SIEMENS

SIMATIC HMI

WinCC flexible 2005 Compact / Standard / Advanced

User's Manual

This manual is part of the documentation package with the order number 6AV6691-1AB01-0AB0

Edition 06/2005
A5E00280169-03

Preface
Introduction to WinCC flexible 1
WinCC flexible Engineering System 2
Working with projects 3
Working with Tags 4
Creating Screens 5
Creating an Alarm System 6
Working with a connection 7
Structure of a recipe management system 8
Logging and displaying tags 9
Working with reports 10
User administration 11
System functions and runtime scripting 12
Structure of Multilingual Projects 13
Project documentation 14
Planning jobs 15
Managing project versions 16
Logging changes 17
Transfer 18
Integration of WinCC flexible in STEP 7 19
Appendix 20

www.ElectricalPartManuals.com

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. These notices shown below are graded according to the degree of danger.

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Copyright Siemens AG 2005. All rights reserved.
The distribution and duplication of this document or the utilization and transmission of its contents are not permitted without express written permission. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Disclaimer of Liability
We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 4848, 90327 Nuremberg, Germany

© Siemens AG 2005
Technical data subject to change

www.ElectricalPartManuals.com

Preface

Purpose of this manual

This user manual is part of the WinCC flexible documentation. The manual provides you with a complete overview of configuring with WinCC flexible. The manual supports you in creating new projects, in the procedure used during configuration and in transferring a project to an HMI device.

The manual is intended for newcomers, operators and configuration engineers involved in configuration, commissioning, installation and service with WinCC flexible.

The help integrated in WinCC flexible, the WinCC flexible Information System, contains detailed information. The information system contains instructions, examples and reference information in electronic form.

Basic Knowledge Requirements

General knowledge in the field of automation engineering is required to understand this manual.

You should also have experience of using PCs running under the Windows 2000 or Windows XP operating systems. A knowledge of VBA or VBS is required for advanced configuration by using scripts.

Scope of the manual

This manual is valid for the WinCC flexible 2005 software package.

Position in the information scheme

This manual is part of the SIMATIC HMI documentation. The information below presents an overview of the information landscape of SIMATIC HMI.

User manual

- WinCC flexible Micro
  - describes the engineering basics based on the WinCC flexible Micro engineering system (ES)
- WinCC flexible Compact/ Standard/ Advanced
  - describes the engineering basics based on the WinCC flexible Compact, WinCC flexible Standard and WinCC flexible Advanced engineering systems (ES)
- WinCC flexible Runtime:
  - Describes how to commission and operate your Runtime project on a PC.

www.ElectricalPartManuals.com

- WinCC flexible Migration:
  - Describes how to convert an existing ProTool project to WinCC flexible.
  - Describes how to convert an existing WinCC project to WinCC flexible.
  - Describes how to migrate ProTool projects with an HMI migration from OP3 to OP 73 or OP 73 micro.
  - Describes how to migrate ProTool projects with an HMI migration from OP7 to OP 77B or OP 77A.
  - Describes how to migrate ProTool projects with an HMI migration from OP 17 to OP 177B.
  - describes how to migrate ProTool projects with HMI migration from RMOS graphic devices to Windows CE devices.
- Communication:
  - Communication Part 1 describes the connection of the HMI device to SIMATIC PLCs.
  - Communication Part 2 describes the connection of the HMI device to third-party PLCs.

Operating Instructions

- Operating instructions for SIMATIC operating units:
  - 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
  - MP 270B
  - MP 370
- Operating instructions for mobile SIMATIC operating units:
  - Mobile Panel 170
- Operating instructions (compact) for SIMATIC operating units:
  - OP 77B
  - Mobile Panel 170

Getting Started

- WinCC flexible for first time users:
  - Based on a sample project, this is a step-by-step introduction to the basics of configuring screens, alarms, and recipes, and screen navigation.
- WinCC flexible for advanced users:
  - Based on a sample project, this is a step-by-step introduction to the basics of configuring logs, project reports, scripts, user management, and multilingual projects, and integration into STEP 7.
- WinCC flexible options:
  - Based on a sample project, this is a step-by-step introduction to the basics of configuring the WinCC flexible Sm@rtServices, Sm@rtAccess and OPC Server options.

www.ElectricalPartManuals.com

Online availability

The following links provide direct access to technical documentation on SIMATIC products and systems in English, German, French, Italian, and Spanish.

- SIMATIC Guide Technische Dokumentation in Deutsch: "http://www.ad.siemens.de/simatic/portal/html_00/techdoku.htm"
- SIMATIC Guide for Technical Documentation in English: "http://www.ad.siemens.de/simatic/portal/html_76/techdoku.htm"

Guide

Structure of this manual:

- Introduction to WinCC flexible – Chapter 1
- Working with WinCC flexible – Chapters 2 -16
- Transferring a project to an HMI device – Chapter 17
- Integration of WinCC flexible in STEP 7 – Chapter 18
- Features of WinCC flexible – Chapter 19

Conventions

A distinction is made in the naming conventions for the configuration and runtime software:

- "WinCC flexible 2005" refers to the configuration software.
- "Runtime" designates the runtime software running on the HMI devices.
- "WinCC flexible Runtime" designates the visualization product for use on standard PCs or panel PCs.

The term "WinCC flexible" is used in the general context. A version name such as "WinCC flexible 2005" is used whenever it is necessary to distinguish it from other versions.

The following conventions are used in the text and will help you to read the manual more effectively:

Notation	Scope
"Add screen"	• Terminology that occurs in the user interface, e.g., dialog names, tabs, buttons, menu commands. • Inputs required, e.g., limit values, tag values • Path information
"File > Edit"	Operational sequences, e.g., menu commands/context menu commands.
<F1>, <Alt + P>	Keyboard inputs

Please observe notes labeled as follows:

Note

Notes containing important information about the product and its use or a specific section of the documentation to which you should pay particular attention.

www . ElectricalPartManuals . com

Trademarks

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

Third parties using for their own purposes any other names in this documentation which refer to trademarks might infringe upon the rights of the trademark owners.

Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

You will find your contact person at:

"http://www.siemens.com/automation/partner"

You will find a guide to the technical documentation offered for the individual SIMATIC Products and Systems here at:

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

The online catalog and order system is found under:

"http://mall.automation.siemens.com/"

Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: "http://www.sitrain.com"

Technical Support

You can reach the Technical Support for all A&D products
via the Web formula for the Support Request

"http://www.siemens.com/automation/support-request"

Phone: + 49 180 5050 222

Fax: + 49 180 5050 223

Additional information about our Technical Support can be found on the Internet pages "http://www.siemens.com/automation/service"

www.ElectricalPartManuals.com

Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

"http://www.siemens.com/automation/service&support"

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.
- The right documents via our Search function in Service & Support.
- A forum, where users and experts from all over the world exchange their experiences.
- Your local representative for Automation & Drives.
- Information on field service, repairs, spare parts and more under "Services".

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

Table of contents

Preface ..... **i**

1 Introduction to WinCC flexible **1-1**

1.1 Introduction to SIMATIC HMI 1-1

1.2 WinCC flexible system overview 1-2
1.2.1 Components of WinCC flexible 1-2
1.2.2 WinCC flexible Engineering System 1-3
1.2.3 WinCC flexible Runtime 1-5
1.2.4 Available options 1-5
1.2.5 Licensing 1-6
1.2.5.1 Licenses and License Key 1-6
1.2.5.2 WinCC flexible without licensing 1-7

1.3 Automation concepts 1-8
1.3.1 Automation concepts with WinCC flexible 1-8
1.3.2 Remote access to HMI devices 1-10
1.3.3 Automatic alarm dispatch 1-11
1.3.4 Distributed HMI 1-12

1.4 Configuration concepts 1-13
1.4.1 Configuration Support 1-13
1.4.2 Scalable Configuration Tools 1-14
1.4.3 PLC-independent configuration 1-15
1.4.4 Use 1-15
1.4.5 Intelligent tools 1-16
1.4.5.1 Bulk data processing 1-16
1.4.5.2 Configuring movement paths 1-17
1.4.5.3 Graphic Configuration of the Screen Navigation 1-18
1.4.6 Totally Integrated Automation 1-19

2 WinCC flexible Engineering System **2-1**

2.1 Basic Principles on the Programming Interface 2-1

2.2 WinCC flexible user interface 2-1
2.2.1 WinCC flexible User Interface Elements 2-1
2.2.2 Menus and Toolbars 2-3
2.2.3 Work area 2-5
2.2.4 Project View 2-6
2.2.5 Property view 2-7
2.2.6 Library 2-8
2.2.7 Output View 2-9
2.2.8 Object view 2-10

2.3 Placing editor-specific operating elements 2-11

2.4 Working with windows and toolbars 2-12

2.5 Working with the Mouse 2-15

2.6 Keyboard control 2-16

www.ElectricalPartManuals.com

2.7 Working with WinCC flexible ... 2-17
2.7.1 Working with WinCC flexible ... 2-17
2.7.2 Working with projects ... 2-17
2.7.3 Editing Multiple Projects with WinCC flexible ... 2-18
2.7.4 Functional scope of a project ... 2-19
2.7.5 Editor Properties ... 2-20
2.7.6 Open Editor ... 2-21
2.7.7 Switching between editors ... 2-23
2.7.8 Displaying Help ... 2-25
2.7.9 Customized setup of WinCC flexible ... 2-26

3 Working with projects ... 3-1

3.1 Basis for working with projects ... 3-1
3.1.1 Working with projects ... 3-1
3.1.2 Component parts of a project ... 3-2

3.2 Types of projects ... 3-2
3.2.1 Types of projects ... 3-2
3.2.2 HMI device dependency of projects ... 3-3
3.2.3 Configuring a project for several HMI devices ... 3-5
3.2.4 Creating a project for use on different operating units ... 3-6
3.2.5 WinCC flexible integrated in SIMOTION and STEP7 ... 3-7

3.3 Multilingual configuration ... 3-8

3.4 Editing projects ... 3-10
3.4.1 Editing projects ... 3-10
3.4.2 Displaying projects ... 3-12
3.4.3 Working in the Project View ... 3-14
3.4.4 Working in the Object View ... 3-15
3.4.5 Migrating existing projects ... 3-16

3.5 Reusing project data ... 3-17
3.5.1 Using libraries ... 3-17
3.5.2 Using faceplates ... 3-18

3.6 Working with the cross-reference ... 3-18

3.7 Internal project find and replace feature ... 3-19

3.8 Basic principles on documentation in WinCC flexible ... 3-20

3.9 Debugging projects ... 3-20

3.10 Transferring projects ... 3-21
3.10.1 Basic Principles of the Transfer Operation ... 3-21
3.10.2 Back transfer of projects ... 3-22

4 Working with Tags ... 4-1

4.1 Basics ... 4-1
4.1.1 External tags ... 4-1
4.1.2 Internal Tags ... 4-2

4.2 Elements and basic settings ... 4-2
4.2.1 Tag editor ... 4-2
4.2.2 Basic Settings for Tags and Arrays ... 4-4

www.ElectricalPartManuals.com

4.3 Working with Tags ... 4-6
4.3.1 Properties of a Tag ... 4-6
4.3.2 Communication with the PLC using external tags ... 4-7
4.3.3 Tag limit values ... 4-8
4.3.4 Start value of a tag ... 4-9
4.3.5 Updating the Tag Value in Runtime ... 4-9
4.3.6 Logging tags ... 4-10
4.3.7 Linear scaling a tag ... 4-12
4.3.8 Indirect addressing of tags ... 4-13
4.4 Array basics ... 4-13
4.5 Cycle basics ... 4-15
4.6 Importing Tags ... 4-16
4.6.1 Importing Tags in WinCC flexible ... 4-16
4.6.2 Settings for the Tag Import ... 4-16
4.6.3 Format of the Connection Data for the Import ... 4-18
4.6.4 Format of the Tag Data for the Import ... 4-20
5 Creating Screens ... 5-1
5.1 Basics ... 5-1
5.1.1 Screen Basics ... 5-1
5.1.2 "Screens" Editor ... 5-3
5.1.3 Procedures ... 5-4
5.2 Configuring the navigation system ... 5-5
5.2.1 Navigating options ... 5-5
5.2.2 Graphic programming of the screen navigation system ... 5-5
5.2.3 Using the navigation control ... 5-7
5.3 Working with objects ... 5-8
5.3.1 Overview of Objects ... 5-8
5.3.2 Object Groups ... 5-12
5.4 Options of assigning dynamic update functions ... 5-12
5.5 Working with function keys ... 5-13
5.6 The Advantage of Layers ... 5-14
5.7 Object libraries ... 5-15
5.8 Working with faceplates ... 5-16
6 Creating an Alarm System ... 6-1
6.1 Basics ... 6-1
6.1.1 Visualization of process and system alarms ... 6-1
6.1.2 User-defined alarms ... 6-2
6.1.2.1 Available Alarm Procedures ... 6-2
6.1.2.2 Acknowledging Alarms ... 6-3
6.1.2.3 Alarm classes ... 6-3
6.1.3 System alarms ... 6-4
6.1.4 Displaying Alarms ... 6-5
6.1.4.1 Displaying Alarms on the HMI Device ... 6-5
6.1.4.2 Logging and reporting alarms ... 6-6
6.1.4.3 System Functions for Alarm Editing ... 6-6

www.ElectricalPartManuals.com

6.2 Elements and basic settings ... 6-8
6.2.1 Alarm Components and Properties ... 6-8
6.2.2 Editors for Configuring Alarms ... 6-9
6.2.2.1 Basic Principles of Editors ... 6-9
6.2.2.2 "Discrete alarms" editor ... 6-11
6.2.2.3 "Analog alarms" editor ... 6-12
6.2.2.4 "System alarms" editor ... 6-13
6.2.2.5 "Alarm classes" editor ... 6-14
6.2.2.6 "Alarm groups" editor ... 6-15
6.2.3 Basic Settings for the Alarm System ... 6-16

6.3 Working with alarms ... 6-17
6.3.1 Reporting alarms ... 6-17
6.3.2 Integrating alarms with the alarm numbering procedure ... 6-17

6.4 Alarm logging ... 6-19
6.4.1 Basic principles of alarm logging ... 6-19
6.4.2 Alarm logging ... 6-20
6.4.3 "Alarm logs" editor ... 6-20
6.4.4 Basic settings for alarm logs ... 6-21
6.4.5 Alarm logging ... 6-23
6.4.6 Displaying logged alarms on screens ... 6-24
6.4.7 Structure of a *.csv file with alarms ... 6-24
6.4.8 Accessing the ODBC log database directly ... 6-26

7 Working with a connection ... 7-1

7.1 Basics ... 7-1
7.1.1 Communication basics ... 7-1
7.1.2 Principles of communication ... 7-2

7.2 Elements and basic settings ... 7-3
7.2.1 Connections Editor ... 7-3
7.2.2 Parameters for connections ... 7-5
7.2.3 Area pointers for connections ... 7-6

8 Structure of a recipe management system ... 8-1

8.1 Basics ... 8-1
8.1.1 Basic principles of recipes ... 8-1
8.1.2 Structure of recipes ... 8-3
8.1.3 Structure of recipe data records ... 8-4
8.1.4 Configuration of recipes ... 8-5
8.1.5 Transfer of recipe data records ... 8-7

8.2 Elements and basic settings ... 8-8
8.2.1 "Recipes" editor ... 8-8
8.2.2 Recipe elements ... 8-9
8.2.3 Recipe data records ... 8-11
8.2.4 Recipe settings ... 8-12

8.3 Viewing and editing recipes in Runtime ... 8-13
8.3.1 Viewing and editing recipes in Runtime ... 8-13
8.3.2 Basic principles of the recipe view ... 8-15
8.3.3 Basic principles of the simple recipe view ... 8-15
8.3.4 Operator control elements of the recipe view ... 8-17
8.3.5 Behavior of the recipe view in Runtime ... 8-18
8.3.6 Configuration options for the recipe view ... 8-18
8.3.7 Basic principles of the recipe screen ... 8-20

www.ElectricalPartManuals.com

8.4 Scenarios ... 8-23
8.4.1 Scenario: Entering recipe data records in Runtime ... 8-23
8.4.2 Scenario: Manual production sequence ... 8-24
8.4.3 Scenario: Automatic production sequence ... 8-26

9 Logging and displaying tags ... 9-1

9.1 Basics ... 9-1
9.1.1 Basic principles for data logging ... 9-1
9.1.2 Trends ... 9-1
9.1.3 Data logging in WinCC flexible ... 9-3

9.2 Elements and basic settings ... 9-5
9.2.1 "Data Logs" editor ... 9-5
9.2.2 Basic settings for data logs ... 9-6

9.3 Logging tags ... 9-8

9.4 Outputting logged data ... 9-9
9.4.1 Outputting tag values in screens ... 9-9
9.4.2 The structure of a *.csv file with tags ... 9-9
9.4.3 Accessing the ODBC log database directly ... 9-11

10 Working with reports ... 10-1

10.1 Principles on the report system ... 10-1

10.2 Structure of reports ... 10-2

10.3 Elements and basic settings ... 10-3
10.3.1 Protocols ... 10-3
10.3.2 Using the toolbox view ... 10-5

10.4 Working with reports ... 10-6
10.4.1 Creating a report ... 10-6
10.4.2 Adapting the report properties ... 10-8
10.4.3 Objects for report creation ... 10-9
10.4.4 Use of report objects ... 10-10

10.5 Reporting alarms ... 10-11
10.5.1 Reporting alarms ... 10-11
10.5.2 Editing output parameters for an alarm report ... 10-12

10.6 Reporting recipes ... 10-15
10.6.1 Reporting recipes ... 10-15
10.6.2 Editing output parameters for a recipe report ... 10-16

10.7 Outputting a report ... 10-18

11 User administration ... 11-1

11.1 Field of application of the user administration ... 11-1

11.2 Structure of the user administration ... 11-2

11.3 Elements and basic settings ... 11-3
11.3.1 "Groups" user administration ... 11-3
11.3.2 User groups work area ... 11-4
11.3.3 "Users" user administration ... 11-5
11.3.4 Users work area ... 11-6

11.4 Working with the user administration ... 11-7
11.4.1 Users in Runtime ... 11-7
11.4.2 Access security ... 11-8

www.ElectricalPartManuals.com

12 System functions and runtime scripting 12-1

12.1 Basics 12-1
12.1.1 System functions and runtime scripting 12-1
12.1.2 System functions 12-2
12.1.3 Use of system functions 12-4
12.1.4 Scripts 12-4
12.1.5 Use of scripts 12-5

12.2 Working with function lists 12-6
12.2.1 Basic principles of the functions list 12-6
12.2.2 Properties of a function list 12-7

12.3 Elements and basic settings 12-8
12.3.1 Scripts 12-8
12.3.2 Properties of the "Script" editor 12-10

12.4 Creating scripts 12-13
12.4.1 Access to tags 12-13
12.4.2 Call up of scripts and system functions in the scripts 12-14
12.4.3 Access to objects 12-15
12.4.4 Synchronization of tags and objects 12-16

12.5 Debugging 12-16
12.5.1 Debugging Scripts 12-16
12.5.2 Integrating the debugger 12-17

12.6 Runtime behavior of functions in runtime 12-21
12.6.1 Completion of the function list in runtime 12-21
12.6.2 Processing of scripts in runtime 12-21
12.6.3 Delivery and return of values 12-22
12.6.4 Changing of object properties in runtime with VBS 12-23
12.6.5 HMI device dependent system functions in the script 12-23

13 Structure of Multilingual Projects 13-1

13.1 Languages in WinCC flexible 13-1
13.1.1 Working with multiple languages 13-1
13.1.2 WinCC flexible terminology 13-2

13.2 Language Settings 13-4
13.2.1 Language settings in the operating system 13-4
13.2.2 Operating system settings for Asian languages 13-5
13.2.3 "Project Languages" editor 13-5

13.3 Creating a project in multiple languages 13-7
13.3.1 Creating a project in multiple languages 13-7
13.3.2 Specific features of Asian and Eastern languages in the engineering system 13-8
13.3.3 Translating project texts in the editor 13-9
13.3.4 "Project texts" editor 13-10
13.3.5 Exchanging texts with translators 13-11

13.4 Working with dictionaries 13-12
13.4.1 Working with dictionaries 13-12
13.4.2 "System dictionary" editor 13-13
13.4.3 "User dictionary" editor 13-14

13.5 Use of language-dependent graphics 13-15
13.5.1 Use of language-dependent graphics 13-15
13.5.2 "Graphics" editor 13-15

www.ElectricalPartManuals.com

13.6 Languages in Runtime 13-17
13.6.1 Languages in Runtime 13-17
13.6.2 Configuring language switching 13-17
13.6.3 Specific features of Asian and Eastern languages in Runtime 13-18

14 Project documentation 14-1

14.1 Basics 14-1
14.1.1 Project documentation 14-1
14.1.2 Structure of a layout 14-2

14.2 Using layouts 14-3
14.2.1 Using layouts 14-3
14.2.2 Editing a layout for the project documentation 14-4

14.3 Creating a project report 14-6
14.3.1 Selecting the data for a project report 14-6
14.3.2 Outputting of data of selected objects 14-6
14.3.3 Selecting Objects for the Project Documentation 14-7

15 Planning jobs 15-1

15.1 Field of application of the scheduler 15-1

15.2 Working with jobs and events 15-2

15.3 Elements 15-4
15.3.1 Scheduler 15-4
15.3.2 Work area of the "Scheduler" editor 15-5

16 Managing project versions 16-1

16.1 Applications for project versioning 16-1

16.2 Basics of version management 16-2

16.3 Trunk 16-3

16.4 Branch 16-4

16.5 Elements 16-5
16.5.1 Version management 16-5
16.5.2 Version Management Work Area 16-6
16.5.3 Property view 16-7

16.6 Working with project versions 16-8
16.6.1 Comparing versions 16-8

17 Logging changes 17-1

17.1 Applications for the change log 17-1

17.2 Change log of a project 17-2

17.3 Change log of a project session 17-3

17.4 Change log of a project under version management 17-5

17.5 Elements 17-6
17.5.1 Change log 17-6
17.5.2 Change log work area 17-7

www.ElectricalPartManuals.com

18 Transfer 18-1

18.1 Basics 18-1
18.1.1 Basic Principles of the Transfer Operation 18-1
18.1.2 Transfer settings 18-2
18.1.3 Back transfer of projects 18-5

18.2 Managing Files on the HMI Device 18-6
18.2.1 ProSave 18-6
18.2.2 Backup of HMI data 18-7
18.2.3 Updating the operating system 18-9
18.2.4 Transferring 18-10
18.2.5 Installation of options 18-10

19 Integration of WinCC flexible in STEP 7 19-1

19.1 Basics 19-1
19.1.1 Basic principles of integration in STEP 7 19-1
19.1.2 Working with the SIMATIC Manager 19-2
19.1.3 Working with HW Config 19-3
19.1.4 Configuring connections 19-3
19.1.5 Working with objects 19-5

19.2 Configuring communication settings 19-6
19.2.1 Configuring communication settings with routing 19-6
19.2.2 Project transfer via S7 routing 19-8

19.3 Tag configuration 19-12
19.3.1 Configuring tags with the Tag editor 19-12
19.3.2 Connecting a tag via the application point 19-13

19.4 Configuring alarms 19-14
19.4.1 Integrating alarms with the alarm numbering procedure 19-14

20 Appendix 20-1

20.1 Open Source Software 20-1

20.2 Performance features 20-1
20.2.1 General Technical Data 20-1
20.2.1.1 Released operating systems 20-1
20.2.1.2 Released databases 20-2
20.2.1.3 Further software versions supported 20-2
20.2.1.4 Recommended printers 20-2
20.2.1.5 Legal characters 20-2
20.2.1.6 Memory requirement of recipes 20-3
20.2.2 System limits 20-4
20.2.2.1 System limits 20-4

www . ElectricalPartManuals . com

1 Introduction to WinCC flexible

1.1 Introduction to SIMATIC HMI

Introduction

Maximum transparency is essential for the operator who works in an environment where processes are becoming more complex, and requirements for machine and plant functionality are increasing. The Human Machine Interface (HMI) provides this transparency.

The HMI system represents the interface between man (operator) and process (machine/plant). The controller is the actual unit which controls the process. Hence, there is an interface between the operator and WinCC flexible (at the HMI device) and an interface between WinCC flexible and the controller. An HMI system assumes the following tasks:

- Process visualization

  The process is visualized on the HMI device. The screen on the HMI device is dynamically updated. This is based on process transitions.

- Operator control of the process

  The operator can control the process by means of the GUI. For example, the operator can preset reference values for the controls or start a motor.

- Displaying alarms

  Critical process states automatically trigger an alarm, for example, when the setpoint value is exceeded.

- Archiving process values and alarms

  The HMI system can log alarms and process values. This feature allows you to log process sequences and to retrieve previous production data.

- Process values and alarms logging

  The HMI system can output alarms and process value reports. This allows you to print out production data at the end of a shift, for example.

- Process and machine parameter management

  The HMI system can store the parameters of processes and machines in recipes. For example, you can download these parameters in one pass from the HMI device to the PLC to change over the product version for production.

www . ElectricalPartManuals . com

SIMATIC HMI

SIMATIC HMI offers a totally integrated, single-source system for manifold operator control and monitoring tasks. With SIMATIC HMI, you always master the process and always keep your machinery and units running.

Examples of simple SIMATIC HMI systems are small touch panels for use at machine level.

SIMATIC HMI systems used for controlling and monitoring production plants represent the upper end of the performance spectrum. These include high-performance client/server systems.

Integration of SIMATIC WinCC flexible

WinCC flexible is the HMI software for future-proof machine-oriented automation concepts with comfortable and highly efficient engineering. WinCC flexible combines the following benefits:

- Straightforward handling
- Transparency
- Flexibility

1.2 WinCC flexible system overview

1.2.1 Components of WinCC flexible

WinCC flexible Engineering System

The WinCC flexible Engineering System is the software for handling all your essential configuring tasks. The WinCC flexible edition determines which HMI devices in the SIMATIC HMI spectrum can be configured.

WinCC flexible Runtime

WinCC flexible Runtime is your software for process visualization. You execute the project in process mode in Runtime.

WinCC flexible options

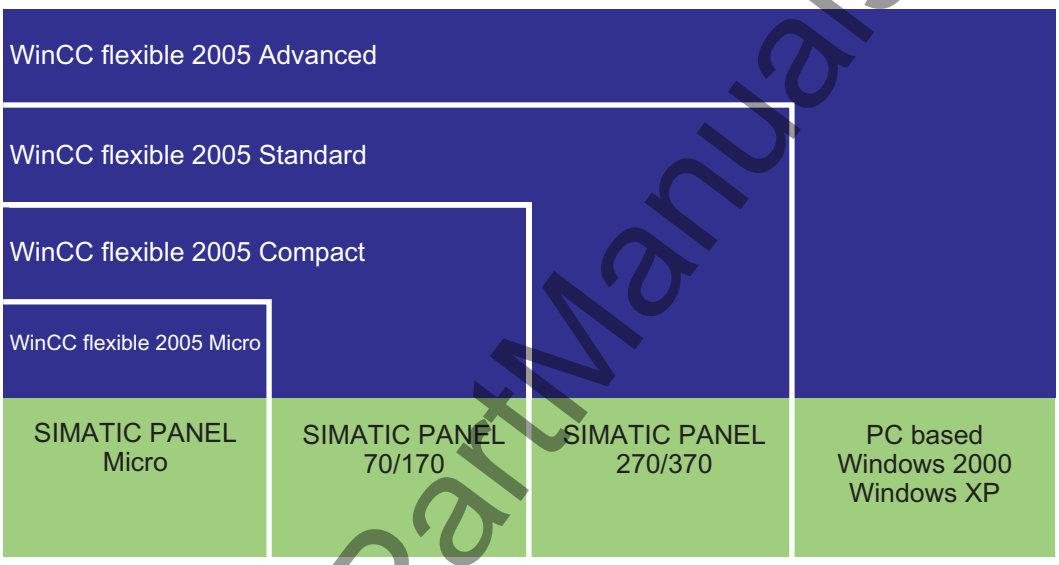
The WinCC flexible options allow you to expand the standard functionality of WinCC flexible. A separate license is needed for each option.

www.ElectricalPartManuals.com

1.2.2 WinCC flexible Engineering System

Introduction

WinCC flexible is an engineering system for all your configuring tasks. WinCC flexible has a modular design. With each higher edition you expand the spectrum of supported devices and WinCC flexible functionality. You can always migrate to a higher edition by means of a powerpack.

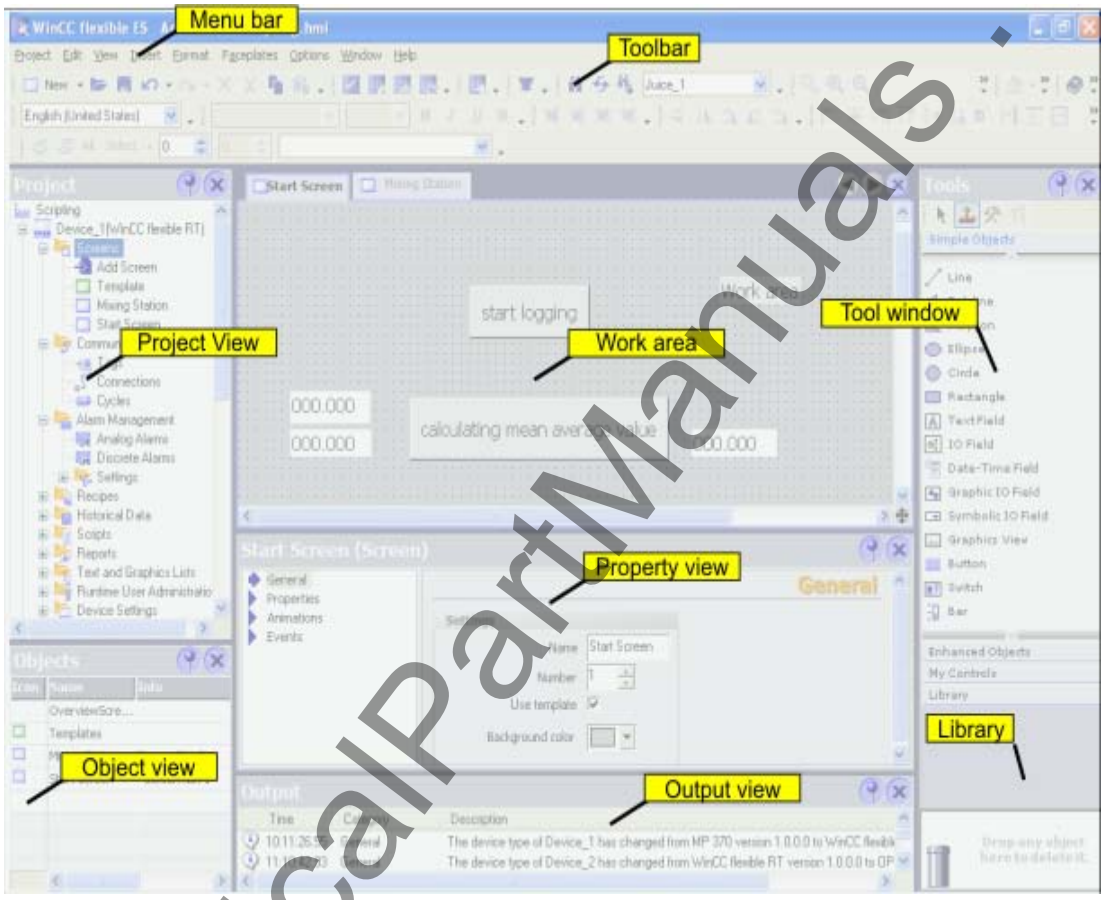


WinCC flexible covers a performance spectrum ranging from Micro Panels to simple PC visualization. The WinCC flexible functionality is thus comparable with that of products of the ProTool family and the TP Designer. You can integrate your existing ProTool projects into WinCC flexible.

www . ElectricalPartManuals . com

Principles

The WinCC flexible workbench opens on the screen of your programming computer when you create a new or open an existing project in WinCC flexible. The project structure is visualized and the project managed in the "Project View."



