえば、ジョブを使用して次の操作を自動化します。

- ログデータの定期スワップアウト
- アラームバッファオーバーフロー発生時のアラームレポートの印刷
- シフト終了時のレポートの印刷

## 16.2 ジョブとイベントを使用した作業

### はじめに

ジョブは、イベントのトリガと"フアクションリスト"で構成されます。



### 定義

スケジューラは、時間ベースのイベントとシステムイベントとを区別します。"毎日 12:00 に開始"などの時間ベースのイベントは特定の時刻に発生します。システムイベントは、[ランタイムの停止]や[ユーザーの変更]などです。

イベントは、[毎日 12:00 に開始]のように周期的に発生するか、[ユーザーの変更]のようにランダムに発生するかのどちらかです。

#### 通知

使用可能なイベントは、HMI デバイスによって異なります。HMI デバイスによってはすべてのイベントをサポートしていない場合があります。

"フアクションリスト"には、システムフアクションまたは行単位のスクリプトが含まれています。

## ジョブシーケンス

イベントが発生すると、スケジューラがそのイベントに関連するジョブを開始します。ジョブは連続して実行させることができます。フアクションリストの各行を実行することによってジョブが実行されます。

システムイベントの場合は、HMI デバイスごとに1つのジョブだけを作成して実行することができます。

---

### 注記

短時間に多数のジョブを実行させると、時間遅延が発生する可能性があります。周期的なイベントの場合は、次のイベントが発生する前にすべてのジョブが実行済みか確認してください。

---

## 時間ベースのイベント用のタイマ

毎日、毎年または一回限りのイベント用に作成した開始時刻をランタイム中にダイナミックに変更するには、内部タグをタイマとして選択します。このタグの値によって、ランタイム中のジョブの開始時刻が決定します。

<b>通知</b>
タグは"DateTime"タイプでなければなりません。

## 16.3 エlement

### 16.3.1 "スケジューラ"エディタ

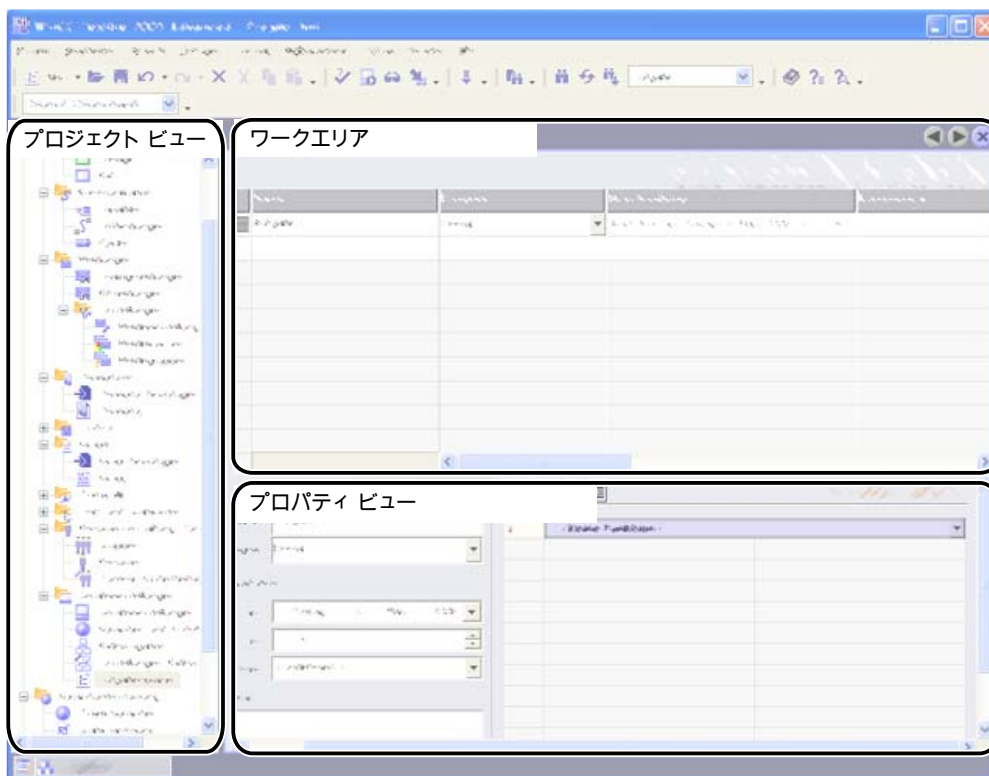
#### はじめに

スケジューラでは、イベントのファンクションリストを作成して、ジョブを計画することができます。

#### 開く

[プロジェクト]ウィンドウでスケジューラを開くには、[スケジューラ]をダブルクリックします。

#### レイアウト



#### 作業エリア

作業エリアには、計画済みのジョブが表示されます。

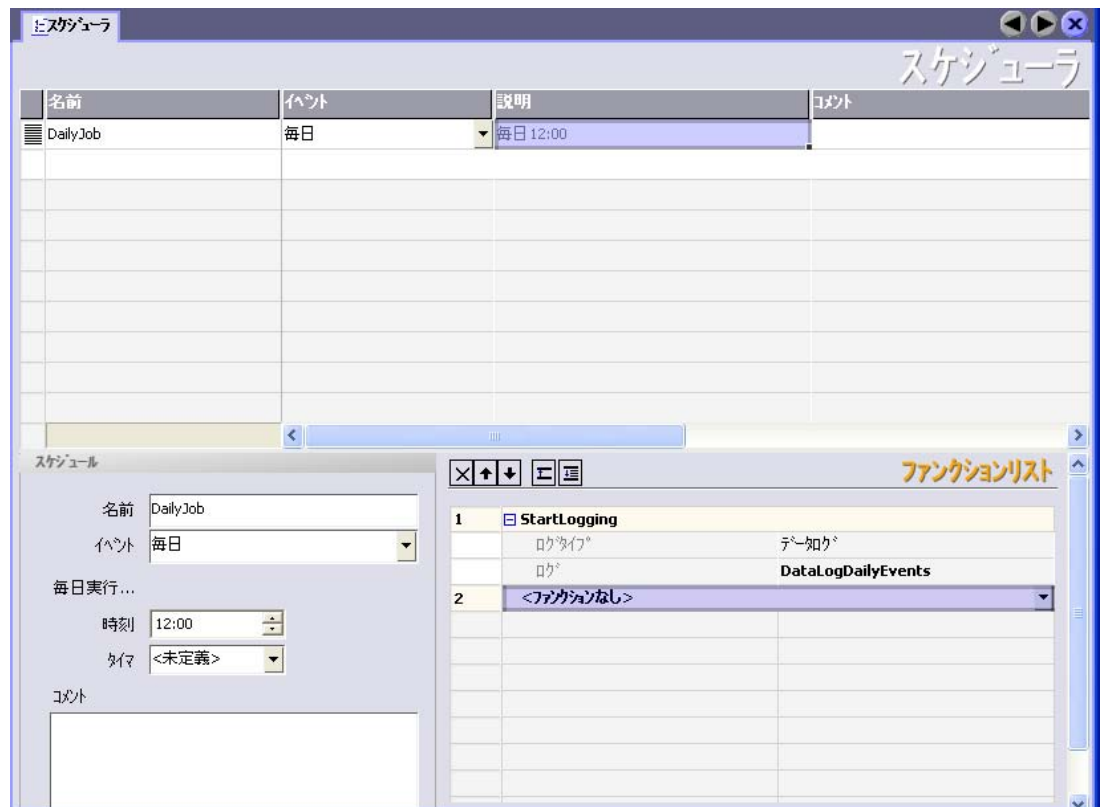
## 16.3.2 [スケジューラ]エディタの作業エリア

### はじめに

作業エリアには、トリガとなるイベントやファンクションリストで構成される計画済みジョブが表示されます。

### レイアウト

作業エリアは、ジョブのテーブル、プロパティ、およびファンクションリストで構成されます。



ジョブのテーブルには、ジョブ、トリガとなるイベント、追加情報が表示されます。ラベルやコメントを割り付けてイベントを選択できます。スケジューラは、ジョブの記述をコンパイルします。

プロパティには、トリガとなるイベントと一緒にジョブも表示されます。時間ベースのイベントはプロパティで指定されます。

ファンクションリストの中に、ジョブで実行すべきファンクションまたはスクリプトを組み込みます。

### 注記

コンパイルされた記述は、ジョブのタイミングを含む、文書化されたジョブの要約を提供します。ユーザーインターフェースで選択したエlementにマウスポインタを合わせて Tooltip ファンクションを使用すれば、さらに詳細な情報を表示できます。



## プロジェクトバージョンの管理

### 17.1 プロジェクトバージョンのアプリケーション

#### 原理

プロジェクトバージョンは、バージョン管理によって定義された保存先に保存されているプロジェクトのコピーです。各バージョンのプロジェクトは、常に固有のプロジェクトステータスを表示しています。以前のバージョンを復元することができます。

#### 適用例

以下の状況でプロジェクトバージョンを使用することができます。

1. 受け取られた参照バージョンのログ: 必要に応じて、以前のバージョンを復元することができます。
2. 旧プロジェクトバージョンを改善するには。たとえば、すでにランタイムで使用中の旧プロジェクトバージョンのエラーを修正したい場合があります。ただし、その間にプロジェクトの開発は続いています。このエラーは旧プロジェクトバージョンで修正されません。現在のバージョンのプロジェクトは、この時点で影響を受けない状態のままになっています。
3. 異なる HMI デバイス上での異なるプロジェクトステータスへのバージョン割り付け: HMI デバイスに故障が発生した場合、対応するプロジェクトバージョンを転送することができます。
4. 既存プロジェクトまたは試験的なプロジェクトの個別バージョン識別: テストバージョン、多様なデバイスやプラントタイプ、あるいは特殊モデルのマシン
5. 異なるメディアへのデータのバックアップ 大容量記憶装置の故障などによるデータの喪失を避けることができます。プラント操作コンポーネントの最新バージョンのプロジェクトは、これにより特に影響を受けます。

## 17.2 バージョン管理の基本

### 定義

バージョン管理は、プロジェクト自体とプロジェクトのバージョンとを区別します。プロジェクトとは、WinCC flexible で編集できるプロジェクトフォルダ内のファイルのことです。プロジェクトバージョンとは、バージョン管理によって"RCS"の保存先に保存されているファイルのことです。



コピーは、プロジェクトフォルダと"RCS"の保存先との間で双方向に交換されます。プロジェクトバージョンが新規作成されると、プロジェクトのコピーが"RCS"の保存先のファイル内に保存されます。旧プロジェクトバージョンを編集するとき、ローカルコピーがプロジェクトフォルダに作成されます。

### 注記

プロジェクトバージョンは、シーケンシャルなバージョン番号で区別できます。バージョン番号は、ランチで重複が起きないように自動的に割り付けられます。

### はじめに

バージョン	状態	ラベル	作成者	日付 - 時刻	コメント
1		<空のラベル>	EUROE02K	2005/08/11 12:54	<空のコメント>
2		<空のラベル>	EUROE02K	2005/08/11 12:57	<空のコメント>
2.1.1		<空のラベル>	EUROE02K	2005/08/11 13:13	<空のコメント>
2.1.2		<空のラベル>	EUROE02K	2005/08/11 13:25	<空のコメント>
3		<空のラベル>	EUROE02K	2005/08/11 12:59	<空のコメント>

プロジェクトは継続的に開発が続けられます。変更は、順次追加されていきます。定期的にプロジェクトのバージョンを追加していくと、シーケンシャルにプロジェクトバージョンが

作成されていきます。1、2、3 などのような整数の番号が付けられているプロジェクトバージョンが、開発のトランク(主要部分)を形成します。

いくつかのブランチが存在する場合もあります。旧プロジェクトバージョン 2 を編集し、定期的にバージョンを追加した場合は、たとえば、2.1.1、2.1.2、2.1.3 のようなブランチが作成されます。

## 17.3 トランク

### 原理

プロジェクトの現在の状態を記録するためにプロジェクトバージョンを新規作成します。新バージョンのプロジェクトは、現在のプロジェクトのコピーです。はじめのプロジェクトバージョンはバージョン番号"1"が割り付けられます。

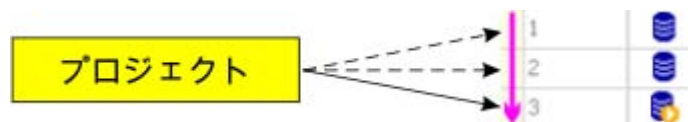
#### 通知

一度プロジェクトバージョンがバージョン管理に保存されると、もはや変更することはできません。変更があるたびに、必ず次のプロジェクトバージョンが作成されます。

プロジェクトは、WinCC flexible で継続的に開発が続けられます。今行っている変更は、バージョン 1 でのプロジェクトの状態に基づいています。バージョン 1 が現在のバージョンです。

開発が続いていけば、やがて新しい段階に到達します。プロジェクトの現在の状態を記録するためにプロジェクトバージョンを新規作成します。バージョン 2 が現在のバージョンです。

次のバージョンのプロジェクトが作成されると、そのプロジェクトの状態はバージョン 3 として保存されます。



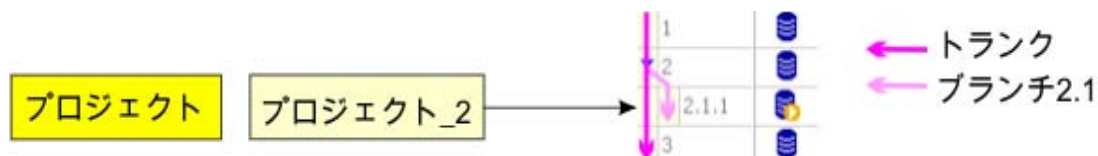
### 旧プロジェクトバージョン



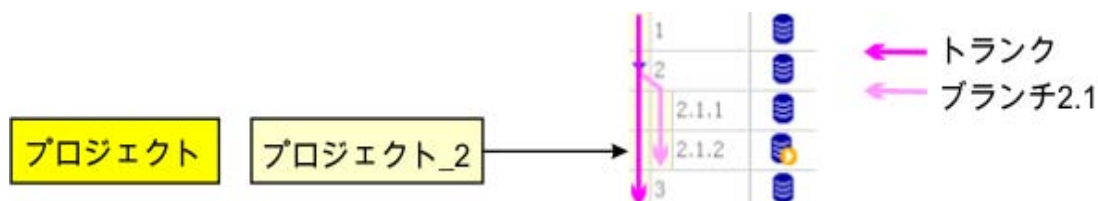
旧プロジェクトの状態にアクセスするには、たとえばバージョン 2 など、それぞれのプロジェクトバージョンを開きます。プロジェクトフォルダー"Project\_2"と WinCC flexible 内にバージョン 2 のコピーが作成されます。これで旧プロジェクトの状態が"Project\_2"で編集できるようになります。変更はバージョン 2 を基準に行われます。バージョン 2 が現在のバージョンです。

## 17.4 ブランチ

### 原理



"Project\_2"のプロジェクト状態をバージョン管理に記録するには、"Project\_2"のプロジェクトバージョンを新規作成します。バージョン3が既に存在しているため、新しいプロジェクトバージョンはバージョン2.1.1として保存されます。バージョン2.1.1が現在のバージョンです。次のバージョンは2.1.2です。これでバージョン2に基づくブランチが、トランクの開発に追加されました。

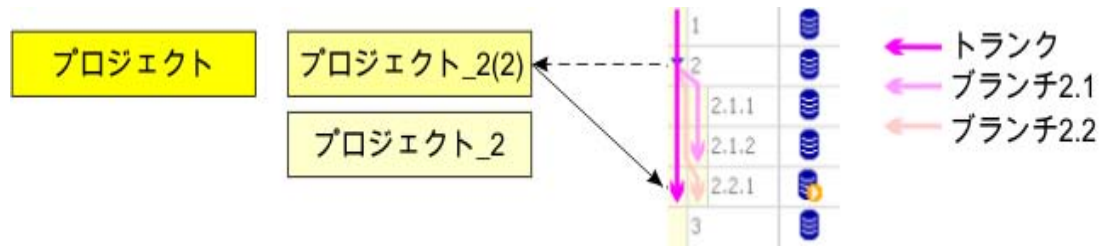


このブランチ(分岐部)の開発を継続するには、プロジェクトフォルダ内の"Project\_2"を必ず開きます。"Project\_2"の新バージョンをもう一つ作成すると、それはバージョン2.1.2として保存されます。この時点で、バージョン2.1.2が現在のバージョンです。ブランチ2.1は、2.1.1、2.1.2、2.1.3などのすべてのプロジェクトバージョンから成っています。

### 注記

1つのブランチは必ずバージョン2などのような1つのバージョンから始まります。複数個のブランチが同じバージョンから派生することもあります。ブランチに、たとえば2.1のように、派生したバージョン番号が必ず割り付けられるのはこのためです。ブランチのプロジェクトバージョンには、たとえば2.1.2のように、バージョン番号として必ず2桁の数字の拡張番号が必要です。

## その他のブランチ



ただし、バージョン管理で再びバージョン 2 を開くときは、"Project\_2"はプロジェクトフォルダ内で上書きされません。その代わりに、"Project\_2(2)"として保存されます。

"Project\_2(2)"では、バージョン 2 の状態を再び編集することができますようになります。

プロジェクトバージョンを新規作成すると、バージョン番号 2.2 の新しいブランチが作成されます。ブランチ 2.1 は既に存在しています。新しいプロジェクトバージョンはバージョン 2.2.1 として保存されます。

ただし、ブランチ 2.1 で作業を継続するには、たとえばバージョン 2.1.2 のような、このブランチで最大の番号をバージョン管理で開くことができます。または、プロジェクトフォルダ内の一番最後に編集した"Project\_2"を開くことができます。


## 17.5 エレメント

### 17.5.1 "プロジェクトバージョン"エディタ

#### はじめに

バージョン管理は、現在のプロジェクトから作成されたプロジェクトのバージョンを示します。新しいプロジェクトバージョンを作成したり、古いプロジェクトバージョンを開いたりすることができます。

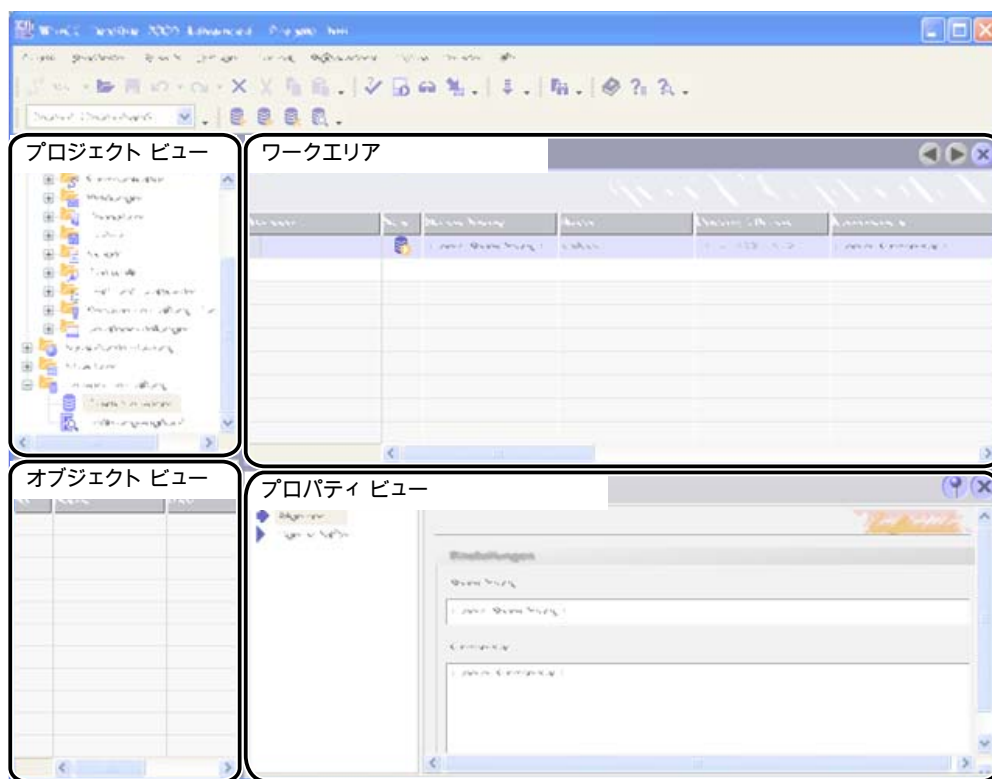
#### 開く

[プロジェクトバージョン]をダブルクリックすると、[プロジェクト]ウィンドウで"プロジェクトバージョン"エディタが開きます。 

#### 注記

プロジェクトバージョンのロケーションを別のプロジェクトに移動した場合は、続いて表示される[プロジェクトバージョンのパスのリセット]ダイアログで新しいロケーションを入力します。

## レイアウト



## メニューバー

メニューバーから、バージョン管理のファンクションを開始することができます。

## ツールバー

[プロジェクトバージョン]ツールバーでは、バージョン管理のファンクションを開始することができます。ツールバーは、デフォルトで表示されています。ツールバーの状況に応じたメニューで、ツールバーを表示したり隠したりすることができます。

作業エリアの状況に応じたメニューで、バージョン管理のファンクションにアクセスすることもできます。

## 作業エリア

作業エリアには、作成済みのプロジェクトのバージョンが表示されます。

## [プロパティ]ウィンドウ

プロジェクトのバージョンが選択されている場合、[プロパティ]ウィンドウでは、その名前と関連コメントを編集できます。現在と次のバージョン番号は、バージョン管理によって割り付けられます。




## 17.5.2 Bedienelemente Projektversionen\_\_ohneVersionsvergleich

### メニューコマンド

[オプション|バージョン管理]メニュー内

メニューコマンド	簡単な説明	ホットキー	使用可能性
プロジェクトバージョンの移動	プロジェクトバージョンの保存先の変更		バージョン管理には、少なくとも1つのプロジェクトバージョンが含まれています。

### [プロジェクトバージョン]ツールバー

シンボル	簡単な説明	使用可能性
	プロジェクトバージョンの作成	
	旧バージョンのプロジェクトを開く	プロジェクトバージョンが選択されています。
	プロジェクトバージョンの削除	プロジェクトバージョンが選択されています。

## 17.5.3 バージョン管理作業エリア

### はじめに

作業エリアには、現在のプロジェクトから作成したプロジェクトバージョンのテーブルが表示されます。新しいプロジェクトバージョンを作成したり、古いプロジェクトバージョンを開いたりすることができます。

#### 注記


作業エリアには、旧プロジェクトバージョンを開いている場合でも、プロジェクトのすべてのバージョンが常に表示されています。


## レイアウト



The screenshot shows a software window titled 'プロジェクトバージョン' (Project Version). The window contains a table with the following columns: 'バージョン' (Version), '状態' (Status), 'ラベル' (Label), '作成者' (Creator), '日付 - 時刻' (Date - Time), and 'コメント' (Comment). The table lists five versions, with version 2.1.2 highlighted in blue, indicating it is the current version. A small blue icon with a white 'P' is visible in the '状態' column for version 2.1.2.