WinCC flexible provides a special editor for each configuring task. For example, you configure the GUI of an HMI device in the "Screens" editor. Or you can use the "Discrete Alarms" editor to configure alarms.

All project configuration data related to a project is stored in the project database.

Migration to another WinCC flexible edition

Your current WinCC flexible edition determines which HMI devices you can configure. To configure an HMI device which is not supported in your current WinCC flexible edition, you can migrate to another WinCC flexible edition. All existing functions remain available.

As of the WinCC flexible Compact edition, you can use the powerpack to upgrade the WinCC flexible edition.

www.ElectricalPartManuals.com

1.2.3 WinCC flexible Runtime

Principle

In runtime, the operator can control and monitor the process. This involves in particular the following tasks:

- Communication with the automation systems.
- On-screen visualization of images
- Operating the process, for example, by setting setpoint values or opening and closing valves.
- Archiving of current runtime data, e.g. process values and alarm events.

Performance spectrum of WinCC flexible Runtime

WinCC flexible Runtime supports a certain number of process variables (powertags) which is determined by your license:

- WinCC flexible Runtime 128: Supports 128 process variables
- WinCC flexible Runtime 512: Supports 512 process variables
- WinCC flexible Runtime 2048: Supports 2048 process variables

You can increase the number of process variables with a Powerpack.

1.2.4 Available options

Principle

The following options are available for WinCC flexible Runtime. The options depend on the target system used.

SIMATIC WinCC flexible RT options	Function	SIMATIC panels	SIMATIC panel PCs
Logs	Archiving functionality in runtime	From Panel 270	x
Recipes	Recipe functionality in runtime	-- (standard feature)	x
Sm@rtAccess	Remote control and remote monitoring as well as communication between different SIMATIC HMI systems.	From Panel 270	x
Sm@rtService	Remote maintenance and servicing of machines/plants via the Internet/Intranet.	From Panel 270	x
OPC server	Use of an HMI device as OPC server	Multi panel	x
ProAgent	Process diagnostics during runtime	From Panel 270	x
Audit	Reporting interactions according to FDA	From Panel 270	x

www . ElectricalPartManuals . com

Options available for the WinCC flexible Engineering System:

SIMATIC WinCC flexible options	Function	Availability
ChangeControl	Version management and modification tracking	from WinCC flexible Advanced

1.2.5 Licensing

1.2.5.1 Licenses and License Key

Principle

All WinCC flexible editions require a license. Certain WinCC flexible editions require a license to be used without restriction.

- License

  You receive your license on paper. The license entitles you to install and use your purchased WinCC flexible edition on a computer. More information about rights of use is available in the electronic catalog CA 01.

- License Key

  The license key is delivered on a separate copy-protected disk. During the installtion, you will be prompted to insert the license key disk.

Licenses for the WinCC flexible Engineering System

Which licensing model is going to be used depends on the WinCC flexible edition:

- WinCC flexible Micro: License agreement
- WinCC flexible Compact: License Contract and License Key Disk
- WinCC flexible Standard: License Contract and License Key Disk
- WinCC flexible Advanced: License Contract and License Key Disk

www . ElectricalPartManuals . com

Licenses for WinCC flexible Runtime

To license WinCC flexible Runtime, use both the license contract as well as the License Key Disk. WinCC flexible Runtime licenses support a different amount of variables:

- WinCC flexible Runtime 128: Supports 128 process variables
- WinCC flexible Runtime 512: Supports 512 process variables
- WinCC flexible Runtime 2048: Supports 2048 process variables

Licenses for the options packages

You receive a license and licence key disk for each option. WinCC flexible Engineering System already contains the functionality of the Runtime options package. You do not need a license on your programming computer for configuring the functionality of a runtime option.

1.2.5.2 WinCC flexible without licensing

Principle

Without a license, WinCC flexible runs in demo mode only and the software operability are restricted. Alarms which need acknowledging appear regularly on the screen both in WinCC flexible Engineering System and in WinCC flexible Runtime.

Ordering a new authorization disk

If the License Key Disk is damaged or if you have misplaced it, contact customer support.

www . ElectricalPartManuals . com

1.3 Automation concepts

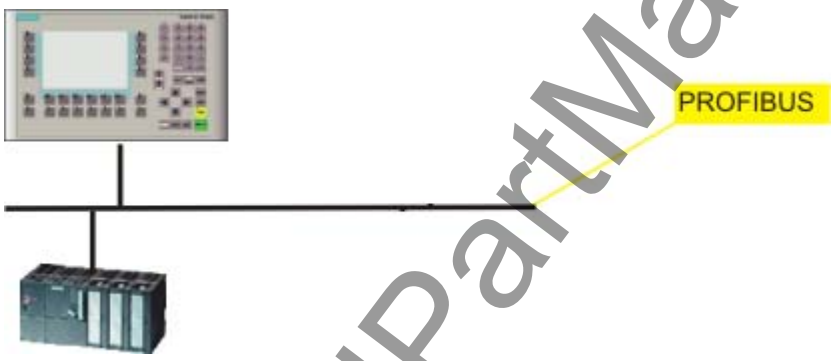
1.3.1 Automation concepts with WinCC flexible

Introduction

WinCC flexible supports the configuration of many different automation concepts. The following automation concepts can be implemented by default using WinCC flexible.

Control with one HMI device

An HMI device which is directly connected to the controller via the process bus is referred to as a single-user system.



Single-user systems are generally used near production, but can also be deployed to operate and monitor independent part processes or system sections.

Controller with several HMI devices

Several HMI devices are connected to one or more controllers via a process bus (e.g. PROFIBUS or Ethernet).

Such systems are deployed, for example, in a production line to operate the plant from several points.

www.ElectricalPartManuals.com

HMI System with centralized functions

An HMI system is connected to a PC via Ethernet. The upstream PC assumes central functions, e.g. recipe management. The necessary recipe data records are provided by the subordinate HMI system.

www . ElectricalPartManuals . com

Support for Mobile Units

Mobile units are mainly implemented in large production plants, long production lines or in conveyor technology, but can also be implemented in systems in which direct visual contact with the process is necessary. The machines to be operated are equipped with several interfaces to which the Mobile Panel 170, for example, can be connected.

The operator or service technician can thus work directly on site. This enables an accurate setting up and positioning, e.g. during the startup phase. In the case of servicing, mobile units ensure shorter downtimes.

1.3.2 Remote access to HMI devices

Introduction

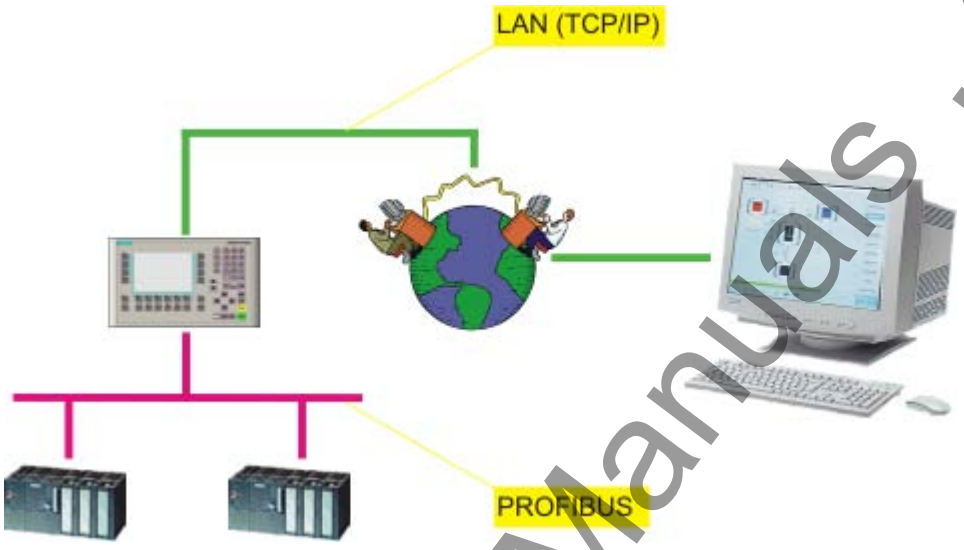
By using the Sm@rtService option, it is possible to connect to an HMI device from a workstation via a network (Internet, LAN).

Example: A medium-sized production company has a service contract with an external service company. When servicing is required, the service technician responsible can remotely access the HMI device and display its user interface directly on his workstation. In this way, updated projects can be transferred more quickly which, in turn, reduces machine downtime.

www . ElectricalPartManuals . com

Application Possibilities

The option "Sm@rtService" is required for implementation.



Remote access via a network can be used for the following applications:

- Remote operation and monitoring

  An HMI device can be operated and running processes monitored from your own workstation.

- Remote administration

  A project can be transferred from a workstation to an HMI device. In this way, projects can be updated from a central point.

- Remote diagnostics

  Each Panel provides HTML pages for accessing the installed software, version or system alarms using a Web browser.

1.3.3 Automatic alarm dispatch

Introduction

A machine which fails due to a fault costs money. An alarm that reaches the service technician in a timely manner helps to minimize unplanned downtime.

Example: Contamination in a feed line reduces the flow of coolant. When the value drops below the configured limit value, the HMI device displays a warning. The warning is also dispatched as an e-mail to the service technician responsible.

www . ElectricalPartManuals . com

Principle

The "Sm@rtAccess" option is required for implementation. In order to send alarms as e-mails, the HMI system must have access to an e-mail server.

The e-mail client sends the alarms via intranet or Internet. The automatic alarm dispatch ensures that all the people involved (e.g. shift foreman and sales manager) are informed of the status of the machine in good time.

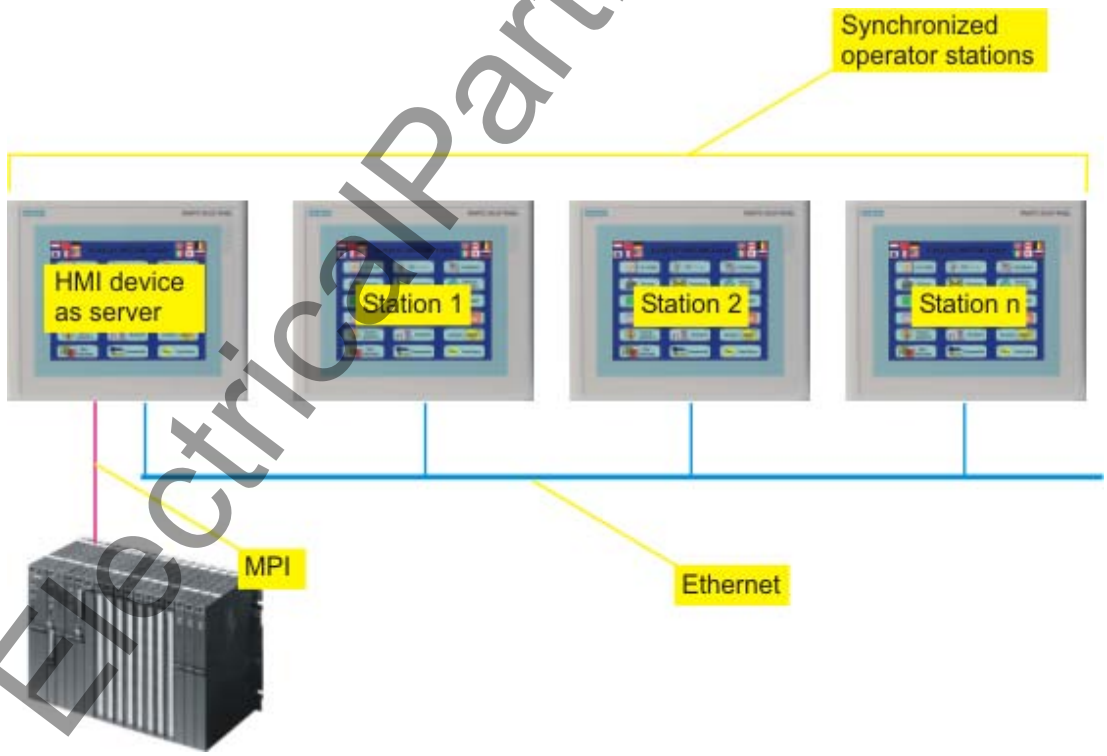
1.3.4 Distributed HMI

Introduction

Distributed HMI enables the operation of a machine from several synchronized operating stations. All the operating stations display the same process screen. The operating authorization is transferred intelligently.

Principle

The "Sm@rtAccess" option is required for implementation.



Only one HMI device contains the configuration data and functions as the server. The server can be controlled from the other operator devices. All the HMI devices display the same screens.

www.ElectricalPartManuals.com

1.4 Configuration concepts

1.4.1 Configuration Support

Introduction

WinCC flexible is used to configure user interfaces to operate and monitor machines and plants. WinCC flexible provides support for the configuration tasks in the form of solution-oriented concepts. For example, this could concern the processing of bulk data, automatic transfers or even the intelligent configuration of movement paths.

Engineering support

Efficient configuration saves time and costs. WinCC flexible supports the following for this:

- Target system dependent configuration

  During configuration, only those functions supported by the selected target system are displayed.
- Engineering independent of the PLC used

  If you use a project for different or several target systems, you only switch over the HMI device in the project. Functionalities of the selected HMI device which are not supported are not displayed.
- Central modification of referenced objects

  Modifications made at a central station are applied to the entire project.
- Use

  The reuse of configuration objects simplifies configuration and reduces overall costs.
- Bulk data processing

  Create an action, for example, with several tags having the same values or chronologically ascending addresses.
- Graphic Configuration of the Screen Navigation

  Create a screen hierarchy from the graphic overview of the configured screens. The required objects for the screen navigation are generated automatically.
- Configuring movement paths

  Configure the movement of objects clearly in the process screen.
- Totally Integrated Automation

  Benefit from the smooth integration of WinCC flexible in the configuration user interface of SIMATIC STEP 7 and SIMOTION SCOUT.

User customization of the programming user interface

The WinCC flexible Workbench can be user customized by moving or hiding windows and toolbars.

www . ElectricalPartManuals . com

1.4.2 Scalable Configuration Tools

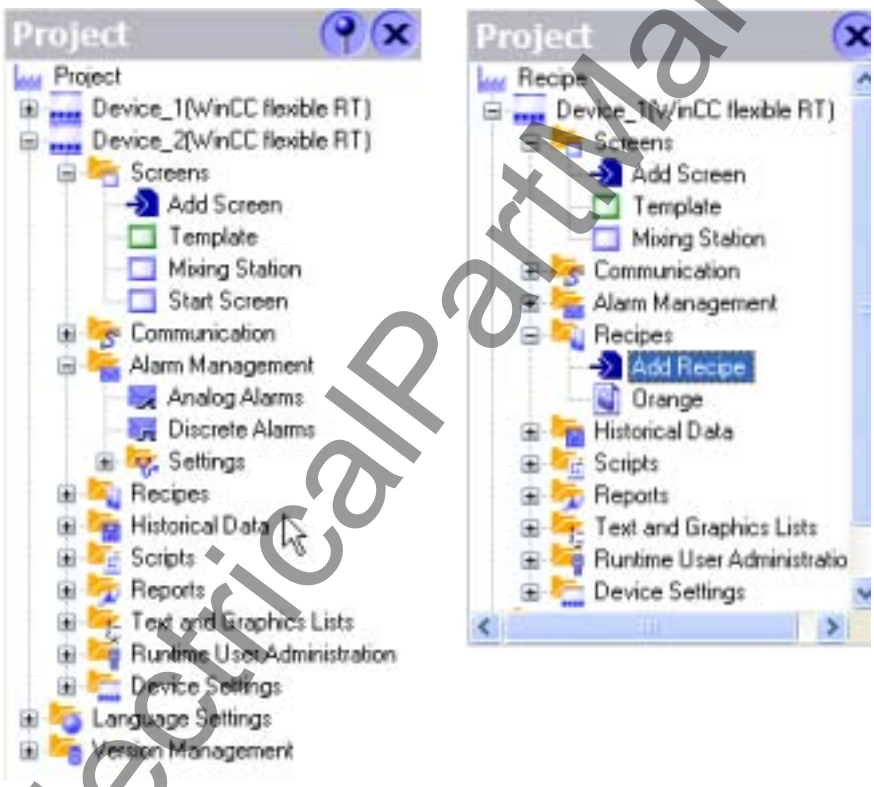
Introduction

If WinCC flexible is used to edit projects for different HMI devices, the functional scope is adapted to the HMI device during configuration. Different functionality is available according to the HMI device.

Custom HMI Device Functions

Custom device functions ensure efficient configuration. You only need to configure those functions which are supported by the specific HMI device.

The editors displayed in the Project View can be used, for example, to quickly detect which functions are supported by the HMI device selected.



You can use a project for different target systems. If the target system is changed, only the view of the project data is modified. When the target system changes, configured objects are not deleted; they are only hidden if some features are not supported by the target system.

www . ElectricalPartManuals . com

Customized setup of the configuration user interface

WinCC flexible allows you to customize the position and reaction of windows and toolbars. This allows you to configure the work environment to meet your special requirements.

The configuration of the WinCC flexible workbench is linked to the user logged on in Microsoft Windows. On saving the project, the positions and behavior of windows and toolbars are automatically saved with it.

When opened again, the positions and behavior of windows and toolbars are identical to when the project was last saved. When the working environment opens, it is identical to the configuration when last closed. This is also the case when a project edited by a different project planner is opened.

1.4.3 PLC-independent configuration

Introduction

WinCC flexible supports you in creating configurations independent on the destination system.

Example: A machine has three operating stations. One HMI device with average performance connected to one of these operating stations is sufficient.

Principle

The project need not be recreated for this HMI device. Simply switch the HMI device in the project instead.

Functions not supported by the HMI device are hidden.

The different HMI devices should not differ too much in their resolution and functionality .

1.4.4 Use

Introduction

Reusing configuration objects facilitates configuration work. Centralized editing saves a considerable portion of the configuration work when an object is changed.

Faceplates

Simple screen objects can be combined as faceplates to form complex objects. For each faceplate, it is possible to define which properties of the screen objects can be changed. By reusing a faceplate stored in the library, modifications can be executed throughout the entire project from a central point.

www . ElectricalPartManuals . com

Libraries

All configuration objects can be stored centrally in libraries. In addition, numerous preconfigured screen objects are supplied which can be used to design process screens appropriately.

Text Libraries

Text libraries can be used to store all configuration texts in several languages. If a project is configured in several languages, the texts can be translated automatically.

1.4.5 Intelligent tools

1.4.5.1 Bulk data processing

Introduction

Bulk data management provides support in the simultaneous creation and editing of several objects. Configuration is more efficient, saving time and costs.

Example: A part of the tag inventory is assumed from an old project but the inventory has the wrong tag type. Using WinCC flexible, the tag type can be modified for all tags in one working step.

Principle

The advantages of bulk data processing can be used when creating and editing specific objects (e.g. tags).

- Automatic address assignment

  If several tags, created with process linking, are stored successively in the controller memory, the address area can be increased automatically for each tag.

- Multiple modification

  Identical modifications for several tags can be executed in one step, e.g. changing the tag type or the controller.

www . ElectricalPartManuals . com

1.4.5.2 Configuring movement paths

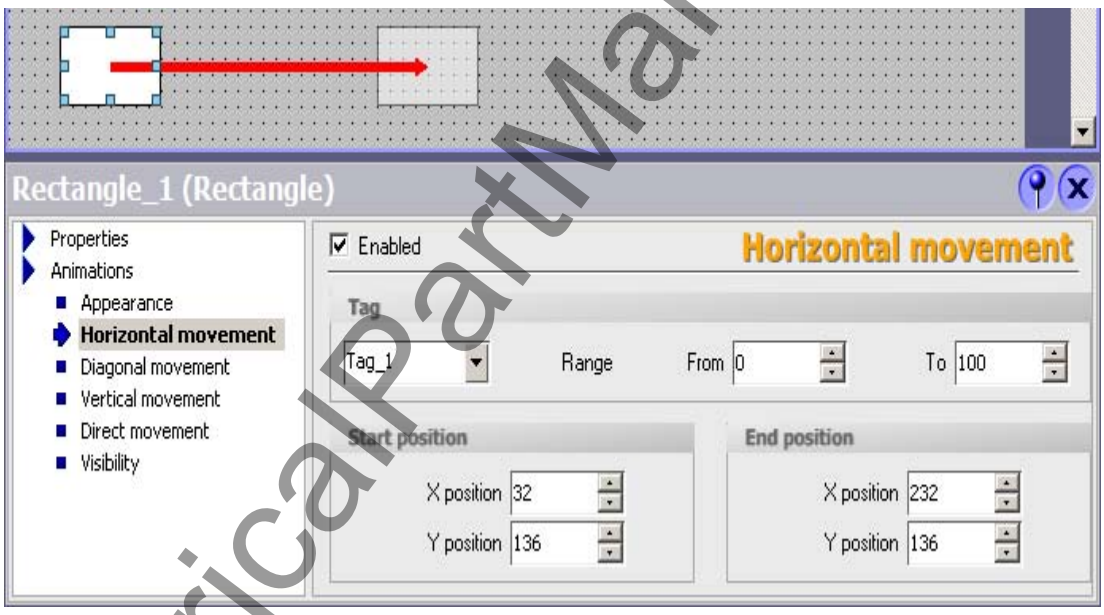
Introduction

Process sequences involving object movement can be clearly displayed on the HMI device, e.g. the transport of a product on a conveyor belt.

Movement paths simplify the configuration of movements of objects in the process screen. The movement process is represented on the screen by a diagram.

Principle

The movement path for an object is defined in the process screen. The movement path is composed of the starting point and finishing point. The movement path is assigned a tag. The tag value defines the relative position of the object on the movement path in runtime.



www . ElectricalPartManuals . com

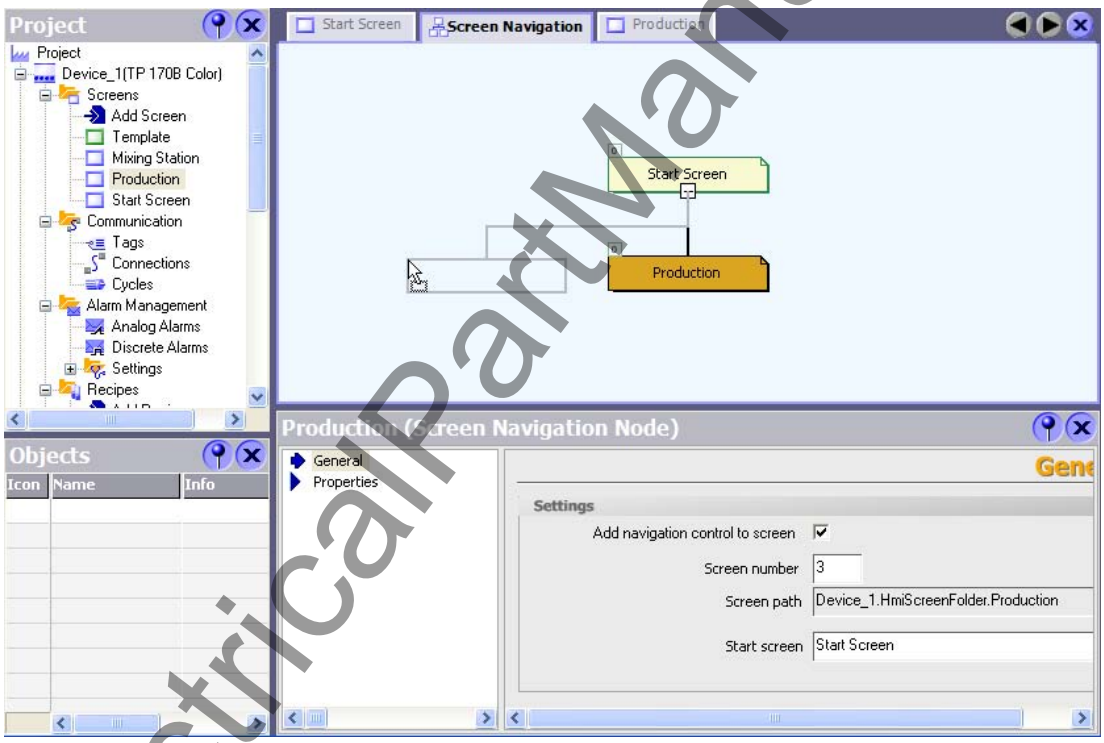
1.4.5.3 Graphic Configuration of the Screen Navigation

Introduction

Screen navigation means configured hierarchy for process screens. Using the screen navigation, a fixed navigation structure is defined for a project. Operators can use the navigation control in runtime to navigate between the various screens of the structure.

Principle

Use the Screen Navigation editor to place screens in the required location in the screen hierarchy using drag-and-drop. You can also create a direct link between screens that are not integrated in the hierarchy. The navigation buttons can be pasted in the process screen.



The creation of a navigation structure offers the following advantages:

- Overview of the navigation structure throughout the entire project.
- Quick creation of direct links between process screens.
- Automatic creation of the basic screen navigation.

www . ElectricalPartManuals . com

1.4.6 Totally Integrated Automation

Introduction

A full automation solution not only involves an HMI system such as WinCC flexible but additional components, e.g. controller, process bus and periphery.

A particularly sophisticated integration is provided by WinCC flexible with components from the SIMATIC product range and the SIMOTION product range.

- Consistent configuration and programming
- Consistent data retention
- Consistent communication

Integration in SIMATIC STEP 7

Process tags provide the link for communication between the controller and HMI system. Without the advantage of the Totally Integrated Automation, each tag would have to be defined twice: once for the controller and once for the HM system.

The integration of SIMATIC STEP 7 in the configuration user interface leads to a lower error frequency and reduced configuration work. During the configuration, direct access is made to the STEP7 icon table and the communication settings.

- The STEP 7 symbol table contains database definitions (e.g. addresses and data types) defined during the creation of the control program.
- The communication settings contain the bus addresses and controller protocols. The communication settings are made in NetPro.

Integration in SIMOTION SCOUT

The integration of SIMOTION SCOUT in WinCC flexible not only provides the advantages of the integration of SIMATIC STEP 7 but also the full integration in the SIMOTION-SCOUT user interface.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

WinCC flexible Engineering System 2

2.1 Basic Principles on the Programming Interface

Principle

WinCC flexible is the HMI software for future-proof machine-oriented automation concepts with comfortable and highly efficient engineering.

You can access all functions supported by the selected HMI device. To start WinCC flexible, either click the desktop icon on the programming device or select it from the Windows Start menu.

WinCC flexible only allows one project to be open at any time. You can work simultaneously on several projects by opening WinCC flexible as many times as necessary.

Note

WinCC also allows you to configure several HMI devices in the same project.

2.2 WinCC flexible user interface

2.2.1 WinCC flexible User Interface Elements

Introduction

The WinCC flexible work environment consists of several elements. Some of the elements are linked to specific editors which means they are not visible unless the corresponding editor is active.

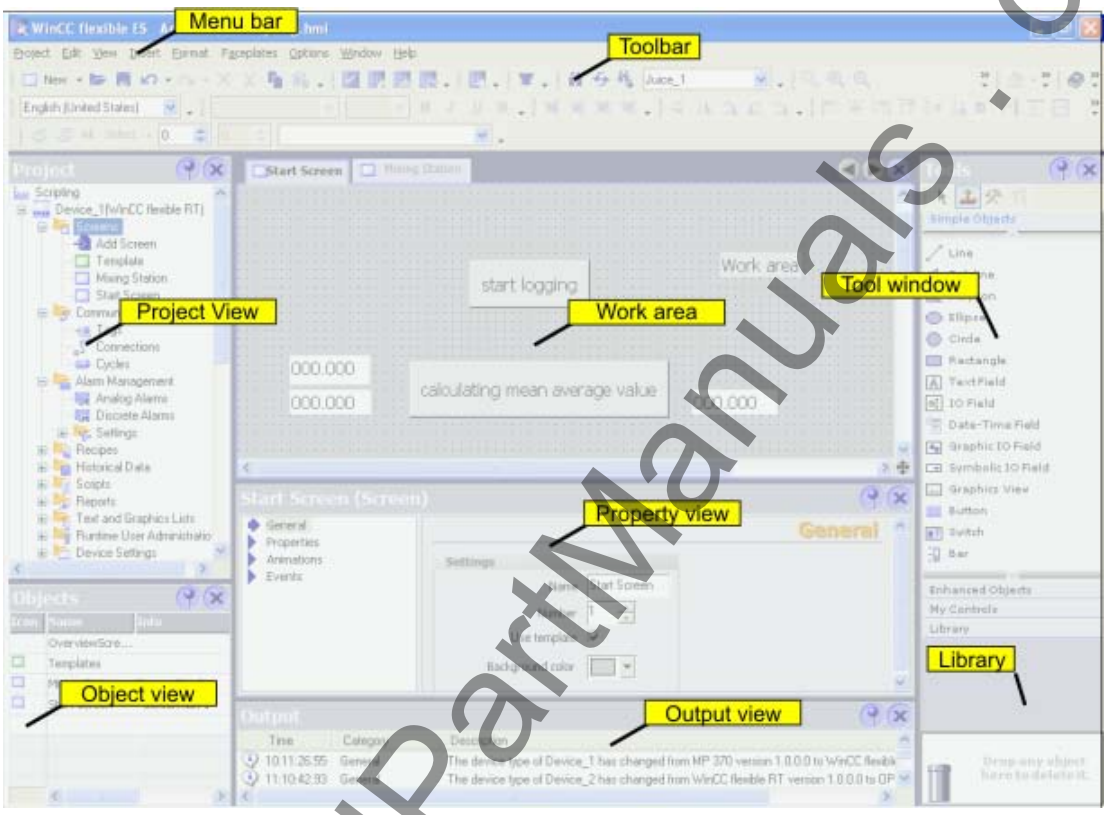
Note

Set the configuration computer operating system to "Small Fonts" while working with WinCC flexible.

www.ElectricalPartManuals.com

Elements of WinCC flexible

WinCC flexible consists of the following elements:



Menus and Toolbars

You can access all the functions provided by WinCC flexible by means of its menus and toolbars. When the mouse pointer is moved over a function, a ToolTip appears.

Work area

Project objects are edited in the work area. All WinCC flexible elements are arranged on the borders of the work area. With the exception of the work area, you can organize, configure and, for example, move or hide any of the elements to suit your individual requirements.

Project View

All component parts and editors available in a project appear in a tree structure in the Project View. Folders are provided as sub-elements of each editor in which you can save objects in a structured way. In addition, direct access to the configured objects is available for screens, recipes, scripts, protocols and user dictionaries. In the project windows you have access to the device settings of the HMI device, the language settings and the version management.

www.ElectricalPartManuals.com

Property view

The Property View is used to edit object properties, e.g. the color of screen objects. The property view is only available in specific editors.

Toolbox

The toolbox contains a selection of objects which you can add to your screens, e.g. image objects or operator control elements. In addition, the toolbox also provides libraries containing object templates and collections of faceplates.

Library

The "Library" is an element of the Toolbox view. The "Library" provides access to screen object templates. You can always add screen objects and thus increase programming efficiency either by multiple use or reuse of object templates. The library is your central database for storing frequently used objects, such as screen objects and tags.

Output View

The output window displays system alarms generated, for example, in a project test run.

Object view

The "Object View" shows all elements of the area selected from the "Project View".

Note

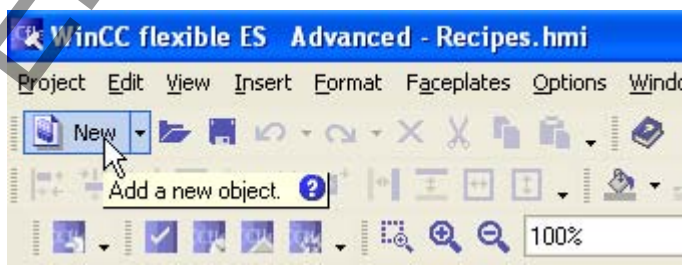
With the exception of the work area, you can show or hide all windows in the "View" menu.

2.2.2 Menus and Toolbars

Introduction

The menus and toolbars provide access to all functions you need to configure your HMI device. When the corresponding editor is activated, menu commands and toolbars specific to that editor appear.

When the mouse pointer is moved over a command, the corresponding ToolTip appears.



www.ElectricalPartManuals.com

Positioning the Toolbars

Menus and toolbars are, as a standard, positioned at the top edge of the screen when creating a new project. The position of menus and toolbars is determined by the user who is logged on in Windows. If the toolbars are moved using the mouse, they revert back to their last 'Exit' position when WinCC flexible is restarted.

Menus

Menus available in WinCC flexible:

Menu	Short description
"Project"	Contains commands for project management.
"Edit"	Contains commands for clipboard and search functions.
"View"	Contains commands for opening / closing elements, and for zoom / layer settings. To reopen a closed element, select the "View" menu.
"Paste"	Contains commands for pasting new objects
"Format"	Contains commands for organizing and formatting screen objects.
"Faceplates"	Contains commands for creating and editing faceplates.
"Tools"	Contains commands for changing the user interface language and configuring the basic settings in WinCC flexible, for example.
"Script"	Contains commands for the synchronization and syntax check of scripts.
"Window"	Contains commands for managing multiple windows in the work area, e.g. for changing to other windows.
"Help"	Contains commands for calling help functions.

The availability of the menus and the scope of their commands depend on the respective editor which is used.

Toolbars

The toolbars provide quick access to important, frequently used functions. The following toolbar configuration options are available:

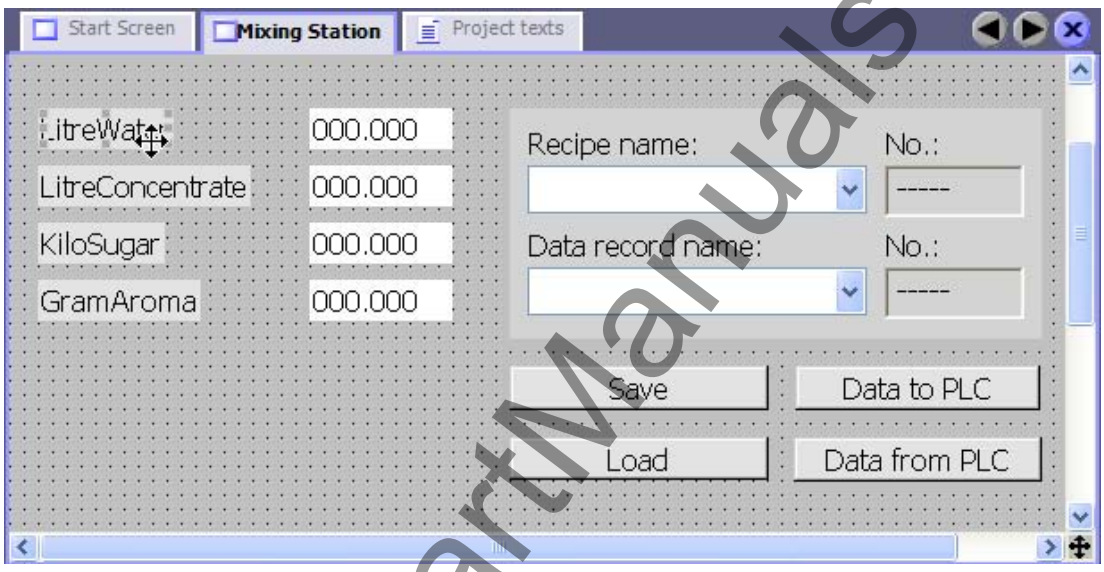
- Adding and removing buttons
- Changing the position

www . ElectricalPartManuals . com

2.2.3 Work area

Introduction

The work area is used to edit project data either in table format, e.g. the tags, or in graphic format, e.g. a process screen.



Description

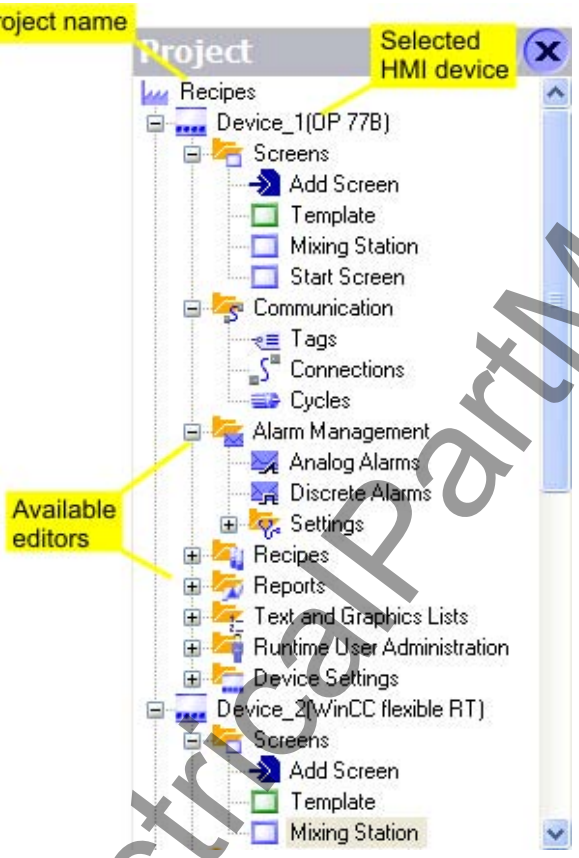
Each editor is opened in a separate tab control on the work area. In the case of graphic editors, each element is displayed on a separate tab control. Only one tab is active when several editors are open simultaneously. To move to another editor, click the corresponding tab. You can open up to 20 editors in parallel.

www . ElectricalPartManuals . com

2.2.4 Project View

Introduction

The project view is the central control point for project editing. All component parts and editors available in a project appear in a tree structure in the Project View. Each editor is assigned a symbol which you can use to identify the corresponding objects. Only those elements which are supported by the selected HMI device are displayed in the project window. In the project windows you have access to the device settings of the HMI device, the language settings and the version management.



Description

The Project View displays the project structure hierarchically:

- project
- HMI devices
- Folder
- Objects

The project view is used to create and open objects for editing. You can organize your project objects in folders to create a structure. Handling the Project View is similar to handling Windows Explorer. Context menus, which consist of the most important commands, are available for all objects.

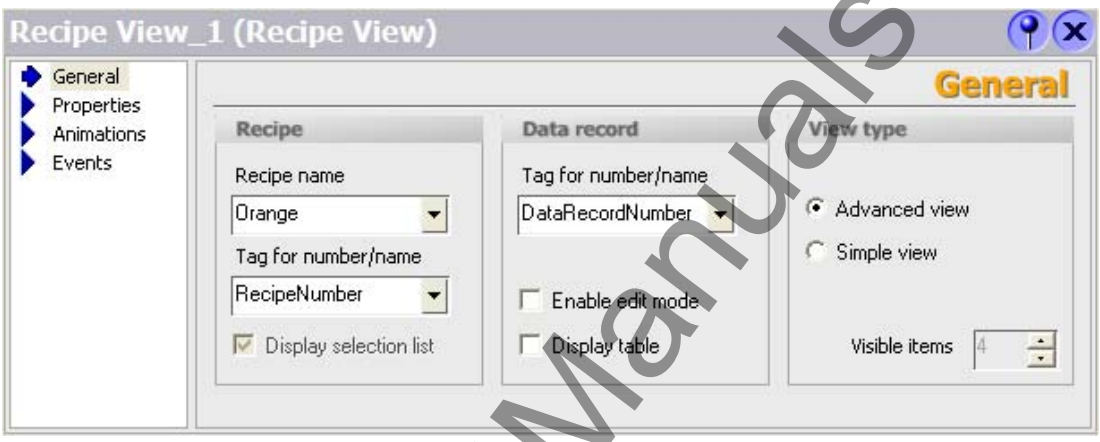
Elements of graphic editors are displayed in the Project View and Object View. Elements of "tabular editors" are shown only in the Object View.

www.ElectricalPartManuals.com

2.2.5 Property view

Introduction

The Property View is used to edit the properties of an object selected from the work area. The content of the property view is based on the selected object.



Description

The "Property View" shows the properties of the selected object organized in categories. The changed values take effect directly after exiting from the input field.

Invalid entries are highlighted with a colored background. A ToolTip will appear to help you correct the entry.

Example

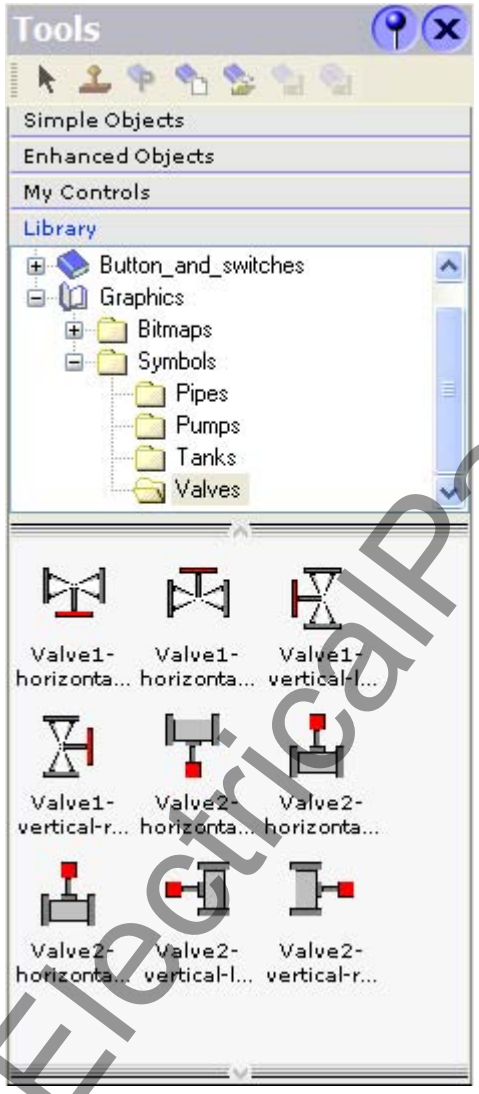
The object property "height" is logically linked to a "Byte" variable. This tag type has a range of values from 0 to 255. When you enter a value of "300" in the "Height" input box of the "Property View", the value is highlighted with a colored background when you exit the box.

www . ElectricalPartManuals . com

2.2.6 Library

Introduction

The "Library" is an element of the Toolbox view. The library is your central database for storing frequently required objects. You need to configure the object stored in the library once only. You can then reuse it as many times as you like. You can always add screen objects and thus increase programming efficiency either by multiple use or reuse of object templates.



www . ElectricalPartManuals . com

Description

WinCC flexible distinguishes between global and project libraries:

- Shared library

  The global library is not saved in the project database. It is written to a file. The file is saved by default in the installation directory of WinCC flexible The global library is available for all projects.

- Project library

  The project library is stored with the project data in the database and is available only in the project for which it was created.

You can create folders in both libraries to generate a structure for the objects they contain. Furthermore, you can always copy elements from a project library to the global library.

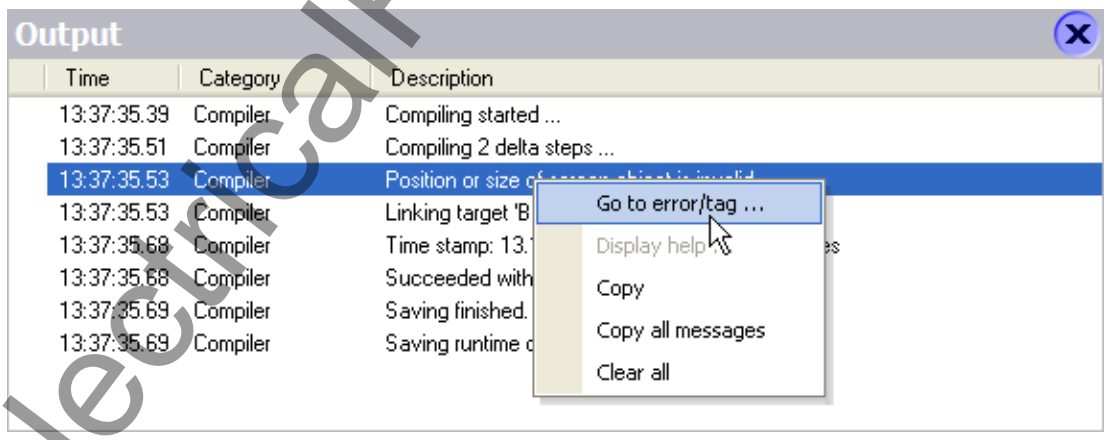
Opening the library in a separate window

You can swap the library from the "Toolbox View" to a separate window. To do so, select the "Library in Toolbox" command from the context menu of the Library view. Select this command again to restore the library to the "Toolbox View".

2.2.7 Output View

Introduction

The output window displays system alarms generated, for example, in a project test run.



Description

The output view normally displays system alarms in the order they occur. The categories define the corresponding WinCC flexible module which has generated a system alarm. For example, system alarms for the "Generator" category are generated during the consistency check.

To sort system alarms, click the header of the corresponding column. The pop-up menu can be used to jump to an error location or a tag, and copy or delete system alarms.

The output view shows all system alarms of the last action. A new action overwrites all previous system alarms. You can still retrieve old system alarms from a separate log file.

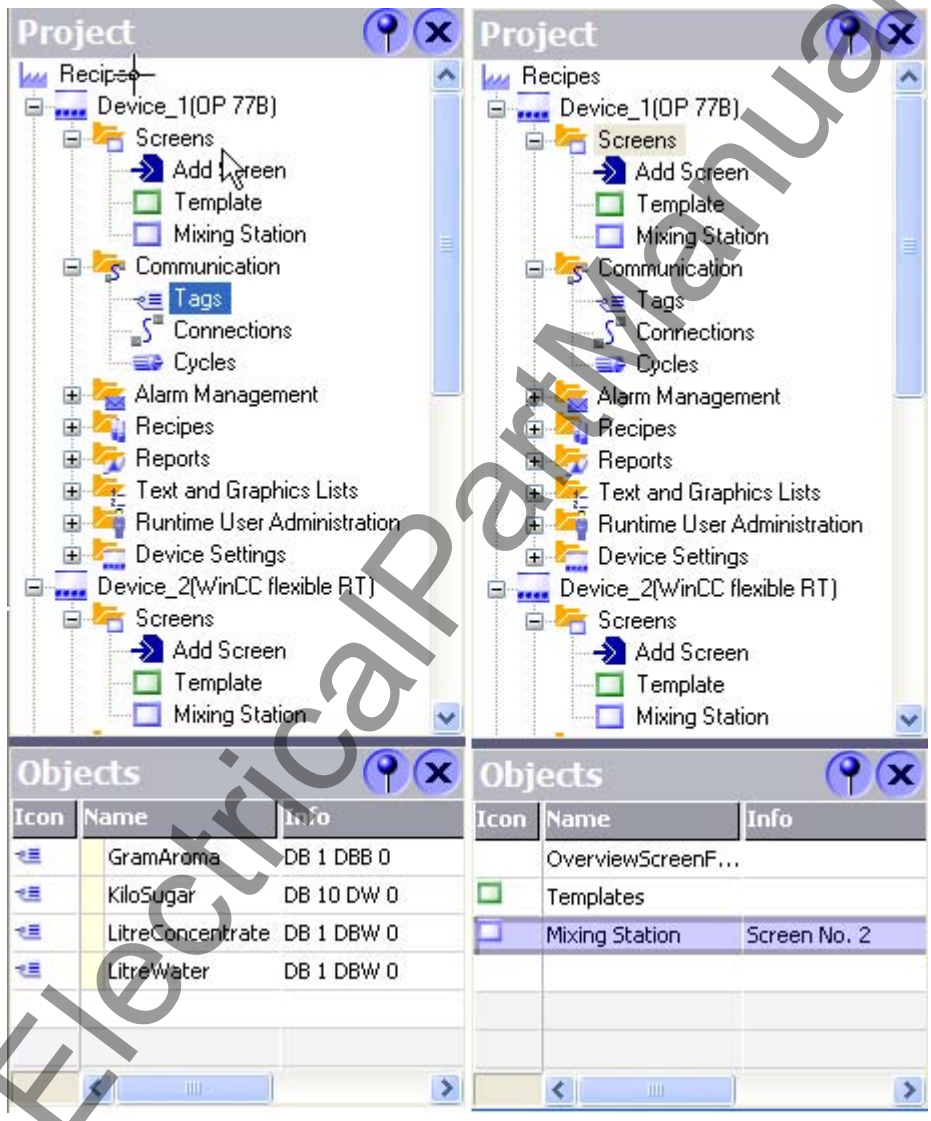
www.ElectricalPartManuals.com

2.2.8 Object view

Introduction

If folders or editors are selected in the Project View, their content is displayed in the Object View.

The following figure illustrates how the selection in the Project View affects the display in the Object View:



www . ElectricalPartManuals . com

Description

Double-click an object in the OBJECT View to open the corresponding editor. Drag-and-drop functions are available for all objects displayed in the object window.

The following drag-and-drop actions, for example, are supported:

- Moving a variable to a process screen in the work area: Creates an I/O field which is linked to the tags.
- Moving a tag to an existing I/O field: Creates a logical link between the variable and the I/O field.
- Moving a process screen to another process screen in the work area: Generates a button with screen change function which is linked to the process screen.

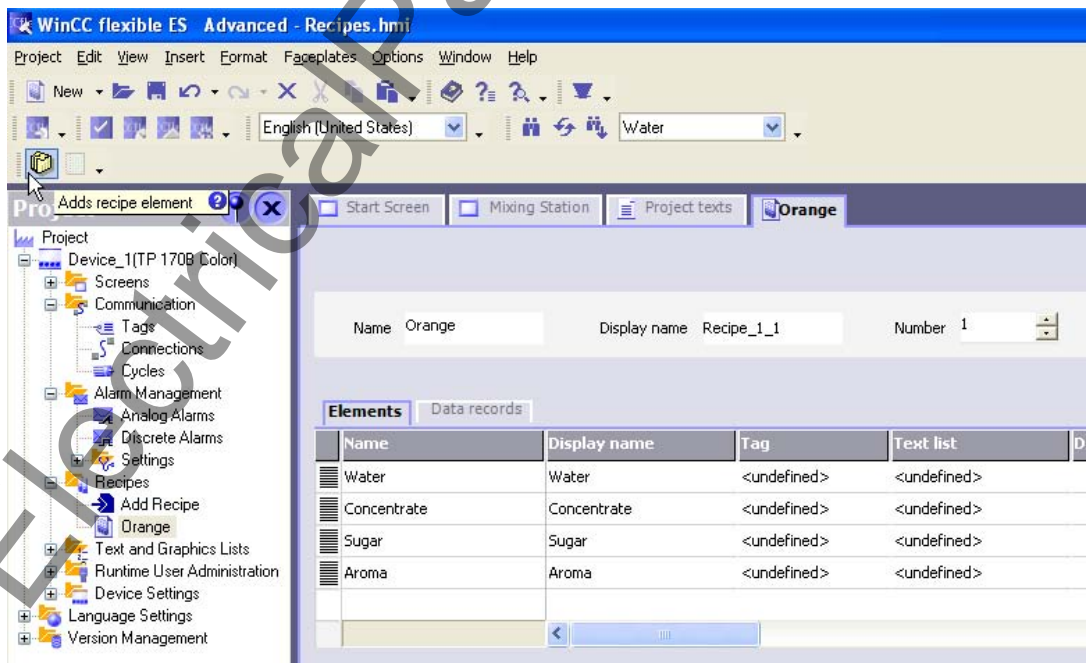
Long object names are abbreviated in the "Object View". After moving the mouse pointer to the object, the full name appears as a ToolTip.

When a large number of objects are available, quickly locate the object you require by entering just the first letter of the object.

2.3 Placing editor-specific operating elements

Introduction

Editor-specific operating elements are only visible in the active work area of the corresponding editor.



Editor-specific operating elements include:

- Toolbars
- Toolbox
- Menu commands

www.ElectricalPartManuals.com

Placement

The default position of editor-specific toolbars is on the right-hand side or below the existing toolbars.

Default position of editor-specific toolbox views is the screen margin on the right-hand side.

Editor-specific commands are added to the corresponding menus.

The positions of editor-specific operating elements are restored the next time you start WinCC if you have rearranged these in a previous session to suit your individual requirements.

2.4 Working with windows and toolbars

Introduction

WinCC flexible allows you to customize the layout of frames and toolbars. You can hide certain frames which are not used frequently in order to enlarge the work area.

The "View" menu can be used to restore the default layout of frames and toolbars.

Operating Elements Available

The table below shows you the operating elements of the frames and toolbars and what they are used for.

Operator control element	Purpose	Where used
	Closes a frame or toolbar	Frames and toolbars (movable)
	Moves and docks frames and toolbars using drag-and-drop	Frames and toolbars (movable)
	Moves a toolbar by means of drag-and-drop	Toolbar (docked)
	Adds or deletes toolbar icons	Toolbar (docked)
	Activates the auto-hide mode for a window	Frame (docked)
	Disables auto-hide mode for a frame	Frame (docked)

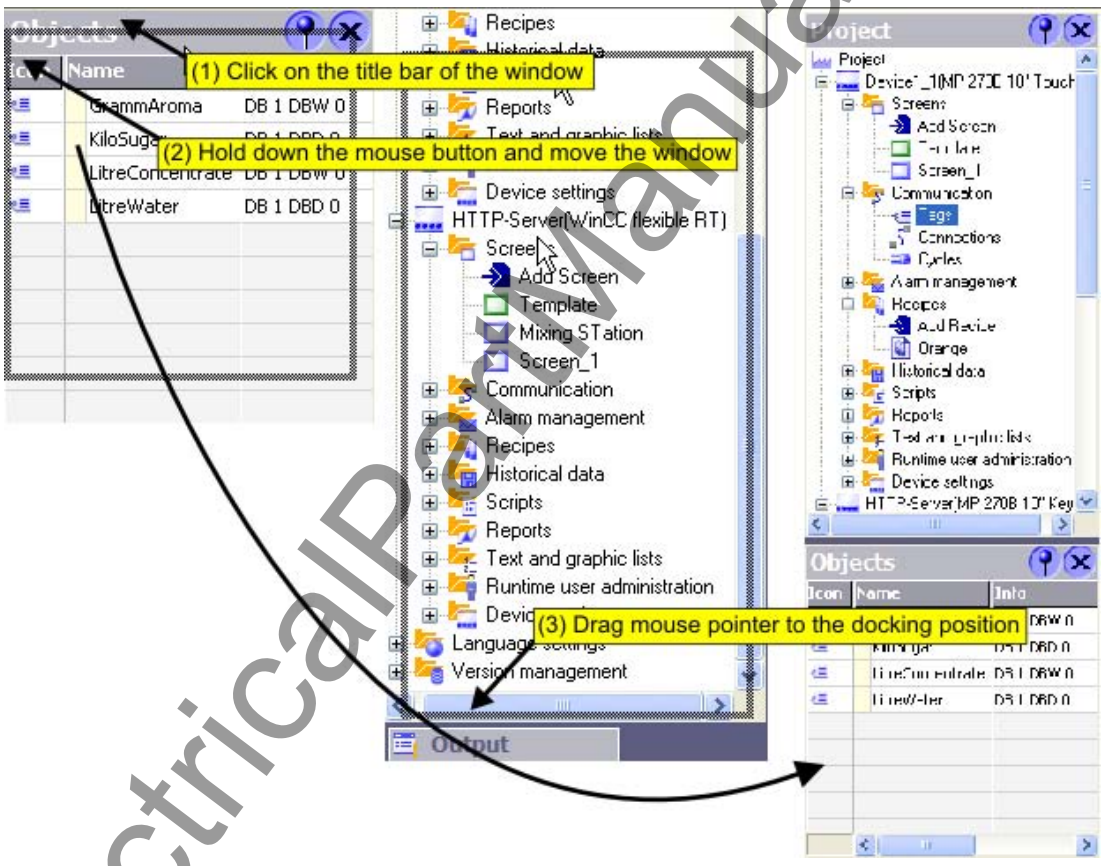
www . ElectricalPartManuals . com

Docking frames or toolbars

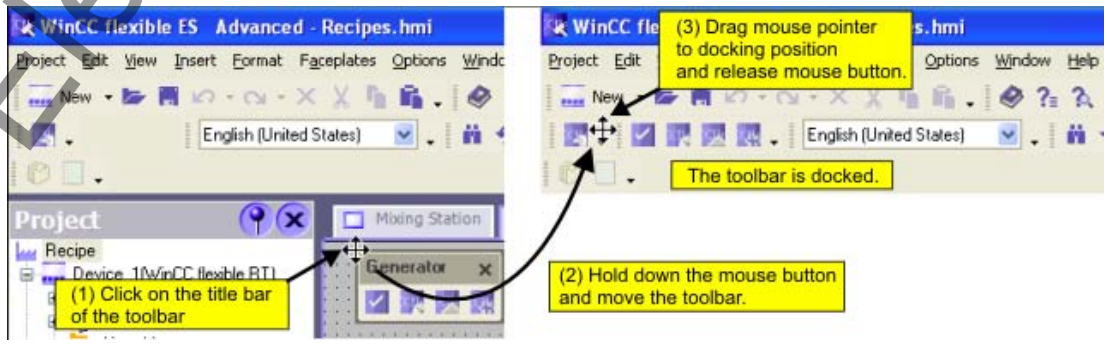
"Docking" refers to the integration of a window into the WinCC flexible workbench. You can automatically hide docked frames in order to increase your workspace.

A freely moveable window can be docked on a window at the following positions:

- Upper edge
- Right edge
- Bottom edge
- Left edge



You can dock a toolbar onto any existing toolbar.



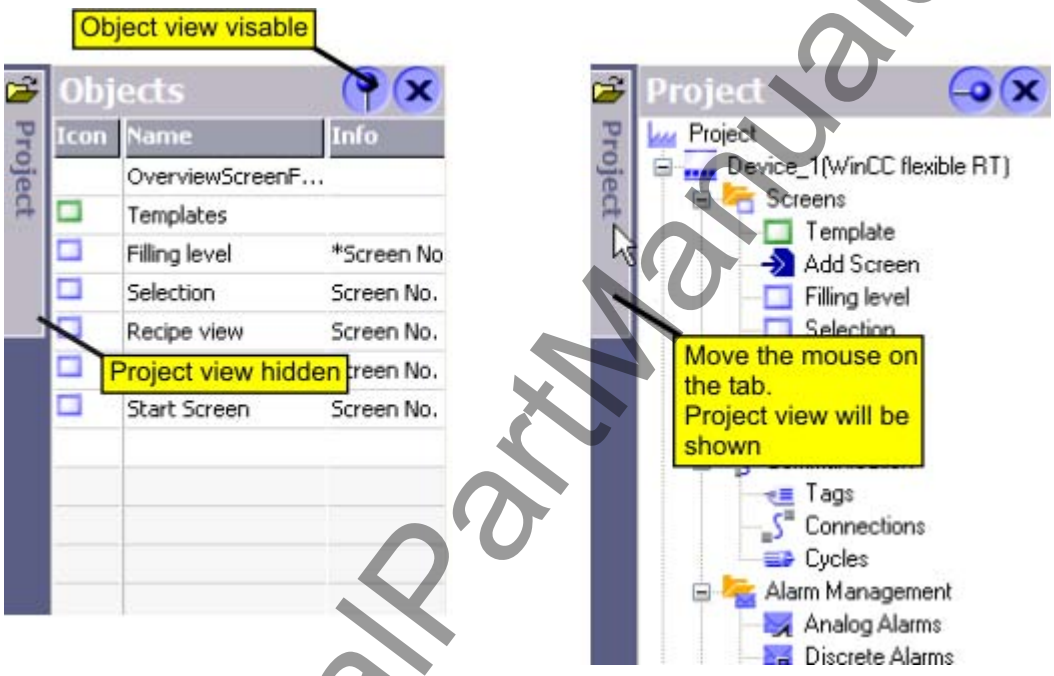
www.ElectricalPartManuals.com

Combined frames

You can combine a frame with another frame. Each frame is represented in the combined frame by a separate tab. To change to a different frame, click the corresponding tab.

Hiding windows automatically

You can automatically hide the windows you do not require frequently. This will increase your work area. To restore the window to the screen, click its title bar.



www . ElectricalPartManuals . com

2.5 Working with the Mouse

Introduction

Work is mainly completed with the mouse in WinCC flexible. Important operating functions in this context are the drag-and-drop function and the call of commands from the context menu.

Drag-and-drop

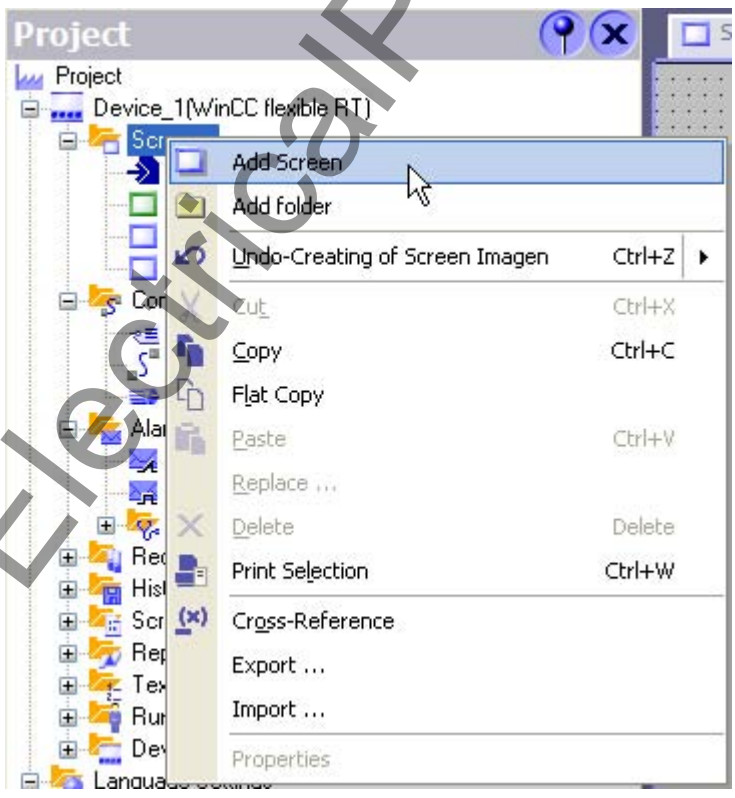
Drag-and-drop makes configuration much easier. For example, when you drag-and-drop a variable from the Object View to a process screen, the system automatically generates an I/O field which is logically linked to the variable. To configure a screen change, drag-and-drop the required process screen onto the process screen shown in the work area. This generates a button configured to contain a corresponding screen change function.