バージョン	状態	ラベル	作成者	日付 - 時刻	コメント
1		<空のラベル>	EUROE02K	2005/08/11 12:54	<空のコメント>
2		<空のラベル>	EUROE02K	2005/08/11 12:57	<空のコメント>
2.1.1		<空のラベル>	EUROE02K	2005/08/11 13:13	<空のコメント>
2.1.2		<空のラベル>	EUROE02K	2005/08/11 13:25	<空のコメント>
3		<空のラベル>	EUROE02K	2005/08/11 12:59	<空のコメント>

作業エリアには、プロジェクトバージョンのツリー表示が含まれています。作業エリアには、プロジェクトバージョンのトランクが最上位レベルとして表示されます。このシンボル  が付いたプロジェクトバージョンは、ブランチの始まりを表しています。Windows エクスプローラでフォルダを開くときと同じ要領で、ブランチのウィンドウを開くことができます。

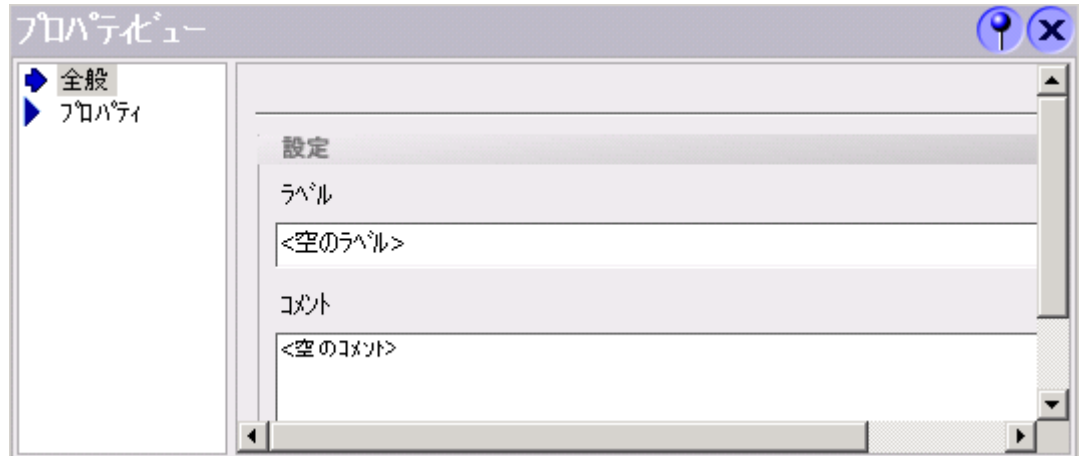
現在のプロジェクトに基づくプロジェクトバージョンには、必ず  アイコンが表示されます。このプロジェクトバージョンが現在のバージョンです。

[作成者]、[日付/時刻]、[バージョン]、[ステータス]情報などが、バージョン管理により割り付けられます。コメントと説明を入力することができます。この情報は、プロジェクトバージョンの保存先に保存されます。コメントまたは名前を変更した場合は、この情報は直ちに更新されます。

## 17.5.4 [プロパティ]ウィンドウ

### はじめに

[プロパティ]ウィンドウには、[全般]グループで選択されたプロジェクトバージョンの"名前"とコメントが表示されます。[プロパティ]グループには、[プロジェクトバージョン]の下に現在のバージョンと次のバージョンが表示されます。



### ラベル

この名前は、多くのプロジェクトにおける異なるプロジェクトバージョンを識別するために使用されます。例: "2001年1月15日リリース"。異なるプロジェクト内でプロジェクトバージョンに対して同じ名前を使用する場合、それら2つのプロジェクトバージョンがグループであり、同じ"2001年1月15日リリース"のステータスを共有していることを明示します。

### 現在のバージョンと次のバージョン

[現在のバージョン番号]と[次のバージョン番号]の下に表示された情報は、現在のプロジェクトに関連しています。[現在のバージョン番号]には、現在のプロジェクトに基づくプロジェクトバージョンのバージョン番号が表示されます。[次のバージョン番号]には、次にプロジェクトバージョンを新規作成したときのバージョンに割り付けられるバージョン番号が表示されます。

## 17.6 プロジェクトバージョンの取り扱い

### 17.6.1 バージョンの比較

#### 原理

現在のプロジェクトが現在のステータスで、他のプロジェクトバージョンと比較されます。バージョンの比較は、現在のプロジェクトが基礎にしている同じプロジェクトバージョンとでも、下位または上位のバージョン番号をもつプロジェクトバージョンとでも可能です。

プロジェクトバージョンが開かれ、現在のプロジェクトステータスと比較されます。比較の結果は、変更ログと同様に[バージョンの比較]テーブルに表示されます。

言語を変更しても、バージョンの比較により決定されるオブジェクト名には影響を及ぼしません。オブジェクト名を新しいユーザーインターフェース言語で表示するには、バージョンの比較を繰り返します

#### はじめに

[バージョンの比較]テーブルには、作成、削除または変更されたすべてのオブジェクトが表示されます。

作成済み: そのオブジェクトは現在のプロジェクト内には存在するが、プロジェクトバージョン内には存在しません。

削除済み: そのオブジェクトはプロジェクトバージョン内には存在するが、現在のプロジェクト内には存在しません。

変更済み: そのオブジェクトは現在のプロジェクト内にもプロジェクトバージョン内にも存在しています。ただし、そのオブジェクトのプロパティは異なります。

[バージョンの比較]テーブルで変更済みのオブジェクトを選択し、このオブジェクトが現在のプロジェクトで変更されている場合は、第2のテーブルが表示されます。第2のテーブルには、選択したオブジェクトでどのプロパティが変更されたかが表示されます。

## 変更のロギング;ヘンコウノロギング

### 18.1 変更ログ用のアプリケーション

#### 定義

変更ログには、連続するテーブル内のプロジェクトに加えられた変更がすべて記載されています。テーブルには、変更済みオブジェクトとオブジェクトのプロパティが入っています。統合されたプロジェクトで、STEP 7 と SIMOTION の変更は明示的には識別されません。

#### 適用例

1. 特定の産業部門では、製品のライフサイクル全体と製品の生産条件の完全で信頼のおける検証に特に関心が寄せられています。だれが何を、いつ、どこで、どのような理由で行ったかという証拠が保存されます。そうすれば、証拠を何年後でも文書化することができます。その1例は製薬業界です。
2. たとえば、米国のFDA(食品医薬品局)は、食品および医薬品に対する規則を規定する責任を担っています。  
プロジェクトの変更を文書化する必要があるのはFDAや各種の技術検査会社だけにとどまらず、他の多くの産業部門や製品もこのような文書化制度の対象となります。
3. 技術系の会社では、その会社のプロジェクト設定内でカスタマーの注文を処理しています。カスタマーが契約の枠外の変更を要求することは、よくあります。変更ログは、このような変更の文書化を手助けし、追加コスト算出の基となるものを作成します。

## 18.2 プロジェクトの変更ログ

### 原理

どのプロジェクトにも、独自の変更ログがあります。プロジェクトの変更は、変更ログが有効である限り記録されます。

### 記録済みのプロジェクトの変更

以下のプロジェクトの変更が記録されます。

- WinCCflexible で作成される新規変更ログ
- 変更ログの有効化
- オブジェクトの作成
- オブジェクトの削除
- オブジェクトの名前の変更
- オブジェクトの変更
- オブジェクトのコピー
- オブジェクトの移動
- プロジェクトを保存する
- プロジェクトの名前の変更
- 変更ログ内のコメントの変更
- 変更ログの無効化

以下のプロジェクトの変更は、バージョン管理の行われているプロジェクトでのみ記録されます。

- プロジェクトバージョンの新規作成
- 旧バージョンのプロジェクトを開く
- プロジェクトバージョンを別の保存先へ移動する



## 例

変更ログを有効にして、最初に"PictureChangeLog"などのオブジェクトを編集すると、[変更]テーブルの新しい行に新しいオブジェクトが挿入されます。

[位置]などの変更されたオブジェクトプロパティすべてが個別の行に入力され、"200"などの新しい値が[詳細]テーブルに入力されます。

---

### 注記

画面では、[高さ]や[幅]などのオブジェクトプロパティは、オブジェクトプロパティの[サイズ]としてまとめられます。高さに加えられた変更は、[サイズ]への変更として記録されません。[対角線方向への移動]などのアニメーションのために終了位置を変更する代わりに開始位置のオフセットをロギングします。

---

## 新規変更セクション

プロジェクトセッションは、1つまたは複数の変更セクションで構成されています。変更セクションは、以下のアクションに限定されます:

- プロジェクトを開く
- プロジェクトバージョンの新規作成
- プロジェクトを保存する
- 変更ログを有効にします

新規変更セクションが開始されます。変更セクションは、以下のアクションで終了します。

- プロジェクトを閉じる
- 旧バージョンのプロジェクトを開く
- プロジェクトを再保存する
- 変更ログを無効にします

## 18.4 バージョン管理下のプロジェクトの変更ログ

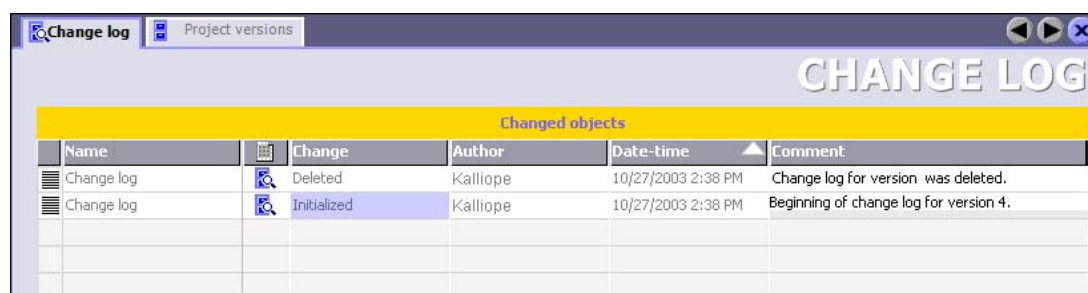
### はじめに

以下のセクションでは、プロジェクトのバージョン管理を行っている場合の変更ログに対する特別な状況について説明します。プロジェクトバージョンごとに、追加エントリをもつ独自の変更ログがあります。

### プロジェクトバージョンの新規作成

新規のプロジェクトバージョン作成時には、変更ログがバージョン管理されている現在のプロジェクトと一緒に保存されます。変更ログは、最後のエントリ、"保存済みプロジェクト"として追加されます。WinCC flexible で変更ログが削除されます。

新規変更ログが、新規プロジェクトバージョンで作成されます。

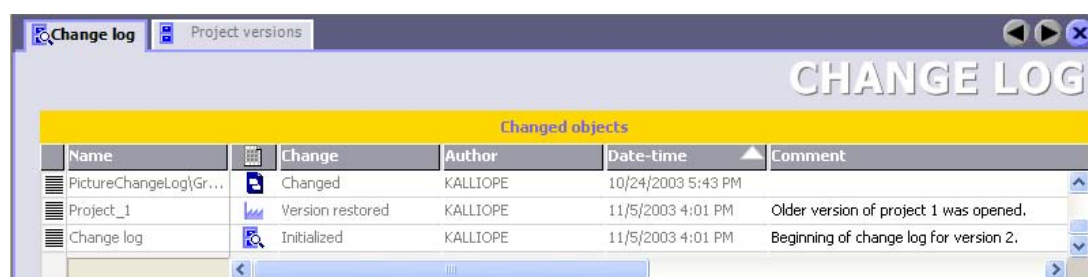


The screenshot shows a window titled "Change log" with a sub-tab "Project versions". The main area is titled "CHANGE LOG" and contains a table with the following data:

Changed objects					
Name	Change	Author	Date-time	Comment	
Change log	Deleted	Kalliope	10/27/2003 2:38 PM	Change log for version was deleted.	
Change log	Initialized	Kalliope	10/27/2003 2:38 PM	Beginning of change log for version 4.	

### 旧バージョンのプロジェクトを開く

古いプロジェクトバージョンが開いている場合、変更ログはプロジェクトと共に開きます。



The screenshot shows a window titled "Change log" with a sub-tab "Project versions". The main area is titled "CHANGE LOG" and contains a table with the following data:

Changed objects					
Name	Change	Author	Date-time	Comment	
PictureChangeLog\Gr...	Changed	KALLIOPE	10/24/2003 5:43 PM		
Project_1	Version restored	KALLIOPE	11/5/2003 4:01 PM	Older version of project 1 was opened.	
Change log	Initialized	KALLIOPE	11/5/2003 4:01 PM	Beginning of change log for version 2.	

プロジェクトバージョンに加えられた変更はすべて、変更ログが有効である限り記録されます。

## 18.5 エlement

### 18.5.1 "ログの変更"エディタ

#### はじめに

変更ログには、プロジェクトで行われたコンフィグレーションの変更が表示されます。ユーザーは、オブジェクトやオブジェクトプロパティの変更者、変更日時、対応するコメントを見ることができます。

#### 開く

[バージョン管理|変更ログ]をダブルクリックし、[プロジェクト]ウィンドウで変更ログを開きます。

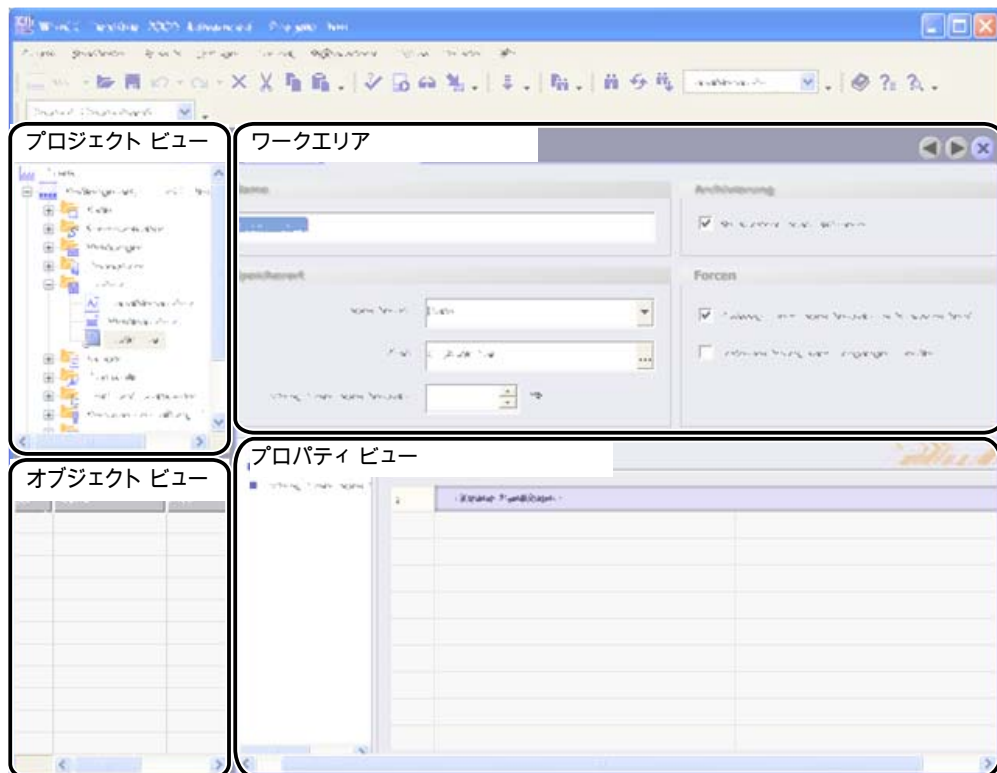
---

#### 注記

以前のプロジェクトバージョンの変更ログを開くには、バージョン管理で必要なプロジェクトバージョンを最初に開きます。

---

#### レイアウト



## メニューバー

メニュー項目[オプション|バージョン管理]で、変更ログを有効、無効にすることができます。

## ツールバー

[プロジェクトバージョン]ツールバーから変更ログを開くことができます。

## 作業エリア

作業エリアには、修正済みオブジェクトとオブジェクトプロパティが表示されます。

## 18.5.2 ログオペレータ制御の変更

### メニューコマンド

[オプション|バージョン管理]メニュー内

メニューコマンド	簡単な説明	ホットキー	使用可能性
[変更ログの有効化]	変更ログを有効にします		変更ログが無効になっています
[変更ログの無効化]	変更ログを無効にします		変更ログが有効になっています



## 転送;テンソウ

### 19.1 基本

#### 19.1.1 転送操作の基本原理

##### 転送

転送操作とは、プロジェクトを実行する HMI デバイスに完全なプロジェクトファイルを転送することを言います。

[プロジェクト]>[コンパイラ]>[コンパイル]または[プロジェクト]>[コンパイラ]>[すべての再構築]メニューコマンドを選択して、設定後のプロジェクトの整合性を確認します。

---

##### 注記

###### すべての再構築

商用プロセスにプロジェクトをリリースする前に[すべての再構築]コマンドを実行して、プロジェクト全体を再コンパイルします。

また、現在のエンジニアリングセッションでの差分データのコンパイル所要時間を削減するために、適切な周期で[すべての再構築]コマンドを実行することをお勧めします。

[プロジェクト]>[コンパイラ]>[すべての再構築]メニューコマンドを選択して、フルコンパイルを実行してください。

複数の HMI デバイスを設定している場合、[すべての再構築]コマンドを実行すると、[生成する HMI デバイスの選択]ダイアログが開きます。

このダイアログから生成する HMI デバイスを選択します。複数選択も可能です。

---

一貫性チェックの完了後に、システムはコンパイル済みのプロジェクトファイルを生成します。このプロジェクトファイルにはプロジェクトと同じ名前が割り付けられますが、拡張子は".fwx"になります。コンパイルされたプロジェクトファイルを、設定した HMI デバイスに転送します。

---

##### 注記

診断メッセージのために fwx ファイルのサイズはかなり大きくなる場合があります。fwx ファイルのサイズが大きすぎて HMI デバイスへ転送できない場合、アラーム設定で診断メッセージを無効にしてください。

---

プロジェクトデータを転送するには、HMI デバイスを設定コンピュータに接続する必要があります。HMI デバイスが PC の場合、ディスクなどのデータ媒体を使用しても転送操作を実行できます。

\*.pwx が見つからない場合にプロジェクトを再コンパイルすると、データの転送中にエラーメッセージが受信されます。

## 基本的な手順

1. WinCC flexible プロジェクトに、個々の HMI デバイスに応じた転送設定を入力します。
2. プロジェクトを転送する HMI デバイスで転送モードを入力します。
3. 設定コンピュータから HMI デバイスに、コンパイルしたプロジェクトファイルを転送します。プロジェクトファイルは、転送設定で選択されている各チェックボックスに対応する HMI デバイスすべてに転送されます。

## 転送モード

転送操作中、HMI デバイスは"転送モード"になっている必要があります。HMI デバイスのタイプに基づいて、転送モードは次のように有効になります。

- Windows CE システム

初めて作動するとき、HMI デバイスは自動的に転送モードで起動します。

HMI デバイスの設定メニューでこの転送オプションが有効になっている場合、転送操作を追加起動するたびに HMI デバイスが転送モードに切り替わります。

転送オプションが無効になっている場合、HMI デバイスを再起動して、[スタート]メニューで転送アプレットを呼び出します。または、プロジェクト内の Change Operating Mode システムファンクションを設定します。

- PC

HMI デバイスが、プロジェクトがまだ格納されていない PC の場合、最初の転送操作の前に"RT Loader"で転送モードを手動で有効にする必要があります。

HMI デバイスで転送モードを設定する方法の詳細については、マニュアルを参照してください。

---

## 注記

### PROFIBUS 上での MP 377 を使ったオペレーティングシステムの転送

PROFIBUS で使用できる画像サイズとポーレートのため、PROFIBUS を使った MP 377 の画像転送には 1 時間かかることがあります。

USB またはイーサネット経由でオペレーティングシステムまたはイメージを転送します。

---

## HMI デバイスバージョン

プロジェクトをオペレータデバイスに転送すると、設定したオペレーティングシステムバージョンが、HMI デバイス上のバージョンに対応しているかどうかチェックされます。バージョンが違っていると、転送が中止されて、メッセージが表示されます。WinCC flexible プロジェクトまたは HMI デバイスのオペレーティングシステムバージョンが異なる場合、次の可能性が考えられます。

- HMI デバイス上のオペレーティングシステムの更新。

詳細については、「Operating System Transfer (オペレーティングシステム転送)」章を参照してください。

### 19.1.2 転送設定;テンソウセッテイ

#### はじめに

プロジェクトの HMI デバイス毎に、個々に転送設定を入力することができます。転送設定には、転送操作を行うための通信設定と HMI デバイスの選択が含まれます。

[転送設定]ダイアログボックスには、選択した HMI デバイスで実際に使用可能な設定だけを入力することができます

#### 転送操作のための HMI デバイスの選択

転送操作を実行すると、設定コンピュータ上の転送設定で、関連のチェックボックスを選択したすべての HMI デバイスに、コンパイル済みプロジェクトファイルが転送されます。

HMI デバイスのショートカットメニューを使用して、この特定の HMI デバイスのみの転送操作を開始する場合も、設定コンピュータの転送設定で関連するチェックボックスを選択する必要があります。

#### 転送モード

HMI デバイスに応じて、以下の転送モードのうち 1 つまたは複数の転送モードが使用できます。

- 直接接続(USB ケーブル(ホスト-トゥ-ホスト)、シリアルケーブル)

転送は、シリアルケーブルまたは USB ケーブルを使って、設定コンピュータと HMI デバイスを接続して行います。

---

#### 注記

シリアルケーブルを使った転送操作では、常に可能な最速の転送速度を選択します。転送速度が遅いと、大量のデータを転送するのに数時間がかかってしまうことがあります。

USB 経由の転送が失敗した場合、USB ケーブルの両端(PC と HMI)を一旦抜いてから再度挿入してください。

---

---

**注記**

Windows CE オペレーティングシステムのデバイス上で稼動する ProSave は、USB/PPI ケーブルを使用したスタンドアロンモードでの転送のみをサポートしています。

USB/PPI ケーブルは OS アップデートのみに承認されており、通信には承認されていません。USB/PPI ケーブルが承認されている HMI デバイスの詳細については、次を参照してください。

- WinCC flexible ヘルプ:  
[WinCC flexible 情報システム] > [サービスおよび開発用ユーティリティ] > [ProSave] > [デバイス依存の転送インターフェース]
  - ProSave ヘルプ:  
[ProSave] > [デバイス依存の転送インターフェース]
- 

- Ethernet ネットワーク接続

設定コンピュータと HMI デバイスは、ネットワーク内に配置されるか、ポイントツーポイント接続されます。設定コンピュータと HMI デバイス間の転送操作は、Ethernet 接続により実行されます。

- MPI/DP

設定コンピュータと HMI デバイスは、MPI ネットワークまたは PROFIBUS DP ネットワーク内に配置されます。転送操作には、対応するプロトコルが使用されます。

- Http

転送は、イントラネットやインターネット経由など、HTTP プロトコルに基づいて行われます。

下で説明している設定は、HTTP 転送の前提条件です。

設定コンピュータの"インターネット設定": 常設のプロキシサーバを HTTP 接続用に設定する必要はありません。

HMI デバイスの"インターネット設定": [Web サーバ]タブで[リモート転送を有効にする]チェックボックスを有効にする必要があります。

"エンジニアリング"の権限を持つユーザーを HMI デバイス上で設定しておく必要があります。

- ファイル

プロジェクトファイルは、指定した転送先ディレクトリに転送されます。

- S7Ethernet

設定コンピュータと HMI デバイスは、ネットワーク内に配置されるか、ポイントツーポイント接続されます。設定コンピュータと HMI デバイス間の転送操作は、Ethernet 接続により実行されます。

S7Ethernet は、ルーティングに使用できる転送チャンネルです。ルーティングは、WinCC flexible の STEP7 に統合された操作でのみ可能です。

設定コンピュータと HMI デバイスは、以下のいずれかのネットワークに接続されています。

- MPI
- PROFIBUS DP
- Ethernet

HMI デバイスがプロジェクト表示で選択されていて、[プロジェクト|転送]メニューのコマンドの 1 つが選択されている場合(バック転送操作の場合や HMI デバイス上でオペレーティングシステムを更新する場合など)、HMI デバイス用の転送モード設定も適用できます。

## OP 73micro、TP 170micro、TP 177micro、OP 73、OP 77A、および TP 177A HMI への転送

プロジェクトデータを OP 73micro、TP 170micro、TP 177micro、OP 73、OP 77A、および TP 177A HMI に転送するには、"RS 232 / PPI Multi Master Cable" (6ES7 901-3CB30-0XA0)シリアルケーブルが必要です。DIL スイッチ 5 を 0 に設定してください。

マイクロオペレーティングパネルではシリアル転送だけが可能です。他のすべての HMI は、シリアルデータ転送または MPI 転送のどちらかをサポートしています。転送設定でチャンネルの転送モードを変更して、対応する他のチャンネルを無効にします。

OP 73、OP 77A、および TP 177A HMI は、WinCC flexible の複数のアクティブなインスタンスから複数の HMI への MPI 経由の同時データ転送をサポートしていません。これらのデバイスは、最初の転送要求を実行するだけです。

## 転送保存先

Windows CE の HMI デバイスでは、フラッシュメモリや HMI デバイスの RAM に、コンパイル済みのプロジェクトファイルを保存できます。

## Windows CE デバイスでのデルタ転送

Windows CE の HMI デバイスで転送時間を省くためにできることは、デルタ(差分)転送だけです。デルタ転送の場合、HMI デバイス上のデータと相対的に変更されるプロジェクトデータだけが転送されます。

デルタ転送中に、RAM メモリにデータを転送できます。旧設定を失わずに新規設定をテストするには、この方法をお勧めします。HMI デバイスのシャットダウン/再起動後、RAM に転送された設定は失われます。このため、フラッシュメモリに保存されている設定を再度適用できます。

"デルタ転送"が、Windows CE の HMI デバイス用デフォルト設定です。転送設定時に、プロジェクト全体を転送するようにデフォルト設定を変更できます。誤動作やデルタ転送後の矛盾が原因で実行可能なプロジェクトファイルが HMI デバイスに残っていない場合など、プロジェクト全体の転送が必要なことがあります。

---

### 注記

HMI デバイスが PC の場合は、常に完全なデータファイルが転送されます。

---

## アップロード

転送時、圧縮ソースデータファイルを、コンパイルしたプロジェクトファイルと一緒に HMI デバイスに転送できます。圧縮ソースデータファイルは、プロジェクトと同じ名前で拡張子\*.pdz を付けて HMI デバイス上に保存されます。

必要に応じて、このソースデータファイルを希望の設定コンピュータにバック転送できます。このため、後ほどオリジナルの設定コンピュータ以外のコンピュータ上でオリジナルのプロジェクトを分析し、引き続き処理できます。