The drag-and-drop function is available for all objects in the project view and "Object view." The mouse pointer shows you whether drag-and-drop is supported at the destination or not:

-  Drag-and-drop is possible
-  Drag-and-drop is not possible

Context menu

In WinCC, you can right-click any object to open a context menu. The context menu contains the commands you can execute in the relevant situation.



www . ElectricalPartManuals . com

Overview: Mouse functions

Function	Effect
Left-click	Activates any object or executes an action such as a menu command or drag-and-drop.
Right-click	Opens a context menu.
Double-click (left mouse button)	Starts an editor in the Project View or Object View or opens a folder.
<Left mouse button+drag-and-drop>	Generates a copy of the object in the project view.
<CTRL+left mouse button>	Selects a number of individual objects from the "Object view" one after the other.
<SHIFT+left mouse button>	Selects all objects within the rectangle lasso you have drawn with the mouse in the "Object view."

2.6 Keyboard control

Introduction

WinCC flexible provides a number of hotkeys which you can use to execute frequently required menu commands. The menu shows whether a hotkey is available for the relevant command or not.

WinCC also integrates all the standard hotkeys provided by Windows.

Important hotkeys

The table shows you the most important hotkeys for use in WinCC flexible.

Hotkeys	Effect
<Ctrl+Tab>/<Ctrl+Shift+Tab>	Activates the next/previous tab in the work area.
<Ctrl+F4>	Closes the active view in the work area.
<Ctrl+C>	Copies a selected object to the clipboard.
<Ctrl+X>	Cuts an object and copies it to the clipboard.
<Ctrl+V>	Inserts the object stored in the clipboard.
<Ctrl+F>	Opens the "Find and Replace" dialog.
<Ctrl+A>	Selects all objects in the active area.
<ESC>	Cancels an action.

www . ElectricalPartManuals . com

2.7 Working with WinCC flexible

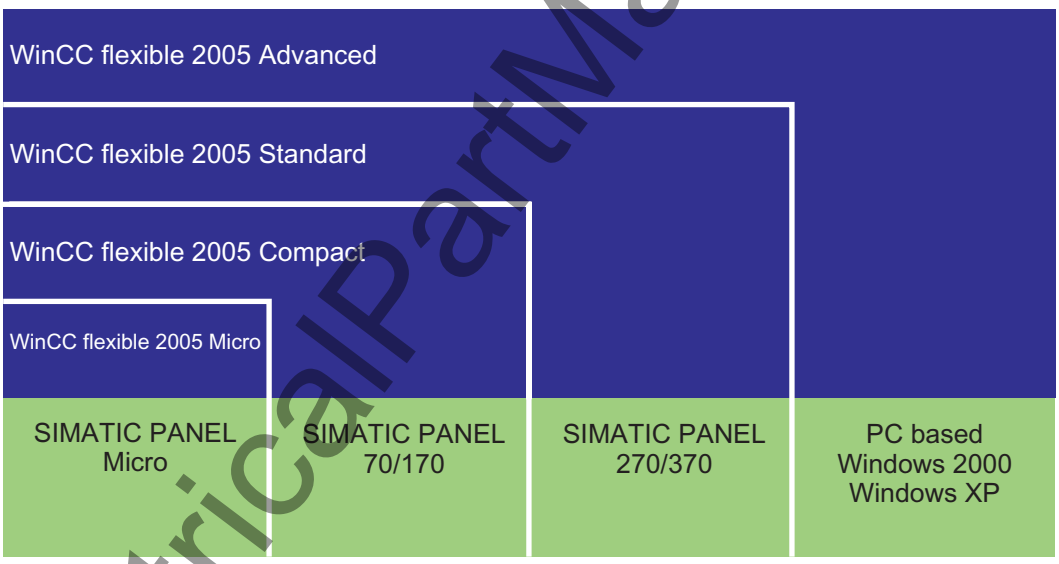
2.7.1 Working with WinCC flexible

Introduction

WinCC flexible offers a range of scalable engineering systems which are optimally adapted to the respective configuration task or can be adapted by the user. Each edition supports a wider range of HMI devices and functions, whereby the "Standard" edition can be used to configure HMI devices from the "Micro" edition. You can always migrate to a higher edition by means of a powerpack.

Functional scope of the individual editions

WinCC flexible is available in the following editions:



2.7.2 Working with projects

Introduction

WinCC flexible is used to configure user interfaces to operate and monitor machines and plants.

Special editors are available for the different configuration tasks. All configuration information is saved in a project.

www . ElectricalPartManuals . com

Creating or Loading a Project

After starting WinCC flexible, a wizard guides the user through all the steps which are necessary to create a new project. The user is prompted, for example, to enter a name for the project and select an HMI device.

If WinCC flexible is already open, select the "New" command to create a new project. In some circumstances, a wizard will appear to guide you through the process.

To load an existing project, select the "Open" command from the "Project" menu.

Device-based dependency

WinCC flexible only provides the functionality supported by the HMI devices selected. The Project View displays the editors available for configuration.

Migration

If an existing ProTool or WinCC project is opened in WinCC flexible, the data is converted. The user is guided through the conversion process and informed of the progress of the conversion.

2.7.3 Editing Multiple Projects with WinCC flexible

Principle

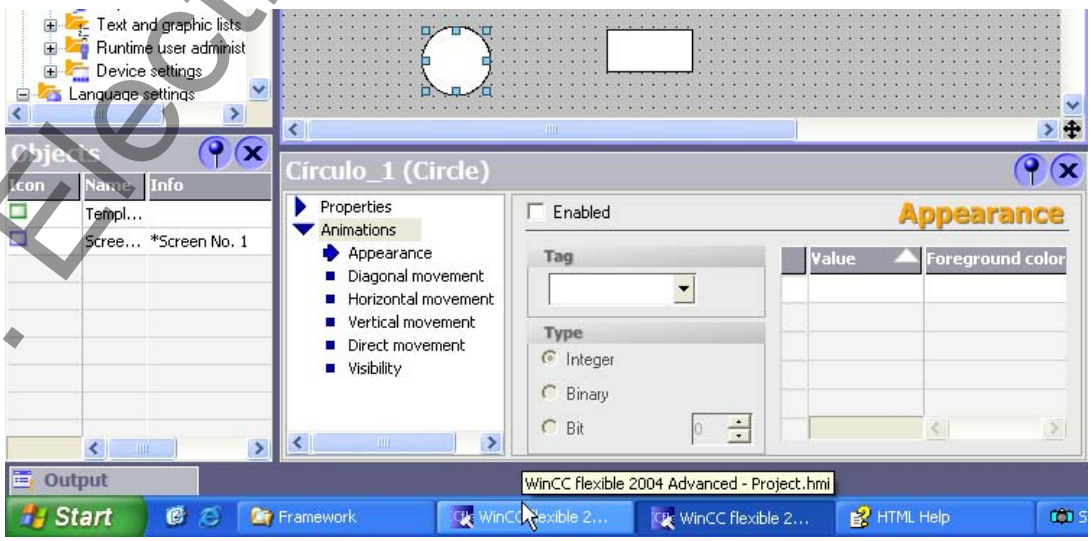
WinCC flexible only allows one project to be open for editing at any time. If projects should be copied globally, for example, restart WinCC flexible and then open the required project.

Note

If both ProTool and WinCC flexible are installed on your PC, you can only open one of the programs at a time.

Several HMI devices can be set up parallel in each project.

Each opened WinCC flexible is shown in the Windows task bar:



www.ElectricalPartManuals.com

2.7.4 Functional scope of a project

Introduction

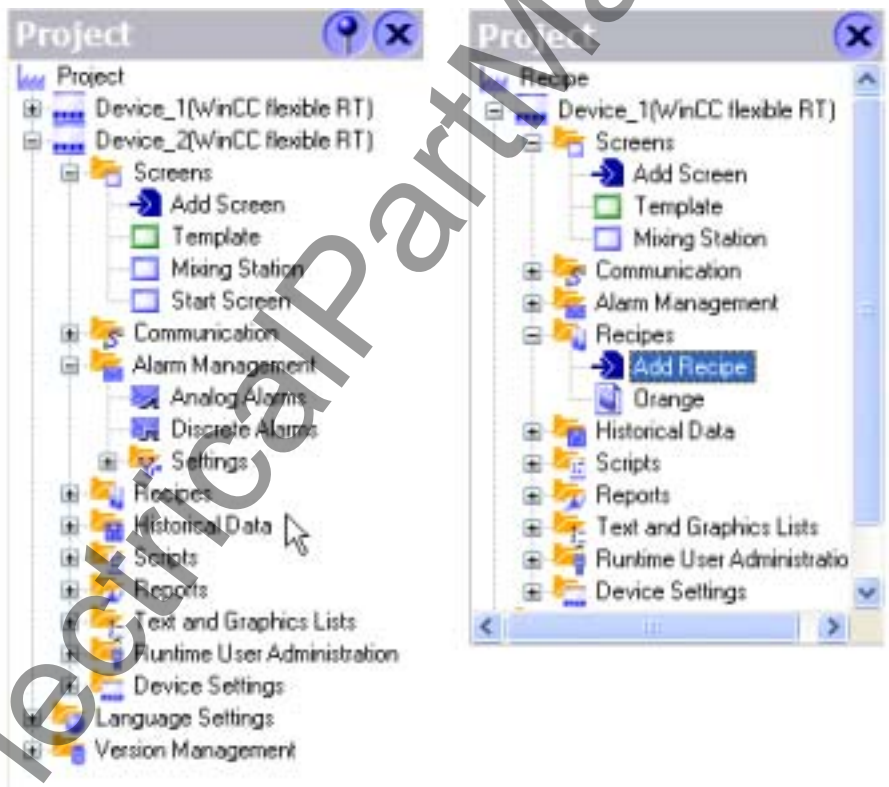
If WinCC flexible is used to edit projects for different HMI devices, the functional scope is not identical for the configuration. Different functionality is available according to the HMI device.

Principle

The functional scope available is dependent on the HMI device selected. Only configure the functions supported by the selected HMI device. This procedure is advantageous for efficient configuration.

The editors displayed in the Project View can be used, for example, to quickly detect which functions are supported by the HMI device selected.

The diagram illustrates the functional scope of two different HMI devices based on the Project View:



www . ElectricalPartManuals . com

2.7.5 Editor Properties

Introduction

WinCC flexible provides a special editor for each configuring task. WinCC flexible differentiates between two different types of editors: graphical editors and tabular editors. You can open up to 20 editors in parallel.

Graphical Editors

Graphical editors, such as the Screen editor, display the elements belonging to both the Project View and Object View. You open each object in the work area with graphical editors.

Tabular editors

Tabular editors, such as the Tag editor, only display the associated objects in the Object View. When a tabular editor is opened to edit the objects, all associated objects are displayed in a table in the work area

Editor properties

The following properties apply to all editors and their objects:

- Changing contents

  Changes take effect directly after exiting an input field and affect projects globally. All the objects affected by a modification are automatically updated.

  If a tag parameter is changed at the place of use in the Screens Editor, for example, the change has a direct effect in the Tag editor.

- Accepting changes to the project data

  The modified project data are transferred to the project database as soon as the project is saved.

- Undo or redo working steps

  Every editor has an internal list in which user actions are saved. In this way, all actions can be reverted (undone) or restored. The relevant commands are in the "Edit" menu. The list is deleted when the editor is closed or the project is saved. Switching to another editor does not affect the actions stored in the list.

www . ElectricalPartManuals . com

2.7.6 Open Editor

Introduction

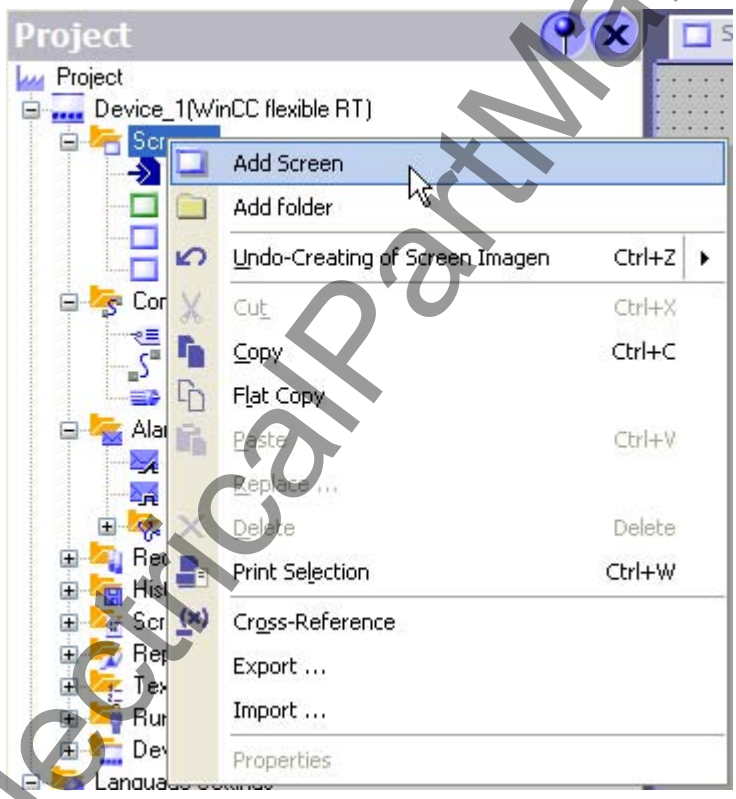
The way in which an editor is started depends on whether it is a graphical editor (e.g. Screen Editor) or tabular editor (e.g. Tag Editor). You can open up to 20 editors in parallel.

Opening a Graphical Editor

A graphical editor is started by either creating a new object or opening an existing object.

To create a new object, proceed as follows:

1. Click the right mouse button on the graphical editor in the Project View in which a new object is to be added.
2. Select "Add screen" in the context menu, for example.



The object, e.g. a screen, is created in the Project View and displayed in the work area.

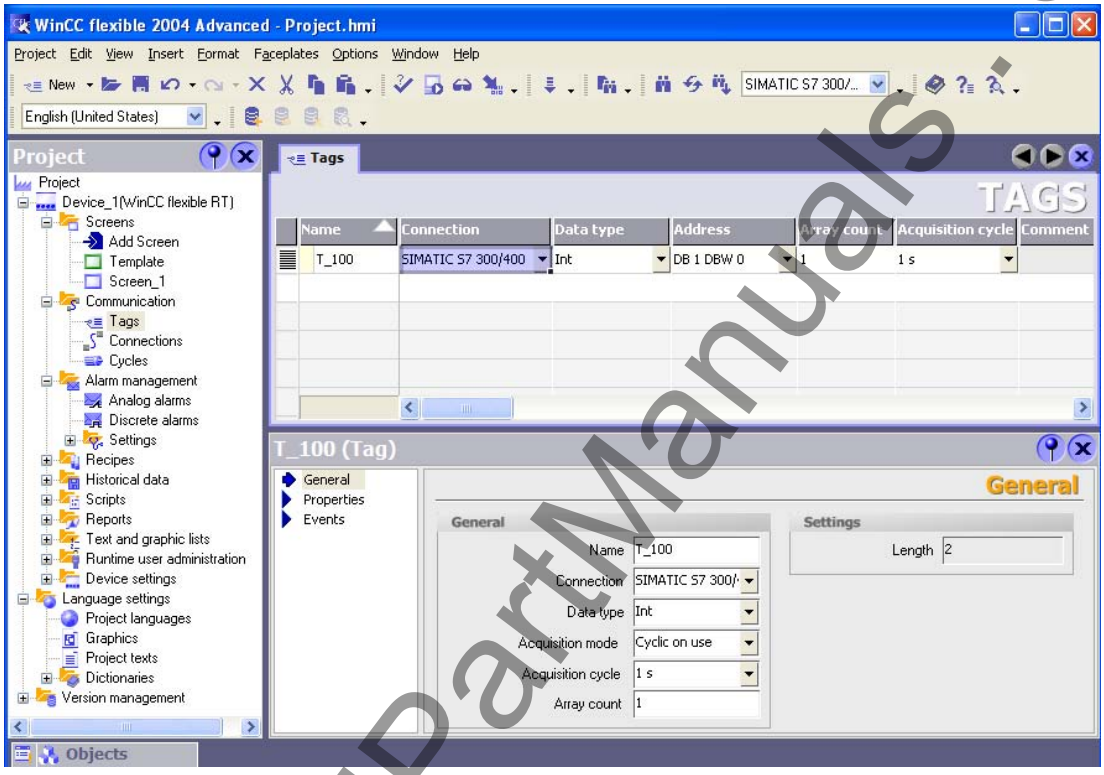
3. To open an existing object, double-click on the object in the Project View or Object View.

   The object, e.g. a screen, is displayed in the work area.

www . ElectricalPartManuals . com

Opening a tabular editor

A tabular editor is opened by double-clicking on the tabular editor in the Project View. The editor appears in the work area.



A tabular editor can also be activated using the associated context menu. To open an existing element in the tabular editor, select the tabular editor in the Project View. Then double-click on the required object in the Object View.

Alternative procedure

To open an editor via the menu, select the "New object in project" command from the "Insert" menu.

www . ElectricalPartManuals . com

2.7.7 Switching between editors

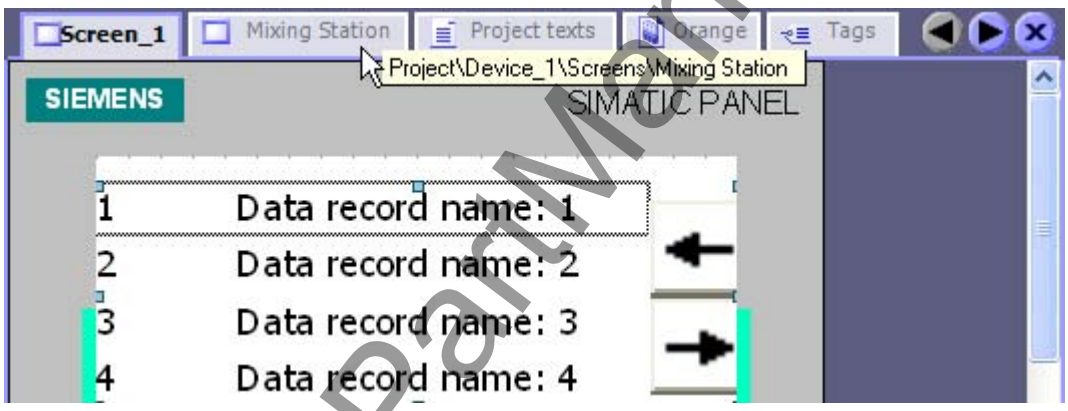
Introduction

Although several editors or their objects can be opened simultaneously in WinCC flexible, only the work area of one editor can be active in the work area.

If several editors are open, they are represented by separate tab controls in the work area.

Tab Controls

To select a different editor, click the relevant tab in the work area. In tabular editors, a tab shows the name of the editor for easy identification. In the case of graphical editors, the name of the current element is indicated, e.g. "Screen1".



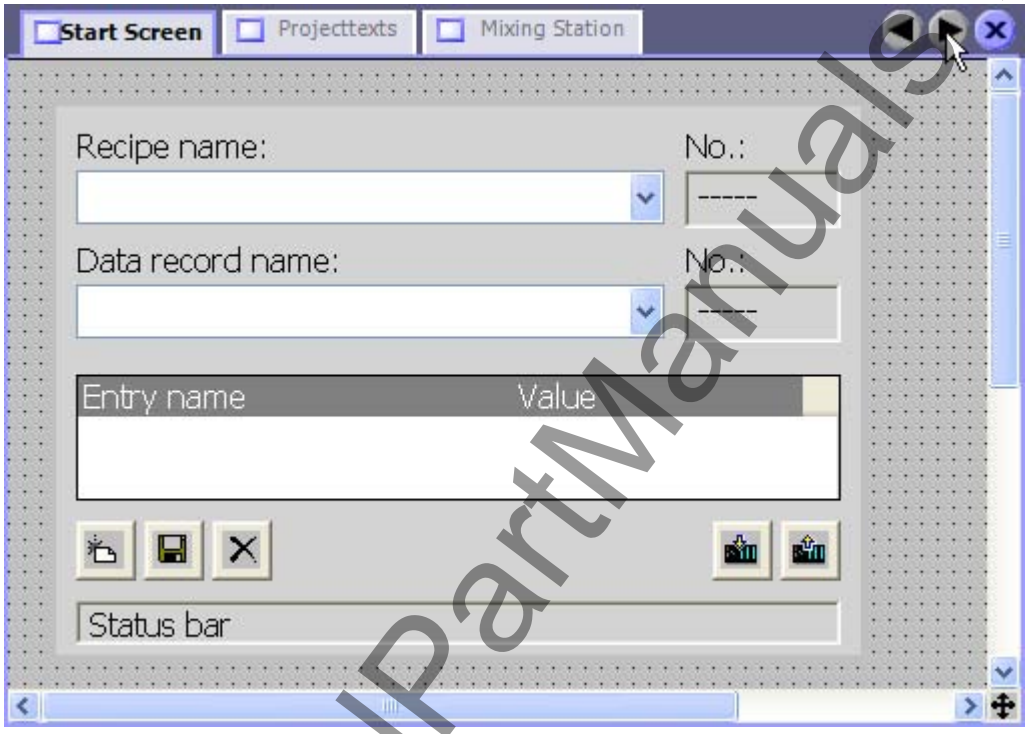
The tooltip indicates which HMI device is being configured in the editor.

www . ElectricalPartManuals . com

Navigation arrows

If the work area runs out of space to show all tabs, the navigation arrows become active in the work area.

To access tabs which are no longer visible in the work area, click the corresponding navigation arrow.



Closing the editor

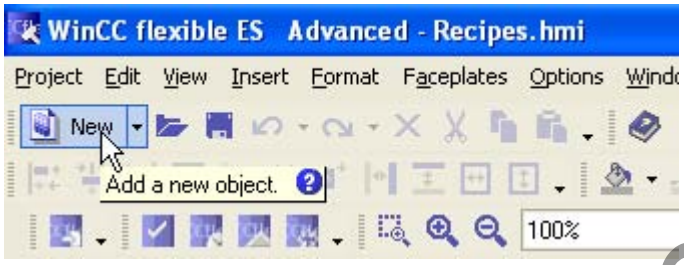
To close an editor, click the symbol in the work area.

www . ElectricalPartManuals . com

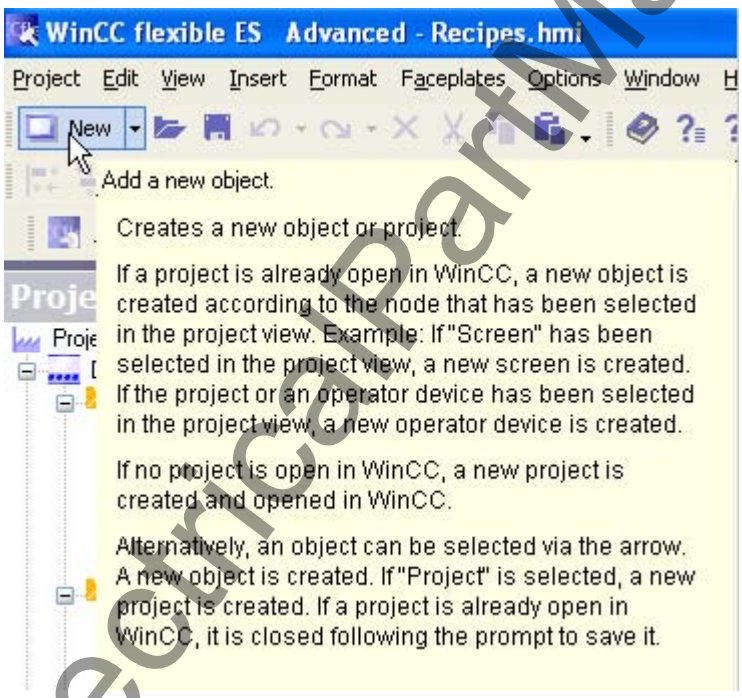
2.7.8 Displaying Help

What's this help

A tooltip will appear after moving the mouse pointer over any object, icon, or dialog element.



A question mark next to the tooltip indicates that a what's this help is available for this user interface element. To call up an additional explanation to the short description, click on the question mark, press <F1> if the tooltip is activated, or move the mouse cursor to the tooltip.



The explanation includes references which refer users to a detailed description in the Online Help.

www . ElectricalPartManuals . com

Online help

In the "Help" command menu you can access the online help. When you use the "Help > Contents" menu command, the WinCC flexible Information System opens with an opened table of contents. Use the table of contents to navigate to the desired topic.

Alternatively select the "Help > Index" menu command. The WinCC flexible Information System opens with an opened index. Use the index to search for the desired topic.

In order to use the full text search across the entire WinCC flexible Information System select the "Help > Search" menu command. The WinCC flexible Information System opens with a search tab. Enter the desired search term.

The WinCC flexible Information System can also be opened via the Start menu in Windows. Select the menu command "Start > SIMATIC > WinCC flexible > WinCC flexible Help System" in the task bar.

The Online Help system is opened in a separate window.

2.7.9 Customized setup of WinCC flexible

Introduction

WinCC flexible allows you to customize the position and reaction of windows and toolbars. This allows you to configure the work environment to meet your special requirements.

Working Environment User-Dependency

The appearance of WinCC flexible is linked to the user logged on in Microsoft Windows. On saving the project, the positions and behavior of windows and toolbars are automatically saved with it.

When the project is opened again, the project status loaded is identical to that when saved. In this way, the working environment is opened as it was when last closed. This is also the case when a project edited by a different project planner is opened.

Resetting the work environment

The positions of views and toolbars can be reset to their original state. To do this, select "Reset layout" in the "View" menu.

www . ElectricalPartManuals . com

3 Working with projects

3.1 Basis for working with projects

3.1.1 Working with projects

Projects in WinCC flexible

The WinCC flexible configuration software is used to configure the user interface for controlling machines and systems. These can involve simple text display units for reading parameters but also complex operating stations for the production line, for example.

WinCC flexible has a modular design. You decide which HMI device types can be configured through the selection of a WinCC flexible edition. If necessary, a WinCC flexible edition can be upgraded to a later edition without any problem.

A WinCC flexible project contains all of your configuration data for a plant or an HMI device. Configuration data includes:

- Process screens, to display the process.
- Tags, to transfer data between PLC and HMI device in runtime.
- Alarms, to display operational states in runtime.
- Logs, to save process values and alarms.

All data related to a project is stored in the database integrated in WinCC flexible.

Application example

WinCC flexible can be used, for example, to process the following scenarios:

- Configuring an HMI device installed permanently at machine level.

  This represents the most common type of configuration in which the HMI device is permanently installed on the system or machine.

- Use of mobile units

  Mobile units are generally used in the case of large production sites or lines, or material conveyor technology. The machines to be operated are equipped with several interfaces to which the Mobile Panel 170, for example, can be connected.

  The operator or service technician can thus work directly on site. This enables an accurate setting up and positioning, e.g. during the startup phase. In the case of servicing, mobile units ensure shorter downtimes.

www . ElectricalPartManuals . com

- Configuring several HMI devices

  It is possible to connect several HMI devices to a system or machine. The system can then be operated from various points. Using WinCC flexible, it is possible to configure several units in one project, even of different types, which can work with the same project data.

3.1.2 Component parts of a project

Principle

A WinCC flexible project consists of all the configuration data which enables the system to by operated and monitored. The configuration data is compiled in WinCC flexible according to topic categories. Each category is processed in an individual editor.

The editors available for use depend on the WinCC flexible edition used and the HMI device to be configured. The working environment of WinCC flexible only displays the editors supported by the HMI device currently in use. This means that configuration is simple and easy to follow.

3.2 Types of projects

3.2.1 Types of projects

Principle

Different types of projects can be created using WinCC flexible. The type of project is dependent on the system configuration, the size of the system or machine, the required representation of the system or machine and the HMI devices used for operating and monitoring.

The following project types can be configured in WinCC flexible:

- Single-user project

  Project which is used for a single HMI device.

- Multi-user project

  Project in which several HMI devices are configured.

- Project for use on different HMI device

Single-user project

In most cases, only one HMI device is configured. During the configuration phase, a project always displays precisely the function range which is supported by the currently selected HMI device.

www.ElectricalPartManuals.com

Multi-user project

If several HMI devices are used to operate a system, WinCC flexible can be used to create a project in which several HMI devices are configured. This type of project is used, for example, when the machine or system controlled is operated from several different points. Common objects can then be used in the project. This method means that a project need not be created for each separate HMI device, but rather all HMI devices are managed in the same project.

A WinCC flexible project consists of all the configuration data which enables the system to by operated and monitored. Each configured unit only displays the functions which the specific unit supports. Functions which are not supported are hidden but remain a component part of the project data.

Project for use on different HMI device

A project can be created for a specific HMI device and loaded onto several different HMI devices. When loading onto the HMI device, only that data is loaded which is supported by the HMI device.

3.2.2 HMI device dependency of projects

Principle

The functions of the HMI device determine project visualization in WinCC flexible and the functional scope of the editors.

Selecting Operating Unit Types

You select the type of the first HMI device when you create a project. The HMI device type can be changed in the Project View context menu of the HMI device.

Note

After switching the HMI device type, all configured data is still contained in the project file. In the Engineering System, only the functions are still available and only the configuration data that are supported by the current HMI device are displayed. This involves, for example, logs, recipes, available objects in screens, available system functions, and available communication protocols.

Functions dependent on the HMI device

Apart from changing the functional scope when switching from one HMI device type to another, the following features must also be considered:

- Colors supported

  When switching from an HMI device with color display to another with a smaller color range, the color is changed automatically. If you change the colors yourself for an HMI device with a smaller color range and you change back to an HMI device with a larger color range, the reduced color range is retained.

www . ElectricalPartManuals . com

- Fonts

  If a "font" which has been configured is not available on the HMI device, it is replaced by a similar one or the configured "standard font." The "standard font" is dependent on the HMI device selected.

- Resolution

  When switching from an HMI device to another with a lower resolution, there are two options available: All screen objects can be automatically scaled. All screen objects can be left in their original size. Objects at the lower or right side of the screen that overlap the displayable screen will not be displayed. To display these hidden objects, select the screen background and select "Display hidden objects" in the context menu. In the dialog that opens, you can select individual objects or all objects and move them to the visible area of the screen by pressing "OK."

Note

Because HMI devices with a display size less than 6" have the same width but different heights, you should turn off the automatic scaling when replacing the HMI device. Because the width remains the same, automatic scaling would change only the height of the objects, causing them to become distorted. To enable or disable automatic scaling, select the "Options > Settings" menu command. In the "Settings" dialog that opens, click "Settings for screens editor" in the "Screens editor" group. Enable or disable the "Fit screens and screen objects to new HMI device" option.

Selecting the operating system version of the HMI device

When you configure a new HMI device, WinCC flexible automatically selects the latest operating system version.

If you want to use a new version on an HMI device with an older operating system version, you must transfer an image of the relevant firmware version to the HMI device. WinCC flexible provides the required images for the supported HMI devices. You can find further information in chapter "Operating System Transfer".

If you must use an older operating system version for compatibility reasons, you must convert the WinCC flexible project to the previous version. The version of the HMI device is automatically set to the previous version during conversion. You can find further information under "Converting a project."

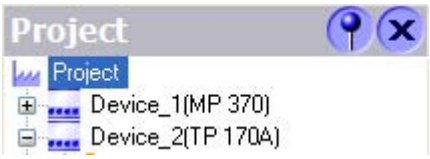
If you want to use an older version on an HMI device with the current operating system version, you must transfer an image of the relevant firmware version to the HMI device. WinCC flexible provides the required images for the supported HMI devices. You can find further information in chapter "Operating System Transfer".

www . ElectricalPartManuals . com

3.2.3 Configuring a project for several HMI devices

Principle

Using a WinCC flexible edition from "Compact" and later, a project can be configured with several HMI devices.



You can delete, copy (also across projects) and rename the HMI devices in the project window.

Application example

This type of configuration is used, for example, in projects designed for large systems which are to be operated by several HMI devices.

Global data and HMI device-specific data

Within a project, in which several HMI devices are configured, some data and objects are available specific to HMI devices and some are globally available throughout the project.

- HMI device-specific data

  Data related to a specific HMI device can be individually setup in the project. HMI device-specific data and objects are all data and objects that are listed in the Project View below the entry "Device", e.g. pictures, communication, recipes or logs.

www . ElectricalPartManuals . com

- Global project data

  Global project data applies to all HMI devices within the entire project. It applies to all data and objects in the Project View at the same level as the "Device" option, e.g. "Language" or "Version management."

3.2.4 Creating a project for use on different operating units

Principle

It is possible to create a single project and to load it on several different HMI devices.

Application example

This type of configuration is typically used for several operating units of a similar type but with different performance, for example.

Special aspects of configuration

Proceed as follows to use a project for different HMI devices:

- Create a project for a specific operating unit type, normally for the operating unit with the smallest functional scope.
- Copy the configuration for the operating unit in the Project View.
- Test the feasibility for other operating units by switching the operating unit type in the project.

Pay particular attention to the following aspects:

- After switching the operating unit type, all configured data are still contained in the project file. However, only that configuration data is displayed which is supported by the operating unit currently in use. This relates to editors, objects and object properties.
- WinCC flexible not only checks the functional scope of an operating unit but also its limitations. If only a specific number of tags can be used on an HMI device, for example, the corresponding error message appears when transferring the project to the HMI device or when testing it in runtime.

www.ElectricalPartManuals.com

3.2.5 WinCC flexible integrated in SIMOTION and STEP7

Introduction

WinCC flexible operation can be operated in SIMATIC STEP 7 and SIMOTION SCOUT starting with the WinCC flexible Compact edition. Integration has the following advantages:

- The tags and texts are imported into the WinCC flexible project.
- Direct access to SIMATIC STEP 7 symbols and SIMOTION SCOUT symbols during process connection.
- The texts and attributes contained in the alarm configuration are imported into WinCC flexible.
- The configuring overhead is reduced thanks to the common use of configuration data.

A condition for integrated operation in SIMOTION SCOUT is that SIMATIC STEP 7 and SIMOTION SCOUT are installed on the configuration computer.
The installation sequence is:

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

WinCC flexible integrated in SIMATIC STEP 7

When installing WinCC flexible, the user defines whether WinCC flexible should be integrated in SIMATIC STEP 7. The integration of SIMATIC STEP 7 in the configuration interface offers the following advantages:

- Better fault tolerance
- Less work for modifications
- Less configuration work

During configuration, direct access is made to the SIMATIC STEP 7 symbol tables, data areas, and controllers from SIMATIC STEP 7. The icon table contains data point definitions (e.g. addresses and data types) defined during creation of the PLC program.

The WinCC flexible project tree is mirrored in the project tree of SIMATIC Manager. The objects are edited, however, in a separate WinCC flexible application with the independent WinCC flexible ES user interface.

Further information on the use of SIMATIC STEP 7 is provided in the documentation of STEP 7.

www . ElectricalPartManuals . com

WinCC flexible integrated in SIMOTION SCOUT

When installing WinCC flexible, the user defines whether WinCC flexible should be integrated in SIMOTION SCOUT.

HMI devices with SIMOTION SCOUT connection are configured in the SIMOTION SCOUT working environment.

When WinCC flexible and SIMOTION SCOUT are installed on a configuration computer, WinCC flexible is integrated in the SIMOTION SCOUT working environment. Work is then performed in a single working environment for all tasks from the SIMOTION SCOUT or WinCC flexible environment.

A WinCC flexible project appears as a node in the SIMOTION SCOUT project tree. All operating units configured in a project appear as sub-entries of the project tree The WinCC flexible editors are opened parallel to the SCOUT editors in the SCOUT user interfaces.

Further information about using SIMOTION SCOUT is provided in the "SIMOTION SCOUT" documentation.

3.3 Multilingual configuration

Multilingual configuration

You can configure your projects in multiple languages using WinCC flexible. WinCC flexible supports the multilingual configuration of practically all objects with texts displayed in runtime.

WinCC flexible can be used for configuration in all languages installed in the operating system.

In addition to direct text input in the editors, WinCC flexible provides easy-to-use export and import features for translating projects. This is of particular advantage when configuring large projects with a large share of text.

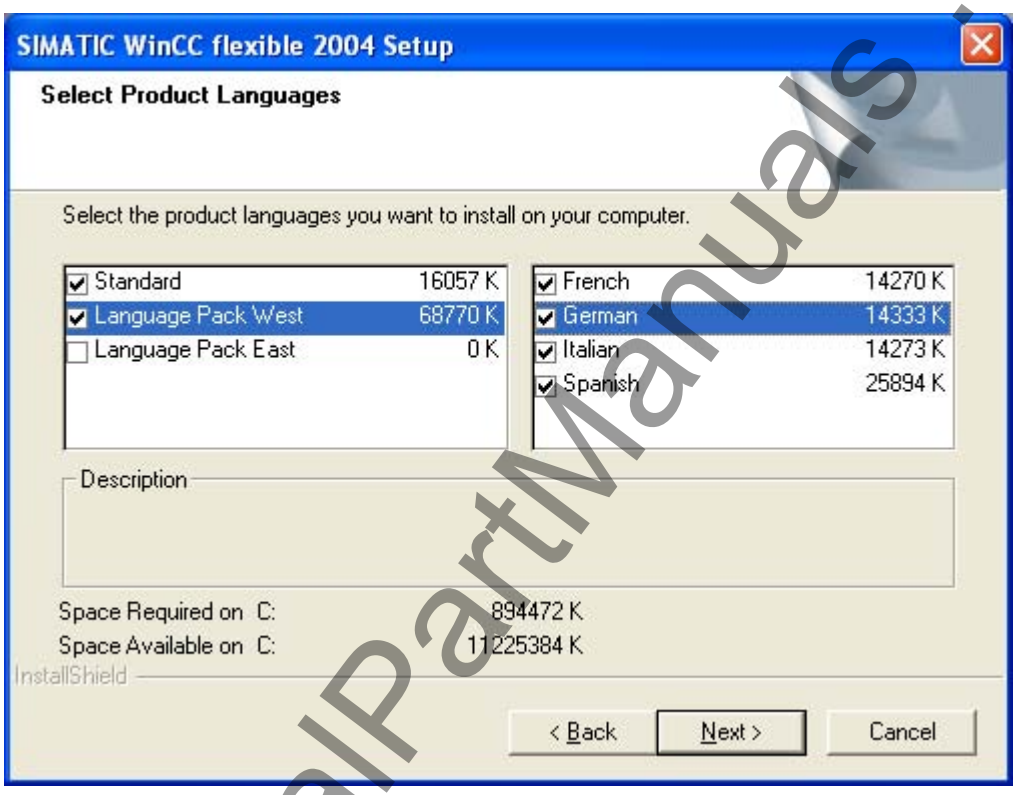
Use the following editors to translate texts in WinCC flexible:

Toolbar	Brief description
Project languages	Managing languages for the project texts
Languages and fonts	Defining the languages and fonts used in runtime
Project texts	Managing language-dependent project texts
Graphic browser	Managing local graphics
Dictionaries	Managing the system and user dictionaries

www . ElectricalPartManuals . com

WinCC flexible multilingual user interface

The language of the user interface in WinCC flexible can be selected, for example, to suit regional requirements of several engineers of different nationality working on a project configuration. During the installation of WinCC flexible, select the languages that will be required later.



English is installed as the default user interface language in any case. The following languages can also be installed:

- Western European languages

  German

  Spanish

  Italian

  French

www . ElectricalPartManuals . com

3.4 Editing projects

3.4.1 Editing projects

Objects and editors

The following objects can be created and edited in WinCC flexible.

- Screens

  Screens are created and edited in the Screens editor. It is possible to define the navigation between screens in the "Screen navigation" editor.

- Faceplates

  Faceplates are groups of objects which can be used as often as required in a project. Faceplates are stored in libraries.

- Graphics list

  In a graphic list, the values of a tag are assigned to various graphics. The graphic lists are created in the "Graphic List" editor and displayed with the "Graphic IO Field" object.

- Text List

  In a text list, the values of a tag are assigned to various texts. The text lists are created in the "Text List" editor and displayed with the "Symbolic IO Field" object.

- Language-dependent texts and graphics

  Using WinCC flexible, projects can be created in different languages:

  - The Project Languages editor is used to manage the languages in which the projects should run.
  - The Project Texts editor is used to manage and translate language-dependent texts centrally.
  - The Graphic editor is used to manage language-dependent graphics.
  - The User Dictionary editor is used to create and manage dictionaries for translating project texts. The System Dictionary editor is used to view the system dictionary integrated in WinCC flexible.

- Tags

  Tags are created and edited in the Tags editor.

- Cycles

  It is possible to configure events in WinCC flexible which reoccur at regular intervals. The time intervals are defined in the Cycles editor.

- Alarms

  Alarms are created and edited in the Analog Alarms and Discrete Alarms editors.

- Logs

  The Alarm Log editor is used to log alarms in order to record operating statuses and faults which occur in a system.

  The Data Log editor is used to compile, process and log process values.

www.ElectricalPartManuals.com

- Protocols

  The Reports editor is used to create reports with which the user prints alarms and process values, for example, in runtime.

- Scripts

  WinCC flexible provides the option of making your projects dynamic using custom scripts. The scripts are managed in the Scripts editor.

The following tasks can also be completed in WinCC flexible:

Task	Editor
Configuration of controllers	Connections
Setting up users, user groups and assigning user rights for operation in runtime	Runtime user administration
Managing task-related jobs. It is possible to execute a job once or several times.	Scheduler
Setting up the device settings, such as start screen, language used.	Device settings
Managing different project versions	Version management

Unit dependency and editors

The representation of the project in the WinCC flexible Project View and the functional scope of the editors are dependent on the HMI device selected. Please refer to your unit manual to determine which objects and editors are available on your HMI device.

Tabular editors and image editors

Graphical editors, such as the Screen editor, display the elements belonging to both the Project View and Object View. You open each object in the work area with graphical editors.

Tabular editors, such as the Tag editor, only display the associated objects in the Object View. When a tabular editor is opened to edit the objects, all associated objects are displayed in a table in the work area

www . ElectricalPartManuals . com

3.4.2 Displaying projects

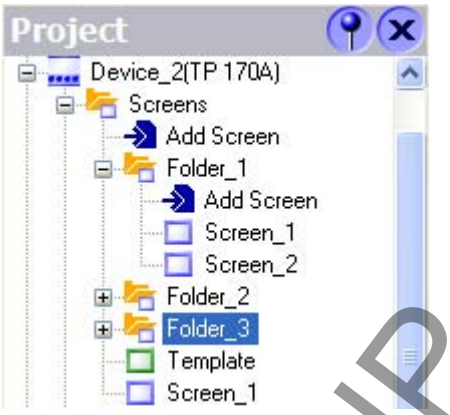
Principle

All component parts and editors available in a project appear in a tree structure in the Project View.

Displaying a project in the Project View

All editors available are displayed under the project node in the Project View. The objects in a project can be edited using the various editors.

Folders are provided as sub-elements of each editor in which you can save objects in a structured way. In addition, direct access to the configured objects is available for screens, recipes, scripts, logs and reports.



The Project View display is dependent on the HMI device selected when the project was created. Only those editors are displayed which are supported by the HMI device selected. If, for example, a "TP170A" is configured, the "Log" editor is not available because the "TP170A" does not have a logging function.

Select the project objects which are to be edited in the Project View. To do this, double-click on the relevant object. The corresponding editor will open.

Displaying objects in the Object View

The Object View displays the content and associated information of the respective folders and editors selected in the Project View. The Object View is displayed below the Project View by default.

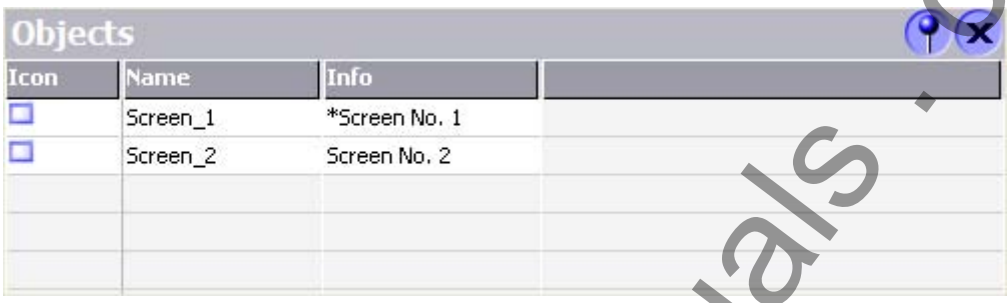
The Object View is comprised of three columns:

- Object type
- "Name"
- Name of the object

www.ElectricalPartManuals.com

- "Info"

  Brief information, e.g. a comment entered by the configuration planner



Objects are displayed in the Object View by the following icons:

Symbol	Brief description	Symbol	Brief description
	Figure		Data log
	Tag		Alarm log
	Analog alarm		Protocol
	Discrete alarms		Connection
	System alarms		Cycle
	Alarm class		Text list and graphic list
	Alarm group		Users
	Recipe		User groups
	Task		

Working with the Object View

Double-click an object in the Object View to open the corresponding editor.

In addition, drag-and-drop actions can be performed on all objects displayed in the Object View. The following drag-and-drop actions, for example, are supported:

- Moving a tag to a process screen in the work area: This creates an I/O field linked to the tag.
- Moving a process screen to another process screen in the work area: Creates a "Change screen" button to the relevant process screen.

www . ElectricalPartManuals . com

3.4.3 Working in the Project View

Principle

The representation of the project in the Project View can be used to edit projects.

The following actions can be executed in the Project View:

- Double-click
- Select a command in the context menu
- Drag-and-drop actions

Double-click

A folder is opened by double-clicking on the folder in the Project View.

The editor is opened by double-clicking on an editor, e.g. "Tag" editor, or on an object, e.g. a screen, in the Project View.

Context menus

After positioning the pointer on an object or folder and clicking the right mouse button, the respective context menu appears. The following actions are available in the context menu:

Action	Description
"Open editor"	Opens the editor
"Add folder"	Creates a new subfolder. The creation of subfolders enables the user to sort the objects according to topics.
"Delete"	Deletes the object or folder selected.
"Rename"	Enables the object or folder selected to be renamed.
"Undo"	Reverts the last process.
"Cut"	Copies the object or folder in the clipboard and deletes it.
"Copy"	Copies the object or folder in the clipboard.
"Paste"	Inserts the object stored in the clipboard.
"Print selection"	Prints the object or folder selected.
"Cross-reference"	Shows all places of use for the selected object or folder.
"Properties"	Shows the properties of the object or folder selected.

www . ElectricalPartManuals . com

Drag-and-drop actions

Drag-and-drop can be used for the following actions:

- Inserting objects in an editor

  Drag an image from the Project View and drop it in another screen. The screen is then assigned a button which, when clicked, switches the screen content back to the first screen.

- Moving or copying objects in subfolders

  If the Project View simultaneously contains objects and subfolders, an object can be moved to a subfolder by means of drag-and-drop or copied.

3.4.4 Working in the Object View

Principle

The Object View provides an overview of the objects.

The following actions can be executed in the Object View:

- Double-click
- Select a command in the context menu
- Drag-and-drop

Double-click

A folder is opened by double-clicking on the folder in the Object View.

After double-clicking on an object (e.g. a screen) in the Object View, the editor opens.

Context menus

The following actions are available in the context menu:

Action	Description
"Open editor"	Opens the editor
"Add folder"	Creates a new subfolder. The creation of subfolders enables the user to sort the objects according to topics.
"Delete"	Deletes the object or folder selected.
"Rename"	Enables the object or folder selected to be renamed.
"Undo"	Reverts the last process.
"Cut"	Copies the object or folder in the clipboard and deletes it.
"Copy"	Copies the object or folder in the clipboard.

www.ElectricalPartManuals.com

Action	Description
"Paste"	Inserts the object stored in the clipboard.
"Print selection"	Prints the object or folder selected.
"Cross-reference"	Shows all places of use for the selected the object or folder.
"Properties"	Shows the properties of the object or folder selected.

Drag-and-drop

Drag-and-drop can be used for the following actions:

- Inserting objects in an editor

  Using drag-and-drop, an object can be dragged from the Object View into any editor when the editor permits editing of the object. An example of its application is the linking of tags to a screen. If a tag is dragged from the Object View into a screen, an I/O field is automatically created.

- Moving or copying objects in subfolders

  If the Object View contains both objects and subfolders, an object can be moved to a subfolder by means of drag-and-drop or copied.

3.4.5 Migrating existing projects

Migrating projects from ProTool and WinCC

Projects can also be opened in WinCC flexible which were created with ProTool or WinCC. Such projects are automatically converted when the WinCC flexible edition installed supports the HMI device defined.

Instead of a file of the type "HmiProjects", open one of the following types in the "Open" dialog:

- ProTool project

  On opening the project, all data is converted. Afterwards, the project can only be saved as a WinCC flexible project.

- WinCC project

  WinCC Version 6 projects can only be migrated to a very limited degree to WinCC flexible.

www . ElectricalPartManuals . com

3.5 Reusing project data

3.5.1 Using libraries

Principle

Libraries enable the multiple use of objects. Libraries can be used to save all types of objects from simple graphics up to complex modules.

WinCC flexible provides different libraries for different tasks:

- Shared libraries

  A global library is saved as a file in the file system independent of a project (in the installation directory of WinCC flexible by default). Global libraries are available for all projects.

- Project libraries

  A project library is saved together with the project data in the database and is only available in the project in which it was created.

It is possible to exchange objects between the two libraries.

Objects in libraries

All objects which can be moved by means of drag-and-drop can be saved in libraries, e.g. graphic objects, screens, alarms and tags.

When a object which has references to other objects is saved in the library, it is possible to select whether the referenced objects should also be saved in the library. A reference object can be a tag, for example, for an I/O field.

Configuration of libraries

The following configuration options are provided for libraries:

- Creating folders to organize the objects
- Changing the display of the library objects

  It is possible, for example, to display small icons or library objects without names.

- Multilanguage configuration of library objects

www . ElectricalPartManuals . com

3.5.2 Using faceplates

Principle

Faceplates are groups of preconfigured objects. Faceplates extend the number of screen objects available and reduce the amount of work for configuration. Faceplates are created and edited in the faceplate designer.

This editor is used to define the faceplate properties which can be configured when put in use. These properties can be the properties of the objects contained.

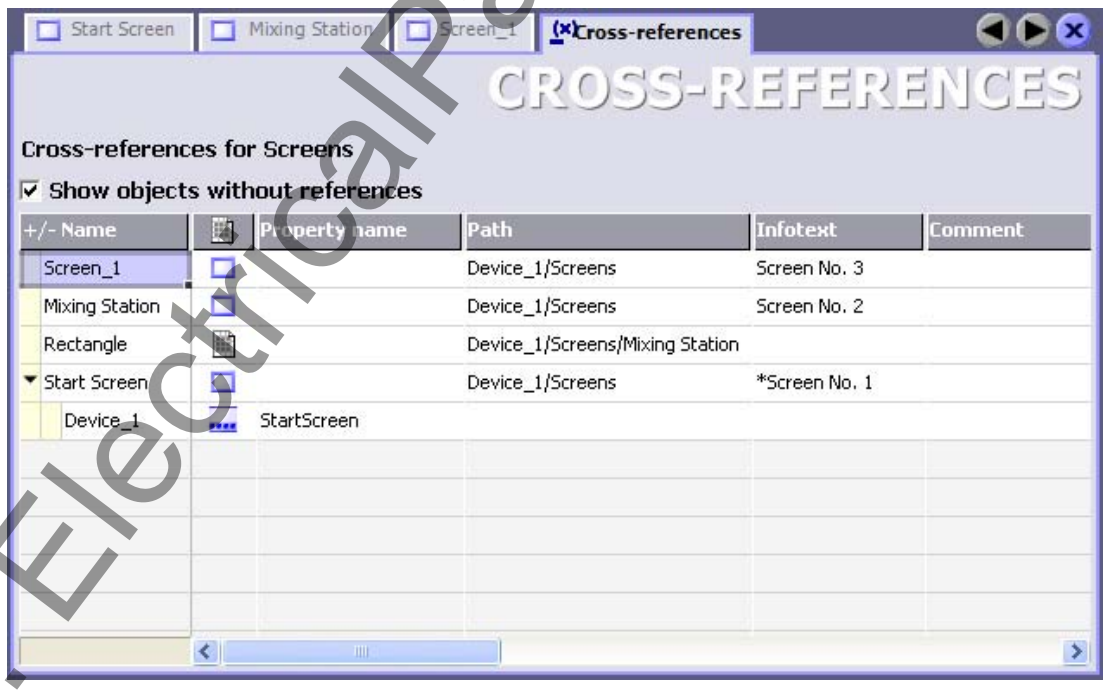
In addition, all of the places where a faceplate is used can be managed centrally in the "Faceplate" editor. After changing the faceplate, either all or only selected place where a faceplate is used can be updated.

3.6 Working with the cross-reference

Principle

The "Cross Reference" editor enables all places of use for specific objects, e.g. tags or screens, to be located and to skip to those points directly.

Cross-reference user interface



The "Cross Reference" editor contains the following elements:

- Object
- Places of use points for the object

www.ElectricalPartManuals.com

The following information is displayed for each object:

- Name

  The name of the object
- Object icon
- Property name

  The name of the property by which the referencing object uses the selected object.
- Path

  The name of the path in the "Project View" where the object is stored, e.g. "Device/screens/motor", when the object has been inserted in the "Motor" screen.
- Operator note

  Operator notes about the object which the user can call in during runtime.
- Comment

  Comment explaining use of the object

Operating options in the cross reference

Use the "Go to usage" command to skip directly to the point of usage in the project. Alternatively, double-click on the point of use.

It is possible to change the view in the "Cross reference" editor. The following commands are available to do this:

- Collapse all

  The "Collapse all" command is used to hide the list of places of use for an object.
- Expand all

  The "Expand all" command is used to recall the list of places of use.

3.7 Internal project find and replace feature

Principle

WinCC flexible enables character strings and objects to be found and replaced.

- Character strings can be searched for and replaced in certain editors. Enter the character string to be searched for in the "Find and replace strings" toolbar Alternatively, use the "Find in work area..." dialog.
- Objects can be found and replaced within entire projects. Use the "Find in project" dialog in this case.

www . ElectricalPartManuals . com

3.8 Basic principles on documentation in WinCC flexible

Principle

Use the project documentation to recall an overview of the configuration data.

The project documentation can be provided as follows:

- Displayed on screen.
- Output as file, e.g. PDF or HTML
- Output via a printer.

If only certain parts of the project data need to be used in the project documentation, select the corresponding objects.

3.9 Debugging projects

Introduction

During configuration, the data entered is automatically tested for its plausibility. When you create a new user, the system indicates that the current password of the user is invalid and you have to assign a new password for the user.

The plausibility test ensures, for example, that value ranges are maintained and incorrect input is indicated during the configuration phase.

There is no check for incorrect parameters in the input, for example, when no tag is assigned in an IO field. The assignment is checked with the "Check consistency" function and displayed as an error.

Consistency test

To locate configuration faults, start the integrity test by clicking the "Check consistency" icon. All faulty points in the project are listed in the Output Window. Then skip to the cause of fault. Work through the fault list from top to bottom.

Testing projects using the simulator

The simulator enables the project to be simulated directly on the configuration computer. The simulation program is an independent program which is installed with WinCC flexible. The simulator enables you to test the response of the configuration by setting values for tags and area pointers.

Tag values can be simulated via a simulation table or enable the system communication with a real PLC can be simulated.

www.ElectricalPartManuals.com

3.10 Transferring projects

3.10.1 Basic Principles of the Transfer Operation

Transfer

A transfer operation refers to the transfer of a complete project file to the HMI devices where the project is to run.

After you have completed a configuration process, check the consistency of the project by using the menu "Project > Compiler > Check Consistency". After completing the consistency check, the system generates a compiled project file. This project file has the same name assigned to it as the project, however with the extension "*.fwx". Transfer the compiled project file to the configured HMI devices.

The HMI devices must be connected to the configuration computer to transfer the project data. If the HMI device is a PC, it is also possible to perform the transfer operation using data media such as diskettes.

Basic procedure

1. Enter the transfer settings for the individual HMI devices in your WinCC flexible project.
2. Enter the transfer mode on the HMI device where the project is to be transferred.
3. Transfer the compiled project file from the configuration computer to the HMI devices. The project file is transferred to all HMI devices for which the respective check box is selected in the transfer settings.

Transfer mode

The HMI device must be in "transfer mode" for the transfer operation. Depending on the type of HMI device, transfer mode is enabled as follows:

- Windows CE systems

  The HMI device starts up automatically in transfer mode when the device is commissioned the first time.

  The HMI device switches automatically to transfer mode at the start of each additional transfer operation if this transfer option is enabled on the configuration menu of the HMI device.

  If not, restart the HMI device and call the transfer applet on the Start menu, or configure the "Change Operating Mode" system function in your project.

- PCs

  If the HMI device is a PC that does not yet contain a project, you must enable the transfer mode in the "RT Loader" manually before the first transfer operation.

Refer to your product manual for more detailed instructions on setting the transfer mode on the HMI device.

www . ElectricalPartManuals . com

HMI device version

When transfering a project onto the operator device, the system checks if the configured operating system version corresponds with the version on the HMI device. If the system finds different versions the transfer is aborted and a message is displayed. You have following possibilities if the version of the operating system in the WinCC flexible project and on the HMI device are different:

- Update the operating system on the HMI device

  You can find further information in chapter "Operating System Transfer".

or

- Select the corresponding HMI device version in the WinCC flexible project.

  You can find further information in chapter "Project HMI device dependency".

3.10.2 Back transfer of projects

Introduction

When transferring, you can transfer the compressed source data file along with the compiled project file to the HMI device. This source data file is required for the project to be back transferred from the HMI device to a configuration computer.

Use for back transfer

Normally, only the executable project is transferred to the HMI device during a transfer operation. The original project data remain on the configuration device and are thus available to develop the project further in future or for error analysis.

However, on Windows CE devices with an external storage medium and on PCs, you can store not only the compiled project file but also the compressed source data file for the project. This data file can be used at a later time to recover the project from the HMI device or device by back transferring the source data file to a configuration computer.

www . ElectricalPartManuals . com

Advantage:

The back transfer operation enables you to subsequently perform analyses and make changes to an existing project even if the original configuration device is not available or the source file (*.pdf) for the project is no longer available on the configuration device.

Note

You can also use WinCC flexible to transfer the source data file of a ProTool V6.0 project back from the HMI device onto a configuration computer. You can then perform a migration of the ProTool project to a WinCC flexible project.

The source data of a ProTool project which was configured for an operating device not supported by WinCC flexible must be transferred back to a configuration computer with ProTool. Save the ProTool project. Then execute a migration using WinCC flexible.

Requirements for back transfer

- The source data file can only be transferred to the HMI device as part of the transfer operation for the compiled project file. The source data file is transferred along with the compiled project file to the HMI device if the "Enable back transfer" check box is selected in the transfer settings for the respective HMI device.
- There must be sufficient memory available on the HMI device to store the compressed source data file. If the source data file for the back transfer operation is provided by a Windows CE device, this device must have an external memory card. If the HMI device does not have a memory card or if there is insufficient memory space, the transfer is terminated. However, the compiled project file is transferred in its entirety beforehand so that runtime can be started with the transferred project data.

  If the source data of a large project should be stored for back transfer and an Ethernet connection is available to the operating device, you can select a network drive as the storage location rather than the memory card of the operating device. This avoids problems with the storage location.
- If there is no project opened in WinCC flexible, you must select the HMI device on which the source data file for the back transfer operation is located and the loading method in the "Communication settings" dialog prior to carrying out the back transfer operation.

  If a project is open in WinCC flexible, the back transfer operation takes place from each selected HMI device. In this case, the transfer mode selected for this HMI device in the "Transfer Settings" dialog in WinCC flexible is applied.

www . ElectricalPartManuals . com

Transfer and back transfer

When a source file is included in the transfer operation, the project is compressed from the source format (*.pdf) and transferred as a *.pdz file to the external storage medium of the HMI device or directly to the PC.

In the case of a back transfer operation, the *.pdz file is saved on the configuration computer. If a project was open in WinCC flexible during the back transfer, you are prompted to save and close it. Then, the project back transferred is decompressed and opened in WinCC flexible. When saving the project, you must assign a name for the back transferred project.

Caution

WinCC flexible cannot check whether the source data file on the operating unit actually belongs to the project running on the device. If you have performed a transfer operation in the interim that did not include the source data file, old project data may still be on the HMI device. Under certain circumstances, the data will then no longer match the project that is currently running.

Note

Use the back transfer process preferably for small and medium sized configurations in order to keep transfer times as short as possible.

You have the following options when there are numerous project files: Transfer the project file as a compressed *.arj file onto a CF card, for example, using the backup function of the project manager.

www . ElectricalPartManuals . com

4 Working with Tags

4.1 Basics

4.1.1 External tags

Introduction

External tags enable the communication (data exchange) between the components of an automation process, e.g. between the HMI device and the PLC.

Principles

An external tag is the image of a defined storage location in the PLC. You have read and write access to this storage location from both the HMI device and the PLC.

Since external tags are the image of a storage location in the PLC, the applicable data types depend on the PLC which is connected to the HMI device.

If you configure integrated in STEP 7 or SIMOTION Scout when creating the external tags, you can directly access all the tags which were created during the programming of the PLC.

Data types

Basic data types are available for all configurations.

In addition, you can also use other data types for external tags which are intended specifically for the PLC to which a connection exists.

A detailed listing of the basic data types and the data types for a connection to S7 PLCs and S5 PLCs can be found under "Data types if connecting to S7" and "Data types if connecting to S5". Information about data types which are available for connection to other PLCs can be found in the documentation about the respective communication drivers.

Note

In addition to the external tags, area indicators can be used for communication between HMI device and PLC. You can set up and activate the area indicators in the "Connections" editor. Detailed information about the area indicators can be found under "Communication."

www.ElectricalPartManuals.com

4.1.2 Internal Tags

Introduction

Internal tags do not have any connection to the PLC.