通知
HMI デバイス上で十分なメモリが外部で使用できる場合だけ、バック転送用に HMI デバイスにソースデータファイルを保存できます。

## ユーザー管理データとレシピの上書き

コンパイルされたプロジェクトファイルを転送すると、HMI デバイスにすでに存在するユーザー管理データとレシピは、対応する設定データで上書きされます。これにより、プロジェクトの一部としてレシピとユーザー管理データを作成し、このデータをプロジェクトが転送される HMI デバイスすべてで使用することが可能になります。転送中、圧縮されたレシピデータは HMI デバイスに転送されます。転送が終了すると、HMI デバイスでランタイムが開始され、レシピデータが解凍されます。このとき、レシピデータはプロジェクトにインポートされます。インポート後に、システムアラームが生成されます。インポートを終了する前に、レシピデータをエクスポートしないでください。インポート/エクスポートが成功したというシステムアラームが発行された後にのみ、HMI デバイスでレシピデータのエクスポートやインポートを開始できます。

該当するチェックボックスをオフにして、ユーザー管理データとレシピの上書きを防止します。データを保護するために、HMI デバイスにあるユーザー管理とレシピデータのバックアップコピーを必ず作成してください。転送が完了したら、このバックアップファイルからユーザー管理とレシピデータを復元できます。

### レシピデータの転送時は、その日付または時間の詳細が記録されます。

レシピの日付または時間の詳細を利用する場合、設定コンピュータと宛先装置で、時間とデータのシステム設定が一致する必要があります。宛先装置に転送したら、その宛先装置の日付または時間の詳細について、レシピをチェックする必要があります。

## 19.1.3 USB 経由の転送

### はじめに

USB 接続を利用すると、コンフィグレーションコンピュータと HMI デバイスを接続でき、両装置間で双方向にプロジェクトを転送できます。そのためには、適切な USB ホストツーホストケーブルと WinCC flexible CD 2 がサポートしている USB ドライバが必要です。USB インターフェース付きの Windows CE ベースの HMI デバイスで、USB を経由する場合だけ、転送できます。

---

#### 注記

WinCC flexible CD 2 がサポートする USB ドライバだけを使用してください。そうしないと、転送中に問題が発生することがあります。

---

コンフィグレーションコンピュータと HMI デバイス間でのデータ交換は、他メーカーの USB ホストツーホストケーブルを使用して正常に行われています。使用するケーブルとそのメーカーおよび供給者に関する詳細情報は、次のサイトを参照してください：  
<http://support.automation.siemens.com/WW/view/de/19142034>

## USB 経由の転送

USB ホストツーホストケーブルでのデータ転送を正常に行うには、次の条件を満たす必要があります：

- USB ホストツーホストケーブルを使用していること。
- WinCC flexible CD 2 からドライバを、既にインストールしていること。
- 使用している HMI デバイスは Windows CE ベースであり、USB インターフェースを持っていること。  
Windows CE ベースの HMI デバイスの概要は、「インターフェースのデバイスベース依存性」の章を参照してください。

条件がすべて満たされると、コンフィグレーションコンピュータを HMI デバイスとリンクでき、データ転送を開始できます。

### 19.1.4 プロジェクトのバック転送

#### はじめに

転送時、圧縮ソースデータファイルを、コンパイルしたプロジェクトファイルと一緒に HMI デバイスに転送できます。HMI デバイスからコンフィグレーションコンピュータにプロジェクトをバック転送するには、このソースデータファイルが必要になります。統合されたプロジェクトのアップロードは、サポートされていません。

#### バック転送の用途

通常、転送操作中、HMI デバイスに転送されるのは実行可能プロジェクトだけです。オリジナルのプロジェクトデータは設定デバイス上に残り、プロジェクトを将来さらに開発したり、エラーの分析に使用できます。

ただし、外部記憶媒体付きの Windows CE デバイス上及び PC 上には、コンパイルされたプロジェクトファイルだけでなく、プロジェクトの圧縮されたソースデータファイルも保存できます。このデータファイルを後ほど使用して、設定コンピュータにソースデータファイルをバック転送すれば、HMI デバイスまたはデバイスからプロジェクトを復旧できます。

#### 利点:

オリジナルの組込みデバイスが使用できない、または組込みデバイスのプロジェクト用のソースファイル(\*.hmi)が使用できない場合でも、バック転送操作によって既存プロジェクトを分析、変更することができます。

---

#### 注記

WinCC flexible を使用しても、HMI デバイスからコンフィグレーションコンピュータ上に ProTool プロジェクトのソースデータファイルをバック転送できます。その後で、ProTool プロジェクトを WinCC flexible プロジェクトに移行できます。

WinCC flexible でサポートされていないオペレーティングデバイス用に作成された ProTool プロジェクトのソースデータは、ProTool を使ってコンフィグレーションコンピュータにバック転送する必要があります。ProTool プロジェクトを保存します。それから、WinCC flexible を使って移行を実行します。

---

## バック転送の必要条件

- ソースデータファイルを HMI デバイスに転送できるのは、コンパイルしたプロジェクトファイルの転送操作の一部とだけです。各 HMI デバイスの転送設定で[バック転送を有効にする]チェックボックスを選択しておく、ソースデータファイルが、コンパイルしたプロジェクトファイルと一緒に HMI デバイスに転送されます。
- 圧縮ソースデータファイルを保存するには、HMI デバイス上に、十分な記憶容量のメモリが装備されている必要があります。バック転送操作のためのソースデータファイルを Windows CE デバイス上に供給するには、このデバイスに外部メモリカードを実装する必要があります。HMI デバイスにメモリカードが実装されていない場合、またはメモリ容量が不十分な場合には、転送が中断します。ただし、コンパイルされたプロジェクトファイルは事前にその全体が転送されるため、そのデータでランタイムを起動することができます。

オペレーティングデバイスに対してイーサネット接続が利用可能で、大きなプロジェクトのソースデータをバック転送用に保存しなければならない場合は、オペレータデバイスのメモリカード以外の保存場所としてネットワークドライブを選択することができます。これによって、保存場所の問題が解消されます。

- WinCC flexible 内でプロジェクトが開いていない場合、バック転送操作を実行する前に、[通信設定]ダイアログで、バック転送用ソースデータファイルを割付ける HMI デバイスとロード方法を選択する必要があります。

WinCC flexible でプロジェクトが開いている場合、選択した HMI デバイスからバック転送操作が実行されます。この場合、WinCC flexible の[転送設定]ダイアログで、HMI デバイスに選択した転送モードが適用されます。

## 転送とバック転送

転送操作にソースファイルが指定されている場合、プロジェクトはソースフォーマット (\*.hmi)から圧縮され、\*.pdz ファイルとして HMI デバイスの外部記憶媒体に転送されるか、PC に直接転送されます。

バック転送操作では、\*.pdz ファイルが設定コンピュータに保存されます。バック転送中に WinCC flexible でプロジェクトが開いていた場合は、そのプロジェクトを保存して閉じるように要求されます。その後で、バック転送されたプロジェクトが WinCC flexible の中で解凍されて開きます。バック転送されたプロジェクトを保存するときに、名前を付ける必要があります。

### 注意

WinCC flexible は、オペレーティングユニット上のソースデータファイルが、そのデバイス上で実行しているプロジェクトに実際に所属しているかどうかチェックできません。その間、ソースデータファイルを組み込まない転送操作を実行すると、旧プロジェクトデータがまだ HMI デバイス上に存在している可能性があります。状況によっては、現在動作中のプロジェクトにデータが一致しない状態になる可能性があります。

### 注記

転送時間を出来るだけ短くするためには、出来れば小さな及び中くらいのサイズの設定に対してバック転送プロセスを使用してください。

大型のプロジェクトファイルには、次のオプションがあります:たとえば、プロジェクトマネージャの backup ファンクションを使用して、圧縮\*.arj ファイルとしてプロジェクトファイルを CF カード上に転送します。

## 19.2 HMI デバイス上でのファイルの管理

### 19.2.1 ProSave

#### はじめに

ProSave サービスツールは、WinCC flexible に付属しています。ProSave の機能は、プログラミングデバイスの WinCC flexible ユーザーインターフェースに統合されます。ProSave は、WinCC flexible がインストールされていないコンピュータに、スタンドアローンプログラムとしてインストールすることもできます ("スタンドアローン操作")。

#### 機能範囲

ProSave には、HMI デバイスへのファイルの転送に必要な機能がすべて備わっています。

- データバックアップとバックアップデータの復元
- Windows CE ベースのデバイスのオペレーティングシステム更新
- オフソリゼーションの転送
- ドライバとオプションのインストールとアンインストール、さらに、インストール済みのオプション、および HMI デバイスにインストール可能なオプションについての説明
- 通信設定

#### 設定コンピュータでの統合された操作

ProSave は、標準 WinCC flexible インストール時に、設定コンピュータにインストールされます。ProSave の機能範囲は、メニュー[プロジェクト|転送]によって WinCC flexible 内に完全に統合されます。

#### コンピュータでのスタンドアローン操作

ProSave は、WinCC flexible CD からコンピュータにインストールしたり、また WinCC flexible をインストールせずに使用したりすることもできます(保守用など)。

たとえば、デバイスを交換する場合、WinCC flexible がインストールされていなくても、ProSave を使ってオリジナルの HMI デバイスからプロジェクトをバックアップし、交換デバイスにそれを復元することができます。

WinCC flexible 外で ProSave を使用している場合には、ユーザーインターフェース言語を変更するオプションがあります。言語を選択するときは、ProSave の[言語]メニューコマンドを使用します。[言語切り替え]を有効にするためには、ProSave を再起動する必要があります。

## 19.2.2 HMI データのバックアップ

### はじめに

HMI デバイスのデータは、定期的にバックアップすることをお勧めします。

データバックアップにより、システム障害の後やデバイスを交換したときに、素早く運転を再開できます。バックアップデータが新規の HMI へ転送されるだけで、オリジナルの状態が再現されます。

### WinCC flexible または ProSave を使ったデータバックアップ

WinCC flexible と HMI デバイスに接続するプログラミングデバイスを使用して、すべての HMI データをバックアップしたり復元することができます。

WinCC flexible がコンピュータにインストールされていない場合は、ProSave を使って集中的なバックアップを行うためのオプションを利用することができます。

### データバックアップの範囲

バックアップおよび復元操作は、HMI デバイスのタイプに依存します。以下のプロジェクトデータを含めることができます。

- 完全なバックアップ(HMI デバイスによって異なる: ランタイム、ファームウェア、オペレーティングシステムイメージ、設定、レシピ、パスワードおよび設定、ただし、オーソリゼーションはバックアップされません)
- レシピのみ
- パスワードのみ

HMI データをバックアップすると、\*.psb という拡張子の付いたバックアップファイルが生成されます。

HMI デバイスとメモリ媒体の間が適切に接続されていれば、データサーバなど、どのメモリ媒体にもバックアップできます。

---

### 注記

ただし、同じ設定ソフトウェアを使って構築したオペレーティングデバイスにあるプロジェクトデータに対してだけ、復元機能を使用してください。

たとえば、ProTool を使って構築されたデバイスに WinCC flexible レシピデータを復元する場合、フラッシュメモリデータを読み込むことはできません。この場合、フラッシュメモリは削除して、ProTool プロジェクトを再度転送します。

---

---

**注記**

Windows CE デバイスの完全データファイルバックアップおよび復元操作実行時は、以下の事項に注意します。

オーソリゼーションはバックアップされません!

完全データ復元完了時には、デバイス上に事前にあったデータは、オーソリゼーションおよびオペレーティングシステムを含めて、すべて完全に削除されます。

データ復元中に割り込みが生じた場合は、データ復元を再開する前に、まず"ブートストラップ"メカニズムを使用して HMI デバイスにオペレーションシステムを再ロードする必要があります。

関連オーソリゼーションを除いて、インストール済みのオプションもすべてバックアップされます。基本的に、"電源オフ"後も有効なオプションのデータはすべてバックアップされます。

---

**注記**

転送時間の短縮につながるので、可能であれば、USB または Ethernet を使ったデータのバックアップおよび復元をお勧めします。

---

**注記**

Windows CE デバイスからは、外部記憶媒体、すなわち CF カードや PC カードへのダイレクトデータバックアップが実行できます。詳細については、『関連操作の説明』を参照してください。

---

### 19.2.3 オペレーティングシステムの更新

#### はじめに

Windows CE デバイス上のオペレーティングシステムのバージョンが設定と互換性がない場合は、設定の転送がキャンセルされます。オペレーティングシステムを更新する必要がある場合、その旨のメッセージが表示されます。

#### オペレーティングシステムの更新

---

**注記**

オペレーティングシステムは、非 PC ベースの HMI デバイスでのみ更新できます。

---

HMI デバイスが設定コンピュータに接続されている場合は、設定コンピュータから WinCC flexible を使用してオペレーティングシステムを更新することができます。

WinCC flexible がコンピュータにインストールされていない場合は、HMI デバイスのオペレーティングシステムを更新するために、ProSave を使うオプションを利用できます。

<b>通知</b>
-----------

オペレーティングシステムの転送には、シリアル接続を使用しないようにしてください。シリアルケーブルを使用した場合、転送操作に最大で 1 時間かかることがあります。
--

オペレーティングシステムが更新されると、更新先のデバイス上のデータはオーソリゼーションを含めてすべて削除されます。したがって、"オーソリゼーション" 機能を使って、事前にオーソリゼーションをライセンスディスクに転送しておいてください。

オペレーティングシステムを更新し、内部フラッシュメモリに保存したユーザーデータ(パスワードやレシピなど)を継続して使用する場合、事前にそのデータを外部データメモリにエクスポートしておき、更新後に HMI デバイスに再ロードする必要があります。

## "ブートストラップ"

オペレーティングシステムの更新が途中で終了した場合、オペレーティングシステムはもはや HMI デバイスでは使用できません。オペレーティングシステムをロードするために利用できる唯一のオプションが、"ブートストラップ"操作です。

オペレーティングシステムの更新時、設定コンピュータと HMI デバイス間の通信は、HMI デバイスのオペレーティングシステムを使って行われます。しかし、"ブートストラップ"操作中は、設定コンピュータは HMI デバイスのブートローダーと通信します。HMI デバイスによっては、次の接続経由の通信が可能です。

- シリアル接続経由
- シリアルポートのない HMI デバイス用の Ethernet 経由

ブートストラップ操作には時間がかかります。

"ブートストラップ"操作が WinCC flexible で開始されると、HMI デバイスがブートローダーを使って通信できるよう、HMI デバイスのスイッチを切り、再度入れなおす(ブート)する必要があります。

## 19.2.4 オーソリゼーションの転送

### ライセンスキーの転送

特定の WinCC flexible ランタイムオプションをインストールするためには、対応するライセンスキーをライセンスキーの保存場所で使用できるようにする必要があります。インストール中に設定 PC を使用して、必要なライセンスキーを HMI デバイスに転送します。

ライセンスキーは、ライセンスキーの保存場所に戻すこともできます。たとえば、別の HMI デバイスでも使用できるようにするためなどです。

#### 注意

ライセンスキーが失われて復元不能にならないように、以下の場合には、HMI デバイスのライセンスキーをライセンスキーディスクに転送します。

- Windows CE HMI デバイスのオペレーティングシステムの更新前
- バックアップコピーからの完全なデータベースの復元前

HMI デバイス上のライセンスキーは、HMI デバイスの設定に応じて、バックアップされます。詳細については、該当する HMI デバイスの取扱説明書を参照してください。

## 19.2.5 オプションのインストール

### 使用可能なオプション

後で、WinCC flexible に添付されている追加オプションを HMI デバイスにインストールできます。同じように、後で、WinCC flexible とは別途購入したオプションをインストールすることもできます。

HMI デバイスのタイプによって、インストールできるオプションが決まります。

使用可能なオプションの概要については、「Introduction to WinCC flexible(WinCC flexible 概論)」を参照してください。



## STEP 7 での WinCC flexible の統合

### 20.1 基本原理

#### 20.1.1 統合されたプロジェクトの制限

##### はじめに

STEP 7 または SIMOTION での WinCC flexible の統合された操作には、以下の制限が適用されます：

- バージョン管理は使用できません。
- アップロードはできません。
- STEP 7 または SIMOTION の変更は、変更ログ中では明示的に識別されません。
- HW Config は、HMI ステーションのインポート/エクスポートをサポートしていません。対応する HMI ステーションを[コピー]と[貼り付け]で追加します。

- シンボルサーバー

STEP 7 プロジェクトを大幅に変更すると、シンボルサーバーに問題が生じる場合があります。STEP 7 プロジェクトを大幅に変更する前に、すべての WinCC flexible プロジェクトを閉じてください。

これが特に当てはまるのは以下の場合です。

- ステーション、CPU、または S7 プログラムの名前を変更する場合。
- 1 つ以上の S7 ステーションを、SIMATIC Manager の S7 プロジェクトへコピーするか、S7 プロジェクトから削除する場合。

- 統合接続のコピー

統合接続をコピーして貼り付けると、[ステーション]、[パートナ]、[ノード]列は空白のままになります。対応する値を手動で入力します。

## 20.1.2 統合されたプロジェクトの変換

### はじめに

統合されたプロジェクトでは、プロジェクトを他のバージョンの WinCC flexible に変換することも可能です。変換したプロジェクトを使用するとき、統合に含まれるプログラムのリリースされたバージョンを使用する必要があります。統合されたプロジェクトでは、以下を区別する必要があります：

- STEP7 に統合されている WinCC flexible プロジェクト  
追加の ProAgent コンポーネントあり
- SIMOTION SCOUT プロジェクトに統合された WinCC flexible プロジェクト

### STEP 7/SIMOTION SCOUT に統合された WinCC flexible プロジェクトの変換

STEP7 または SIMOTION SCOUT に統合された WinCC flexible プロジェクトの変換については、「STEP7 に統合された WinCC flexible プロジェクトの変換」の章を参照してください。

### ProAgent コンポーネントを使用したプロジェクトの変換

WinCC flexible プロジェクトを変換するとき、そこに含まれる ProAgent コンポーネントも自動的に変換されます。

以前のバージョンの WinCC flexible プロジェクトを新しいバージョンの WinCC flexible で開くと、そこに含まれている ProAgent コンポーネントを含むプロジェクトは、新しいバージョンに変換されます。

WinCC flexible プロジェクトを以前のバージョンで保存するとき、そこに含まれる ProAgent コンポーネントを含むプロジェクトは、選択したバージョンに変換され、保存されます。元のプロジェクトおよびそこに含まれる ProAgent コンポーネントに、変化はありません。

選択した WinCC flexible バージョンに適した ProAgent のバージョンは、自動的に選択されます。

最新の ProAgent バージョンの新しい機能は、以前のバージョンの ProAgent では使用できません。シーケンスを、変換後に再ビルドする必要があります。

### 20.1.3 STEP 7 への統合の基本原理

#### はじめに

SIMATIC PLC を使用してシステムに STEP 7 設定ソフトウェアをインストールしている場合は、WinCC flexible を STEP 7 に統合できます。

#### STEP 7 に統合したときの利点

統合された設定中に、STEP 7 での PLC 設定時に作成した STEP 7 設定データに、アクセスします。これには以下の利点があります。

- SIMATIC Manager は、SIMATIC PLC および WinCC flexible プロジェクトの作成、処理、および管理の、中心的な場所として使用できます。
- WinCC flexible のプロジェクトを作成すると、PLC の通信パラメータが事前に割り当てられます。STEP 7 で変更を行うと、WinCC flexible 内の通信パラメータが更新されます。

The screenshot shows the configuration window for an MP 370 12" Key PLC. The interface is organized as follows:

- MP 370 12" Key**: Hardware model.
- インターフェース**: Interface, set to IF1 B.
- Station**: Station icon.
- HMIデバイス**:
  - タイプ: TTY, R5232, R5422, R5485, Simatic (selected).
  - ポート: 187500.
  - アドレス: 5.
  - アクセスポイント: S7ONLINE.
  - バス上にマスタのみ
- ネットワーク**:
  - プロファイ: DP.
  - 最大ステーションアドレス(HSA): 126.
  - マスタの数: 1.
- PLCデバイス**:
  - アドレス: 4.
  - 拡張スロット: 3.
  - ラック: 0.
  - サイクリックオペレーション

STEP 7 への統合の際、システムによって作成された接続パラメータ: ネットワークパラメータとパートナーパラメータは事前に割り付けられます。

- タグとエリアポイントを設定すると、WinCC flexible で STEP 7 シンボルに直接アクセスできます。WinCC flexible では、タグをリンクする STEP 7 シンボルを選択するだけで済みます。STEP 7 でシンボルを変更すると WinCC flexible でも更新されます。
- STEP 7 でシンボル名を一度割り付けると、STEP 7 と WinCC flexible で使用できます。
- STEP 7 で設定された ALARM\_S と ALARM\_D のアラームは、WinCC flexible でサポートされ、HMI デバイスに出力できます。
- WinCC flexible プロジェクトは、STEP 7 に統合せずに作成し、後で STEP 7 に統合することができます。
- 統合化されたプロジェクトを STEP 7 から削除し、スタンドアロンプロジェクトとして使用できます。
- STEP 7 マルチプロジェクトでは、プロジェクト間に通信接続を設定できます。

## インストールの必要条件

WinCC flexible を STEP 7 に統合するには、特定のインストール手順に従う必要があります。まず、STEP 7 ソフトウェアをインストールしてから、次に WinCC flexible をインストールします。WinCC flexible をインストールすると、既にインストールされている STEP 7 が検出され、STEP 7 に統合するためのサポートが自動的にインストールされます。

ユーザーが手動でインストールする場合は、[STEP 7 への統合]オプションを有効にする必要があります。

WinCC flexible が既にインストールされた状態で STEP 7 をインストールする場合、STEP 7 のインストールが完了してから WinCC flexible をアンインストールして、再インストールする必要があります。

STEP 7 が既にインストールされていて、Service Pack などの更新をインストールしようとしている場合、[STEP 7 に統合]オプションを WinCC flexible に再インストールする必要があります。

WinCC flexible セットアップを、コントロールパネルを使って開き、インストールを更新します。STEP 7 の統合を無効にします。セットアップを再起動して、STEP 7 の統合を有効にします。

## 新しい STEP 7 プロジェクトへの STEP 7 シンボルの相互接続

別の WinCC flexible プロジェクトがまだ開いている間に、該当の STEP 7 プロジェクトを作成した場合、STEP 7 シンボルと WinCC flexible タグの相互接続に問題が生じます。

新しい STEP 7 プロジェクトを作成する前には、すべての WinCC flexible プロジェクトを閉じてください。

### 20.1.4 SIMATIC Manager の取り扱い

#### はじめに

STEP 7 で統合された WinCC flexible を使用する場合、WinCC flexible プロジェクトで SIMATIC Manager を使用できます。STEP 7 プロジェクトでは、SIMATIC Manager が WinCC flexible プロジェクトを含めたプロジェクト管理の中心となる場所となります。SIMATIC Manager を使用すると、自動化システムの設定、オペレータ制御とモニタレイヤーの設定にアクセスできます。

#### 必要条件

WinCC flexible が SIMATIC STEP 7 に統合されていること。

## SIMATIC Manager の操作法

統合されたプロジェクトの場合、SIMATIC Manager には以下のオプションが用意されています。

- WinCC flexible Runtime による HMI や PC ステーションの作成
- WinCC flexible オブジェクトの挿入
- WinCC flexible フォルダの作成
- WinCC flexible プロジェクトを開く
- WinCC flexible プロジェクトのコンパイルおよび転送
- 翻訳テキストのエクスポートおよびインポート
- 言語設定の指定
- WinCC flexible プロジェクトのコピーまたは移動
- STEP 7 プロジェクトのフレームワーク内での WinCC flexible プロジェクトのアーカイブおよび検索

### 20.1.5 HW Config の取り扱い

#### はじめに

パラメータをハードウェアに設定したり割り付けたりするために STEP 7 に HW Config エディタが用意されています。ドラッグアンドドロップ操作で必要なハードウェアを構成します。ハードウェアの構成要素を選択するためにカタログが用意されています。設定の際、アドレスパラメータを持つ設定テーブルが自動的に作成されます。後で STEP 7 または WinCC flexible で編集するとき、システムがこの設定テーブルにアクセスし、用意されたパラメータを受取ります。

#### HW Config の使用;HW Config ノシヨウ

HW Config を使用し、新しいステーション用ハードウェア設定を作成するか、既存のステーションに必要なモジュールを追加します。HW Config には、使用できるモジュールや事前に設定された構成要素とステーションを持つカタログが用意されています。HW Config によって挿入しようとしているオブジェクトの有用性がチェックされます。このようにして、使用できない、または不正なオブジェクトが挿入できないようになっています。HW Config で挿入されたオブジェクトのプロパティを直接編集します。関連するオブジェクトのメニューを開き、[オブジェクトプロパティ]を選択します。表示されるダイアログボックスでオブジェクトのプロパティを直接編集します。

たとえば、SIMATIC Manager で PC ステーションを作成できます。HW Config で設定用のステーションを開きます。WinCC flexible Runtime アプリケーションを挿入します。通信インターフェースを選択し、それを挿入します。HW Config で通信インターフェースの設定を編集します。WinCC flexible Runtime アプリケーションは、HW Config では開けません。このアプリケーションを開くには、SIMATIC Manager を使用します。