Principles

Internal Tags are stored in the memory of the HMI device. Therefore, only this HMI device has read and write access to the internal tags. You create internal tags, for example, in order to execute local calculations.

You can use all basic data types for internal tags. A detailed list of the data types can be found under "Basic data types."

4.2 Elements and basic settings

4.2.1 Tag editor

Introduction

In the tag editor you can create and configure tags.

Open

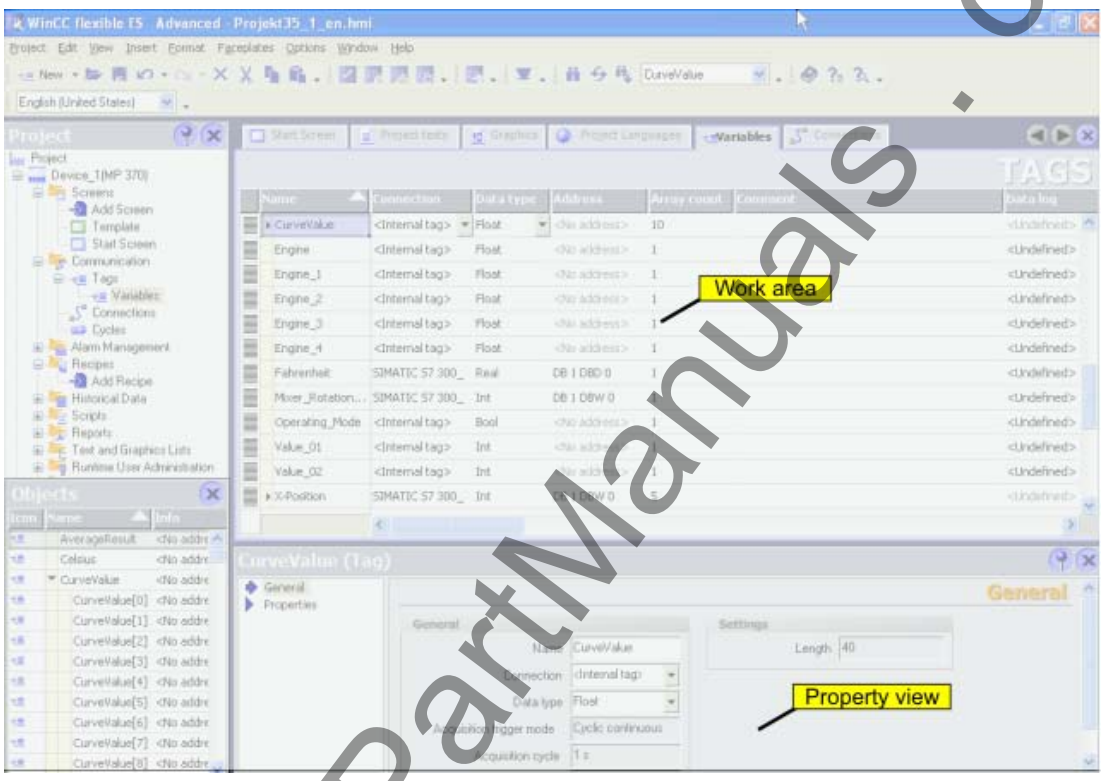
To open the tag editor, create a new tag or edit a tag displayed in the Object window.

As an alternative, you can open the tag editor by double-clicking on the entry "Tags" in the Project window.

www . ElectricalPartManuals . com

Layout

The tag editor displays all tags that are in a folder.



Work area

All tags are displayed in a table in the work area. You can edit the properties of the tags in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

You can configure the selection of columns to suit your needs. Some columns are not available, depending on the HMI device for which you are configuring. The configured column selection will be saved whenever the project is saved. It is linked with the user name that you used when logging into Microsoft Windows.

Property view

Here you configure tags. The property view offers the same information and settings as the work area table.

The property view has a tree structure on the left from which you can select the various property categories. The fields for configuring the selected properties category are shown on the right in the properties window.

www . ElectricalPartManuals . com

4.2.2 Basic Settings for Tags and Arrays

Introduction

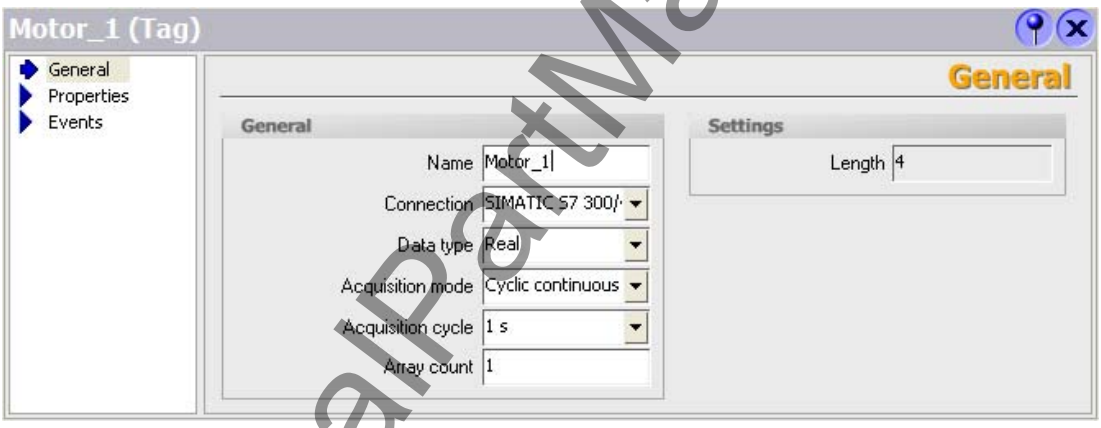
You can configure the properties of tags and array elements in the tabular editors and also in the corresponding property view.

The respective property view offers the same information and settings as the tag editor.

Structure of the property view

All property views have a tree structure on the left from which you can select the various property categories. The fields for configuring the currently selected properties category are shown on the right in the properties view.

Property view for tags



In the property view for tags you can configure the following properties of the selected tag:

Entry in the tree structure	Fields
"General"	"Name" "Connection" "Data type" "Acquisition cycle" "Array count" "Length"
"Properties"	
"Addressing" (for external tags only)	"Icon" (in integrated configuration only) "Area" "DB" "DBW"

www.ElectricalPartManuals.com

Entry in the tree structure	Fields
"Limits"	"Hi limit - disabled" "Hi limit constant" "Upper limit - tag" "Lo limit - disabled" "Lo limit constant" "Lower limit - tag" Limit check Generate alarms
"LinearScaling" (for external tags only)	"PLC end value" "PLC start value" "HMI device - end value" "HMI device - start value"
"Basic settings"	"Update code" "Continuous update"
"Comment"	Text field for entering the comment
"Logging"	"Data log" "Type of sampling" "Logging cycle"
"Logging limits"	"Hi limit constant" "Upper limit - tag" "Upper limit - no limit value" "Lo limit constant" "Lower limit - tag" "Lower limit - no limit value"
"Events"	
"Hi limit exceeded"	List of functions that will be processed if the hi limit is exceeded
"Change"	List of functions that will be processed if the process value changes
"Lower limit exceeded"	List of functions that will be processed if the value drops below the lower limit

www . ElectricalPartManuals . com

4.3 Working with Tags

4.3.1 Properties of a Tag

Introduction

In WinCC flexible, certain properties can be configured for every tag.

The properties determine how you can use the tag in your projects.

Principle

The following properties can be set for tags:

- "Name"

  Every tag has a name which you can choose. Note, however, that the name may only occur once within the tag folder.

- "Connection" to PLC and tag "Logging cycle"

  For external tags, you must specify the PLC to which the HMI device is connected since these tags represent memory locations in the PLC. The available data types for a tag and their address in the PLC memory depend on the type of PLC.

  Furthermore, you must specify how often the tag should be updated.

- "Data type" and "Length"

  The data type of a tag determines which type of values will be stored in a tag, how these are saved internally and the maximum value range that can be held by the tag.

  Two simple examples of data types are "Int" for saving integers or "String" for saving character strings.

  For text tags of the type "String" or "StringChar", you can also set the "Length" of the tag in bytes. For all other data types, the value of "Length" is fixed.

- "Array count"

  You can assemble tags from a number of the same type of array elements. Array elements are saved in consecutive memory locations.

  Array tags are primarily used when working with larger quantities of the same form of data, e.g. for a curve buffer or in the definition of a recipe.

- "Comment"

  You can enter a comment for each tag to provide for a more exact documentation of your project.

- "Limits"

  You can specify a value range with an upper and lower limit range for each tag. If the process value, which should be stored in the tag, enters one of the limit ranges, alarm messages can be sent. If the process value lies outside the value range, a function list for sending messages can be processed.

www.ElectricalPartManuals.com

- "Start value"

  You can configure a start value for every tag. The tag will be set to this value at runtime start. In this manner, you can ensure that the project will begin in a defined state every time it is started.

- "Logging" and "Logging limits"

  To ease documentation and later evaluation, data can be stored in different logs.

  You can set the frequency and mode of logging.

  Furthermore, in WinCC flexible it is possible to limit logging to data that is within or outside specified logging limits.

All properties which were configured when the tag was created can be modified with the object list later where the tag is used.

Example: Create a tag and configure its limit values. Link this tag to an IO field. The limit values which were set when the tag was created can be modified with the object list later when the IO field is configured.

4.3.2 Communication with the PLC using external tags

Introduction

External tags are used to exchange data between an HMI device and PLC.

Principle

An external tag is the image of a defined memory location in the PLC. You have read and write access to this storage location from both the HMI device and the PLC.

The fact that the HMI device can access data on the PLC affects which properties are available when you configure the tags. The configuration possibilities supported by the following tag properties depend on the PLC that is connected to the HMI device:

- "Addressing"
- "Data type"

With linear scaling, you can adjust the value range of external tags to suit the requirements of the configuration.

Addressing

If you create an external tag in WinCC flexible, you must specify the same address as it has in the PLC program. This enables both the HMI device and the PLC to access the same memory location.

Note

When you create the external tag in an integrated configuration environment, you can directly access the icon in the symbol table which was created when the PLC was programmed using STEP 7 or SIMOTION Scout. In this case, you need only select the icon which represents the tag. All other settings will then be made by WinCC flexible in accordance with the PLC program.

www.ElectricalPartManuals.com

Data type

Since external tags represent an image of a specific memory location in the PLC, the data types available depend on the PLC that is connected to the HMI device.

A detailed listing of the basic data types and the data types for a connection to S7 PLCs and S5 PLCs can be found under "Data types if connecting to S7" and "Data types if connecting to S5". Information about data types which are available for connection to other PLCs can be found in the documentation about the respective communication drivers.

Acquisition cycle

The acquisition cycle determines when the HMI device will read the process value of an external tag. Normally, the value is updated at regular intervals as long as the tag is shown in the process screen or is logged. The interval for regular updates is set with the acquisition cycle. You can either choose a default acquisition cycle, or define a user-specific cycle.

An external tag can also be updated independent of the display in the process screen, for example, by triggering a value change for the tag functions. Please note that frequent read operations increase communication load.

Linear Scaling

For numeric data types you can configure a linear scaling. The data in a PLC for an external tag can be mapped to a specific value range in the WinCC flexible project.

Example: The user enters length dimensions in centimeters although the controller expects inches. The entered values are automatically converted before they are forwarded to the controller. Using linear scaling, the value range [0 ...100] on the PLC can be mapped to the value range [0 ...254] on the HMI device.

4.3.3 Tag limit values

Introduction

You can define a value range for numerical tags.

Principle

You can specify a value range with an upper and lower limit range for numerical tags.

If the process value of a tag falls into one of the limit ranges, you can have an analog alarm, e.g. a warning, sent.

If the process value exceeds the value range, you can configure this to trigger an analog alarm message or a function list. If the operator enters a value for the tag that is outside the configured value range, the input is rejected and the value will not be entered.

Note

The text of the analog alarm message, which is sent when the limit value is exceeded, can be changed using the analog alarms editor.

www.ElectricalPartManuals.com

Application example

Use the limit values, e.g. to warn the operator in time, if the value of a tag enters a critical range.

4.3.4 Start value of a tag

The value of a tag at the beginning of a project

You can configure a start value for numerical tags. The tag will be preset to this value at runtime start. In this manner, you can ensure that the project begins in a defined state.

In the case of external tags, the start value will be displayed on the HMI device until it is overwritten by the PLC or an operator input.

Application example

You can preset an IO field to a default value. Enter the desired default value as start value for the tag that is linked to the IO field.

4.3.5 Updating the Tag Value in Runtime

Introduction

Tags contain data which change during runtime. Value changes are handled differently at internal and external tags.

Principle

If a start value has been configured for the tag, the tag will be set to this value at runtime start. Tag values change in runtime.

In runtime, you have the following options of changing the tag value:

- by executing a system function, for example, "SetValue."
- by operator input, for example, at an IO box
- A value assignment in a script
- A value change in an external tag in the PLC

www . ElectricalPartManuals . com

Updating the value of external tags

Method for updating the value of an external tag:

- Updating after an acquisition cycle

  Normally, tags are updated after an acquisition cycle as long as the tag appears in a picture or is logged. The acquisition cycle determines the update cycle for tag value updates on the HMI. You can either choose a default acquisition cycle, or define a user-specific cycle.

- When the setting "Cyclic continuous" is activated

  If this setting is activated, the tag will be updated in runtime, even if it is not found in the currently open screen. This function is set, for example, for tags which are configured to trigger a function list in the event of a change in their value.

  Only use the "Cyclic continuous" setting for tags that must truly be updated. Frequent read operations increase communication load.

4.3.6 Logging tags

Introduction

In runtime, tag values can be stored in logs for later evaluation. For the logging of a tag, you must specify the log in which the values are to be stored, how often this should happen and whether only the tag values in a specific value range are to be saved.

Note

The main purpose of data logging is to log the values of external tags. However, you can also log the values of internal tags.

www . ElectricalPartManuals . com

Principle

Several steps are involved in data logging:

- Creating and configuring data logs

  When creating a data log, you must define the following:

  - General settings, e.g. name, size, storage location
  - Behavior at runtime start
  - Behavior when the log is full

- Configuring the logging of tags

  You can specify a data log for every tag. This log records the values of the tags in runtime and other information, e.g. the time the value was logged.

  Furthermore, you can define when and how often the value of the tag should be logged. To perform the latter, you have the following options:

  - "On request":

    The tag values are logged by calling the "LogTag" system function.

  - "On change":

    The tag values are logged, as soon as the operator device detects a change of value in the tag.

  - "Cyclic continuous":

    The tag values are logged at regular intervals. In addition to the standard cycles available in WinCC flexible, you can add cycles of your own, which are based on the standard cycles.

    Furthermore, you can restrict the logging to those values that are within or outside of a tolerance band. In this manner, you can distribute tag values specifically to different logs for separate analysis later.

- Processing logged tag values further

  The logged process tag values can be evaluated directly in your WinCC flexible project, e.g. in a trend view, or with another application, e.g. Excel.

www . ElectricalPartManuals . com

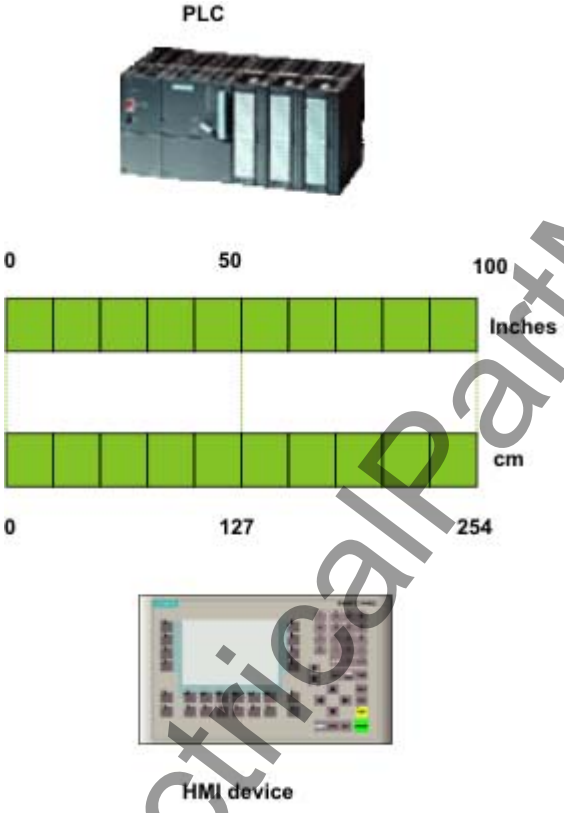
4.3.7 Linear scaling a tag

Introduction

Numeric data types can be processed with linear scaling. The data in a PLC for an external tag can be mapped to a specific value range in the WinCC flexible project.

Principle

To apply linear scaling to a tag, you must specify one value range on the HMI device and one on the PLC. The value ranges will be mapped to each other linearly.



As soon as data from the HMI device is written to an external tag, it will be automatically mapped to the value range of the PLC. As soon as data from the HMI device is read from the external tag, a corresponding transformation will be performed in the other direction.

Note

You can also use the system functions "LinearScaling" and "InverseLinearScaling" to automatically convert process values.

www.ElectricalPartManuals.com

Application example

The user enters length dimensions in centimeters although the controller expects inches. The entered values are automatically converted before they are forwarded to the controller. Using linear scaling, the value range [0 ...100] on the PLC can be mapped to the value range [0 ...254] on the HMI device.

4.3.8 Indirect addressing of tags

Principle

In multiplexes, a type of indirect addressing, the tag used is first determined at runtime. A list of tags is defined for the multiplex tags. The relevant tag is selected from the list of tags in runtime. The selection of the tag depends on the value of the index tag.

In runtime, the system first reads the value of the index tag. Then the tag which is specified in the corresponding place in the tag list is accessed.

Application example

Using indirect addressing, you could configure the following scenario:

The operator selects one of several machines from a selection list. Depending on the operator's selection, data from the selected machine will be displayed in an output field.

To configure such a scenario, configure the index tag for a symbolic IO field. You configure the multiplex tag for an IO field. Configure the tag list of the multiplex tag to reflect the structure of the selection list.

If the operator selects another machine, the value of the index tag will change. The selection field will then display the content of the tag which is pointed to in the tag list (in the multiplex tag) by the new index value.

4.4 Array basics

Introduction

Create a tag array consisting of several elements of the same type to save large volumes of data of the same type. Array elements occupy a continuous address area.

A complex tag containing array elements is referred to as an array tag. Array tags are used, for example, to display process values from different points in time in a trend. Using an index tag, you can control which array element will be accessed.

www . ElectricalPartManuals . com

Principle

Array tags consist of a configurable number of array elements in which data of the same type can be stored. Each array tag element requires the same memory space. All array tag elements are saved consecutively in memory.

Note

Read and write operations always access all array tag elements. The contents of an array tag which is interconnected with a PLC are always transferred whenever there is a change. This is why the HMI device and the PLC can not concurrently write access the same array tag.

Note

The VBS function "IsArray()" cannot be used for array tags that you have created in the tag editor.

Array element properties

The individual array elements take the majority of their properties from the array tag. The properties include, for example, the first part of the array element name, the data type, the length of the array element or the process value log.

Application example

Use array tags whenever you need to continually read a value or when you need to configure multiple tags of the same type. Some examples include:

- Profile trends
  To selectively access the value of the profile trend, configure an array tag. By incrementing the index tag, you can output all values of the profile trend.
- Recipes
  If you have many tags of the same type, you can configure an array tag with the corresponding number of array elements instead. In this manner, you can save time since you only need to configure a single tag. Additionally, in runtime the performance will be better when transferring the process data.

License rule for runtime

In WinCC flexible Runtime, an array will be counted as 1 tag regardless of the number of array elements.

www . ElectricalPartManuals . com

4.5 Cycle basics

Introduction

Cycles are used to control project sequences that are run at regular intervals. Common applications are the acquisition cycle, the logging cycle and the update cycle. Besides the cycles predefined in WinCC flexible, you can also define your own cycles.

Principle

In runtime, actions that are performed at regular intervals are controlled by cycles. Typical applications for cycles include the acquisition of external tags, the logging of data and the updating of screens.

- Acquisition cycle

  The acquisition cycle determines when the HMI device will read the process value of an external tag from the PLC. Set the acquisition cycle to suit the rate of change of the process values. The temperature of an oven, for example, changes much more slowly than the speed of an electrical drive.

  If the acquisition cycle is set too low, it will strongly increase the communication load on the process.

- Logging cycle

  The logging cycle determines when data will be saved in the log database. The logging cycle is always an integer multiple of the acquisition cycle.

- Update cycle

  The update cycle determines how often a screen will be refreshed.

The smallest possible value for the cycle depends on the HMI device that will be used in your project. For most HMIs, this value is 100 ms. The values of all other cycles are always an integer multiple of the smallest value.

If the standard cycles predefined in WinCC flexible do not satisfy the requirements of your project, it is possible to define your own cycles. These custom cycles must, however, be based on the standard cycles.

Application example

Use cycles, for example, to log your process at regular intervals or to provide reminders of the maintenance intervals.

www . ElectricalPartManuals . com

4.6 Importing Tags

4.6.1 Importing Tags in WinCC flexible

Introduction

WinCC flexible 2005 allows you to import tags from an external data source. To successfully accomplish this, the data to be imported must meet the requirements described in this section. You export the tag data to an Excel file from a PLC program. The exported data must be prepared according to rules and can then be imported in WinCC flexible. There are utilities available for preparing the exported tag data from some PLC programs for the import into WinCC flexible. The first version of such a utility for Allan Bradley is available on the product CD. The latest versions of these applications can be be downloaded from the Internet at the following address:

http://support.automation.siemens.com/ww/view/en/16502367/133100

Procedure for Tag Importing

Two files are needed for a complete import of tag data. One file contains the information for connecting to the PLC and the other contains the data from the tag. The connection information is first imported to allow the data type and address parameters to be checked. Then the tag data is imported. Before beginning the import, you can specify whether or not to overwrite existing connections or tags with the same name.

4.6.2 Settings for the Tag Import

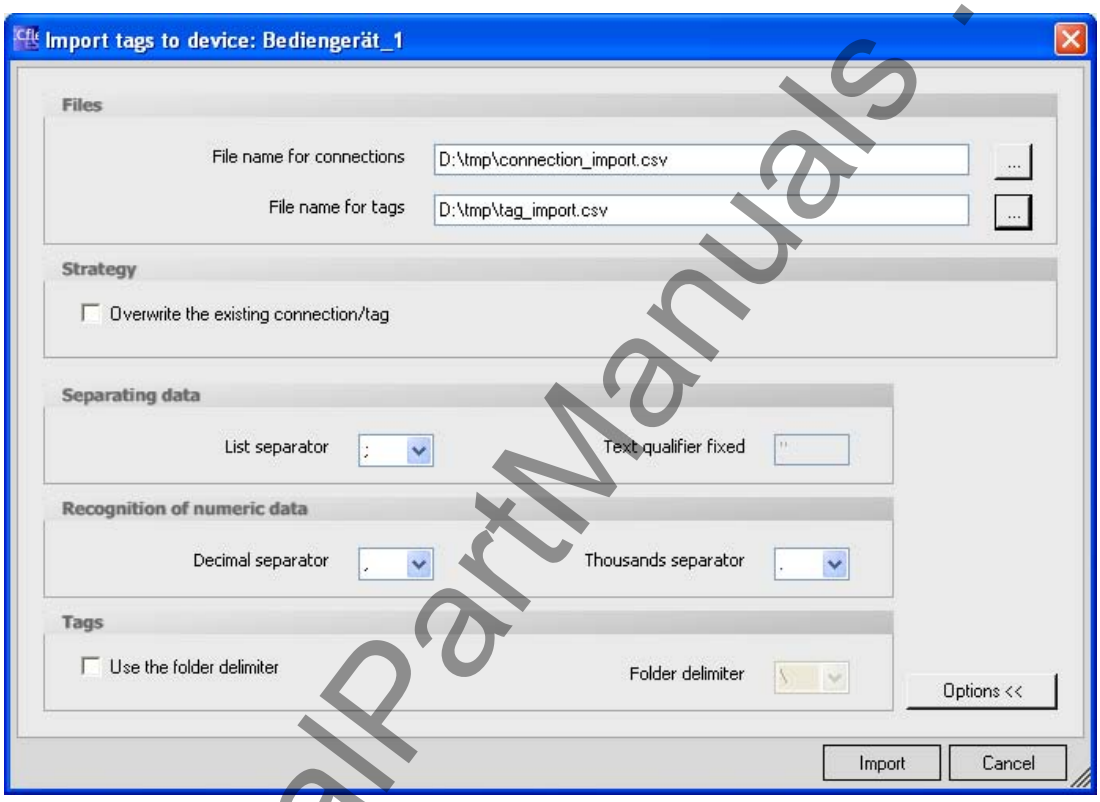
Introduction

The "Import tags to device" dialog is provided for importing tags. Make the required settings in this dialog so that the import files can be correctly interpreted.

www . ElectricalPartManuals . com

Importing Tags into an HMI Device

To open the "Import tags to device" dialog, mark the desired device in the project window and select the context menu command "Import Tags...". You can also mark the HMI device and select the menu command "Project > Import Tags". Click on the "Options" button to make the settings for the import.



In the "Files" area, enter the location of the import files or navigate to the storage location and select the desired files. The import files must be in "*.csv" format. The validity of the file names is checked before the import begins.

If you enable the "Overwrite the existing connection/tag" check box, previous connections or tags with the same name will be overwritten by the import. If you disable the check box, connections and tags with the same name in WinCC flexible are not imported.

In the "List separator" field, select a character to be used to separate the individual parameters for connections and tags. Refer to the sections "Format of the Connection Data for the Import" and "Format of the Tag Data for the Import" for more information.

The text qualifier is used to identify text and strings. The characters enclosed in quotation marks will be interpreted as text. For example, when you import text that includes characters used as control characters for the import, they should be enclosed in quotation marks. Quotation marks are the default text qualifiers but other characters can also be used.

www . ElectricalPartManuals . com

You can specify the separators for decimals and thousands to identify numeric data. Select from the list of characters offered in the respective fields. These two characters cannot be the same. Quotation marks cannot be used for these separators.

The "Use folder delimiter" option allows you create a folder tree with the tag names. The folder tree is created in WinCC flexible and the tags saved in the folders. Select the separator for the folder tree in the "Folder delimiter" field.

Example:

The tag name is "Folder1\Tag_01". The folder separator is "\". In WinCC flexible, "Folder1" is created in the project window under "Communication/Tags" and "Tag_01" is saved in it.

4.6.3 Format of the Connection Data for the Import

Introduction

This section describes the format required for the file with connection data used for imports. The connection data file must be in "*.csv" format.

Format of Connection Data

Each connection is on a separate line in the import file. The import file with the connection data must have the following format:

<Name of connection><list separator character>**<Name of communication driver>**<list separator character>**<Comment>**<Line break (carriage return)>

Meaning of entries

List entry	Meaning
Name of the connection	Specifies the configured name of a connection. This entry is needed to match up to the corresponding entry in the import file of the tags. The list entry for "Name" cannot be empty. The name cannot contain an apostrophe (').
List separator character	The list separator character separates the individual entries in the list. You can select the character to be used for the list separator in the dialog for importing. The following characters are offered for use: Semicolon ";", comma "," and period ".". Use the selection field to select a character other than that shown, if needed.

www . ElectricalPartManuals . com

List entry	Meaning
Name of the communication driver	Specifies the name of the communication driver used in WinCC flexible. It must exactly match the name used in WinCC flexible. The following names are available: • Allen Bradley DF1 • Allen Bradley DH485 • GE Fanuc SNP • LG GLOFA-GM • Mitsubishi FX • Mitsubishi protocol 4 • Modicon MODBUS • Omron Hostlink/Multilink • OPC • SIMATIC S5 AS511 • SIMATIC S5 DP • SIMATIC S7200 • SIMATIC S7300/400 • SIMATIC 500/505 serial • SIMATIC 500/505 DP • SIMOTION • SIMATIC HMI HTTP Protocol
Comment	Any comment about the connection. You can use up to 500 characters.
Line break	The line break (carriage return) separates the entries for one connection from the entries of the next.

Format of the Import File for Connections

An import file for connections has the following format:

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

A comma, for example, can be used as the list separator character. If a list entry is empty, there are two consecutive list separator characters. When you are finished defining the entries on a line, you do not have to put a list separator character at the end of the line.

Note

An example of an import file is located in the "Support\Tag Import" folder on the WinCC flexible CD.

Editing the Import File

You can use programs such as MS Excel or a text editor to edit the import file. Do not double-click on the import file to open it in MS Excel because this may change the data format and the import will fail. Instead, start MS Excel and then use the "Open" command from the "File" menu. Select "Text files" (*.prn; *.txt; *csv)" from the "File type" list. Open the file in a simple text editor to check that the import file has the correct format.

www.ElectricalPartManuals.com

4.6.4 Format of the Tag Data for the Import

Introduction

This section describes the format required for the file with tag data used for imports. The tag data file must be in "*.csv" format.

Format of Tag Data

Each tag is on a separate line in the import file. The import file with the tag data must have the following format:

<Tag name><list separator character>**<Connection name>**<list separator character> **<Tag address>**<list separator character>**<Data type>**<list separator character> **<Length of tag in bytes>**<list separator character>**<Array number>**<list separator character> **< Acquisition trigger mode>**<list separator character>**<Acquisition cycle>**<list separator character> **<Upper limit>**<list separator character>**<High low limit>**<list separator character>**<Low low limit>**<list separator character>**<Lower limit>**<list separator character>**<Linear scaling>**<list separator character>**<Upper PLC scaling value>**<list separator character>**<Lower PLC scaling value>**<list separator character>**<Upper HMI scaling value>**<list separator character>**<Lower HMI scaling value HMI>**<list separator character>**<Start value>**<list separator character>**<Tag ID>**<list separator character>**<Comment>**<line break (carriage return)>

Meaning of entries

List entry	Meaning
Tag name	Specifies the configured name of a tag. You can prefix the tag names with a folder tree you have set with the folder separator character, for example "Foldername1\Foldername2\tagname". If you have enabled the "Use folder delimiter" check box, the folder tree is created when you perform an import into WinCC flexible. The list entry for "Name" cannot be empty. The name cannot contain an apostrophe (').
List separator character	The list separator character separates the individual entries in the list. You can select the character to be used for the list separator in the dialog for importing. The following characters are offered for use: Semicolon ";", comma "," and period ".". Use the selection field to select a character other than that shown, if needed.
Connection name	Specifies the configured name of a connection. This entry is needed to match up to the corresponding entry in the import file of the connections. Each external tag should have a valid entry for the name of the connection. If no name is specified for the connection, an internal tag is created.
Tag address	Specifies the tag address in the PLC. The tag address must exactly match the one used in WinCC flexible, for example, "DB 1 DBW 0" and not "DB1, DBW0". The tag address is empty for internal tags.

www . ElectricalPartManuals . com

List entry	Meaning
Data Type	Specifies the data type of a tag. The data types allowed depend on the the communication driver being used. The following are examples of possible values: 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 bit Block, +/- Double, +/- int, 16 bit group, 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, etc.. For more information about the data types permitted for the various communication drivers, refer to the "Communication" section of the communication driver documentation.
Length of the tag in bytes	Specifies the length of the tag in bytes. The entry is usually only used for string tags; for other data types it is otherwise empty.
Array number	Specifies the number of arrays of a tag. You can define an array with this value. If the entry is empty, WinCC flexible sets the value to "1".
Acquisition trigger mode	Specifies the acquisition trigger mode of a tag. The acquisition trigger mode is represented by numbers. 1 = on demand 2= cyclic on use (default value) 3 = cyclic continuous
Acquisition cycle	Specifies the acquisition cycle of a tag. The acquisition cycle must exactly match the one in WinCC flexible. The value does not depend on the language and therefore must be identical in every language. The default value is "1 s". If the acquisition trigger mode of the tag is "on demand", the acquisition cycle is undefined.
Upper limit High low limit Low low limit Lower limit	The limits can only be set for numeric values. Only constant numeric values can be used as limits, variables cannot. The default value for limits is "No limit". The following conditions apply to limits: "Upper limit" ≥ "High low limit" ≥ "Low low limit" ≥ "Lower limit"
Linear scaling	Specifies if linear scaling is enabled. The entry can only be used for external tags. Default value is "Disabled". The value for linear scaling can be given as numbers or text. Permitted values are: "false" or "0" for "disabled "true" or "1" for "enabled"
Upper PLC scaling value Lower PLC scaling value Upper HMI scaling value Lower HMI scaling value	The upper and lower values can only be set for numeric values. The following conditions apply to upper and lower values: "Upper scaling value" ≥ "Lower scaling value"
Start value	Specifies the start value of a tag. Default values are: 0 for numbers, space for characters, empty string for string tags, current value for time and date.
Tag ID	When the value of a tag changes in the PLC, the tag ID indicate the number of the tag.
Comment	Any comment about the tag. You can use up to 500 characters.
Line break	The line break (carriage return) separates the entries for one connection from the entries of the next.

www . ElectricalPartManuals . com

Format of a tag import file

An import file for tags has the following format:

"tag","Connection","DB 1 DBD 0","Real",,1,3,"1 min",20,10,2,1,1,100,10,10,1,15.5,33,Comment for tag

A comma, for example, can be used as the list separator character. If a list entry is empty, there are two consecutive list separator characters. When you are finished defining the entries on a line, you do not have to put a list separator character at the end of the line. The default value is used for a list entry with no value.