詳細については、HW Config のマニュアルを参照してください。

## 20.1.6 接続を設定する

### はじめに

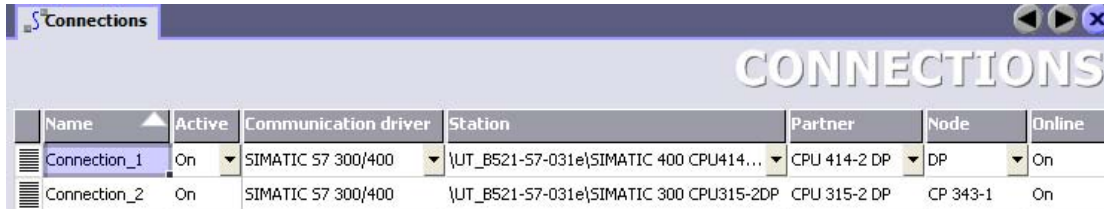
WinCC flexible と自動化レイヤー間のデータ交換には、通信を行うための接続が必要です。統合プロジェクトでは、以下のアプリケーションを使用して接続を作成できます。

- WinCC flexible;WinCC flexible
- NetPro

この設定は、WinCC flexible または NetPro のいずれかを使用して行うことができます。

### WinCC flexible を使用する場合

新しい接続を作成することも、既存の接続を編集して使用することもできます。統合プロジェクトでは、"ステーション"、"パートナー"および"ノード"列も、接続設定用のエディタで提供されます。



Name	Active	Communication driver	Station	Partner	Node	Online
Connection_1	On	SIMATIC S7 300/400	\\UT_B521-57-031e\SIMATIC 400 CPU414...	CPU 414-2 DP	DP	On
Connection_2	On	SIMATIC S7 300/400	\\UT_B521-57-031e\SIMATIC 300 CPU315-2DP	CPU 315-2 DP	CP 343-1	On

接続を作成する場合、ステーション、パートナーおよび接続ノードを選択リストから選択します。必要な接続パラメータが STEP 7 に自動的に取り入れられます。設定が完了したら、プロジェクトを保存する必要があります。WinCC flexible で設定した接続は、WinCC flexible でのみ編集が可能で、NetPro に移行させることはできません。

### NetPro を使用する場合

NetPro は、大規模なプロジェクトでの使用に推奨されます。NetPro では、図表によってサポートされたインターフェースの接続を設定します。NetPro を起動すると、STEP 7 プロジェクトのデバイスおよびサブネットが表示されます。NetPro には、その他のデバイスまたはサブネットを挿入するために使用するネットワークオブジェクトのカタログがあります。統合プロジェクトでは、このカタログに SIMATIC HMI ステーションオブジェクトも含まれています。NetPro の作業エリアにあるカタログからドラッグアンドドロップ操作でオブジェクトを挿入します。個々のステーションをドラッグアンドドロップして、サブネットに接続します。プロパティダイアログボックスを使用して、ノードおよびサブネットの接続パラメータを設定します。NetPro に設定を保存して、WinCC flexible プロジェクトにおけるデータ管理を更新します。NetPro で設定した接続設定を読み取ることができるのは、WinCC flexible だけです。また、WinCC flexible だけで、接続名を変更したり、接続用のコメントを入力したり、接続を"オンライン"に設定したりすることができます。接続の編集ができるのは、NetPro だけです。

### 注記

データ転送率などのサブネットプロパティは、サブネットのすべてのノード用に STEP 7 で自動的に設定されます。ユーザー自身でサブネットプロパティを作成または修正する場合、それらの設定がサブネットの各ノードに適用されることを確認してください。このトピックについての詳細は、NetPro のマニュアルを参照してください。

**注記**

新規 HMI ステーションを STEP 7 に設定する場合、MPI/DP ノードはシステムにより MPI および Address 1 に設定されます。HMI ステーションがネットワーク接続されていないために、HMI ステーションを異なるサブステーションタイプ経由でネットワーク接続する場合、接続パラメータを NetPro または HW 設定で変更する必要があります。

## 20.1.7 オブジェクトの取り扱い

### はじめに

以下のステップを実行し、統合化 WinCC flexible プロジェクトを作成します。

- SIMATIC Manager で新規 HMI ステーションを作成します
- WinCC flexible プロジェクトを STEP 7 に統合します。

### HMI ステーションの作成

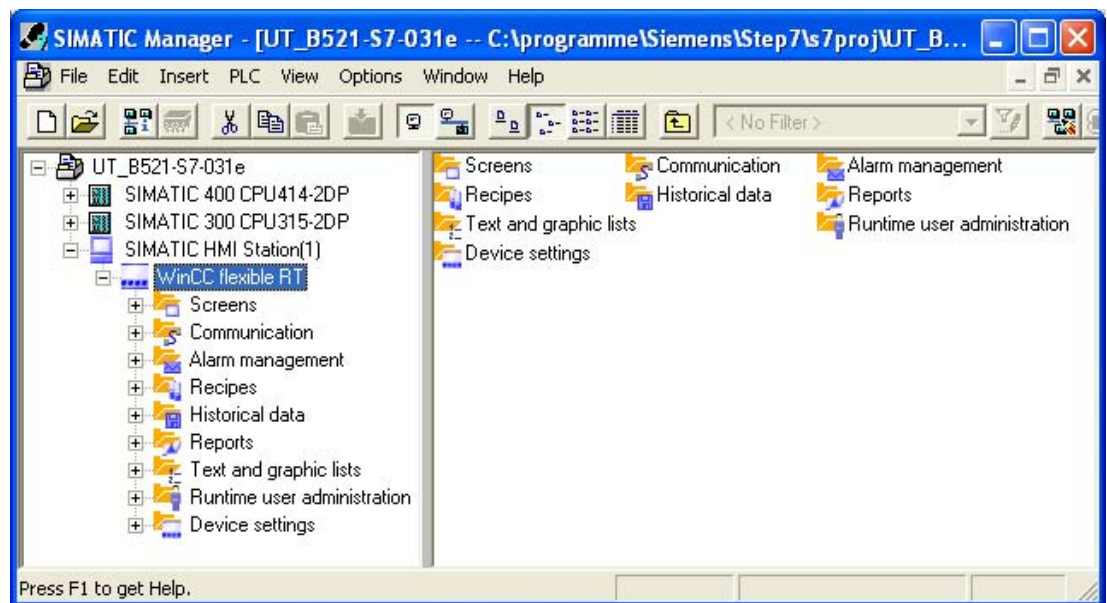
SIMATIC Manager で HMI ステーションを作成すると、基本的に WinCC flexible プロジェクトが新規に作成されます。

### 複数の HMI デバイスの WinCC flexible プロジェクトへの挿入

WinCC flexible プロジェクトで複数の HMI デバイスが必要な場合は、WinCC flexible のプロジェクトに HMI デバイスを挿入する必要があります。

### WinCC flexible オブジェクトの挿入;WinCC flexible オブジェクトノソウニユウ

WinCC flexible プロジェクトが STEP 7 に統合されると、SIMATIC Manager のプロジェクトウィンドウにプロジェクトが表示されます。WinCC flexible プロジェクトは、SIMATIC Manager のプロジェクトウィンドウに、WinCC flexible のプロジェクトウィンドウと同じように表示されます。プロジェクトウィンドウで WinCC flexible エレメントを選択すると、WinCC flexible プロジェクトのオブジェクトが作業エリアに表示されます。



ここから、既存のプロジェクトを開いたり、新規プロジェクトを作成したりできます。WinCC flexible オブジェクトを SIMATIC Manager で作成したり開いたりすると、オブジェクトを編集するために WinCC flexible が自動的に起動します。

たとえば、"画面"エレメントを選択し、WinCC flexible 画面を SIMATIC Manager で直接作成します。直ちに新規画面が作成されて開き、WinCC flexible で編集できます。または、プロジェクトのすべての変更が連続した表に記録された、変更のログを作成します。STEP 7 の変更は、変更ログ中では、明示的に識別されません。

#### 注記

統合されたプロジェクトの操作は、ログオンしている Windows ユーザーの権限によって異なります。この操作には、少なくとも書込みアクセスが必要です。Windows ユーザーが読み出しアクセスのみで統合されたプロジェクトを開いた場合、変更はできません。読み出しアクセスでは、STEP 7 と WinCC の間でのデータの同期は、得られません。

### WinCC flexible オブジェクトの再表示

SIMATIC Manager で編集することのできないグローバルプロジェクトエレメントは、表示されません。バージョン管理や言語の設定などがその例です。

WinCC flexible でテーブルエディタを使って編集するデータは、SIMATIC Manager にシンボルとして表示されます。SIMATIC Manager を介してこのようなシンボルを開くと、WinCC flexible が開き、データを編集できます。たとえば、"タグ"エレメントを選択すると、すべての WinCC flexible タグのシンボルが SIMATIC Manager の作業エリアに表示されます。個々の WinCC flexible タグは、SIMATIC Manager には表示されません。SIMATIC Manager で新規タグを作成すると、そのタグは WinCC flexible で作成され、WinCC flexible で開いて編集できます。

STEP 7 の詳細については、SIMATIC Manager のマニュアルを参照してください。

### STEP 7 データブロック名の変更

STEP 7 データブロック名を変更する場合、WinCC flexible のオブジェクトリストに古い名前が表示されることがあります。変更後の STEP 7 データブロック名がオブジェクトリストに表示されていることを確認するには、STEP 7 に統合された WinCC flexible プロジェクトで、リストを再び開きます。

### WinCC flexible プロジェクトの名前変更

SIMATIC Manager に統合された WinCC flexible プロジェクトの名前変更は、一時的なものに過ぎません。WinCC flexible でプロジェクトを開いて名前変更した後で、再び閉じる前にプロジェクトを保存しないと、変更は失われます。

## 20.1.8 統合されたプロジェクトの変換

### 20.1.8.1 STEP 7 に統合された WinCC flexible プロジェクトの変換

#### はじめに

STEP7 に統合された WinCC flexible プロジェクトは、バージョンの異なる WinCC flexible に保存できません。異なる製品バージョンにプロジェクトを保存すると、プロジェクトは変換されます。統合されていないプロジェクトと比較して、多数の特殊な機能に注意してください。WinCC flexible プロジェクトが STEP 7 に統合されている場合、WinCC flexible プロジェクトは、STEP7 プロジェクトの不可欠なコンポーネントです。WinCC flexible プロジェクトの変換は、WinCC flexible で実行されます。変換処理中は、WinCC flexible プロジェクトだけが変換され、STEP7 周辺プロジェクトには変化はありません。

#### 変換条件

STEP7 に統合されているプロジェクトを変換する場合、対応するバージョンの WinCC flexible が統合されている STEP7 を、インストールする必要があります。

例:

WinCC flexible 2005 SP1 を使用して、STEP 7 に統合されたプロジェクトを作成し、WinCC flexible 2004 SP1 バージョンでプロジェクトを保存します。

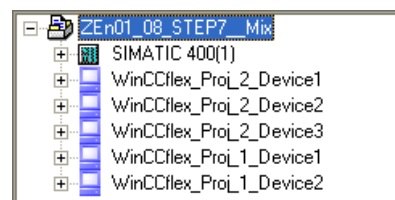
プロジェクトをこのバージョンで編集するには、STEP 7 に統合された WinCC flexible 2004 SP1 が必要です。STEP 7 のバージョンは WinCC flexible の両バージョンに共通です。

STEP 7 は、バージョン V5.3 SP2 以降で使用できます。新しいバージョンを使用する場合は、サポートしている WinCC flexible バージョンに関する情報に従ってください。

複数のバージョンの WinCC flexible を、STEP 7 のインストールに統合することは不可能です。WinCC flexible のそれぞれのバージョンに、STEP 7 をインストールした 1 台の PC が必要です。別の PC への STEP 7 プロジェクトの転送については、STEP 7 のマニュアルを参照してください。

STEP 7 プロジェクトに複数の WinCC flexible プロジェクトが含まれている場合、個々の WinCC flexible プロジェクトを、別々に変換する必要があります。WinCC flexible プロジェクトの変換時に、すべての HMI デバイスに含まれるデータが変換されます。

WinCC flexible プロジェクトに複数の HMI デバイスが含まれる場合、STEP 7 のプロジェクトウィンドウに、個別の HMI ステーションとして表示されます。



したがって、複数の HMI デバイスを含む単一の WinCC flexible プロジェクトの変換は、STEP 7 に表示された複数の HMI ステーションに影響する可能性があります。変換されたプロジェクトと共に、有効な WinCC flexible バージョンが影響を受けた各 HMI ステーションのプロパティに書き込まれます。プロパティを表示するには、HMI ステーションのショートカットメニューを開き、メニューコマンドの[プロパティ]を選択します。使用する HMI は、[デバイス]エントリの後に表示されます。HMI デバイスのバージョンは括弧の中に表示され、有効な WinCC flexible のバージョン（たとえば、"MP 370 12"キー（7.1.0.0, 2005））は、コンマの後に表示されます。WinCC flexible のバージョンは、変換されていない HMI ステーションには表示されません。

変換済みの WinCC flexible プロジェクトを現在のバージョン環境で再度開く場合は、現在のバージョンへの再変換のダイアログが開きます。複数の WinCC flexible プロジェクトを変換する場合は、必ずすべてのプロジェクトを同じバージョンに変換してください。

現在のバージョンの WinCC flexible で作成されていないプロジェクトは、STEP 7 または NetPro の HMI デバイスにコピーした後で変換する必要があります。システムによって、プロジェクトの変換を確定するように要求されます。現在のバージョンに変換されていないプロジェクトの Runtime をコピーすることはできません。Runtime は、開始したコピー操作から除外されます。したがって HMI デバイスのコピーは不完全であり、削除する必要があります。

### 変換したプロジェクトのシステム要件

変換したプロジェクトを編集するには、変換中に選択したバージョンの WinCC flexible が必要です。指定したバージョンのサービスパックを、インストールする必要があります。オペレーティングシステムに必要なサービスパックとシステム要件にも注目してください。システム要件に関する情報は、各バージョンの WinCC flexible または STEP 7 のマニュアルを参照してください。無効となった HMI デバイスに関しては、ハードウェアサポートパッケージが WinCC flexible の対応するバージョンに対して利用可能であるかを確認してください。詳細については、「新しい HMI デバイスの事後インストール」の章を参照してください。

### 20.1.8.2 統合されたプロジェクトの現在の WinnCC flexible バージョンへの変換

#### はじめに

以前のバージョンから統合された WinCC flexible プロジェクトは、新しいバージョンの WinCC flexible で開くと、自動的に変換されます。変換が開始される前に、確認のメッセージが表示されます。

#### 必要条件

- 以前のバージョンの WinCC flexible からの WinCC flexible プロジェクトが統合されている、STEP 7 プロジェクトがあること。
- STEP 7 および新しいバージョンの WinCC flexible がインストールされていること。
- STEP 7 の周辺プロジェクトが、SIMATIC Manager で開かれていること。

#### 手順

1. SIMATIC Manager のプロジェクトウィンドウで、WinCC flexible プロジェクトのプロジェクトノードを開き、[WinCC flexible RT]エントリを選択します。
2. ショートカットメニューを開き、[オブジェクトを開く]コマンドを選択します。  
WinCC flexible が始動します。  
次の変換を警告するダイアログが開きます。
3. 変換を開始するには、[OK]で警告ダイアログを確定します。  
プロジェクトは、現在のバージョンの WinnCC flexible に変換されます。
4. [キャンセル]をクリックすると、処理がキャンセルされ、プロジェクトは開きません。

#### 結果

統合された WinCC flexible プロジェクトは変換され、現在のバージョンの WinCC flexible で編集可能です。対象バージョンの WinCC flexible の機能も完全に変換され、事後設定は必要ありません。対象バージョンでサポートされていない機能は、変換後は使用できません。事後設定が必要な場合があります。

---

#### 注記

プロジェクトに含まれる HMI デバイスは、必要により、WinCC flexible のバージョンに適合する HMI デバイスバージョンに変換されます。

---

### 20.1.8.3 統合されたプロジェクトの、以前の WinCC flexible バージョンへの変換

#### はじめに

現在のバージョンの統合されている WinCC flexible プロジェクトを、以前のバージョンの WinCC flexible へ保存できます。変換が開始される前に、確認のメッセージが表示されます。

---

#### 注記

##### Windows Vista オペレーティングシステム

Windows Vista では、MSDE (Microsoft SQL Server Desktop Engine)が使用できないため、前のバージョンの WinCC flexible でのプロジェクト保存はサポートされていません。

---

#### 必要条件

- 現在のバージョンの WinCC flexible からの WinCC flexible プロジェクトが統合された STEP 7 プロジェクトがあること。
- STEP 7 および現在のバージョンの WinCC flexible がインストールされていること。
- "バージョンを付けて保存のサポート"コンポーネントが、WinCC flexible のセットアップ中にインストールされていること。
- STEP 7 の周辺プロジェクトが、SIMATIC Manager で開かれていること。

#### 手順

1. SIMATIC Manager のプロジェクトウィンドウで WinCC flexible プロジェクトのプロジェクトノードを開き、[WinCC flexible RT]エントリを選択します。
2. ショートカットメニューを開き、[オブジェクトを開く]コマンドを選択します。プロジェクトが WinCC flexible で開きます。
3. WinCC flexible で、メニューコマンドの[プロジェクト | バージョンを指定して保存]を選択します。変換に関する指示のダイアログが表示されます。
4. ダイアログでは、ドロップダウンリストボックスの[バージョンを指定して保存]で希望の WinCC flexible バージョンを選択します。
5. 変換を開始するには、ダイアログで[OK]をクリックします。プロジェクトは選択したバージョンの WinCC flexible に変換されます。プロジェクトが閉じていること。
6. [キャンセル]をクリックすると、処理がキャンセルされ、プロジェクトは変換されません。

#### 結果

統合された WinCC flexible プロジェクトに変換され、選択したバージョンの WinCC flexible で編集可能です。対象バージョンの WinCC flexible の機能も完全に変換され、事後設定は必要ありません。対象バージョンでサポートされていない機能は、変換後は使用できません。事後設定が必要な場合があります。

---

#### 注記

プロジェクトに含まれる HMI デバイスは、WinCC flexible のバージョンに適合する HMI デバイスバージョンに変換されます。HMI デバイスが以前のバージョンの WinCC flexible で使用できない場合、[デバイスの選択]ダイアログが表示されます。別の HMI を選択します。最新の WinCC flexible バージョンの新しい機能は、以前のバージョンの WinCC flexible ではサポートされません。

---

## 20.1.9 PC ステーションでの WinCC flexible の統合

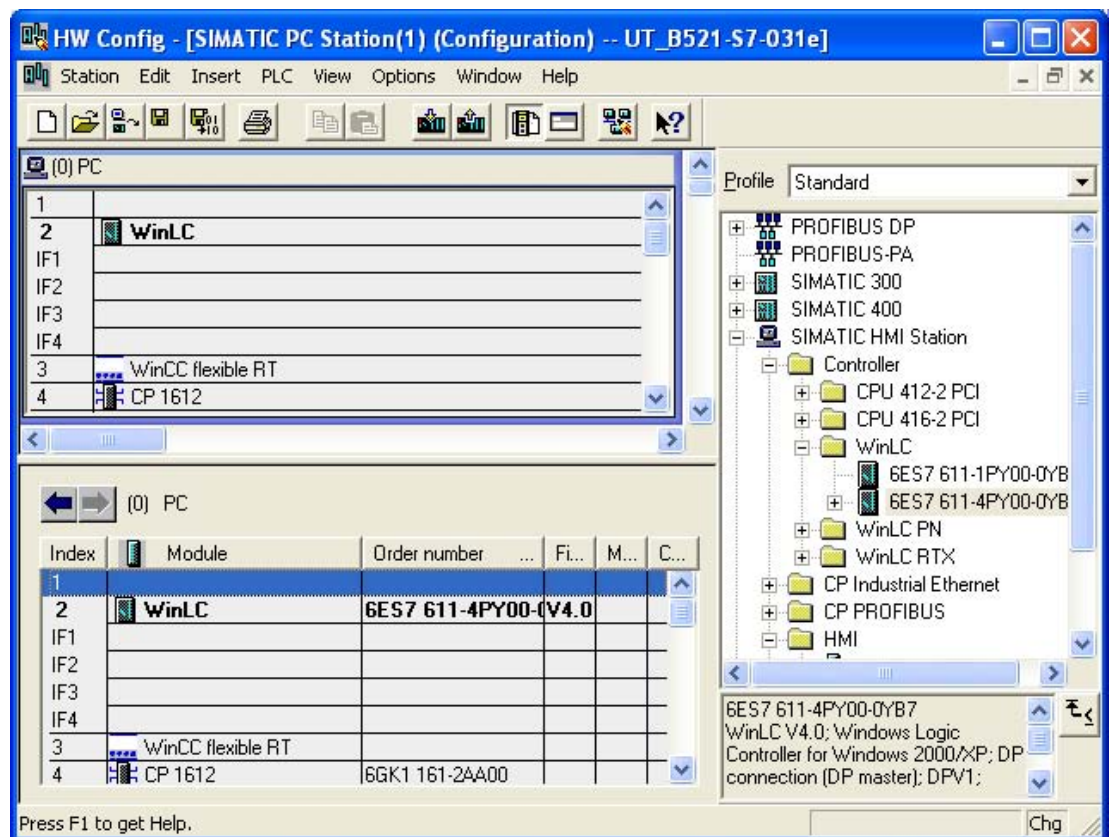
### はじめに

SIMATIC PC ステーションとは、WinCC flexible Runtime、スロット PLC、またはソフト PLC などのオートメーションタスクのための SIMATIC コンポーネントを含む、PC または OS ステーションを示します。WinCC flexible Runtime は、PC ステーションの HMI ソフトウェアとして統合され、構築されます。

STEP 7 には、PC ステーションを構築するために、"HW Config"エディタが用意されています。

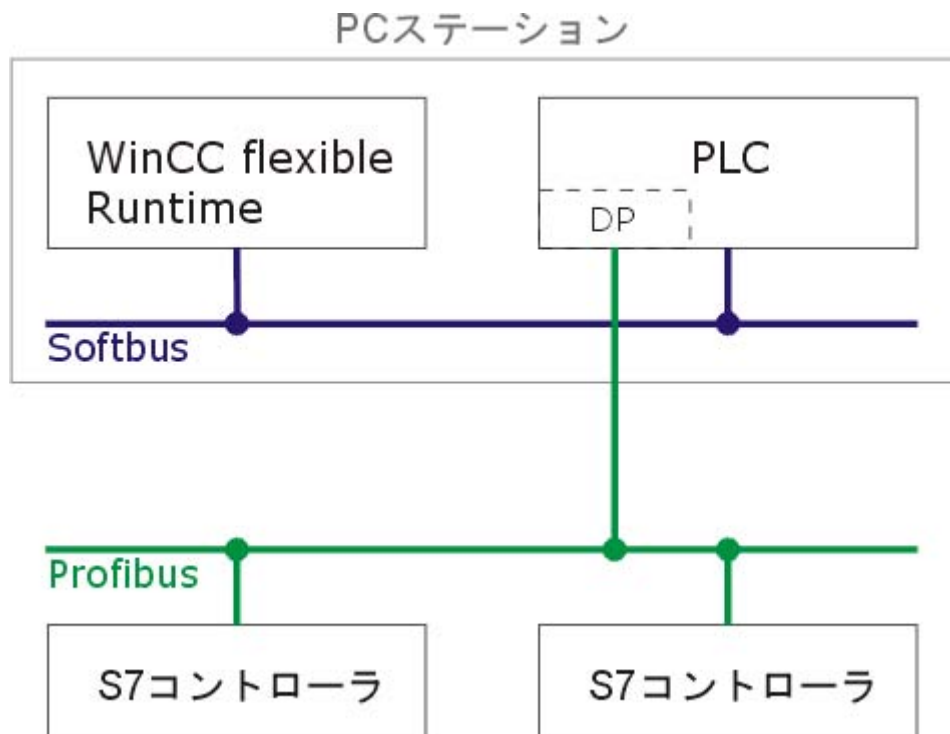
### PC ステーションの設定

WinCC flexible には、事前設定済みの PC ステーションが用意されています。新規の PC ステーションを構築するには、STEP 7 プロジェクトに SIMATIC HMI ステーションを挿入し、HMI デバイスとして [PCWinCC flexible RT] を選択します。システムによって、"PC"タイプ HMI デバイスを持つ SIMATIC HMI ステーションが作成されます。スロット PLC またはソフト PLC の追加などのような、PC ステーションの追加の構成設定は、HW Config で行います。統合されたプロジェクトでは、ドラッグアンドドロップ操作を使用して、PC 設定のすべてに必要なコンポーネントを挿入できるように、HW Config のカタログが拡張されています。



## 内部通信と外部通信

PC ステーションでは、WinCC flexible RT と含まれている PLC デバイス間の内部通信はソフトバス経由で実行されます。ソフトバスの内部接続は、WinCC flexible で構築されます。ソフトバスへの接続は自動です。接続の"ステーション"列で、PLC デバイスを選択するだけです。外部オートメーションデバイスのある通信は、PLC デバイスの統合済みインターフェース、または[HW Config]を用いて挿入した別の通信モジュールのどちらかを使って行います。



通信はステーションマネージャで管理します。ステーションマネージャで管理できるようにするには、[S7RTM]フラグが PC ステーションプロパティで設定されている必要があります。承認されたバージョンの SIMATIC Net ソフトウェアを、ターゲットステーションにインストールしておく必要があります。

このトピックについての詳細は、『SIMATIC Net マニュアル』にあります。

## 20.2 通信設定のコンフィグレーション

### 20.2.1 ルーティング経由の通信設定の構成

#### はじめに

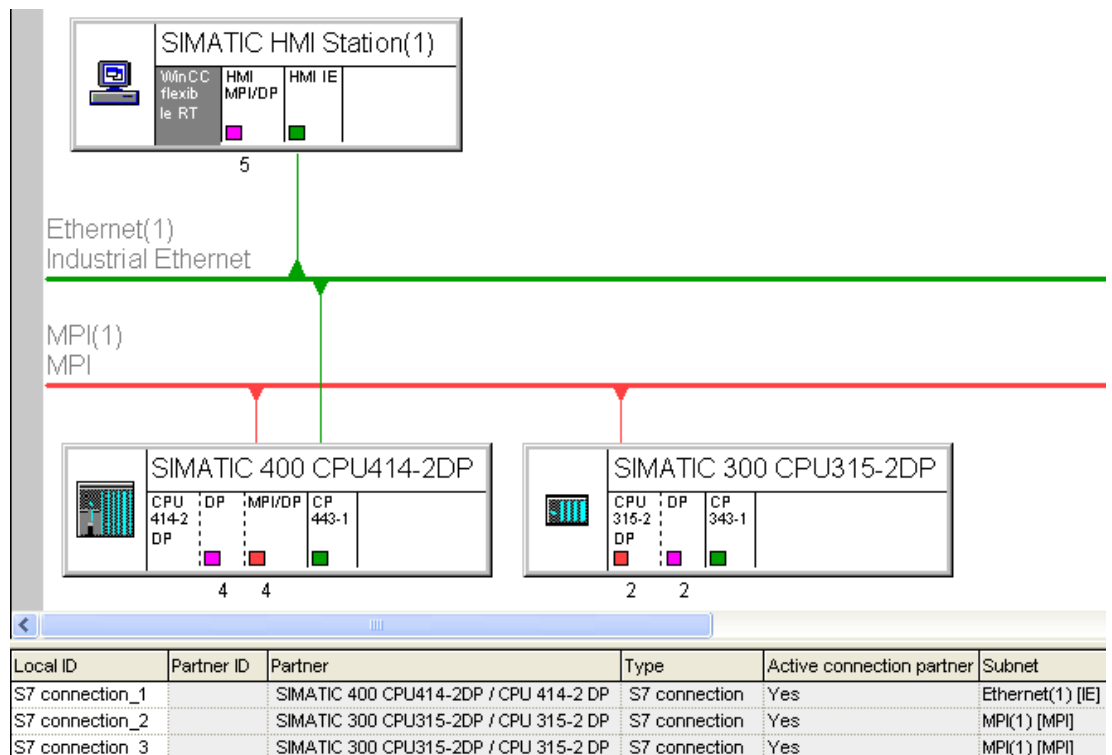
自動化システムのすべてのステーションが同じバス(サブネットワーク)に接続されているわけではない場合、これらのステーションにオンラインで直接アクセスすることはできません。これらのデバイスにコネクションを確立するには、ルーターを間に設置する必要があります。この場合、さまざまなサブネットワークに対して適切なインターフェースを持っているならば、SIMATIC ステーションがルーターとして動作できます。サブネットワーク間のゲートウェイを確立するために使用される通信能力を持つモジュール(CPU または CP)は、ルーティング能力を持つ必要があります。

ルーティングパスは、Runtime でシステムによって決定され、ユーザーは変更できません。このため、設定時に、エラー接続に関する情報を出力することはできません。

このルーティングパス内のルーティング HMI が、この接続の正しい動作を妨げる場合があります。通常、HMI デバイスはルータとして使用できません。S7-RTM フラグが設定された PC は例外です。これには SIMATIC NET ソフトウェアが必要です。

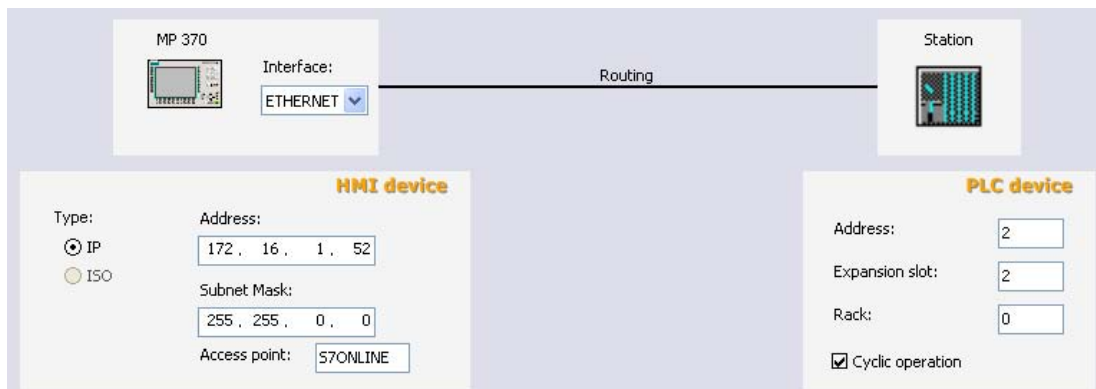
#### ルーティング接続;ルーティングセツゾク

ルーティング接続を作成するには、すべての通信パートナーが設定されて STEP 7 プロジェクトにロードされる必要があります。



ルーティング接続のあるハードウェア構成図。

上図では、ルーティング接続が SIMATIC HMI ステーション(1)と SIMATIC 300 自動化デバイス間に確立されています。SIMATIC 400 自動化デバイスはルーターとして動作しています。統合化されたプロジェクトでは、このタイプのルーティング接続を直接確立することができません。これは、SIMATIC HMI ステーションに接続をセットアップしたり、SIMATIC 300 自動化デバイスを接続パートナーとして直接選択したりして行います。ルーティング接続はシステムによって自動的に検出されます。接続は WinCC flexible の接続プロパティにルーティング接続として表示されます。



SIMATIC HMI ステーションと自動化デバイス間のルーティング接続は、統合化されたプロジェクトでのみ作成できます。

## 20.2.2 S7 ルーティング経由のプロジェクト転送

### はじめに

WinCC flexible はさまざまなサブネットワークを経由した、設定コンピュータから HMI デバイスへの WinCC flexible プロジェクトのダウンロードをサポートしています。さまざまなサブネットワーク間で接続を確立するには、ルーターを間に設置する必要があります。この場合、さまざまなサブネットワークに対して適切なインターフェースを持っているならば、SIMATIC ステーションがルーターとして動作できます。サブネットワーク間のゲートウェイを確立するために使用される通信能力を持つモジュール(CPU または CP)は、ルーティング能力を持つ必要があります。

プロジェクトを転送するには、WinCC flexible エンジニアリングステーションを、MPI バス、PROFIBUS または Ethernet に接続する必要があります。プロジェクトの転送先になる HMI デバイスも、MPI バス、PROFIBUS または Ethernet に接続する必要があります。

転送用のルーティング接続は、WinCC flexible プロジェクトの HMI デバイスと自動化デバイス間の接続設定とは無関係です。この章で説明されている接続は、ルーティング接続を経由した WinCC flexible プロジェクトから HMI デバイスへの転送の場合に限られます。

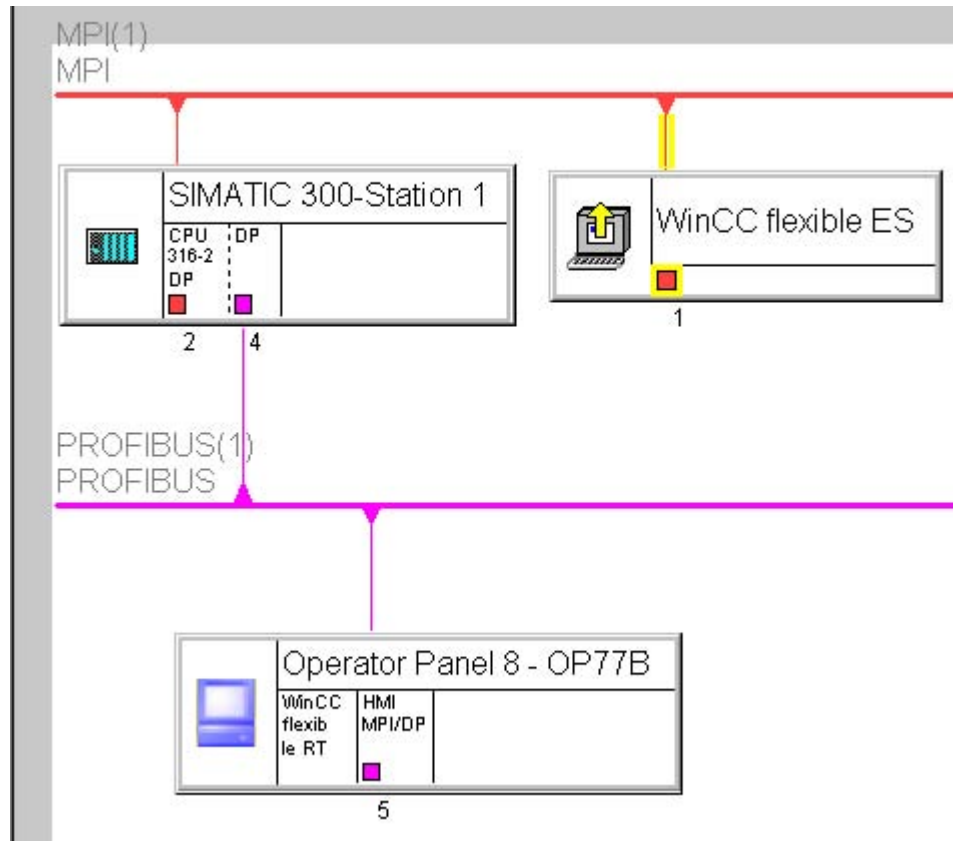
### 注記

コンポーネントをルーティングするかどうかを決定するには、各コンポーネントの技術文書を参照してください。別の方法として、NetPro または HW Config の、コンポーネントのオブジェクトプロパティを開いてください。[全般]タブに、プロパティの簡単な説明があります。

OP 73、OP 73micro、OP 77A、TP177A、および TP 177micro HMI デバイスは、プロジェクト転送の S7 ルーティングをサポートしていません。

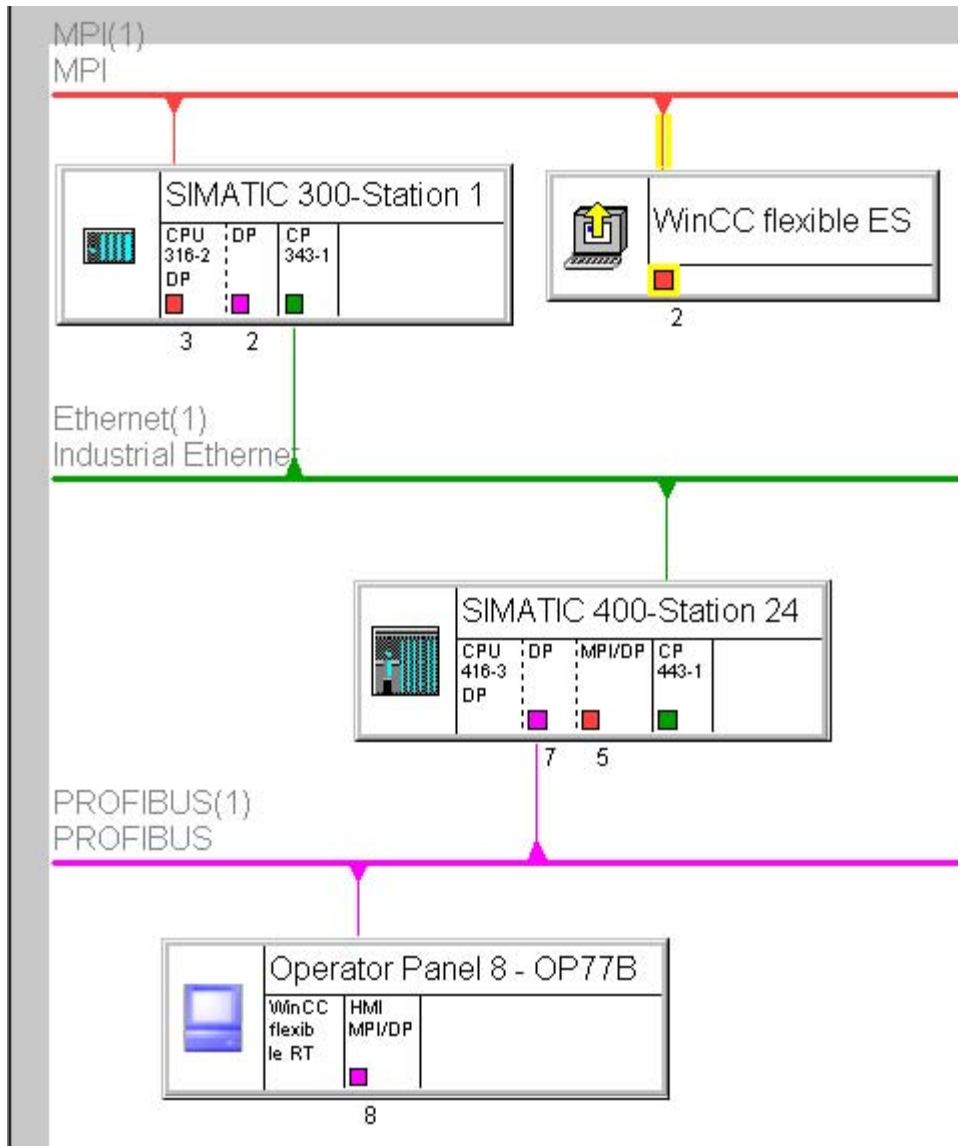
### 転送操作のルーティング接続

ルーティング接続を作成するには、すべてのステーションが設定されて STEP 7 プロジェクト内にロードされる必要があります。ルーティング接続経由でターゲットデバイスを初期化することはできません。



転送操作のルーティング接続の図

上の図では、"WinCC flexible ES" WinCC flexible エンジニアリングステーションと"オペレータパネル 8 - OP77B" HMI デバイス間のルーティング接続が確立されています。  
 "SIMATIC 300-ステーション 1"自動化デバイスは、ルーターとして機能します。関連するデバイス間の転送接続は、NetPro で設定します。設定コンピュータのインターフェースを割り付ける必要があります。関連付けは、サブネットワークへの黄色の接続線およびステーションのシンボル内の黄色の矢印によって示されます。NetPro での設定の後、プロジェクトを保存して再コンパイルします。



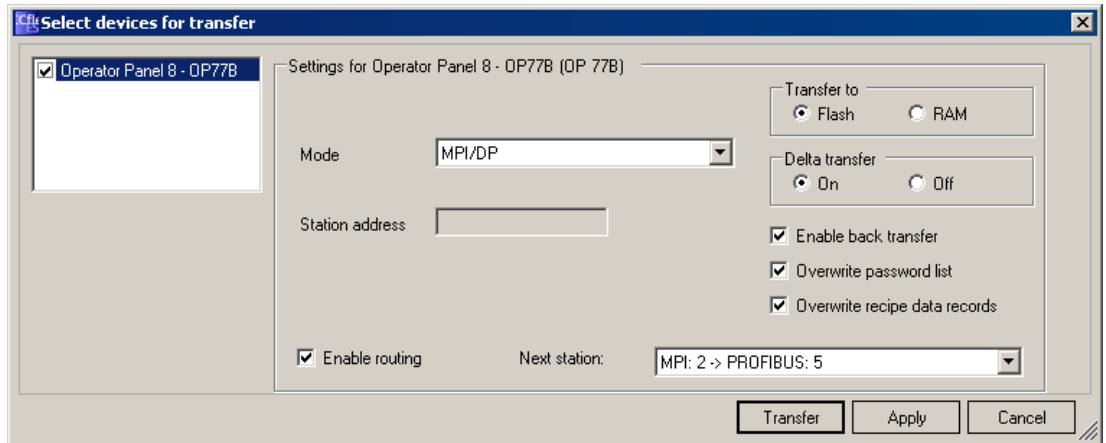
複数のステーションを経由したルーティング

複数のステーションを経由したルーティングの必要条件:

- WinCC flexible エンジニアリングステーションを、MPI バス、PROFIBUS または Ethernet に接続する必要があります。
- 転送操作の実行先となる HMI デバイスを、MPI バス、PROFIBUS または Ethernet に接続する必要があります。

## WinCC flexible の転送の開始

STEP 7 の設定が完了したら、WinCC flexible で HMI ステーションを開きます。転送をトリガするには、メニューコマンド[プロジェクト]>[転送]>[転送設定]を選択します。



### 転送設定

[モード]フィールドで、[MPI/DP]を設定する必要があります。

[ルーティングの有効化]ボックスをチェックする必要があります。

[次のステーション]フィールドには、次と最後の接続のバスタイプ、および次のルーティングパートナーとターゲットデバイスのネットワークアドレスが表示されます。利用可能な中間ルーティングパートナーは、ここには表示されません。

[転送]ボタンをクリックすると、直ちに転送が開始されます。

ルーティング設定は、[モード]でバスタイプを"S7Ethernet"の[MPI/DP]に設定した場合のみ使用可能です。ルーティングの設定が表示されない場合は、システムが永続的なルーティング接続を識別できません。関連するステーションの設定とネットワークアドレスをチェックしてください。設定されたパラメータが、システム上のステーションの設定と一致する必要があります。

転送用のルーティング接続は、統合されたプロジェクトでのみ確立することができます。

### 注記

ステーションマネージャが有効化された PC ベースの HMI デバイスへのルーティング転送はできません。

## 20.3 タグのコンフィグレーション

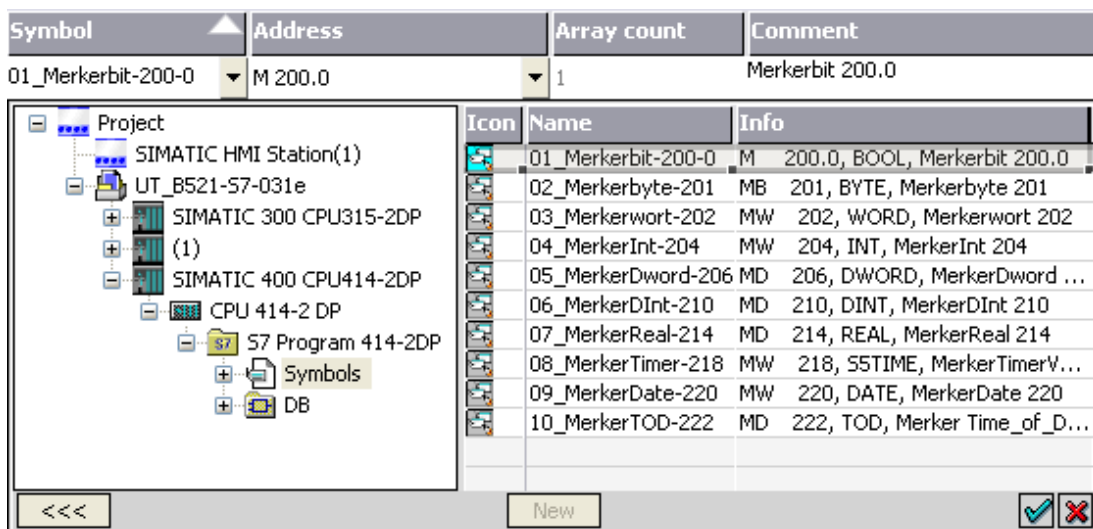
### 20.3.1 "タグ"エディタによるタグの設定

#### はじめに

編集を簡略化するために、STEP 7 では、オペランドの絶対アドレスにはシンボル名(シンボル)が付きます。これらのシンボルとその関連付けは、シンボルテーブルにリストされます。統合されたプロジェクトでは、WinCC flexible タグが STEP 7 プロジェクトからのシンボルに直接接続されます。関連するオペランドが自動的に指定されます。シンボル選択によって、データブロック(DB)内のシンボルへのダイレクトアクセスも可能になります。直接 DB にアクセスするタグを再配線するためには、[再配線]ダイアログボックスのアイコンをダブルクリックして DB を直接 STEP 7 で開きます。開いた DB のタグ接続を変更します。

#### STEP 7 からのタグの受け取り

STEP 7 からタグを受け取る時は、WinCC flexible で"タグ"エディタを開きます。統合されたプロジェクトの"タグ"エディタに、"シンボル"列が追加されます。"タグ"エディタで新規のタグを挿入します。"シンボル"列にあるフィールド上でマウスポインタをクリックし、選択ボタンを表示します。選択ボタンを押して[選択]ダイアログを開き、必要な PLC の S7 プログラムに移動します。シンボルリストまたはデータブロックから必要なシンボルを選択します。



コマンドボタンをクリックします。STEP 7 からのシンボル名はタグ名として処理されます。シンボルテーブルまたはデータブロックに関係するデータは、WinCC flexible のタグに統合されます。

STEP 7 から WinCC flexible プロジェクトに転送されたタグの名前は、一般的な STEP 7 シンボルのコンポーネントから生成されます。たとえば、タグ名"Motor.Speed"は"Motor.Speed"から生成されます。

それぞれを識別するために同じタグには、"1"から始まるインデックスが割り付けられます。タグ名の中の非サポート文字は、アンダースコア("\_")に置換されます。

## STEP 7 からの配列の転送

SIMATIC S7 300、SIMATIC S7 400、または SIMOTION PLC を使用している場合は、タグに加えて STEP 7 からすべての配列を受け取れます。

SIMATIC 300/400 コントロールプロトコルを使用していて、WinCC flexible ですべての配列を受け取る場合は、以下の手順に従います。

1. WinCC flexible で新しいタグを作成します。
2. このタグの"シンボル"列にマウスポインタを置いてクリックします。表示される選択ダイアログボックスを開くボタンを押します。
3. 必要な PLC に移動し、受け取る配列を選択します。配列エレメントの数に対応するタググループが作成されます。

## 接続の変更

ノード、プログラムまたはステーションなどを変更して、接続を変更しても、タグとシンボルの関係は失われません。タグの関連付けは、自動的に STEP 7 シンボルに再割り付けされます。

アドレスまたはシンボルが存在しないためにタグを割り付けることができない場合は、以下のオプションを利用することができます。

- 関連付けを保存  
タグは不良としてラベル付けされます。該当するタグは手動で接続する必要があります。
- タグをシンボルから分離  
タグと STEP 7 シンボルとは、手動でしか比較できなくなります。

## 20.3.2 アプリケーションポイント経由のタグの接続

### はじめに

コントロール層での WinCC flexible オブジェクトとオペランドの接続は、接続された PLC 内でシンボルを選択すると、簡単に設定できます。

### STEP 7 からのタグの受け取り

タグに接続できるすべての WinCC flexible オブジェクトは、アプリケーションポイント経由のタグの受け取りに使用できます。たとえば、IO フィールドダイナミック作成時に、IO フィールドのプロパティウィンドウでタグの選択ダイアログを開きます。必要な PLC 内の S7 プログラムへ移動します。シンボルリストまたはデータブロックから必要なシンボルを選択します。 コマンドボタンをクリックします。システムは、WinCC flexible タグを自動的に作成し、そのタグを STEP 7 の関連オペランドに接続します。

STEP 7 からのシンボル名はタグ名として処理されます。シンボルテーブルまたはデータブロックに関係するデータは、WinCC flexible のタグに統合されます。

STEP 7 から WinCC flexible プロジェクトに転送されたタグの名前は、一般的な STEP 7 シンボルの構成要素から生成されます。たとえば、タグ名"Motor.Speed"は"Motor.Speed"から生成されます。

それぞれを識別するために同じタグには、"1"から始まるインデックスが割り付けられます。タグ名の中の非サポート文字は、アンダースコア("\_")に置換されます。

## 接続の変更

ノード、プログラムまたはステーションなどを変更して、接続を変更しても、タグとシンボルの関係は失われません。タグの関連付けは、自動的に STEP 7 シンボルに再割り付けされます。

アドレスまたはシンボルが存在しないためにタグを割り付けることができない場合は、以下のオプションを利用することができます。

- 関連付けを保存  
 タグは不良としてラベル付けされます。該当するタグは手動で接続する必要があります。
- タグをシンボルから分離  
 タグと STEP 7 シンボルとは、手動でしか比較できなくなります。