Note

An example of an import file is located in the "Support\Tag Import" folder on the WinCC flexible CD.

Editing the Import File

You can use programs such as MS Excel or a text editor to edit the import file. Do not double-click on the import file to open it in MS Excel because this may change the data format and the import will fail. Instead, start MS Excel and then use the "Open" command from the "File" menu. Select "Text files" (*.prn; *.txt; *csv)" from the "File type" list. Open the file in a simple text editor to check that the file has the correct format.

www . ElectricalPartManuals . com

5 Creating Screens

5.1 Basics

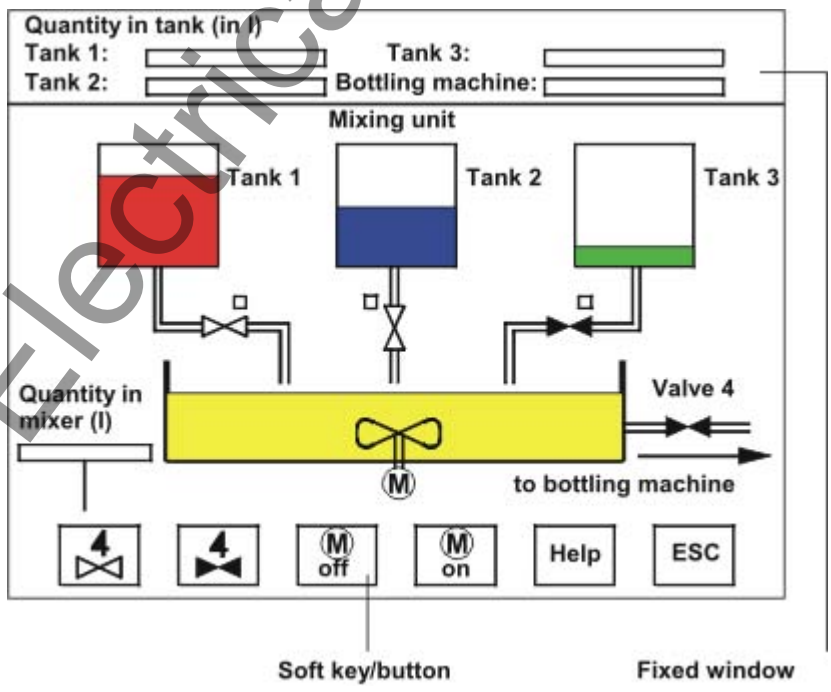
5.1.1 Screen Basics

Introduction

In WinCC flexible, you create screens which an operator can use to control and monitor machines and plants. When you create your screens, the object templates included support you in visualizing processes, creating images of your plant, and defining process values.

Application example

The figure shows a process screen which was created in WinCC flexible. With the help of this screen, you could operate and monitor the mixing unit of a fruit juice manufacturing system. Fruit juice base is supplied from various tanks to a mixing unit. The screen indicates the filling levels of the tanks and of the mixer. The screen also contains operator control elements for the valve units and for the mixer motor.



www . ElectricalPartManuals . com

Screen design

You add screen elements you need for process visualization, and configure these to suit the requirements of your process.

A screen may consist of static and dynamic elements.

- Static elements such as text or graphic objects do not change their status in runtime. The tank labels shown in this example of a mixing plant are such static elements.
- Dynamic elements change their status based on the process. They visualize current process values which are output from the memory of a PLC or operator station, in the form of alphanumeric displays, trends and bar graphs. Operator input boxes also belong to the category of dynamic elements. The filling level values of the tanks in our example of a mixing plant also belong to the category of dynamic screen objects.

The PLC and the operator station exchange process values and operator input data by means of tags.

Screen properties

The screen layout is determined by the features of the HMI device you are configuring. It corresponds with the layout of the user interface of this device. It contains an image of the function keys, for example, provided the selected HMI device is equipped with such keys. Other properties such as the screen resolution, fonts and colors are also determined by the characteristics of the selected HMI.

Function keys and softkeys

A function key is a physical element of the operator station. You can assign one or several functions in WinCC flexible. These functions are triggered when the operator presses the relevant key on the HMI device..

A function key can be assigned global or local functions.

Global function keys always trigger the same action, regardless of the currently displayed screen.

Function keys assigned local functions are softkeys. They trigger different actions, based on the currently displayed screen on the operator station. This assignment applies only to the screen in which you have defined the softkey. The operator control elements for the valve and the motor in our example of a mixing plant represent softkeys.

Navigation

All configured screens must be integrated into the operator control system, in order to enable runtime access to these at the operator station. You have various options of configuring these functions:

- Use the "Screen Navigation" editor to define the screen structure and configure the entire screen navigation system.
- You use the "Screen" editor to configure buttons and function keys for calling other screens.

www.ElectricalPartManuals.com

5.1.2 "Screens" Editor

Introduction

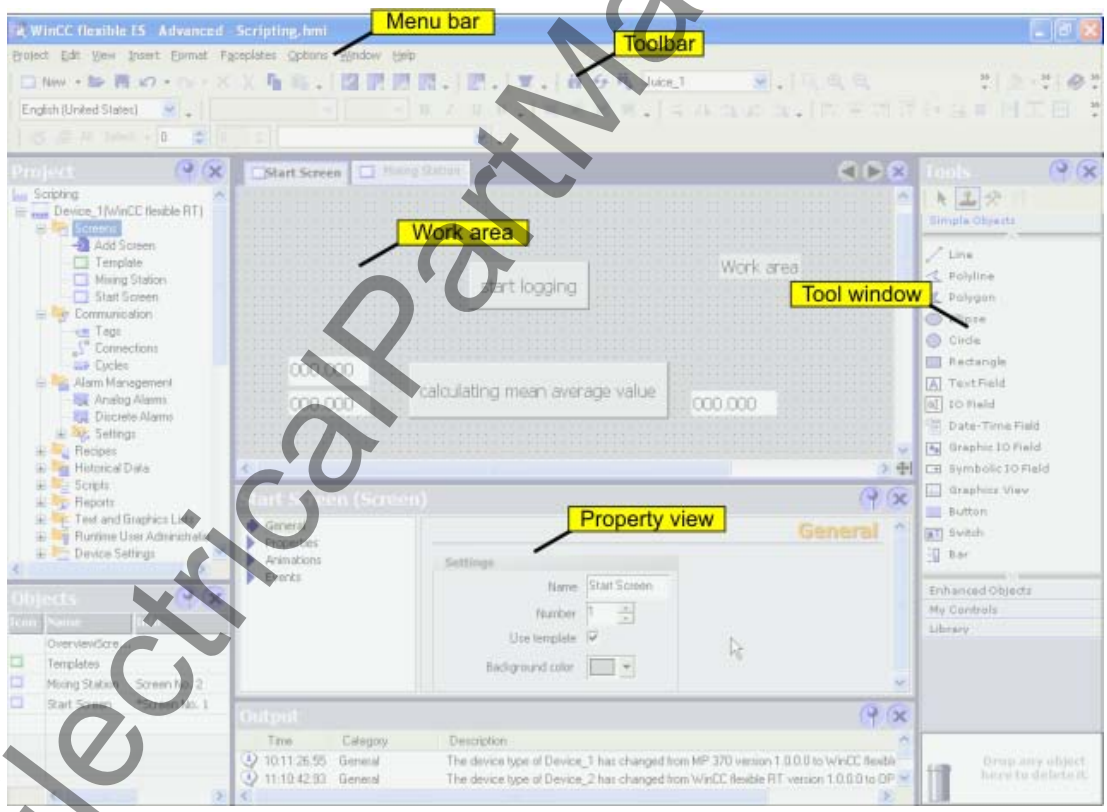
Programming screens in the screen editor. This editor is formed by the combination of a graphic programming software and a process visualization tool. You can access the screen editor in the project view.

Open

Double-click "Add screen" in the "Screens" group of the project view,

The work area opens with a new screen.

Layout



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available shortcut keys are indicated next to the menu commands.

Toolbars

You can hide or show a specific toolbar.

www . ElectricalPartManuals . com

Work area

You configure screens in the work area.

Toolbox

The toolbox contains a selection of simple and complex objects which you can add to your screens, for example, graphic objects or operating elements. In addition, the toolbox also provides libraries containing object templates and collections of faceplates.

Property view

The content of the property view is determined by the object you have currently selected in the work area.

- The properties of a selected object can be viewed and edited in the properties dialog box.
- If you have not selected an object on the active screen, the properties of this screen are shown and can be edited in the property view.

5.1.3 Procedures

Procedures

To create screens, you need to take the following initial steps:

- Create a draft of the process visualization structure, i.e. define the structure and the number of screens.

  Example: Process partitions can be visualized in separate screens and merged in a master screen.
- Define your screen navigation control strategies.
- Adapt the template.

  The template which is stored in WinCC flexible for the selected HMI device applies to all your project screens. In this template, you can define objects locally and assign global functions keys. For some of the HMI devices, you can store the objects you want to integrate into all screens in the permanent window.
- Create the screens. Use the following options of efficient screen creation:
  - Create a screen structure in the "Screen Navigation" editor.
  - Working with libraries
  - Working with faceplates
  - Working with layers

www . ElectricalPartManuals . com

5.2 Configuring the navigation system

5.2.1 Navigating options

Introduction

A WinCC flexible project which consists of multiple screens offers the following screen navigation options in runtime:

- Navigation by means of navigating buttons
- Navigating with the help of function keys
- Navigation by means of the navigation control

WinCC flexible offers the following programming options:

- By programming buttons or function keys
- By graphical configuration via the "Screen Navigation" editor and the navigation control

Note

If you have set the "Visibility" animation to "Hidden" in the project view of a screen, this screen cannot be called up in Runtime.

5.2.2 Graphic programming of the screen navigation system

"Screen Navigation" editor

The "Screen Navigation" editor is used for the graphics configuration of the navigation between several screens. This editor allows you to organize your project screens in a hierarchical structure. Operators can use the navigation control in runtime to navigate between the various screens of the structure, for example, to change to the parent screen, or to the neighboring screen.

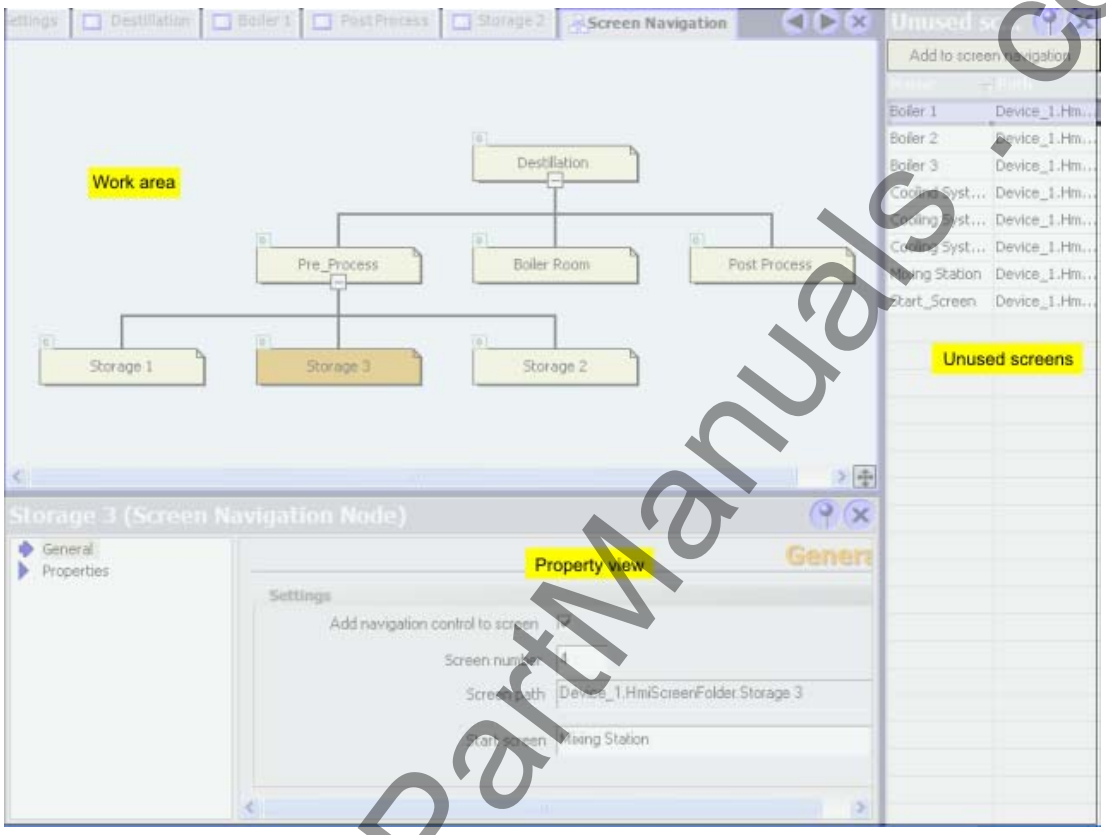
In addition to these structured connections, the "Screen Navigation" editor also allows you to generate direct screen connections, regardless of the given structure.

Open

To open the "Screen Navigation" editor, double-click on the "Screen Navigation" are under the "Device Settings" in the project view.

www . ElectricalPartManuals . com

Layout



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available shortcut keys are indicated next to the menu commands.

Work area

The view of the "Screen Navigation" editor shows the screen structure. The various screens are indicated by rectangles.

The screen interconnections correspond with your navigating options in runtime. The various interconnection types are identified by means of colored lines:

- Black lines reflect the structured screen interconnections.
- Green arrows represent direct screen connections, irrespective of the structure.

Context menu

The context menu contains commands you can use to configure the "Screen Navigation" editor, and to create, open, delete, copy or rename screens.

www.ElectricalPartManuals.com

Customizing the layout of the view

You have several options of customizing the view of the "Screen Navigation" editor:

- You can zoom in or out in the view in order to show either a larger or smaller section of the "Screen Navigation" editor.
- You can move this section using the icon in order to display another section of your view.
- You can show a single screen, including all its child screens.
- You can hide or show all child screens.
- You can toggle between a horizontal and a vertical view.

"Unused screens"

The "Unused screens" view contains all the project screens which are not included in your navigation system. You can drag-and-drop "Unused screens" from this dialog box onto your view, and interconnect these with other screens.

Property view

The "Properties" dialog box of a screen allows you to enable the navigation control, to change the screen number, and to configure direct screen connections.

5.2.3 Using the navigation control

Using the navigation control

You can open a navigation control in all screens. The navigation control contains a number of preconfigured screen navigation buttons. These buttons can be used in runtime to call further screens of the project.

Customizing the navigation control

The navigation control is adapted in the "Screen Navigation" editor:

- You can display or hide the navigation control.
- You can configure the navigation control and the command buttons contained in it

The button is disabled if it is not assigned a target. In this case, the button appears in runtime without a label.

www . ElectricalPartManuals . com

5.3 Working with objects

5.3.1 Overview of Objects

Introduction

Objects are graphic elements which you use to design the process graphics of your project.

The "Toolbox" contains various types of objects which are frequently required for use in process screens.

The Toolbox view can be faded in and out using the "Toolbox" command in the "View" menu. The Toolbox view can be moved to any position on the screen.

The "Toolbox" contains various object groups, depending on the currently active editor. When the "Screens" editor is opened, the toolbox provides objects in the following object groups.

- "Simple objects"

  Simple objects are graphic objects such as the "Line" or "Circle" and standard operator control elements, such as the "I/O field" or "Button".

- "Enhanced objects"

  These objects provide an enhanced functional scope. One of their purposes is to display processes dynamically, e.g. integrating bars or Active X controls in the project, such as the Sm@rtClient view.

- "User-specific controls"

  In this object group, you can add ActiveX controls which are registered in the Windows operating system of your PG / PC to the toolbox, and thus integrate them into your project.

- "Graphics"

  Graphical objects e.g. of machinery and plant components, measuring equipment, control elements, flags and buildings are displayed thematically in a directory tree structure. You can also create shortcuts to your graphic files. External graphics that are placed un this directory and parent directories are shown in the toolbox window and thereby integrated into the project.

- "Library"

  A library contains object templates such as graphics of pipes, pumps, or default buttons. You can integrate multiple instances of a library object in your project, without any need to reconfigure these.

  The WinCC flexible software package includes libraries. You may also store user-defined objects and faceplates in user libraries.

- The "Faceplates" represent preconfigured object groups. Some of their properties, but not all of these, can be configured at the relevant place of their application. The faceplates can be edited from a central location. The use of faceplates helps you to reduce the work involved in configuration and ensure uniform project design.

www.ElectricalPartManuals.com

Simple objects

Symbol	Object	Instructions
	"Line"	You can select straight, rounded or arrow-shaped line ends.
	"Polyline"	A polyline consists of linked paths and can have any number of corners. The corner points are numbered in the order of their creation. The corner points can be modified or deleted individually. You can select straight, rounded or arrow-shaped polyline ends. The polyline is an open object. Although the start and end points may have the same coordinates, the area they enclose cannot be filled in.
	"Polygon"	The corner points of a polygon are numbered in the order of their creation. The corner points can be modified or deleted individually. You can fill a polygon area with a color or a pattern.
	"Ellipsis"	You can fill an ellipsis with a color or a pattern.
	"Circle"	You can fill the circle with a color or a pattern.
	"Rectangle"	The corners of a rectangle can be rounded. You can fill the rectangle with a color or a pattern.
	"Text box"	You can enter one or several lines of text in a "Text box" and define the font and the font color. You can add a background color or pattern to a text box.
	"I/O box"	An I/O box may have the following runtime functions: • Output of the values in a tag • Operator input of values; these input values are saved to a tag. • Combined input and output; the operator can here edit the output value of the tag and thus set a new value. You can define limits for the tag values shown in the IO field. Set "Hide input" if you want to hide operator input in runtime.
	"Date-time box"	A "Date / time box" may have the following runtime functions: • Output of the date and time • Combined input and output; the operator can here edit the output values in order to reset the date and time. The system time or a corresponding tag may be used as source to define the date and time. The date can be output in extended format, for example, Tuesday, December 31, 2003, or in short format, for example, 12.31.2003.

www . ElectricalPartManuals . com

Symbol	Object	Instructions
	"Graphic I/O box"	A "Graphic I/O box" may have the following runtime functions: • Output of graphic list entries • Combined input and output; the operator can here select a graphic from an graphic list in order to change the content of the "Graphic IO field." Example of its use as output field: To indicate the runtime status of a valve, the "Graphic I/O box" outputs the image of a closed or open valve.
	"Symbolic I/O box"	The "Symbolic I/O box" may have the following runtime functions: • Output of text list entries • Combined input and output; the operator can here select a text from a text list in order to change the content of the "Symbolic I/O box." Example of its use as combination I/O box: To control a motor in runtime, the operator selects the text "Motor OFF" or "Motor ON" from the text list. The motor is either started or stopped as selected, and the "Symbolic IO field" indicates the current status of the motor (motor OFF / motor ON.)
	"Graphic view"	The "Graphic view" shows you on one screen all of the graphic objects created by means of an external graphic programming tool. Graphic objects can be shown in the following formats: "*.emf", "*.wmf", "*.dib", "*.bmp", "*.jpg", "*.jpeg", "*.gif" and "*.tif". In the "Graphic view", you can also integrate graphic objects of other graphic programming tools as OLE (object linking and embedding) objects, for example. OLE objects opened and edited in the graphic program in which they were created directly from the property view of the graphic view.
	"Button"	The operator can use a button to control a process. You can configure functions or scripts for a button.
	"Switch"	The switch is used in runtime to input and visualize two states, for example, ON and OFF, or pressed and not pressed. It can be labeled with text or a graphic that indicates the runtime status of the switch.
	"Bar"	The "Bar" represents a process value in the form of a scaled bar graph. A bar graph allows you to visualize, for example, dynamic values of filling levels.

www . ElectricalPartManuals . com

Enhanced objects

Symbol	Object	Description
	"Slider"	The "Slider" is used for operator input and monitoring of numeric values. • When used as display instrument, the slider position indicates a process value which is output by the controls. • The operator inputs values by changing the slider position. You can customize the slider, so that it operates only in vertical direction.
	"Clock"	On your HMI device, you can view the clock in runtime either in digital or in analog format.
	"Status force"	The "Status / control" functions provide direct read / write access to specific address areas of the connected SIMATIC S7 or SIMATIC S5 CPU.
	"Sm@rtClient view"	The operator can monitor and operate a remote operator station by means of the "Sm@rtClient view ."
	"HTML browser"	The operator can view pages in HTML format by means of the "HTML browser."
	"User view"	In WinCC flexible, you can use passwords to control access to screen objects. In the "User view", an administrator can manage users on the HMI device in Runtime. In the "User view", user who do not have administrator privileges can change their password in runtime.
	"Gauge"	The "Gauge" dial can display numerical values in runtime. The layout of the "Gauge" is configurable. You can customize the background image or the dial layout, for example.
	"Trend view"	In the "Trend view", you can show a group of trends which represent process values read from the PLC or from a log. The trend coordinates are configurable, i.e. the scaling, units etc.
	"Recipe view"	The operator can use the "Recipe view" in runtime to view, edit and manage data records.
	"Alarm view"	In the alarm view, the operator can view selected alarms or alarm events in the alarm buffer or the alarm log in runtime.
	"Alarm window"	In the "Alarm window", the operator can view selected alarms or alarm events in the alarm buffer or the alarm log in runtime. You always edit the template to configure the alarm window.
	"Alarm indicator"	The "Alarm indicator" warns the operator of alarm events which are not acknowledged yet. You always edit the template to configure the alarm indicator.

Note

Some of the toolbox objects are either available with restricted functionality or not at all. This depends on the HMI device you are configuring. Objects not available in the toolbox are grayed out and cannot be selected.

www.ElectricalPartManuals.com

5.3.2 Object Groups

Principle

You can organize multiple object in a group. In your screen, you edit an object group in the same way you edit a single object. You can also edit any object of the group separately.

In contrast to the multiple selection function which shows the selection rectangles of single objects, the system displays only one selection rectangle for the complete group.

You can edit any object of the group separately. To do so, change to the single-object editing mode. In this mode, you can access all of the properties of a single object you have selected from the group.

5.4 Options of assigning dynamic update functions

Introduction

All objects for input and output demonstrate dynamic reactions in Runtime. You can also assign dynamic properties for objects. An example of this feature is the graphic of a tank filling level, which is dynamically updated based on the respective process value. Another example of dynamic object properties is a button which triggers a particular function.

Dynamic objects

You can assign dynamic properties to any graphic object. Programming options:

- The object changes its appearance: Color or flashing properties.
- The screen object is animated.
- The object is shown or hidden.

The following additional options are available for the operator control elements:

- Operator control of an object is enabled or locked.
- The operator control, e.g. clicking, of an object triggers an event which is configured to execute a function list.

Dynamic control and object properties

Dynamic update is an element of the object properties. Which dynamic update functions and events are actually available depends on the selected object. When you copy an object, its dynamic update functions are included.

www . ElectricalPartManuals . com

5.5 Working with function keys

Introduction

The function key is a physical key on your HMI device and can be assigned user-defined functions. In WinCC flexible, you can assign one or several functions to a function key. You can assign a press or release trigger to the function key.

A function key can be assigned global or local functions.

Global assignment

Global function keys always trigger the same action, regardless of the currently displayed screen.

Global function keys are configured once in the template. A global assignment applies to all screens of the selected HMI device which are based on this template.

Global function keys reduce programming effort considerably, because there is no need to assign these global keys to each individual screen.

Local assignment

Function keys assigned local functions are softkeys. They can trigger a different action in every screen. Softkeys are assigned locally in the screen. This assignment applies only to the screen in which you have defined the softkey.

You can overwrite the global assignment of a function key with a local assignment.

Note

Local function keys remain active in runtime, even if the screen in which they are assigned is overlayed by an alarm view or an alarm window. This may occur particularly with HMI devices with a small display (e.g. OP 77B).

Hotkey assignment

You can assign hotkeys to operator control objects such as buttons. The hotkeys available depend on the HMI device.

Graphics

When a function key is placed directly next to the display, you can assign a graphic to it to make the function of the softkey more clear.

www . ElectricalPartManuals . com

Visualizing the assignment

While you are programming the keys, the assignment of the function keys is indicated by the following icons:

Softkey	Description
	Unassigned
	Used globally
	Used locally
	Used locally (local assignment overwrites global assignment)
	Assigning hotkeys to buttons

5.6 The Advantage of Layers

Layers

The layers and the nesting depth of the objects form a feature which allows differentiated visualization and editing of screen objects. A screen consists of 32 layers. You can add objects to any one of these layers. The layer assignment of the object determines its nesting depth on the screen. Objects of the layer 0 are located at the screen background, while objects of the layer 31 are located in the foreground.

The objects within the various layers are also nested. When a process screen is created, the objects within the layer are always organized in the order of their creation. The object which was inserted first lies completely at the back within the level. Each further object is placed one position towards the front. The position between objects within the layer can be changed.

www . ElectricalPartManuals . com

Principle of the layer technique

There is always one active layer. New objects you add to the screen are always assigned to the active layer. The number of the active layer is indicated in the "Layer" toolbar. The active layer is highlighted in color in the layer pallet.

When you open a screen, all 32 layers of the screen are displayed. Using the open layer pallet, you can hide all layers except the active layer. This allows you to explicitly edit objects of the active layer.

Application examples

You can use the layer technique, for example, to hide the labels of objects you are currently editing.

Alarm windows should be configured within their own layer in the template. This layer can be hidden later when the screens are configured.

5.7 Object libraries

Introduction

Libraries are a collection of screen object templates. They enhance the collection of available screen objects and programming efficiency, because library objects are always available for reuse without the need to reconfigure them. Your WinCC flexible software package is supplied with comprehensive libraries which contain, for example, "Motor" or "Valve" objects. You may, however, define your own library objects.

Project library

There is one library for each project. Objects of the project library are stored alongside with the project data and are available only for the project in which the library was created. When you move the project to a different computer, the project library created therein is included. The project library remains hidden as long as it does not contain any objects. In the context menu of the library view, select the command "Display project library" or drag the screen object into the library view to display the project library.

www . ElectricalPartManuals . com

Shared libraries

In addition to the objects from the project library, you can also incorporate objects from shared libraries in your projects. A shared library is stored independent of project data in a separate file with the extension *.wlf.

When a shared library is used in a project, you generate only one reference to this library in the relevant project. The shared libraries are not automatically included when you move the project to a different computer. The interconnection between the project and the shared library may be lost in the course of this action. This interconnection will also be lost if the shared library is renamed in a different project or in an application other than WinCC flexible.

A project can access multiple shared libraries. A shared library may be used concurrently in multiple projects.

When a library object is changed by a project, this library is opened in all other projects in this modified state.

Among the shared libraries you will also find the libraries supplied with your WinCC flexible package.

Categories

To sort library objects by topics, you can either split a library into categories, or create several shared libraries. A particular shared library may contain all of the objects you need to configure motor controls, for example. Another shared library may contain all of the objects you need to configure the pump controls.

Library objects

A library may contain all the WinCC flexible objects, e.g. screens, tags, graphic objects or alarms.

To use a library object in a project, copy the object and all referenced objects to the project. The copied object looses its interconnection with the library. Changes in the library do not affect any of the copied library objects.

If you wish to use multiple instances of configurable object groups and edit these centrally, you need to create faceplates.

5.8 Working with faceplates

Introduction

A faceplate is a preconfigured object group which can be edited locally. Faceplates expand your screen object resources, reduce programming effort and, at the same time, ensure a consistent layout of your projects.

Faceplates are created and edited in the faceplate designer. The faceplates you create are added to the "Project library", and can be inserted into screens in the same way as other objects.

www.ElectricalPartManuals.com

Faceplate properties

When you configure a faceplate, you determine which of its properties can be edited and which events can be assigned to it.

The faceplate properties can be derived from the properties of its embedded objects. Moreover, you can also define a range of new properties.

Application example

You may configure a "Bar display" faceplate, for example The faceplate may consist of a bar and a text field for the name.

You can define properties for this faceplate that you want to link to properties of the individual objects, e.g. the foreground color or maximum value.

Using faceplates

After you have created it, the faceplate appears as an object in your project library. You can now insert the faceplate into your process screens, and configure its properties in the properties dialog box to suit the requirements of the relevant application.

An inserted faceplate is automatically updated when you change its properties in the library.

Reusing Instances

Inserting a faceplate into a process screen creates an instance of this faceplate. Since interconnections between a faceplate and tags, scripts, etc. would only apply to the current project, you need to make these interconnection to an instance of the faceplate. You can preconfigure and reuse an instance of a faceplate so that every new instance created from it will use the same script or the same tags, for example. You can drag and drop the final configured instance of a faceplate into a library and reuse it later. This preconfigured instance can continue to be used even after the faceplate type has been changed. This means, however, that the interface of the faceplate must remain essentially unchanged.

Configuring Security Levels

Security levels cannot be assigned with faceplates since a preconfigured security level would only apply to the faceplate type and not to the project in which the instances are used. To configure the security levels, connect the "User authorization" property of the objects in the faceplate to the interface. Then assign the user authorization to each instance of the faceplate in use.