## 20.4 アラームのコンフィグレーション

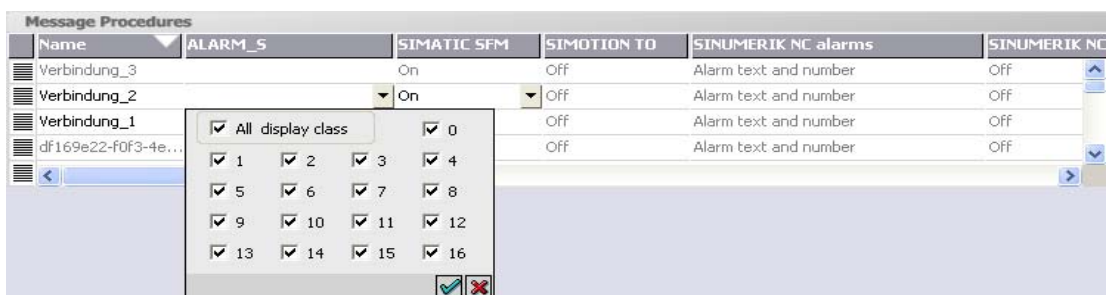
### 20.4.1 アラームナンバリング手順によるアラームの統合化

#### SIMATIC STEP 7 での設定

ALARM\_S および ALARM\_D は、アラームナンバリング手順です。アラーム番号は、STEP 7 の設定時に自動的に割り付けられます。これらの番号は、アラームメッセージを一意に割り付けるために使用されます。

STEP 7 でアラームを設定するとき、保存されたアラームと属性は STEP 7 設定データに保存されます。WinCC flexible は自動的に必要なデータをインポートし、それらを後で HMI デバイスに転送します。

表示クラスを使用して、ALARM\_S アラーム WinCC flexible の出力をフィルタリングします。プロジェクトウィンドウで[アラーム]設定]の順に選択した後、[アラーム設定]をダブルクリックします。[アラーム手順]エリアに既存の接続が表示されます。



必要な接続の行で[ALARM\_S 表示クラス]列にあるフィールドを選択し、選択ボタンを押して選択ダイアログボックスを開きます。必要な表示クラスを選択します。[OK]ボタンを押して選択ダイアログボックスを閉じます。

リンクの[SFM Alarms]列でシステムエラーを表示する必要があるか指定します。詳細については、STEP 7 のマニュアルを参照してください。

## ALARM\_S アラームの最大数

WinCC flexible における ALARM\_S アラームの最大数は 32767 です。実際には、構成可能なアラームの最大数は、HMI デバイスの利用可能なメモリーで制限されます。

## アラームクラスレイアウト

ALARM\_S と ALARM\_D の各アラームは、STEP 7 内の特定アラームクラスに割り付けられます。これらのアラームクラスの表示オプションを編集するには、WinCC flexible の[プロジェクト]ウィンドウで[アラーム|設定|アラームクラス]の順に選択します。ショートカットメニューを開き、[エディタを開く]コマンドを選択します。アラームクラス名に S7 という接頭語が付いているものが、アラームクラスです。

ALARM CLASSES							
Name	Acknowledgment	Log	I color	IO color	IA color	IOA color	
S7Alarm	On "incoming"	<No log>	<span style="color: red;">■</span>	<span style="color: orange;">■</span>	<span style="color: yellow;">■</span>	<span style="color: green;">■</span>	
S7NoAlarm	Off	<No log>	<span style="color: magenta;">■</span>	<span style="color: cyan;">■</span>	<input type="checkbox"/>	<input type="checkbox"/>	
S7OperationMessage	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7OperatorInputRequest	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7ProcessControlMaintainance	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7ProcessControlSystemMessageOs	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7ProcessControlSystemMessagePlc	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7ProcessMessageAlarm	On "incoming"	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7ProcessMessageEvent	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7StatusMessage	On "incoming"	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7Tolerance	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
S7Warning	Off	<No log>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

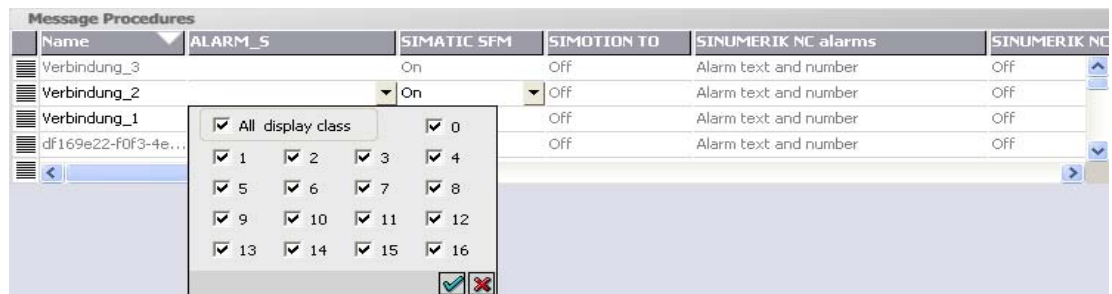
"アラームクラス"エディタを使うと、アラームクラスの表示オプションを設定できます。

### SIMOTION の Alarm\_S アラームの実装

Alarm\_S アラームも SIMOTION で使用できます。"アラーム設定"エディタを使用して、SIMOTION SCOUT で Alarm\_S アラームを設定します。

WinCC flexible は、STEP 7 の Alarm\_S アラームと同様に SIMOTION の Alarm\_S アラームを処理します。

表示クラスを使用して、ALARM\_S アラーム WinCC flexible の出力をフィルタリングします。プロジェクトウィンドウで[アラーム]設定]の順に選択した後、[アラーム設定]をダブルクリックします。[アラーム手順]エリアに既存の接続が表示されます。



SIMOTION デバイスへの接続を含む[ALARM\_S 表示クラス]列のフィールドと行を選択します。選択ボタンを押して選択ダイアログボックスを開きます。必要な表示クラスを選択します。ボタンを押して選択ダイアログボックスを閉じます。

この接続の[アラームへ]列へ行き、SIMOTION のプロセスアラームを表示するかどうかを定義します。詳細については、SIMOTION のドキュメントを参照してください。

アラームクラスの表示は、STEP 7 の Alarm\_S のアラームクラスと同様に設定されます。

## 付録

### 21.1 オープンソースソフトウェア

#### オープンソースソフトウェア

WinCC flexible には以下のオープンソースソフトウェアが含まれています。

名前	著作権 / ライセンス所有者	
Xerxes	アパッチソフトウェアライセンス、バージョン 1.1 による	<a href="http://www.apache.org">http://www.apache.org</a>

### 21.2 特徴

#### 21.2.1 一般的な技術データ

##### 21.2.1.1 対応するオペレーティングシステム

#### 対応するオペレーティングシステム

以下のオペレーティングシステムに対応する WinCC flexible がリリースされています。

- Windows XP Home SP2, Windows XP Home SP3<sup>2)</sup>
- Windows XP Professional SP2, Windows XP Professional SP3<sup>1)</sup>
- Windows Vista Business
- Windows Vista Ultimate<sup>1)</sup>

<sup>1)</sup> 多言語設定の場合は、MUI (Multilingual User Interface)バージョンのオペレーティングシステムを使用します。マイクロソフトのウェブサイト"<http://www.Microsoft.com>"を参照してください。

<sup>2)</sup> WinCC flexible Micro のみ

---

#### 注記

インストールされている Windows のバージョンを確認するには、[スタート]>[設定]>[コントロールパネル]>[システム]にある[全般]タブを参照してください。

---

### 21.2.1.2 対応するデータベース

#### 対応するデータベース

Logging with WinCC flexible Runtime は以下のデータベース用に提供されます。

- MS SQL Server: MS SQL Server 2005 Express Edition はテスト済みです
- MS Data Engine: MS Data Engine 2000 および MS Data Engine XP はテスト済みです

### 21.2.1.3 その他のソフトウェアバージョンサポート

#### 必要なその他のソフトウェアバージョン

WinCC flexible 2008 は、以下のソフトウェアで動作します。

ソフトウェア	バージョン
STEP 7	バージョン 5.4 SP3
SIMOTION SCOUT	V4.1 SP1
SIMATIC NET	V7.0 (2007)
ProTool	V6.0 SP3
WinCC	V7

### 21.2.1.4 推奨プリンタ

#### 推奨プリンタ

HMI デバイスとともに使用する推奨プリンタの現在のリストは、インターネットの以下のアドレスでご覧になれます。

[プリンタの現在のリストへのリンク](#)

<http://support.automation.siemens.com/WW/llisapi.dll?aktprim=0&lang=en&referer=%2fWW%2f&func=cslib.csinfo&siteid=csius&caller=view&extranet=standard&viewreg=WW&nodeid=10805558&objaction=csopen>

---

#### 注記

PC およびパネル PC 以外のすべての HMI デバイスは、複数のポートがある場合でも、USB ポート経由のプリンタを 1 台しかサポートしません。

---

### 21.2.1.5 有効な文字

#### はじめに

WinCC flexible は ASCII 文字セットを完全にサポートしています。ただし、各言語固有の特殊な文字の使用は控えることをお奨めします。特にスクリプトで使用されるオブジェクト名には、これらの特殊な文字を使用しないようにしてください。

#### 無効な文字

以下の文字を使用することはできません。'

### 21.2.1.6 レシピの必要メモリ

#### はじめに

以下のレシピの必要メモリの計算は、Windows CE デバイスのみに有効です。

#### 必要メモリの計算

各レシピで必要なメモリ容量(KB)は、 $D1 + D2 + D3$  の合計から算出されます。

有効なのは

- $D1 = (\text{エントリ数} \times 5 + M + 8) : 1024$

M に適用

M = すべてのタグ名を累積した長さ = エントリで使用されるすべてのタグ名の総文字数 (UTF8 コード、タグ名ごとに最高 255 バイト)

- $D2 = [(\text{データレコードの数} \times 12) + 4] : 1024$
- $D3 = [\text{データレコードの数} \times (\text{データレコードの長さ} + N) + 4] : 1024$

N に適用

すべての言語での各データレコード名の総長さ(言語当たり最高 255 バイト)+データレコード当たりのオーバーヘッド(1 バイト+言語数\*3 バイト)

D1、D2 および D3 は次に大きな数に丸められます。

### 配列を使用するための必要メモリ

各レシピの必要メモリ(KB)は、D1 + D2 + D3 の合計から算出されます。

有効なのは

- $D1 = (\text{エントリ数} \times 5 + M + 8) : 1024$   
使用されるタグ配列の各工元素は、単一のエントリとして計算されます。  
M に適用  
 $M = (\text{配列タグ名の長さ} + K) \times \text{配列工元素の数}$   
K に適用  
K = 3: その配列に 2~9 個の工元素  
K = 4: その配列に 10~99 個の工元素  
K = 5: その配列に 100~999 個の工元素  
K = 6: その配列に 1000~9999 個の工元素  
K = 7: その配列に 10000~12000 個の工元素
- $D2 = [(\text{データレコードの数} \times 12) + 4] : 1024$
- $D3 = [\text{データレコードの数} \times (\text{データレコードの長さ} + N) + 4] : 1024$   
N に適用

すべての言語での各データレコード名の総長さ(言語当たり最高 255 バイト)+データレコード当たりのオーバーヘッド(1 バイト+言語数\*3 バイト)

D1、D2 および D3 は次に大きな数に丸められます。

---

#### 注記

レシピでタグと配列の両方を使用している場合は、両方の計算式の結果を加算して必要メモリの合計を割り出します。

---

#### 21.2.1.7 特殊デバイスのレシピの必要メモリ容量

##### はじめに

以下のレシピの必要メモリの計算は、OP 77A および TP 177A デバイスのみに有効です。

##### 制限事項

この HMI デバイスには、レシピ用に 39 KB のメモリ容量があります。このメモリ容量を超えることはできません。レシピ用のメモリ容量の合計は、以下のように計算されます。すべてのレシピの合計 + 最大必要メモリのレシピ。

各レシピは、最大メモリ容量である 19 KB を超えることはできません。

## 必要メモリの計算

各レシピで必要なメモリ容量(KB)は、 $D1 + D2 + D3$  の合計から算出されます。

規則:

- $D1 = \text{データレコードの数} \times M$

M の規則(データレコードのサイズ):

$M = 1 \times 1 \text{ バイトのエLEMENTの数} + 2 \times 2 \text{ バイトのエLEMENTの数} + 4 \times 4 \text{ バイトのエLEMENTの数} + 8 \times 8 \text{ バイトのエLEMENTの数} + K$

K の規則(文字列ELEMENTのサイズ):

$K = \text{文字列ELEMENTの数} \times (\text{文字列の長さ} + 1) \times 2$

- $D2 - \text{データレコードサイズ}$

$D2 = 4 + \text{言語の数} \times 8 + \text{言語の数} \times (4 + 4 \times \text{データレコードの数} + (\text{データレコード名の長さ} + 1) \times 2 \times \text{データレコードの数}) + 8 + 8 \times \text{データレコードの数}$

または、以下のようにも書けます。

$D2 = 12 + 8 \times \text{データレコードの数} + \text{言語の数} \times (12 + \text{データレコードの数} \times (4 + (\text{データレコード名の長さ} + 1) \times 2))$

- $D3 - \text{共有メモリ}$

$D3 = 14 + \text{ELEMENTの数}$

---

## 注記

配列と単一ELEMENTは、上記のように計算できます。

---

## 21.2.2 システム制限; システムセイゲン

### 21.2.2.1 システム制限

#### はじめに

以下のシステム制限の表は、特定のプロジェクトが特定の HMI デバイスのシステム制限内にあるかどうかを判断する手助けとなります。

指定された最大値が追加されるわけではありません。つまり、これ以上オブジェクトが使用されない場合のみ、4000 アラームまで作成できます。ただし、4000 のアラームと 300 の画面があり、各画面に 40 の画面オブジェクトがある場合、これを同時に出力するのは不可能なことに注意してください。

指定した制限に加えて、設定メモリリソースの制限のために余裕を取る必要があります。

#### エンジニアリングシステム

エンジニアリングシステムの設定は、メインメモリリソースによって制限されます。WinCC flexible は、オペレーティングシステムによって、最大 2 GB のメインメモリを必要とします。

以下で示している設定は、メインメモリの負荷に大きな影響を及ぼします。

- 多数のアニメーション
- 銘板の使用
- 大きなグラフィックオブジェクトの使用
- 1 つのプロジェクト設定内に複数のデバイス

それでも、大容量のメモリを必要とするアプリケーションを同時にいくつも実行する場合には、PC に 2 GB 以上の RAM を搭載しておく価値があります。

#### 概要

##### マイクロパネル

	OP 73micro	TP 170micro	TP 177micro
<b>タグ</b>			
プロジェクト内のタグ数	500	250	250
PowerTags 数	--	--	--
配列ごとのエレメント数	50	100	100
ローカルタグ数	--	--	--
<b>アラーム</b>			
アラームクラス数	32	32	32
ディスクリートアラーム数	250	500	500
アナログアラーム数	3	--	20
アラームの文字列長	80	80	80
アラームごとのプロセス値の数	8	8	8

	OP 73micro	TP 170micro	TP 177micro
アラームバッファのサイズ	100	128	128
キューにあるアラームイベント数	30	16	32
<b>画面</b>			
画面数	250	250	250
画面ごとのフィールド数	20	20	20
画面ごとのタグ数	20	20	20
画面ごとの複合オブジェクト数	5	5	5
<b>レシピ</b>			
レシピ数	--	--	--
レシピごとのエレメント数	--	--	--
データレコードごとのバイト単位でのユーザーデータ長	--	--	--
レシピごとのデータレコード数	--	--	--
プロジェクト内のレシピエレメント数	--	--	--
内部フラッシュ内のデータレコード専用メモリ	--	--	--
<b>ログ</b>			
ログ数	--	--	--
ログファイルごとのエントリ数(すべてのログセグメントを含む)	--	--	--
ログセグメント数	--	--	--
タグロギング用のサイクリックトリガ	--	--	--
<b>トレンド</b>			
トレンド数	--	--	25
<b>テキストリストとグラフィックリスト</b>			
グラフィックリスト数	--	--	100
テキストリスト数	150	--	150
リストの合計数	150	--	150
テキストまたはグラフィックリストごとのエントリ数	30	--	30
グラフィックオブジェクト数	250	500	500
テキストエレメント数	1000	500	500
<b>スクリプト</b>			
スクリプト数	--	--	--
<b>通信</b>			
接続数	1	1	1
"SIMATIC HMI HTTP プロトコル"に基づいた接続数	--	--	--
Sm@rtClients (サービスクライアントを含む)に接続した最大数	--	--	--
<b>ヘルプシステム</b>			
ヘルプテキストの文字数	320	--	320
<b>言語</b>			

21.2 特徴

	OP 73micro	TP 170micro	TP 177micro
ランタイム言語数	5	5	5
<b>スケジューラ</b>			
タスク	--	--	--
<b>ユーザー管理</b>			
ユーザーグループ	1	1	1
オーソリゼーション	2	2	2
パスワード	1	1	1
<b>プロジェクト</b>			
プロジェクトファイル "*.fwx"のサイズ	128 KB	256 KB	256 KB

モバイルパネル

	Mobile Panel 170	Mobile Panel 177	Mobile Panel 277	Mobile Panel 277 IWLAN	Mobile Panel 277F IWLAN
<b>タグ</b>					
プロジェクト内のタグ数	1000	1000	2048	2048	2048
PowerTags 数	--	-	--	--	--
配列ごとのエレメント数	1000	1000	1000	1000	1000
ローカルタグ数	500	500	1000	1000	1000
<b>アラーム</b>					
ディスクリートアラーム数	2000	2000	4000	4000	4000
アナログアラーム数	50	50	200	200	200
アラームの文字列長	80	80	80	80	80
アラームごとのプロセス値の数	8	8	8	8	8
アラームバッファのサイズ	256	256	512	512	512
キューにあるアラームイベント数	64	64	250	250	250
<b>画面</b>					
画面数	500	500	500	500	500
画面ごとのフィールド数	50	50	200	200	200
画面ごとのタグ数	50	50	200	200	200
画面ごとの複合オブジェクト数	5	5	10	10	10
<b>レシピ</b>					
レシピ数	100	100	300	300	300
レシピごとのエレメント数	200	200	1000	1000	1000
データレコードごとのバイト単位でのユーザーデータ長	800	800	4000	4000	4000
レシピごとのデータレコード数	200	200	500	500	500
プロジェクト内のレシピエレメント数	--	--	--	--	--
内部フラッシュ内のデータレコード専用メモリ	32 KB	32 KB	64 KB	64 KB	64 KB

	Mobile Panel 170	Mobile Panel 177	Mobile Panel 277	Mobile Panel 277 IWLAN	Mobile Panel 277F IWLAN
<b>ログ</b>					
ログ数	--	--	20	20	20
ログファイルごとのエントリ数 (すべてのログセグメントを含む)	--	--	10000	10000	10000
ログセグメント数	--	--	400	400	400
タグロギング用のサイクリック トリガ	--	--	1 秒	1 秒	1 秒
<b>トレンド</b>					
トレンド数	50	50	300	300	300
<b>テキストリストとグラフィックリスト</b>					
グラフィックリスト数	100	100	400	400	400
テキストリスト数	300	300	500	500	500
リストの合計数	300	300	500	500	500
テキストまたはグラフィックリ ストごとのエントリ数	30	30	256	256	256
グラフィックオブジェクト数	1000	1000	1000	1000	1000
テキストエレメント数	2500	2500	10000	10000	10000
<b>スクリプト</b>					
スクリプト数	--	--	50	50	50
<b>通信</b>					
接続数	4	4	6	6	6
"SIMATIC HMI HTTP プロトコ ル"に基づいた接続数	--	4	8	8	8
Sm@rtClients (サービスクライ アントを含む)に接続した最大 数	--	2	2	2	2
<b>モバイルワイヤレス</b>					
ゾーン数	--	--	--	254	254
有効範囲数	--	--	--	--	127
最低 1 ゾーンまたは 1 有効範 囲に割り当てられたトランスポ ンダ数	--	--	--	255	255
設定可能なトランスポンダ数				65534	65534
<b>ヘルプシステム</b>					
ヘルプテキストの文字数	320	320	320	320	320
<b>言語</b>					
ランタイム言語数	5	5	16	16	16
<b>スケジューラ</b>					
タスク	10	10	48	48	48
<b>ユーザー管理</b>					
ユーザーグループ	50	50	50	50	50
オーソリゼーション	32	32	32	32	32

	Mobile Panel 170	Mobile Panel 177	Mobile Panel 277	Mobile Panel 277 IWLAN	Mobile Panel 277F IWLAN
パスワード	50	50	50	50	50
<b>プロジェクト</b>					
プロジェクトファイル "*.fwx" のサイズ	768 KB	2 MB	6 MB	6 MB	6 MB

## Basic Panels

	KTP400 Basic	KTP600 Basic	KTP1000 Basic	TP1500 Basic
<b>タグ</b>				
プロジェクト内のタグ数	128	128	256	256
PowerTags 数	--	--	--	--
配列ごとのエレメント数	100	100	100	100
ローカルタグ数	--	--	--	--
<b>アラーム</b>				
アラームクラス数	32	32	32	32
ディスクリートアラーム数	200	200	200	200
アナログアラーム数	15	15	15	15
アラームの文字列長	80	80	80	80
アラームごとのプロセス値の数	8	8	8	8
アラームバッファのサイズ	128	128	128	128
キューにあるアラームイベント数	64	64	64	64
<b>画面</b>				
画面数	50	50	50	50
画面ごとのフィールド数	30	30	30	30
画面ごとのタグ数	30	30	30	30
画面ごとの複合オブジェクト数	30	30	30	30
<b>レシピ</b>				
レシピ数	5	5	5	5
レシピごとのエレメント数	20	20	20	20
データレコードごとのバイト単位でのユーザーデータ長	--	--	--	--
レシピごとのデータレコード数	20	20	20	20
プロジェクト内のレシピエレメント数	--	--	--	--
内部フラッシュ内のデータレコード専用メモリ	40 KB	40 KB	40 KB	40 KB
<b>ログ</b>				
ログ数	--	--	--	--
ログファイルごとのエントリ数(すべてのログセグメントを含む) <sup>1)</sup>	--	--	--	--
ログセグメント数	--	--	--	--
タグロギング用のサイクリックトリガ	--	--	--	--

	KTP400 Basic	KTP600 Basic	KTP1000 Basic	TP1500 Basic
ログ可能なタグ数 <sup>4)</sup>	--	--	--	--
<b>トレンド</b>				
トレンド数	25	25	25	25
<b>テキストリストとグラフィックリスト</b>				
グラフィックリスト数	100	100	100	100
テキストリスト数	150	150	150	150
リストの合計数	150	150	150	150
テキストまたはグラフィックリストごとのエントリ数	30	30	30	30
グラフィックオブジェクト数	500	500	500	500
テキストエレメント数	500	500	500	500
<b>スクリプト</b>				
スクリプト数	--	--	--	--
<b>通信</b>				
接続数	4	4	4	4
"SIMATIC HMI HTTP プロトコル"に基づいた接続数	--	--	--	--
<b>ヘルプシステム</b>				
ヘルプテキストの文字数	320	320	320	320
<b>言語</b>				
ランタイム言語数	5	5	5	5
<b>スケジューラ</b>				
タスク	--	--	--	--
<b>ユーザー管理</b>				
ユーザーグループ	50	50	50	50
オーソリゼーション	32	32	32	32
パスワード	50	50	50	50
<b>プロジェクト</b>				
プロジェクトファイル "*.fwx"のサイズ	1024 KB	1024 KB	1024 KB	1024 KB

パネル

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	TP 270 OP 270	TP 277 OP 277
<b>タグ</b>									
プロジェクト内のタグ数	1000	1000	1000	500	1000	500	1000	2048	2048
PowerTags 数	--	--	--	--	--	--	-	--	--
配列ごとのエレメント数	50	100	1000	100	1000	250	1000	1000	1000
ローカルタグ数	--	--	500	250	500	--	500	1000	1000
<b>アラーム</b>									

付録

21.2 特徴

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	TP 270 OP 270	TP 277 OP 277
アラームクラス数	32	32	32	32	32	32	32	32	32
ディスクリートアラーム数	500	1000	1000	1000	2000	1000	2000	4000	4000
アナログアラーム数	3	10	50	--	50	20	50	200	200
アラームの文字列長	80	80	80	80	80	80	80	80	80
アラームごとのプロセス値の数	8	8	8	8	8	8	8	8	8
アラームバッファのサイズ	256	256	256	128	256	256	256	512	512
キューにあるアラームイベント数	16	64	64	16	64	64	64	250	250
<b>画面</b>									
画面数	500	500	500	250	500	250	500	500	500
画面ごとのフィールド数	20	30	30	20	50	30	50	200	200
画面ごとのタグ数	20	30	30	20	50	30	50	200	200
画面ごとの複合オブジェクト数	5	5	5	5	5	5	5	10	10
<b>レシビ</b>									
レシビ数	--	10	100	--	100	10	100	300	300
レシビごとのエレメント数	--	50	200	--	200	50	200	1000	1000
データレコードごとのバイト単位でのユーザーデータ長	--	800	800	--	800	800	800	4000	4000
レシビごとのデータレコード数	--	20	200	--	200	20	200	500	500
プロジェクト内のレシビエレメント数	--	--	--	--	--	--	--	--	--
内部フラッシュ内のデータレコード専用メモリ	--	40 KB	32 KB	--	32 KB	40 KB	32 KB	64 KB	64 KB
<b>ログ</b>									
ログ数	--	--	--	--	--	--	--	20	20
ログファイルごとのエントリ数(すべてのログセグメントを含む)	--	--	--	--	--	--	--	10000	10000
ログセグメント数	--	--	--	--	--	--	--	400	400
タグロギング用のサイクリックトリガ	--	--	--	--	--	--	--	1 秒	1 秒
ログ可能なタグ数 <sup>4)</sup>	--	--	--	--	--	--	--	20	--
<b>トレンド</b>									
トレンド数	--	--	--	--	50	25	50	300	300
<b>テキストリストとグラフィックリスト</b>									

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	TP 270 OP 270	TP 277 OP 277
グラフィックリスト数	--	--	--	--	100	--	100	400	400
テキストリスト数	150	300	300	--	300	300	300	500	500
リストの合計数	150	300	300	--	300	300	300	500	500
テキストまたはグラフィックリストごとのエントリ数	30	30	30	--	30	30	30	256	256
グラフィックオブジェクト数	500	1000	1000	1000	1000	1000	1000	1000	1000
テキストエレメント数	2500	2500	2500	1000	2500	1000	2500	10000	10000
<b>スクリプト</b>									
スクリプト数	--	--	--	--	--	--	--	50	50
<b>通信</b>									
接続数	2	4	4	4	4	4	4	6	6
"SIMATIC HMI HTTP プロトコル" に基づいた接続数	--	--	--	--	--	--	4	8	8
Sm@rtClients (サービスクライアントを含む) に接続した最大数	--	--	--	--	--	--	2	6": 3 10": 2	6": 3
<b>ヘルプシステム</b>									
ヘルプテキストの文字数	320	320	320	--	320	320	320	320	320
<b>言語</b>									
ランタイム言語数	5	5	5	5	5	5	5	5	16
<b>スケジューラ</b>									
タスク	--	--	10	--	10	--	10	48	48
<b>ユーザー管理</b>									
ユーザーグループ	25	50	50	1	50	50	50	50	50
オーソリゼーション	32	32	32	2	32	32	32	32	32
パスワード	25	50	50	1	50	50	50	50	50
<b>プロジェクト</b>									
プロジェクトファイル "*.fwx" のサイズ	256 KB	256 KB	1 MB	320KB	768 KB	512 KB	2 MB	2 MB	4 MB

マルチパネル

	MP 177	MP 270B	MP 277	MP 370	MP 377
<b>タグ</b>					
プロジェクト内のタグ数		2048	2048	2048	2048
PowerTags 数		--	--	--	--

	MP 177	MP 270B	MP 277	MP 370	MP 377
配列ごとのエレメント数		1000	1000	1000	1000
ローカルタグ数		1000	1000	2000	2000
<b>アラーム</b>					
アラームクラス数		32	32	32	32
ディスクリートアラーム数		4000	4000	4000	4000
アナログアラーム数		200	200	200	200
アラームの文字列長		80	80	80	80
アラームごとのプロセス値の数		8	8	8	8
アラームバッファのサイズ		512	512	1024	1024
キューにあるアラームイベント数		250	250	500	500
<b>画面</b>					
画面数		500	500	500	500
画面ごとのフィールド数		200	200	400	400
画面ごとのタグ数		200	200	400	400
画面ごとの複合オブジェクト数		10	10	20	20
<b>レシピ</b>					
レシピ数		300	300	500	500
レシピごとのエレメント数		1000	1000	1000	1000
データレコードごとのバイト単位でのユーザーデータ長		4000	4000	4000	4000
レシピごとのデータレコード数		500	500	1000	1000
プロジェクト内のレシピエレメント数		--	--	--	--
内部フラッシュ内のデータレコード専用メモリ		64 KB	64 KB	128 KB	128 KB
<b>ログ</b>					
ログ数		20	20	50	50
ログファイルごとのエントリ数(すべてのログセグメントを含む) <sup>1)</sup>		10000	10000	50000	50000
ログセグメント数		400	400	400	400
タグロギング用のサイクリックトリガ		1 秒	1 秒	1 秒	1 秒
ログ可能なタグ数 <sup>4)</sup>		20	--	50	--
<b>トレンド</b>					
トレンド数		300	300	400	400
<b>テキストリストとグラフィックリスト</b>					
グラフィックリスト数		400	400	500	500
テキストリスト数		500	500	500	500
リストの合計数		500	500	500	500
テキストまたはグラフィックリストごとのエントリ数		256	256	256	256
グラフィックオブジェクト数		1000	1000	2000	2000
テキストエレメント数		10000	10000	30000	30000

	MP 177	MP 270B	MP 277	MP 370	MP 377
<b>スクリプト</b>					
スクリプト数		50	50	100	100
<b>通信</b>					
接続数		6	6	6	6
"SIMATIC HMI HTTP プロトコル" に基づいた接続数		8	8	8	8
Sm@rtClients (サービスクライアントを含む)に接続した最大数		6": 最大 3 10": 最大 2	8": 最大 3 10": 最大 2	12": 最大 3 15": 最大 2	12": 最大 3 15": 最大 2 19": 最大値: 1
<b>ヘルプシステム</b>					
ヘルプテキストの文字数		320	320	320	320
<b>言語</b>					
ランタイム言語数		5	16	5	16
<b>スケジューラ</b>					
タスク		48	48	48	48
<b>ユーザー管理</b>					
ユーザーグループ		50	50	50	50
オーソリゼーション		32	32	32	32
パスワード		50	50	50	50
<b>プロジェクト</b>					
プロジェクトファイル "*.fwx"のサイズ		4 MB	6 MB	7 MB	12 MB

1) すべてのセグメントのサイクリックログのエントリ数は、"セグメントサークリックログ"のロギング方法に適用されます。サイクリックログ回数と、このログのデータレコード数を乗算した値を超えてはなりません。

#### WinCC flexible Runtime

	WinCC flexible Runtime
<b>タグ</b>	
プロジェクト内のタグ数	2048
PowerTags 数	128 -2048
配列ごとのエレメント数	1600
ローカルタグ数	2000
<b>アラーム</b>	
アラームクラス数	32
ディスクリートアラーム数	4000
アナログアラーム数	500
アラーム長	80
アラームごとのプロセス値の数	8
アラームバッファのサイズ	1024
キューにあるアラームイベント数	500
<b>画面</b>	

	WinCC flexible Runtime
画面数	500
画面ごとのフィールド数	400
画面ごとのタグ数	400
画面ごとの複合オブジェクト数	40
<b>レシピ</b>	
レシピ数	999
レシピごとのエレメント数	2000
データレコードごとのバイト単位でのユーザーデータ長	8000
レシピごとのデータレコード数	5000
プロジェクト内のレシピエレメント数	--
内部フラッシュ内のデータレコード専用メモリ	--
<b>ログ</b>	
ログ数	100
ログファイルごとのエントリ数(すべてのログセグメントを含む) <sup>1)</sup>	500000
ログセグメント数	400
タグロギング用のサイクリックトリガ	1 秒
ログ可能なタグ数 <sup>4)</sup>	100
<b>トレンド</b>	
トレンド数	800
<b>テキストリストとグラフィックリスト</b>	
グラフィックリスト数	500
テキストリスト数	500
リストの合計数	500
テキストまたはグラフィックリストごとのエントリ数	3500
グラフィックオブジェクト数	2000
テキストエレメント数	30000
<b>スクリプト</b>	
スクリプト数	200
<b>通信</b>	
接続数	8
"SIMATIC HMI HTTP プロトコル"に基づいた接続数	16
Sm@rtClients (サービスクライアントを含む)に接続した最大数	5 <sup>3)</sup>
<b>ヘルプシステム</b>	
ヘルプテキストの文字数	320
<b>言語</b>	
ランタイム言語数	16
<b>スケジューラ</b>	
タスク <sup>2)</sup>	48
<b>ユーザー管理</b>	
ユーザーグループ	50
オーソリゼーション	32

WinCC flexible Runtime	
パスワード	100
<b>プロジェクト</b>	
プロジェクトファイル "*.fwx"のサイズ	

- 1) すべてのセグメントのサイクリックログのエントリ数は、"セグメントサークリックログ"のロギング方法に適用されます。サイクリックログ回数と、このログのデータレコード数を乗算した値を超えてはなりません。
- 2) 時刻でトリガされるタスクにのみ有効です。イベントでトリガされるタスクシステム制限に関連しません。
- 3) 最大3つの Sm@rtClient まで、Panel PC 477 上の Sm@rtServer と相互接続できます。
- 4) CE デバイス(TP 270、OP 270、MP 270B、MP 370)の情報：  
HMI デバイスは主にサイクリックデータロギングのために設計されているというよりも、非サイクリックエラーステータスをロギングするために使用されます(エラーやオペレーティングアラームなど)。HMI デバイスは、大量のデータや高性能の要件を伴う複雑なロギングタスク向きには設計されていません。



# 索引

"

"グラフィック"エディタ, 362  
"データログ"エディタ, 274  
"ランタイムセキュリティ設定"エディタ, 307

\*

\*.pwx, 122, 412

[

[アラーム]ウィンドウ, 192  
    フィルタリング, 193  
[アラーム]画面, 192  
[スクリプトウィザード], 325  
[プロジェクト]ウィンドウ  
    HMIデバイス依存データ, 93  
[プロパティ]ウィンドウ  
    タグ, 132  
[ユーザー]ウィンドウ, 169, 311, 312  
[レシピ]ウィンドウ, 251  
    オペレーティングエレメント, 259  
    キーボードによる操作, 259  
    ドロップダウンリスト形式での使用, 254  
    ファンクションキーを使用した操作, 255  
    マウスによる操作, 259  
    レシピデータレコード, 234, 235, 255  
    動作, 255  
    更新, 252  
    画面を変更したときの動作, 255  
    簡易, 251  
    表示された値のみ, 252  
    設定可能なイベント, 254  
    詳細, 251  
[有効範囲]エディタ, 382

A

ALARM\_D アラーム  
    STEP 7 での設定, 207, 446  
    STEP 7 での設定, 207, 446

ALARM\_S アラーム

    SIMOTIONでの設定, 209, 448  
    STEP 7 での設定, 207, 446  
    STEP 7 での設定, 207, 446  
    表示のフィルタリング, 207, 446  
    表示のフィルタリング, 207, 209, 446, 448

C

CSV エクスポート  
    タグ, 151  
csv ファイル  
    例, 216, 279  
    レイアウト, 216, 279  
CSV ファイル  
    アドレス多重化を使用したタグ, 158  
    プロジェクトテキスト, 356

E

Ethernet, 226

H

HMI システム  
    タスク, 19  
HMI ステーション  
    作成, 431  
    作成, 431  
HMI デバイス  
    システム制限, 454  
    選択, 90  
    データの復元, 420  
    データバックアップ, 420  
    バージョン, 123, 413  
    パーソナル, 34  
    パフォーマンス特性, 454  
    複数のHMIデバイスの設定, 89  
    複数の挿入, 431  
    リモートアクセス(コンセプト), 31  
    同期, 33  
    複数のHMIデバイスのプロジェクト, 93  
    複数のHMIデバイス用のプロジェクトの使用, 95  
    複数の挿入, 431

- HMI デバイスの依存性
  - 原理, 64
  - スクリプトでの, 333
- HMIデバイスの依存性, 90
- HMIデバイスバージョン, 92, 95
- HMI の交換
  - DP設定, 92
- HTMLブラウザ, 169
- HW Config の使用
  - 統合されたプロジェクトで, 429
- I**
- IntelliSense, 326
- IOフィールド, 169, 289
- N**
- NetPro
  - 接続の編集, 430
- O**
- OLE オブジェクト
  - ファイルからの作成, 174
  - 再作成, 174
- OP 77A と TP 177A
  - レシピの相違点, 241
- P**
- PageDown, 452
- PC ステーション
  - 外部通信, 438
  - 内部通信, 438
  - 設定, 437
- PROFISafe, 375
- ProSave, 419
  - 言語切り替え, 419
- ProTool プロジェクト
  - 移行, 107
- S**
- S7Ethernet
  - 転送, 414
- SIMATIC HMI
  - WinCC flexible, 20
  - タスク, 19
  - はじめに, 19
  - 定義, 19
- SIMATIC Manager
  - WinCC flexibleオブジェクトの編集, 431
  - 操作法, 429
- SIMATIC STEP 7, 96
- SIMATIC STEP 7 との統合, 41
- SIMATICログオン, 309
- SIMOTION
  - Alarm\_Sアラーム, 209, 448
  - プロセスアラーム, 209, 448
- SIMOTION SCOUT, 96
- SIMOTION SCOUTとの統合, 41
- SmartClientの表示, 169
- STEP 7
  - WinCC flexibleでのタグの処理, 444, 445
  - WinCC flexibleでのタグの有効化, 444
  - WinCC flexibleへの配列転送, 445
  - WinCC flexibleへの配列転送, 445
- STEP 7 からの配列転送, 445
- STEP 7 からの配列転送, 445
- STEP 7 の統合, 427
- STEP 7 への統合
  - 利点, 427
  - 必要条件, 427
- T**
- TIA, 41
- U**
- USB
  - 転送, 417
  - 転送, 413
- USB経由の転送, 413
- V**
- VBS
  - オブジェクトプロパティの変更, 344
  - ヘルプファンクション, 329
- W**
- WinCC
  - オプション, 20
  - 自動化コンセプト, 27
- WinCC flexible, 20, 44
  - エディション, 62
  - エンジニアリングサポート, 35
  - エンジニアリングシステム, 22

- クイックスタート, 85
  - 個々の設定, 84
  - 作業, 62
  - 接続の編集, 430
  - ユーザーインターフェースのカスタマイズ, 37
  - ライセンスがない場合, 26
  - ランタイムソフトウェア, 23
  - 一貫した設定, 21
  - 用途, 20
  - 複数言語ユーザーインターフェース, 98
  - 起動, 85
  - 開始, 85
  - WinCC flexible, 44
  - WinCC flexible内の文書, 120
  - WinCC flexible の開始
    - 必要な書込み許可, 86
  - WinCC プロジェクト
    - 移行, 107
  - WLAN, 375
  - WLANエリア, 377
- X**
- XLS ファイル
    - プロジェクトテキスト, 356
- A**
- アクション
    - [オブジェクト]ウィンドウでの, 106
  - アクセス
    - Runtimeオブジェクトモデル, 334
    - スクリプト内のスクリプト, 332
    - スクリプトのシステムファンクション, 332
    - タグ, 331
  - アジア言語
    - 設定, 352
  - アジア言語オペレーティングシステム, 349
  - アジア言語文字
    - HMIデバイスでの入力, 364
    - メモリ要件, 364
    - 解釈, 364
  - アップロード
    - HMIデバイスから, 124, 417
    - プロジェクトファイル, 415
    - 統合されたプロジェクト, 124, 417, 425
  - アドレス指定
    - タグ、間接アドレス指定, 140
    - 多重化, 140
  - アドレス多重化を使用したタグ
    - インポート, 158
  - アナログアラーム
    - アナログアラームエディタ, 202
    - アナログアラームエディタ, 202
    - アナログアラーム手順, 188
    - アプリケーション
      - プロジェクト文書, 367
      - [レシビ]ウィンドウ, 258
      - レポートオブジェクト, 292
    - アプリケーションのフィールド, 387
      - ジョブの計画, 387
    - アプリケーションのフィールド, 375, 393
      - プロジェクトバージョンの管理, 393
      - ユーザー管理, 301
      - 変更のロギング, 403
    - アラーム
      - HMIデバイス上の表示, 192
      - PLCによる確認, 198
      - イベント, 196
      - 印刷, 194
      - エディタ, 199
      - 基本設定, 206
      - コンポーネント, 197
      - システムアラーム, 191
      - システムファンクション, 195
      - 出力, 215
      - プロパティ, 197
      - レポート機能, 194, 207, 292
      - ロギング, 194, 215
      - 確認, 189
      - アラームインジケータ, 169, 192
      - [アラーム]画面, 169
      - アラームクラス, 190, 210
        - アラームクラスエディタ, 204
      - アラームクラスエディタ, 204
      - アラームグループ
        - アラームグループエディタ, 205
      - アラームグループエディタ, 205
      - アラームステータス, 188
        - レイアウト, 190
      - アラームテキスト, 197
      - アラーム手順, 188
      - アラーム配信
        - 自動, 32
        - 電子メールを介して, 32
      - アラーム番号, 197
      - アラーム番号手順, 188
      - アラームレポート
        - コンフィグレーション, 293
      - アラームロギング, 210
        - ログの動作, 211
        - 記憶媒体, 211
      - アラームログ
        - アラームの表示, 215
        - 基本原理, 210

"アラームログ"エディタ, 212  
 アラーム行, 192  
 アンインストール  
   ユーザー辞書, 358

## イ

イーサネット, 229  
 移行, 63  
   原理, 63  
 移動バス, 39  
 イベント, 388, 389  
   時間ベースのイベント, 389  
   設定, 254  
 イメージ  
   エディタの構造, 362  
 イメージバージョン, 92, 95, 123, 413  
 印刷アラーム  
   出力パラメータのコンフィグレーション, 293  
 印刷レシピ  
   出力パラメータのコンフィグレーション, 297  
 インスタンス  
   再使用, 185  
 インストール  
   Audit Viewer, 24  
   オプション, 423  
 インデックスタグ, 140  
 インポート, 311  
   プロジェクトテキスト, 356  
   レシピデータレコード, 238

## ウ

ウィンドウ, 58  
   折りたたみ, 59  
   組み合わせ, 58  
   ドッキング, 58

## エ

エクスポート, 311  
   タグ, 151  
   プロジェクトテキスト, 356  
   レシピデータレコード, 238  
 エディタ, 68  
   エディタの簡単な説明, 100  
   可能なWinCC flexibleエディタ, 89  
   グラフィカルエディタ, 65, 101  
   クロスリファレンスリスト, 118  
   システム辞書, 359  
   データログ, 274

テーブル形式エディタ, 199  
 テーブル形式エディタ, 65, 100  
 閉じる, 69  
 開く, 66  
 プロジェクト文書, 120  
 プロパティ, 65  
 画面, 100  
 言語依存のオブジェクトを用いて, 353

エディタ, 65  
 エラー  
   ランタイムエラー, 336  
   論理エラー, 336  
 エリアポインタ  
   データレコード, 243  
   接続エディタ, 225  
 エレメントリスト, 251  
 円, 169  
 エンジニアリングサポート, 35  
   概要, 35

## オ

オートメーション  
   パーソナルHMIデバイス, 34  
   リモートアクセス, 31  
 オブジェクト, 289  
   ActiveXコントロールの追加, 174  
   VBSでのプロパティの変更, 344  
   アクセス, 334  
   オブジェクトの位置変更, 173  
   サイズ変更, 173  
   スクリプト内の同期化, 335  
   タブ順序, 174  
   デフォルトのプロパティのカスタマイズ, 174  
   ファイルからのOLEオブジェクト作成, 174  
   プロジェクトレポート用の選択, 374  
   参照, 334  
   反転, 174  
   同じタイプの挿入, 174  
   回転, 173  
   挿入, 113, 173  
   整列, 173  
   新しいOLEオブジェクトの作成, 174  
   置換, 114  
   複数のオブジェクトの位置変更とサイズ変更, 174  
   複数のオブジェクトの選択, 174  
 [オブジェクト]ウィンドウ  
   アクション, 106  
   プロパティ, 55  
 オブジェクトグループ  
   プロパティ, 178  
 オブジェクトの編集

- SIMATIC Managerで, 431
- SIMATIC Managerでの, 431
- オブジェクトの選択
  - 作成データの出力, 373
- オブジェクトリスト
  - 作業, 70
  - 開く, 70
- オプション, 24
  - インストール, 423
  - ライセンス, 25
- オペレータ制御エレメント
  - エディタ固有の, 56
  - エディタ固有の配置, 56
  - ツールバーの, 57
  - フレームの, 57
- オペレータ制御
  - 簡易[レシピ]ウィンドウ, 260
- オペレーティングエレメント
  - [レシピ]ウィンドウ, 259
- オペレーティングシステム
  - HMIデバイス上での更新, 421
  - アジア言語設定, 349
  - ヨーロッパ言語設定, 349
- 折りたたみ, 59
  - ウィンドウ, 59
- オンライン
  - レシピタグ, 240
- オンラインヘルプ, 84
  - 表示, 84

## カ

- 開始, 66
  - エディタ, 66

## ガ

- 外部イメージファイル
  - フォルダリファレンスの作成, 341
- 外部通信
  - PCステーションによる, 438

## カ

- 確認タグ
  - 書込み, 198
  - 読み取り, 198
- カバーシート
  - レイアウト, 369

## ガ

- 画面
  - 作成の作業手順, 164
- 画面エディタ, 163
- 画面ナビゲーション
  - 作業エリア, 167
- "画面ナビゲーション"エディタ, 166

## カ

- 簡易[レシピ]ウィンドウ
  - 操作, 260
  - 動作, 261
- 間接アドレス指定, 140

## キ

- キーボードによる操作
  - [レシピ]ウィンドウ, 259
- 基準言語, 347
- 機能
  - HMIデバイスタイプ依存性, 90
- 機能範囲
  - ProSave, 419
- 基本設定
  - アラーム, 206
  - スクリプト, 329
- 切り替え, 68

## ク

- クイックスタート
  - WinnCC flexibleの, 85
- 組み合わせ, 58
  - ウィンドウ, 58

## グ

- グラフィック
  - イメージブラウザからイメージを使用する, 175
  - 管理, 176
  - 透明な背面, 175
- グラフィックIOフィールド, 169, 289
- [グラフィック]ウィンドウ, 169
- グラフィックオブジェクト ; グラフィックオブジェクト, 169
- グラフィックブラウザ
  - イメージの使用, 175
- グラフィックリスト, 78, 81, 82
  - アプリケーション, 78

デフォルト値, 81  
最低重要度ビットセット, 82  
設定, 78  
グラフィック表示, 289  
グループ, 177  
グローバルライブラリ, 182  
グローバル割り付け  
ファンクションキーの, 180

## ク

クロスリファレンスリスト, 118  
エディタ, 118  
作業, 118

## ゲ

ゲージ, 169  
限界範囲  
タグ, 136  
言語依存性  
システムファンクション, 333  
言語切り替え  
ランタイム中, 344

## ケ

検索, 119  
オブジェクト, 119  
文字列, 119

## コ

更新  
HMIデバイスのオペレーティングシステム, 421  
構成要素  
プロジェクトの, 89  
構文の強調表示, 327  
変更, 330  
コピー  
概要, 112  
フラット, 113  
統合接続, 425  
コントローラ  
タグのリンク, 134  
コントロールエレメント  
プロジェクトバージョンの管理, 399  
変更のロギング, 409  
コンフィギュレーション  
[移動], 39  
移動パス, 39

オブジェクトの同時編集, 38  
タグ, 136

## サ

サイクリックログ, 211  
サイズ  
オブジェクト, 173  
複数のオブジェクトの編集, 174  
再配線, 119  
作業エリア, 48  
画面エディタの, 163  
画面ナビゲーション, 167  
ジョブの計画, 391  
スクリプトエディタ, 325  
タグエディタ, 131, 202, 205, 212, 359  
プロジェクトバージョンの管理, 399  
ユーザー管理, 305, 306  
構造エディタ, 147  
作業環境  
ユーザー依存性, 84  
リセット, 84  
作業中  
[オブジェクト]ウィンドウでの, 106  
クロスリファレンスリストで, 118  
作業手順  
画面の作成, 164  
作成  
HMIステーション, 431  
参照  
オブジェクト, 334

## シ

シーケンス, 389  
タスク, 389  
四角形, 169  
システムアラーム, 191  
システムアラームエディタ, 203  
システムアラームエディタ, 203  
システム辞書, 358  
エディタ, 359  
エディタの構造, 359  
システム辞書エディタ, 359  
システムファンクション, 315, 317  
アプリケーション, 318, 319  
言語依存性, 318, 333  
使用, 319  
スクリプトで, 319  
スクリプトでの, 332, 344  
スクリプトでの呼び出し, 332  
パラメータの受け渡し, 333

ファンクションリストでの, 319, 323  
 呼び出しにおける特殊性, 333  
 システム制限, 454  
   HMIデバイス, 454  
 システム辞書エディタ, 360

## ジ

実行  
   ランタイム中のスクリプト, 342  
 自動化  
   自動アラーム配信, 32  
 自動翻訳, 358

## シ

シミュレーション, 121

## ジ

事務所, 6

## シ

[出力]ウィンドウ, 52  
   プロパティ, 53  
 取得サイクル  
   タグ, 135  
 使用  
   システムファンクション, 319  
   スクリプト, 321  
 ショートカットメニュー  
   アクセス, 60

## ジ

ジョブの計画, 387, 390  
   アプリケーションのフィールド, 387  
   作業エリア, 391  
   適用例, 387

## シ

シングルユーザープロジェクト, 89  
 シンボルIOフィールド, 289  
 シンボルサーバ  
   統合されたプロジェクト, 440  
 シンボルサーバー  
   統合されたプロジェクト, 425  
 シンボルのIOフィールド, 169

シンボルライブラリ, 169

## ス

スイッチ, 169  
 スクリプト, 316, 320  
   HMIデバイスの依存性, 333  
   アプリケーション, 321  
   基本設定, 329  
   コードフォーマットの変更, 330  
   システムファンクションの使用, 344  
   スクリプトでの, 332  
   スクリプトでの呼び出し, 332  
   整列, 320  
   タグ値の更新, 128  
   パラメータの受け渡し, 333  
   反復レベル, 317  
   ファンクションリストでの, 323  
   プロパティ, 320  
   ヘルプファンクション, 329  
   戻り値, 343  
   呼び出しにおける特殊性, 333  
   ランタイム中の処理, 342  
 スクリプトエディタ, 324  
   構文の強調表示, 330  
   作業エリア, 325  
   設定の編集, 329  
   プロパティ, 326  
 スケーラビリティ, 36  
 スケーリング  
   線形スケーリング, 135  
   タグ、線形スケーリング, 139  
 スケジューラ, 387  
 スタートセンター  
   メニューコマンド, 85  
   使用不可, 86  
 ステータス情報  
   ファンクションリスト, 323  
 ステータスの強制, 169

## す

すべての再構築  
   プロジェクト, 121

## ス

スライダ, 169

## セ

- 整列
  - スクリプト, 320
- 設計
  - プロトコル, 282
- 接続
  - 変更, 445, 446
- 接続の編集
  - NetProを使った, 430
  - WinCC flexibleを使った, 430
- 設定
  - アラーム, 206

## ソ

- 操作
  - 簡易[レシビ]ウィンドウ, 260
  - ホットキー, 61
  - [レシビ]ウィンドウ, 258

## ゾ

- ゾーン, 378, 380
  - 作業エリア, 381
- ゾーンエディタ, 380

## ソ

- ソフトバス
  - 設定, 438

## ダ

- ダイナミックコントロール
  - 銘板, 186
  - 銘板インスタンス, 186
  - 銘板に含まれるオブジェクト, 186
- ダイナミックな操作, 179

## タ

- タイマ, 387, 389

## ダ

- 代理店, 6
- 楕円, 169

## タ

- 多角折線, 169
- 多角形, 169
- タグ
  - [プロパティ]ウィンドウ, 132
  - CSVエクスポート, 151
  - OP 77A, 242
  - PLCとの通信, 134
  - TP 177A, 242
  - VBSによるアクセス, 331
  - インデックスタグ, 140
  - インポート, 150
  - エクスポート, 150, 151
  - 外部タグ, 127, 128
  - 間接アドレス指定, 140
  - 限界範囲, 136
  - コンフィギュレーション, 136
  - 取得サイクル, 135
  - スクリプト内の同期化, 335
  - 線形スケーリング, 135
  - 多重化, 140
  - データログ, 138, 277
  - 内部タグ, 129
  - プロパティ, 133
  - ローカル, 331
  - ロギング, 138, 277
  - ロギングサイクル, 144
  - 初期値, 137
  - 取得サイクル, 137, 144
  - 構造, 145
  - 構造エレメント, 145
  - 線形スケーリング, 139
  - 許容範囲, 138, 277
  - 連続読取り, 137
  - 配列, 141
  - タグエディタ, 130
    - 作業エリア, 202, 205, 212
    - 作業エリア, 203, 204, 274, 361
  - タグデータ
    - インポート用フォーマット, 155
  - タグのインポート
    - タグデータのフォーマット, 155
    - 手順, 150
    - 接続データのデータ構造, 153
  - タグの受け入れ
    - STEP 7, 444
    - STEP 7からの, 445
  - タグの接続
    - アプリケーションポイントを介した, 445
    - タグエディタを介した, 444
  - タグの同期化
    - [レシビ]ウィンドウで, 257

タグの有効化  
STEP 7, 444  
タグリスト  
間接アドレス指定, 140  
タグ値  
出力, 278  
タグ接続  
タグエディタを使用, 444  
多重化, 140  
タスク, 388  
シーケンス, 389  
タブ, 68

## チ

置換, 119  
オブジェクト, 119  
文字列, 119

## ツ

通信  
PLCとタグ間, 134  
エリアポイントの使用, 220  
タグの使用, 220  
通信ドライバ, 221  
ツールバー, 47, 58  
ドッキング, 58  
位置付け, 46  
言語サポート, 353  
設定, 47  
順序, 173  
ツールヒント, 83  
ツールボックス, 163  
書込み禁止のライブラリ, 182

## デ

ディスクリートアラーム  
ディスクリートアラームエディタ, 201  
ディスクリートアラームエディタ, 201  
ディスクリートアラーム手順, 188  
データ  
HMIデバイス固有の, 93  
グローバルプロジェクト, 94  
データタイプ  
外部タグ, 135  
内部タグ, 129  
データの一括処理, 38  
利点, 38  
データの選択

プロジェクトレポートの, 373  
データの復元  
HMIデバイス, 420  
データバックアップ  
HMIデバイス, 420  
データフロー, 236  
データメールボックス  
レシピ用, 243  
データレコードリスト, 251  
データレコード名, 247  
データレコード番号, 247  
データロギング, 269  
アプリケーション, 269  
ロギングサイクル, 270  
取得サイクル, 270  
記憶媒体, 270  
データログ  
エディタ, 274  
タグ, 138, 277  
タグ値の出力, 278  
基本設定, 275  
データ交換, 226  
データ媒体  
プロジェクトレポート, 368

## テ

テキストエントリ, 74  
テキストリスト, 74  
テキストフィールド, 169, 289  
テキストリスト, 73  
アプリケーション, 73  
テキストリスト  
設定, 73  
テキストリスト, 73  
テキストリスト  
デフォルト値, 76  
テキストリスト, 76  
テキストリスト  
最低重要度ビットセット, 76  
テキストリスト, 76  
テキストリスト  
レシピデータレコードの使用, 246  
テキスト参照機能, 353

## デ

デバイスの交換  
DP設定, 92  
ランタイムに動作不能なデバイス, 92  
デバイスベース依存性, 63, 89, 90  
原理, 63

デバイスベース依存性  
画面の, 161  
デバッグ  
エラータイプ, 336  
デフォルトプロパティ  
変更, 285  
デフォルト値, 76  
デルタダウンロード, 415

## テ

転送  
HMIデバイスのライセンスキー, 422  
OP 73, 415  
USB経由, 417  
オブジェクト名を転送しない, 330  
スクリプトからのコメントの削除, 330  
デルタダウンロード, 415  
レシピ, 416

## ト

統合されたプロジェクト  
アップロード, 124, 417  
バージョンを付けて保存, 436  
バージョンを指定して保存, 436

## ト

動作  
簡易[レシピ]ウィンドウ, 261

## ト

トグル  
ランタイム言語間, 364  
時計, 169

## ト

ドッキング, 58  
ウィンドウ, 58  
ツールバー, 58  
ドラッグアンドドロップ, 59, 328

## ト

トランク, 395  
トランスポンダ, 378  
トレーニングセンター, 7

トレンド, 272  
[トレンド]ウィンドウ, 272  
[トレンド]ウィンドウ, 169

## ナ

内部通信  
PCステーションの, 438  
ナビゲーション, 165  
ナビゲーションコントロール, 168  
ナビゲーション矢印, 69  
ナビゲーション構造体, 40  
特定, 40

## バ

バー, 169  
バージョン  
HMIデバイス, 123, 413  
バージョンの比較, 402  
バージョンを付けて保存  
バージョンの概要, 108  
変換オプション, 108  
バージョンを指定して保存  
STEP 7 に統合されたプロジェクト, 435, 436  
バージョン管理, 393  
統合されたプロジェクトで, 425

## バ

パーソナル HMI デバイス  
使用, 34

## ハ

配列  
間接アドレス指定, 140

## は

はじめに  
プロジェクト文書, 367

## パ

パスワード  
初期化, 416  
変換後, 110

## バ

バックアップ  
HMIデータ, 420

## パ

パフォーマンス特性  
HMIデバイス, 454  
パラメータ  
接続エディタ, 224  
パラメータの受け渡し  
システムファンクション, 333  
スクリプト, 333

## ハ

反復レベル, 317

## ヒ

日付/時刻フィールド, 169  
表示クラスの定義  
ALARM\_Sアラームのための, 207, 446  
標準レイアウト, 370  
開く, 66  
エディタ, 66

## フ

ファンクション  
タグ値の更新, 128  
ファンクションキー, 160, 180  
ファンクションリスト, 71, 322  
HMIデバイスの依存性, 323  
完了, 323  
システムファンクション, 323  
スクリプト, 323  
ステータス情報, 323  
同期実行, 341  
非同期実行, 341  
プロパティ, 323  
ランタイム中の実行, 341  
設定, 71  
フィルタリング  
[アラーム]ウィンドウの, 193

## ブ

ブートストラップ  
オペレーティングシステム, 422

## フ

フェイスプレート  
アプリケーション, 185  
インスタンスの再使用, 185  
フェイスプレートインスタンス, 185  
フェイスプレートインスタンス, 185  
再使用, 185  
復元  
HMIデータ, 420  
複数選択, 177

## ブ

ブランチ, 396, 397

## プ

プリント  
選択, 373  
プロジェクト, 63, 87, 404  
HMI デバイスの依存性, 89  
機能範囲, 64  
作業, 63  
シミュレータによるテスト, 121  
新規作成, 63  
すべての再構築, 121  
バージョン管理下, 407  
複数のHMIデバイス, 90  
複数のプロジェクト, 63  
ロード, 63  
移行, 107  
編集, 100  
複数HMIデバイス, 95  
複数言語のプロジェクト, 98  
[プロジェクト]ウィンドウ, 49  
HMIデバイスタイプの選択, 90  
作業, 49  
[プロパティ]ウィンドウ  
プロパティ, 50  
プロジェクト言語, 346  
プロジェクトコンフィギュレーションの変更, 404  
記録済みのプロジェクトの変更, 404  
プロジェクトセッション, 405  
プロジェクトタグ, 331  
プロジェクトデータ, 65  
更新, 65  
プロジェクトテキスト  
アクセス先, 354  
エディタ, 354  
外部への変換, 356  
プロジェクトテキストエディタ, 354

プロジェクトナビゲーション, 40  
プロジェクトの変換  
    アラームクラス群の表示, 109  
    ライブラリのHMIデバイス, 109  
    変換後のシステム要件, 109  
プロジェクトバージョン, 393, 395, 407  
    旧プロジェクトバージョン, 395  
    新規プロジェクトバージョン, 407  
    旧プロジェクトバージョン, 407  
    次のバージョン, 401  
    現在のバージョン, 401  
    管理, 397  
プロジェクトバージョンの管理  
    コントロールエレメント, 399  
    作業エリア, 399  
プロジェクトバージョンの管理, 393  
    [プロパティ]ウィンドウ, 401  
    アプリケーションのフィールド, 393  
    適用例, 393  
プロジェクトライブラリ, 117, 182  
プロジェクトレポート, 367  
    オブジェクトの選択, 374  
    コンテンツページのレイアウト, 369  
    出力完了, 369  
    出力コンパクト, 369  
    データの選択, 373  
    データ媒体, 368  
    レイアウト, 368  
    出力, 374  
    単一オブジェクトの場合, 373  
    編集可能なもの, 372  
プロジェクト文書  
    アプリケーション, 367  
    はじめに, 367  
プロジェクト言語  
    エディタ, 349  
プロジェクト言語エディタ, 349  
プロセスアラーム  
    有効化, 209, 448  
プロセス画面  
    変更, 40  
プロトコル  
    設計, 282  
    適用例, 281  
プロパティ  
    アラーム, 197  
    タグ, 133  
    ファンクションリスト, 323  
    レイアウト, 370  
[プロパティ]ウィンドウ, 50

## へ

ヘルプ, 83  
    表示, 83  
ヘルプインジケータ, 169  
ヘルプファンクション, 329  
変更, 68  
    オブジェクトプロパティ, 344  
    構文の強調表示, 330  
    スクリプトでのコードフォーマット, 330  
変更のロギング, 408  
変更のログ  
    統合されたプロジェクトで, 432  
変更ログ, 404  
編集  
    レイアウト, 370, 371  
編集可能なもの  
    プロジェクトレポート, 372  
編集言語, 347

## ボ

ボタン, 169

## ホ

## 翻訳

エディタ, 345  
自動, 358  
ワークフロー, 351

## マ

マウスによる操作  
    [レシピ]ウィンドウ, 259  
マウス機能, 60  
マルチユーザープロジェクト, 90

## メ

銘板, 117  
メニュー, 46  
    コマンド, 47  
メニューコマンド  
    簡易[レシピ]ウィンドウ, 261

## モ

戻り値, 343  
モバイルユニット  
    使用, 30

モバイルワイヤレス, 375, 377, 380, 381, 382, 384  
 アプリケーションのフィールド, 375  
 作業エリア, 381, 384  
 動作, 377

## ユ

ユーザー依存性  
 作業環境, 84  
 ユーザーインターフェース言語, 346  
 WinCC flexible, 402  
 ユーザー管理, 303, 304  
 作業エリア, 305  
 ユーザー辞書, 358  
 アンインストール, 358  
 ユーザーデータ, 311  
 インポート, 311  
 エクスポート, 311  
 ユーザーパスワード  
 変換後の変更, 110  
 ユーザー権限  
 WinCC flexibleの初期起動時, 86  
 ユーザー管理, 301, 302  
 SIMATICログオン, 309  
 アプリケーションのフィールド, 301  
 作業エリア, 306  
 一元的なユーザー管理, 309  
 構造, 302  
 目的, 301  
 ユーザー辞書  
 エディタ, 360  
 エディタの構造, 360

## ラ

ライセンス  
 WinCC flexible ES用, 25  
 WinCC flexible Runtime用, 25  
 オプション用, 25  
 注文, 26  
 ライセンスキー  
 HMIデバイスへの転送, 422  
 ライセンスキーディスク, 422  
 ライセンスキーの保存場所, 422  
 ライブラリ, 51, 117, 182  
 グローバル, 52  
 ツールボックス内の表示, 182  
 プロジェクト関連の, 52  
 ライブラリオブジェクト, 182  
 ライン, 169, 289  
 ランタイム, 385  
 VBSでのオブジェクトプロパティの変更, 344

言語切り替え, 344  
 スクリプティング, 315  
 スクリプトの処理, 342  
 タスク, 23  
 ファンクションリストの実行, 341  
 ユーザー管理, 301  
 有効な範囲, 385  
 ランタイムエラー, 336  
 ランタイム言語, 347  
 ランタイムスクリプト, 316  
 ランタイムセキュリティの設定  
 エディタ, 307  
 ユーザー管理のエクスポート, 308  
 構造, 307  
 開く, 307  
 ランタイム言語, 363

## リ

リストエントリ, 74, 79  
 グラフィックリスト, 79  
 テキストリスト, 74  
 リモートアクセス, 31  
 アプリケーションの可能性, 31

## ル

ルーティング接続, 439  
 作成, 439  
 ルーティング接続  
 転送用の, 441

## レ

レイアウト  
 カバーシート, 369  
 プロジェクトレポート, 368  
 プロパティ, 370  
 編集, 370, 371  
 日付、時刻、通貨、および数値の地域フォーマット  
 , 348  
 簡易[レシピ]ウィンドウ, 260  
 レシピ, 231, 233  
 [レシピ]ウィンドウ, 250  
 [レシピ]画面, 250  
 TP 177AとOP 77Aの相違, 241  
 タグ, 242  
 データフロー, 236  
 データレコード, 233  
 テキストリストの使用, 246  
 転送, 416

表示, 232  
 レポート用出力データ, 296  
 使用, 232  
 入力, 234  
 基本, 233  
 基本原理, 231  
 必要メモリ, 451, 452  
 設定, 248  
 設定オプション, 239  
 転送中の初期化, 416  
 適用例: ロット生産, 232  
 適用例: マシンパラメータ割り付け, 232  
 [レシピ]ウィンドウ, 169, 258  
   アプリケーション, 258  
   操作, 258  
 レシピエディタ, 247  
   エレメントタブ, 245  
   データレコード, 247  
   作業エリア, 244  
 レシピ画面  
   レシピタグ, 235  
 レシピ構造の変更, 263  
 レシピタグ  
   オンライン, 240  
   同期化, 239  
 レシピデータ  
   ロードと保存, 237  
   転送中の上書き, 416  
 レシピデータレコード  
   インポートとエクスポート, 238  
   テキストリストの使用, 246  
   転送オプション, 236  
 レシピデータレコード名, 247  
   タグへの書き込み, 253  
 レシピデータレコード番号, 247  
   タグへの書き込み, 253  
 レシピの出力データ, 296  
 レシピの操作  
   レシピ構造の変更, 263  
 レシピリスト, 251  
 レシピレポート  
   コンフィグレーション, 297  
 レシピ値の転送  
   HMIデバイスとPLC間, 237  
 レシピ名  
   タグへの書き込み, 253  
 レシピ画面  
   タグの同期化, 257  
   画像マシンシミュレーション, 256  
 レシピ番号  
   タグへの書き込み, 253  
 レシピ設定, 248  
 レポートオブジェクト

アプリケーション, 292  
 レポートシステム, 281  
   概要, 281  
 レポートプロパティ  
   編集, 288

## □

ローカルタグ, 331  
 ローカル割り付け  
   ファンクションキーの, 180  
 ログギング  
   アラーム, 215  
   基本原理, 210  
   サイクリックログ, 211, 270  
   タグ, 138, 277  
   タグ値, 269  
   ログタイプ, 211, 270  
   一定期間のサイクリックログ, 211, 270  
 ログギングサイクル  
   タグ, 144  
 ログタイプ, 211  
   サイクリックログ, 270  
   レベル依存の, 270  
   一定期間のサイクリックログ, 270  
 ログデータベース  
   ODBCによる直接アクセス, 217, 280  
 ログの内容  
   表示, 211, 271

## ワ

ワイヤレス, 375  
 ワイヤレスネットワーク, 375

## 一

一元的なユーザー管理, 309  
 一定期間のサイクリックログ, 211  
 一貫した設定, 21  
 一貫性チェック  
   生成, 120

## 位

位置  
   オブジェクト, 173  
   複数のオブジェクトの編集, 174

**作**

## 作成

HMIステーション, 431  
 新規プロジェクトバージョンの作成, 407

## 作業エリア, 384

ゾーン, 381  
 タグエディタ, 201, 203, 204, 274, 361  
 モバイルワイヤレス, 381, 384  
 変更のロギング, 410  
 有効な範囲, 384

## 作業中

プロジェクトの, 87

## 作業手順, 65

元に戻す, 65  
 復元, 65

**使**

## 使用, 232

レシピの, 232  
 銘板, 37

**信**

## 信号, 187

基本原理, 187

**全**

## 全体的に統合された自動化, 41

SIMATIC STEP 7, 41  
 SIMOTION SCOUT, 41

**共**

## 共有ライブラリ, 117

**内**

## 内部タグ, 331

**円**

## 円, 289

**再**

## 再使用

テキストライブラリ, 38

フェイスプレート, 185

ライブラリ, 38

**出**

## 出力

プロジェクトレポート, 374

**分**

分散HMIデバイス, 33

**初**

## 初期値

タグ, 137

## 初期化

パスワード, 416  
 レシピ, 416

**割**

## 割り当て

ファンクションキー, 180

**動**

## 動作

[レシピ]ウィンドウ, 255

**単**

単純コピー, 113

単純なオブジェクト, 289

**印**

## 印刷

選択, 374

**反**

## 反転

オブジェクト, 174

**取**

取得サイクル

タグ, 137, 144

## 同

### 同期化

[レシピ]ウィンドウと[レシピ]画面, 235  
PLCとの, 243  
スクリプト内のオブジェクトの, 335  
スクリプト内のタグの, 335  
レシピタグ, 239  
手動, 335

## 四

四角形, 289

## 回

### 回転

オブジェクト, 173

## 基

### 基本設定

データログ, 275

## 変

### 変換

STEP 7 に統合されたプロジェクト, 435, 436  
パスワードは変更されます。 , 110

### 変換オプション

バージョンを付けて保存, 108

### 変更

接続, 445, 446  
デフォルトプロパティ, 285  
接続, 445

### 変更のロギング, 403

アプリケーション, 403  
アプリケーションのフィールド, 403  
コントロールエレメント, 409  
作業エリア, 410

### 変更のログ, 407

統合されたプロジェクトで, 432  
統合されたプロジェクトで, 403, 425

変更ログ, 403, 405

## 外

外部イメージファイル

管理, 174  
外部タグ, 331

## 多

多角形, 289  
多角折線, 289

## 安

### 安全対策注意事項

バックグラウンドでのレシピデータレコード, 252

## 必

### 必要メモリ

レシピ, 451, 452

## 指

### 指定

オブジェクトグループのプロパティ, 178

## 挿

### 挿入

オブジェクト, 113  
オブジェクトの挿入, 173

## 接

### 接続

変更, 445  
接続工データ, 222  
接続データ  
インポート用フォーマット, 153

## 操

### 操作

マウスを使用した, 59

## 整

### 整列

画面のオブジェクト, 173

**文**

文字セット  
その国特有の文字, 353  
設定, 365

**日**

日付/時刻フィールド, 289

**更**

更新サイクル, 144

**書**

書込み許可  
WinCC flexibleの初期起動用, 86

**最**

最低重要度ビットセット, 76, 82

**有**

有効な範囲, 385  
ランタイム, 385  
作業エリア, 384  
概要, 385  
有効範囲, 378, 382, 384, 385

**東**

東洋言語文字  
HMIデバイスでの入力, 364

**楕**

楕円, 289

**概**

概要  
レポートシステム, 281

**構**

構造, 145, 302  
"データログ"エディタ, 274

コピー, 148  
タグ, 145  
ユーザー管理, 302  
ランタイムセキュリティの設定, 307  
削除, 148  
名前の変更, 148  
挿入, 148  
銘板, 145  
構造エディタ  
作業エリア, 147  
構造エレメント  
コピー, 149  
タグ, 145  
削除, 149  
名前の変更, 148  
挿入, 149  
編集, 149

**特**

特殊文字, 353

**生**

生成  
一貫性チェック, 120

**画**

画面  
デバイススペース依存性, 161  
フォント, 162  
画面ナビゲーション, 40

**確**

確認  
アラーム, 189

**移**

移行  
WinCCまたはProToolプロジェクト, 107

**管**

管理  
グラフィック, 176  
プロジェクトバージョン, 393, 397

## 簡

簡易[レシピ]ウィンドウ, 251  
コントロールオブジェクト, 260  
メニューコマンド, 261  
レイアウト, 260

## 統

統合されたプロジェクト  
HW Configの使用, 429  
STEP 7 シンボル, 428  
アップロード, 425  
シンボルサーバ, 440  
シンボルサーバー, 425  
バージョンを指定して保存, 435, 436  
バージョン管理, 425  
変更のログ, 432  
変更のログ, 425, 432  
変更ログ, 403  
統合接続  
コピー, 425

## 編

編集  
プロジェクト, 100  
レポートプロパティ, 288

## 翻

翻訳  
外部でのプロジェクトテキストの翻訳, 356

## 自

自動での同期化, 335  
自動化  
1台のHMIデバイスを使用した制御, 27  
コンセプト, 27  
シングルユーザーシステム, 27  
モバイルユニット, 30  
分散HMI, 33  
複数のHMIデバイスのPLC, 28  
集中ファンクションを使用したHMIシステム, 29

## 表

表示クラスの定義  
ALARM\_Sアラームのための, 207, 209, 446, 448

## 言

言語  
個々のエディタにおいて, 353  
言語サポート  
ツールバー, 353  
言語によって異なるフォーマット, 348  
言語切り替え, 364  
ProSave, 419

## 設

設定  
HMIデバイスに依存しない, 37  
HMIデバイス依存, 36  
PCステーション, 437  
アジア言語, 352  
イベント, 254  
オペレーティングシステムにおける言語, 348  
ソリューションベースのコンセプト, 35  
ツールバー, 47  
レシピ, 248  
レシピ, 248  
対象デバイスに依存しない, 37  
対象デバイス依存, 36  
画面切替え, 40  
転送の, 413

## 許

許容範囲  
タグ, 138, 277

## 詳

詳細[レシピ]ウィンドウ, 251

## 論

論理エラー, 336

## 起

起動  
WinCC flexible, 85

## 転

転送  
\*.pwx, 122, 412

OP 73micro, 415  
OP 77A, 415  
S7Ethernet, 414  
TP 177A, 415  
TP 177micro, 415  
アップロード, 415  
モード, 413  
ルーティング接続経由の, 441  
レシピデータの上書き, 416  
基本原理, 122, 411  
転送設定, 413  
転送モード  
HMIデバイス上, 123, 412

## 透

透過性  
グラフィックで, 175

## 通

通信, 375  
イーサネット, 226, 229  
モバイル通信, 375  
通信パートナー, 219

## 連

連続読取り  
タグ, 137

## 適

適用例, 387  
オフラインでのレシピデータの入力, 264  
レポート用, 281  
手動製造シーケンスのレシピ, 265

## 選

選択  
プロジェクトレポートのオブジェクト, 374  
印刷, 373, 374  
複数のオブジェクト, 174

## 配

配列, 141  
配列エレメント, 141  
配列タグ, 141

## 重

重要度, 76, 82

## 銘

銘板  
ダイナミックコントロール, 186  
構造, 145

## 開

開く  
"データログ"エディタ, 274  
ランタイムセキュリティの設定, 307  
レシピエディタ, 244  
旧バージョンのプロジェクトを開く, 407