Reusing faceplates in multiple projects

WinCC flexible allows you to add faceplates to a shared library. This means you can use the faceplates in other projects. When you add a faceplate from the shared library to the screen, the system automatically saves a copy of it to the project library. Changes will only take effect if they are made at the faceplate in the project library.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

6 Creating an Alarm System

6.1 Basics

6.1.1 Visualization of process and system alarms

Introduction

- User-defined alarms

  You configure alarms to display process states or measure and report process data that you receive from the PLC on the HMI device.

- System alarms

  System alarms are predefined in these devices to display particular system states in the HMI device or the PLC.

Both user-defined alarms and system alarms are triggered by the HMI device or the PLC and can be displayed on the HMI device.

Tasks of the alarm system

- Visualization on the HMI: To report events or states that occur in the plant or the process.

  A state is reported as soon as it occurs.

- Reporting: Alarm events are output to a printer.
- Logging: Alarm results are saved for further editing and evaluation.

www . ElectricalPartManuals . com

6.1.2 User-defined alarms

6.1.2.1 Available Alarm Procedures

Alarm methods in WinCC flexible

An alarm method identifies the type of information that triggers an alarm and therefore the alarm properties.

WinCC flexible supports the following alarm procedures:

- Discrete alarm procedure

  The HMI device triggers an alarm if a particular bit is set in the PLC. Discrete alarms are configured for this purpose in WinCC flexible.

- Analog alarm procedure

  The HMI device triggers an alarm if a particular "tag" violates a "limit value." Analog alarms are configured for this purpose in WinCC flexible.

- Alarm number procedure

  The PLC transfers an alarm number (and any associated alarm text) to the HMI device. Various alarms can be configured for this purpose in the configuration software of the PLC:

  - In SIMATIC STEP 7:
    ALARM_S alarms
  - In SIMOTION SCOUT:
    ALARM_S alarms and technological alarms

Acknowledging Alarms

For alarms displaying critical or hazardous operating and process states, a stipulation can be made requiring the plant operator to acknowledge the alarm.

Alarm states

The following alarm statuses exist for discrete alarms and analog alarms:

- When the condition for triggering an alarm is satisfied, the alarm status is "Activated." Once the operator has acknowledged the alarm, it assumes the "Activated/acknowledged" status.
- When the condition for triggering an alarm no longer applies, the alarm status is "Activated/deactivated." Once the operator has acknowledged the deactivated alarm, it has "Activated/deactivated/acknowledged" status.

Each occurrence of an alarm status can be displayed and logged on the HMI device as well as printed out.

www.ElectricalPartManuals.com

6.1.2.2 Acknowledging Alarms

Introduction

For discrete and analog alarms displaying critical or hazardous operating and process states, a stipulation can be made requiring the plant operator to acknowledge the alarm.

Mechanisms for acknowledging alarms

An alarm can be acknowledged either by the operator on the HMI device or by the control program. When an alarm is acknowledged by the operator, a bit can be set within a tag.

The following options are useful for acknowledgment by the operator:

- Acknowledgement key <ACK> (only available on certain HMI devices)
- Function keys, softkeys or buttons in screens

In addition, alarms can be acknowledged through system functions in function lists or scripts.

Alarms requiring acknowledgment

The alarm class determines whether or not the alarm must be acknowledged.

Alarm classes essentially define how alarms will appear when shown on the HMI device as well as the acknowledgement behavior. WinCC flexible has both predefined alarm classes and the option to configure user-defined alarm classes.

Acknowledgment by the PLC

A discrete alarm can be acknowledged by setting a specific bit within a tag in the PLC.

Acknowledging alarms collectively

When configuring alarms, you can specify whether alarms must be acknowledged individually by the operator or whether alarms in the same alarm group can be acknowledged together. It is helpful to use alarm groups, for example, when alarms are caused by the same error.

6.1.2.3 Alarm classes

Alarm classes

Alarm classes mainly determine how alarms will appear when they are displayed on the HMI device. Alarm classes are also used to group alarms for various means of display.

- WinCC flexible has both predefined alarm classes and the option to configure user-defined alarm classes.

www.ElectricalPartManuals.com

Available alarm class settings

The following settings can be defined for each alarm class:

- Acknowledge: Alarms in this class must be acknowledged.
- Texts, colors, and flash modes to identify each alarm status when alarms are displayed
- An alarm log for logging all events related to alarms in this class.
- A text placed in front of the alarm number to indicate the alarm class when alarms are displayed on the HMI device.
- An e-mail address to which all messages about events related to the alarms in this class will be sent.

Predefined alarm classes in WinCC flexible

- "Error" for discrete and analog alarms that indicate critical or hazardous operating and process states. Alarms in this class must always be acknowledged.
- "Event" for discrete and analog alarms that indicate regular operating states, process states, and process sequences. Alarms in this class do not require acknowledgement.
- "System" for system alarms that notify the operator about the operating states of the HMI device and the PLCs. This alarm class cannot be used for user-defined alarms.

Only very specific properties can be changed for predefined alarm classes.

6.1.3 System alarms

Introduction

System alarms notify the operator about the operating states of the HMI device and the PLCs. System alarms can range from notes to serious errors.

Triggering of system alarms

The HMI device or the PLC triggers an alarm if a certain system status or an error occurs in one of these devices or during communication between two devices.

A system alarm consists of the number and the alarm text. The alarm text can also contain internal system tags that indicate the cause of the alarm more precisely. Only certain properties can be configured for system alarms.

Types of system alarms

There are two types of system alarms:

- HMI system alarms

  HMI system alarms are triggered by the HMI device if certain internal states occur or an error occurs during communication to the PLC.
- System alarms by the PLC

  These system alarms are generated by the PLC and cannot be configured in WinCC flexible.

www.ElectricalPartManuals.com

Displaying system alarms on the HMI device

In the basic settings for the alarm system, you can specify the type of system alarms to be displayed on the HMI device and how long a system alarm will be displayed.

To display system alarms on the HMI device, use the "alarm view" and "alarm window" objects.

Select the "System" alarm class setting each time one of these objects is configured in a screen or the template.

Device-specific system alarms

The instruction manual for your HMI device contains a list of the possible system alarms along with the cause and available countermeasures.

If you contact online support due to an HMI system alarm, you will need the alarm number and any system alarm tags.

6.1.4 Displaying Alarms

6.1.4.1 Displaying Alarms on the HMI Device

Options for displaying alarms on the HMI device

WinCC flexible offers the following options for displaying alarms on the HMI device:

- Alarm view

  The alarm view is configured for a certain screen. More than one alarm can be displayed simultaneously, depending on its configured size. More than one alarm view can be configured for different alarm classes and in different screens.

  The alarm view can be configured in such a way that it includes only one alarm line.

- Alarm window

  The alarm window is configured in the screen's template and is thus a component of all screens in a project. More than one alarm can be displayed simultaneously, depending on its configured size. An event can trigger closing and reopening of the alarm window. Alarm windows are saved in their own layer for the practical reason that this allows them to be specifically hidden during the configuration.

Additional signal: Alarm indicator

The alarm indicator is a configured graphic symbol that is displayed on the screen when an alarm activates. The alarm indicator is configured in the screen's template and is thus a component of all screens in a project.

The alarm indicator can have one of two states:

- Flashing: At least one unacknowledged alarm is pending.
- Static: The alarms are acknowledged but at least one of them is not yet deactivated.

Function lists can be used to configure HMI device responses.

www . ElectricalPartManuals . com

6.1.4.2 Logging and reporting alarms

Evaluation and documentation of alarms

In addition to real time displays of alarm events in the "alarm view" and "alarm window," WinCC flexible offers the following options for evaluating and documenting alarms:

- Alarm events can be printed out immediately upon occurrence.
- Alarm events from the alarm buffer can be printed out in report form.
- Alarm events can be logged in an alarm log.
- Logged alarm events can be displayed on the HMI device or printed out in report form.

Printing Alarms Immediately

You can enable or disable the printing of alarms for the entire project in the basic settings for the alarm system. In addition, printing of each individual alarm can be enabled.

Logging of alarms

Alarm classes are used to configure assignment of alarms to an alarm log. An alarm log can be specified for each alarm class. All events related to the alarms of this alarm class are logged in the specified alarm log.

Reporting of alarms

The properties of the "Print alarm" object are used to configure the assignment of alarms to a report. In addition to the data source (alarm buffer or alarm log), filtering is also possible on the basis of alarm classes.

6.1.4.3 System Functions for Alarm Editing

System functions

System functions are predefined functions you can use to implement many tasks during runtime even without having any programming knowledge. You can use system functions in a function list or in a script.

The table shows all of the system functions available for editing alarms and manipulating their display.

System function	Effect
EditAlarm	Triggers the "Edit" event for all selected alarms.
ClearAlarmBuffer	Deletes alarms from the alarm buffer on the HMI device.
ClearAlarmBufferProTool	Function like "ClearAlarmBuffer". This system function has been retained to ensure compatibility and used the old ProTool numbering.
AlarmViewEditAlarm	Triggers the event "Edit" for all alarms selected in the given alarm screen.

www.ElectricalPartManuals.com

System function	Effect
AlarmViewAcknowledgeAlarm	Acknowledges the alarms selected in the given alarm view.
AlarmViewShowOperatorNotes	Displays the configured operator notes of the alarm selected in the given alarm screen.
AcknowledgeAlarm	Acknowledges all selected alarms.
SetAlarmReportMode	Switches the automatic reporting of alarms on the printer on or off.
ShowAlarmWindow	Hides or shows the alarm window on the HMI device.
ShowSystemAlarm	Displays the value of the delivered parameter as a system alarm on the HMI device.

For details on these system functions, refer to "Working with WinCC flexible > Reference > System functions."

Events for alarms and objects for alarm indication

The following events can occur during runtime in the case of alarms and in the case of objects for alarm displays. A function list can be configured for each event.

Object	Configurable events
Discrete alarms	Activate Deactivate Acknowledge Edit
Analog alarm	Activate Deactivate Acknowledge Edit
Alarm view	Enable Disable
Alarm window	Enable Disable
Alarm indicator	Click Click when flashing

For details on these events, refer to "Working with WinCC flexible > Reference > System functions."

www . ElectricalPartManuals . com

6.2 Elements and basic settings

6.2.1 Alarm Components and Properties

Properties of alarms

An alarm always comprises the following components:

- Alarm text

  The alarm text contains a description of the alarm. Character formats supported by the relevant HMI device can be used to format the alarm text on a character-by-character basis.

  The operator note can contain output fields for the current values of tags or text lists. The alarm buffer retains the instantaneous value at the time at which the alarm status changes.

- Alarm number

  The alarm number is used to reference an alarm. Each alarm number is unique within the following types of alarms:

  - Discrete alarms
  - Analog alarms
  - HMI system alarms
  - Alarms from the PLC within a CPU

- Alarm triggers
  - For discrete alarms: A bit within a tag
  - For analog alarms: The limit value for a tag

- Alarm class

  The alarm class of an alarm determines whether or not the alarm has to be acknowledged. It can also be used to determine how the alarm appears when it is displayed on the HMI device. The alarm class also determines whether and where the corresponding alarm is logged.

Note

If you want to integrate a project in SIMATIC STEP 7, you can configure a maximum total of 7 alarm classes in WinCC flexible and STEP 7.

These components are freely selected or entered for each alarm.

www . ElectricalPartManuals . com

Optional alarm properties

The behavior of an alarm can also be defined by the following properties:

- Alarm group

  If an alarm belongs to an alarm group, it can be acknowledged along with other alarms in the same group in a single operation.

- Infotext

  Operator notes can contain additional information about an alarm. Opertor notes are displayed in a separate window on the operator device when the operator presses the <HELP> button.

- Automatic reporting

  In addition to the option of enabling and disabling automatic reporting of alarms for the entire project, it is also possible to enable reporting for each individual alarm.

- Acknowledging by the PLC "Acknowledgement write tag""

  A discrete alarm can be acknowledged by the PLC program by setting a particular bit within a tag.

- Sending an acknowledgement to the PLC "Acknowledgment read tag""

  When a discrete alarm is acknowledged by the operator, a particular bit can be set within a tag.

6.2.2 Editors for Configuring Alarms

6.2.2.1 Basic Principles of Editors

Editors for configuring alarms

WinCC flexible contains the following tabular editors for configuring alarms:

- "Discrete alarms" for creating and changing discrete alarms
- "Analog alarms" for creating and changing analog alarms
- "System alarms" for changing alarm texts of system alarms
- "Alarm classes" for creating and changing alarm classes
- "Alarm groups" for creating and changing alarm groups

www . ElectricalPartManuals . com

Changing the column display

The column display can be configured as follows:

- You can show or hide individual columns using the context menu (right-click) for the column header.

  This function is not available in the "Alarm groups" editor because this editor consists of only two columns.
- You can change the column width by dragging the right margin of a column header.
- By dragging a column header, you change the order of the columns.

  This function is not available in the "Alarm groups" editor.
- You can sort the table according to the entries in a column by clicking on the column header. Click the same column header again to reverse the sort order.

  The corresponding column header is marked with an arrow. The arrow direction determines the sort order.

Deleting and copying objects

One or more whole objects can be deleted or copied if you select the entire table row for each object using the icon on the left side of the row.

Filling in multiple table rows automatically by dragging

In tabular editors of WinCC flexible, you can fill in multiple table rows in one operation.

- Creating multiple new objects (alarms, alarm classes, or alarm groups) with similar properties:
  - Sort the table so that the table row to be copied is at the bottom.
  - Select the first element in the row to be copied.
  - Keeping the left-hand mouse button pressed, drag the lower right-hand corner of the selected table element downwards into the blank part of the table.
- Transferring a property to more than one existing object (for example, changing the trigger tags):
  - Select the table element with the relevant property.
  - Keeping the left-hand mouse button pressed, drag the lower right-hand corner of the selected table element downwards across the table rows to be modified.

Dragging and dropping within a table

A drag-and-drop operation can be used to copy an individual property (such as an alarm text or a color) from one table element to another.

Dragging and dropping from the object window.

A drag-and-drop operation can be used to move an object (such as a tag) from the object window to a table cell, provided the object is permissible in this table cell.

www.ElectricalPartManuals.com

6.2.2.2 "Discrete alarms" editor

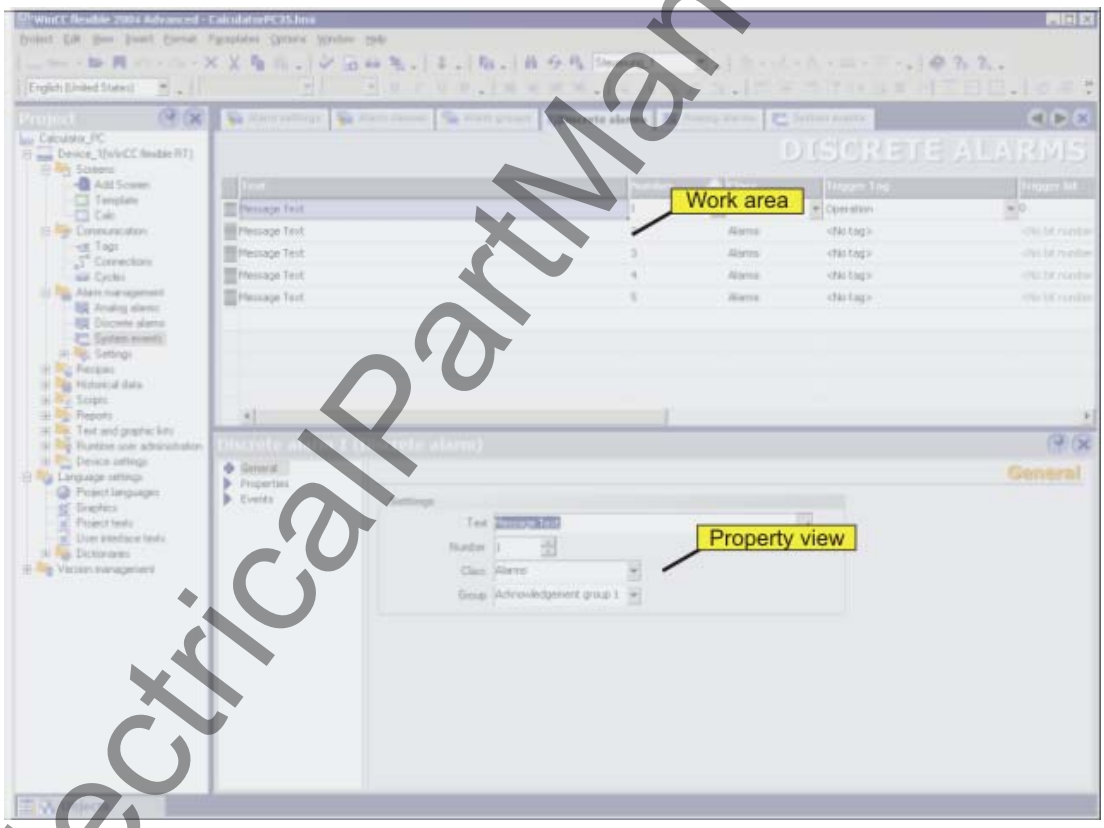
Introduction

You create discrete alarms and specify their properties in the "Discrete alarms" tabular editor.

Open

In the project view, double-click "Discrete alarms" in the "Alarms" group.

Layout



Work area

All the discrete alarms are displayed in a table in the work area. You can edit the properties of the discrete alarms in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

Property view

Here you configure discrete alarms. The property view offers the same information and settings as the work area table.

www.ElectricalPartManuals.com

6.2.2.3 "Analog alarms" editor

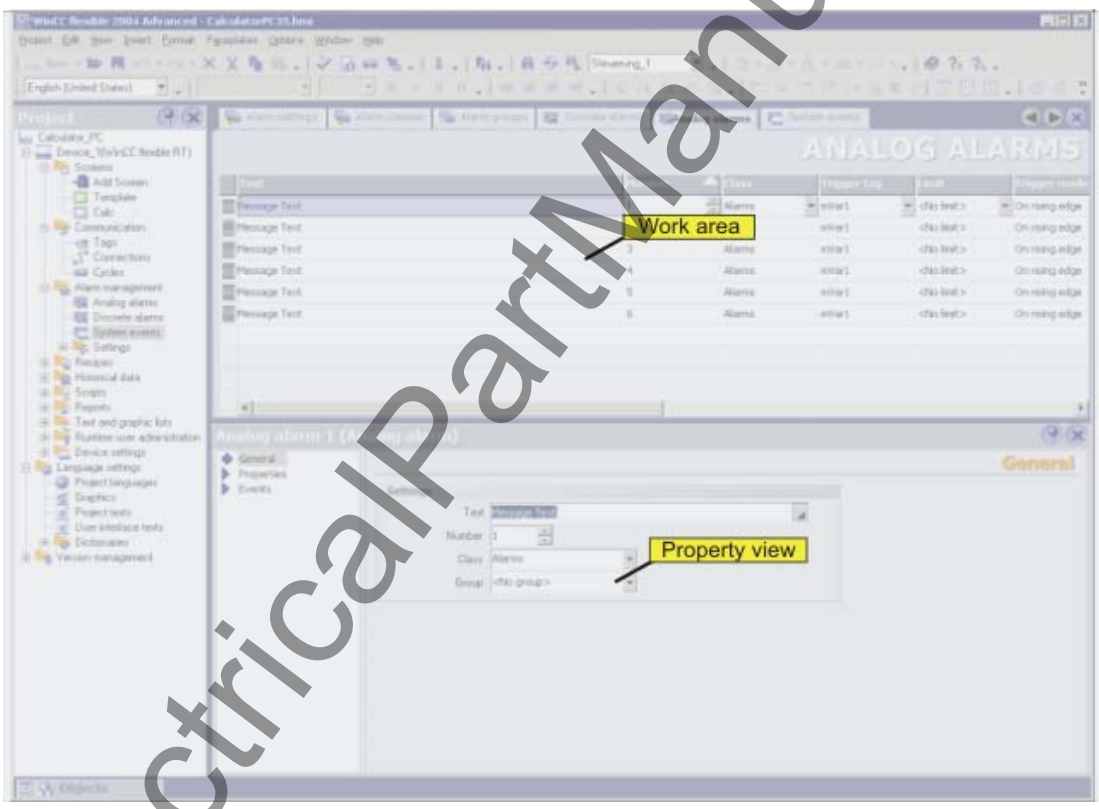
Introduction

You create analog alarms and specify their properties in the "Analog alarms" tabular editor.

Opening the "Analog alarms" editor

In the project view, double-click "Analog alarms" in the "Alarms" group.

Layout



Work area

All the analog alarms are displayed in a table in the work area. You can edit the properties of the analog alarms in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

Property view

Here you configure analog alarms. The property view offers the same information and settings as the work area table.

www.ElectricalPartManuals.com

6.2.2.4 "System alarms" editor

Introduction

You can view all HMI system alarms and change the alarm texts in the "System alarms" tabular editor.

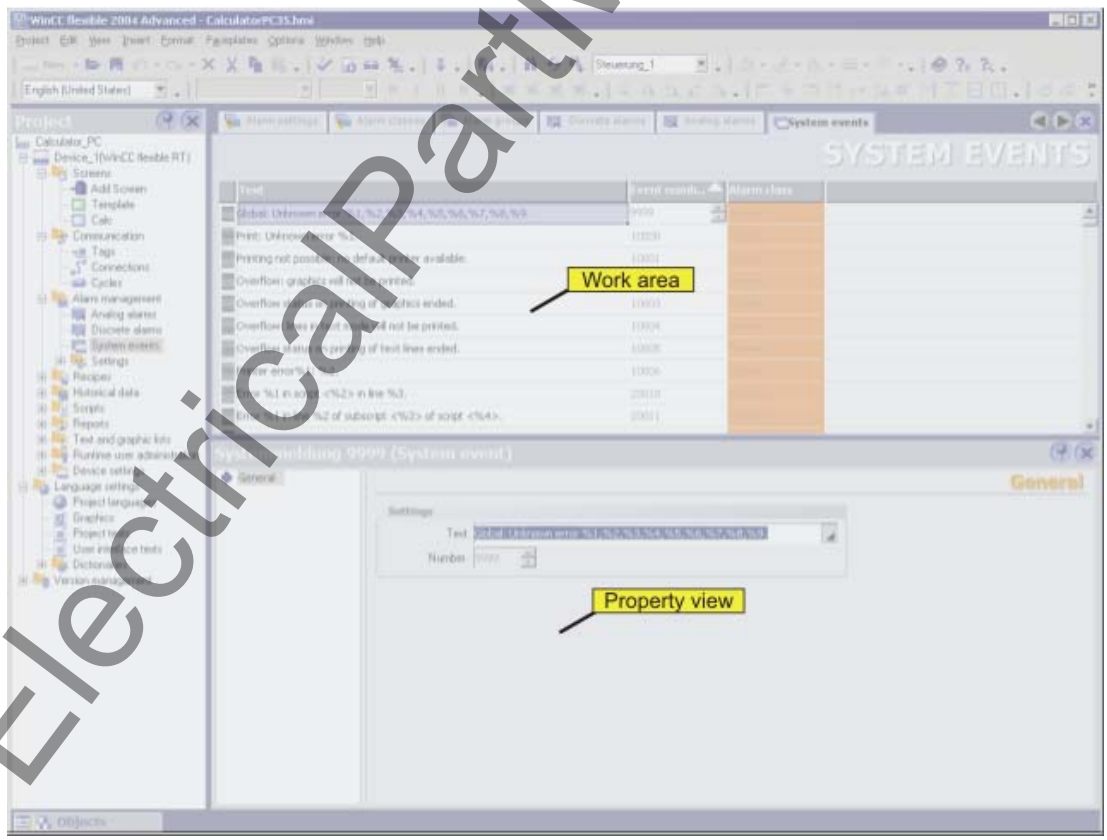
Opening the "System alarms" editor

In the project view, double-click "System alarms" in the "Alarms" group.

The "System alarms" entry is not visible with the default settings of WinCC flexible. To display the entry, proceed as follows:

1. Select "Settings" from the "Options" menu.
2. Open the "Workbench > Settings for project view" category in the "Settings" dialog.
3. Select the "Display all items" option in the "Change the mode in which the project tree is shown" field.

Layout



Work area

All the system alarms are displayed in a table in the work area. You can edit the alarm text of the system alarm in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

www . ElectricalPartManuals . com

Property view

The alarm text of the system alarm is modified in the properties view The alarm number and alarm class are assigned by the system.

6.2.2.5 "Alarm classes" editor

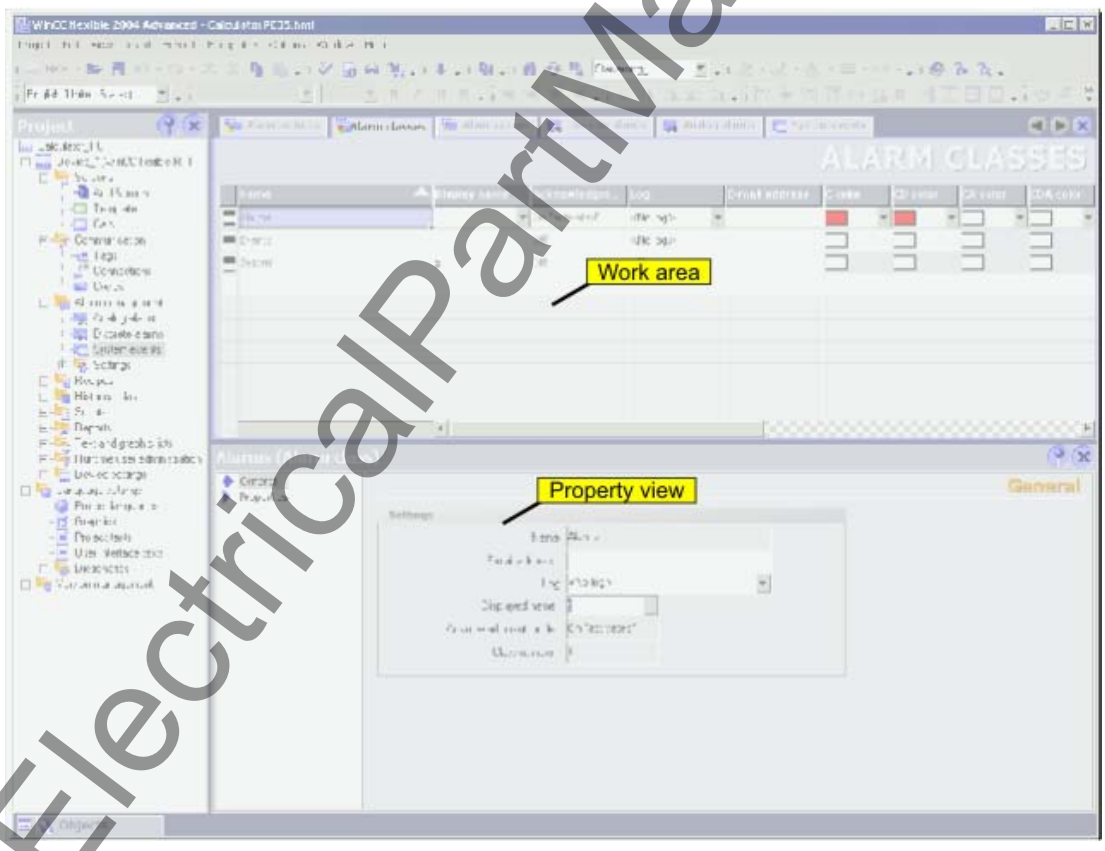
Introduction

You create alarm classes and specify their properties in the "Alarm classes" tabular editor.

Opening the "Alarm Classes" editor

In the project view, double-click "Alarm classes" in the "Alarms > Settings" group.

Layout



Work area

All the alarm classes are displayed in a table in the work area. You can edit the properties of the discrete alarms in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

www.ElectricalPartManuals.com

Property view

Here you configure alarm classes. The property view offers the same information and settings as the work area table.

6.2.2.6 "Alarm groups" editor

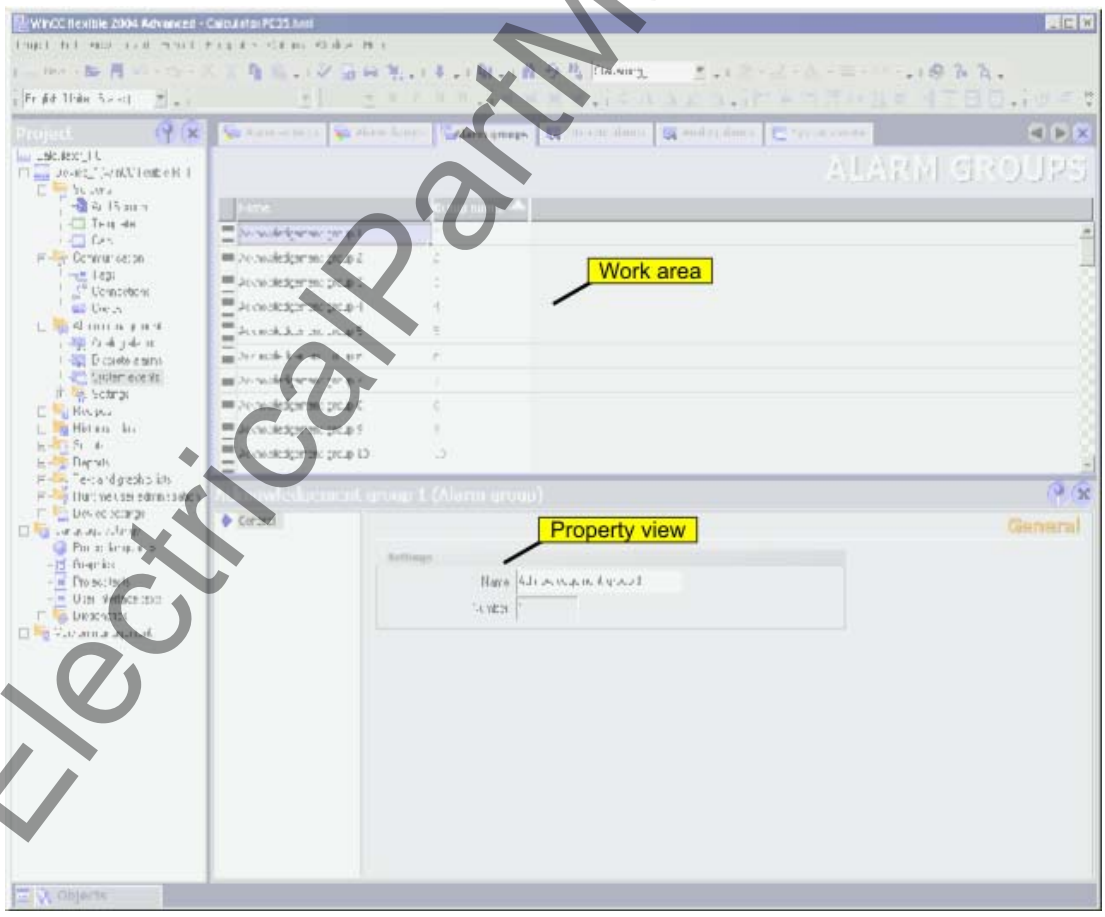
Introduction

You create alarm groups and specify their properties in the "Alarm groups" tabular editor.

Opening the "Alarm groups" editor

In the project view, double-click "Alarm groups" in the "Alarms > Settings" group.

Layout



Work area

All the alarm groups are displayed in a table in the work area. You can edit the properties of the alarm groups in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

www . ElectricalPartManuals . com

Property view

The name of the alarm group is modified in the properties view. The number is assigned by the system.

6.2.3 Basic Settings for the Alarm System

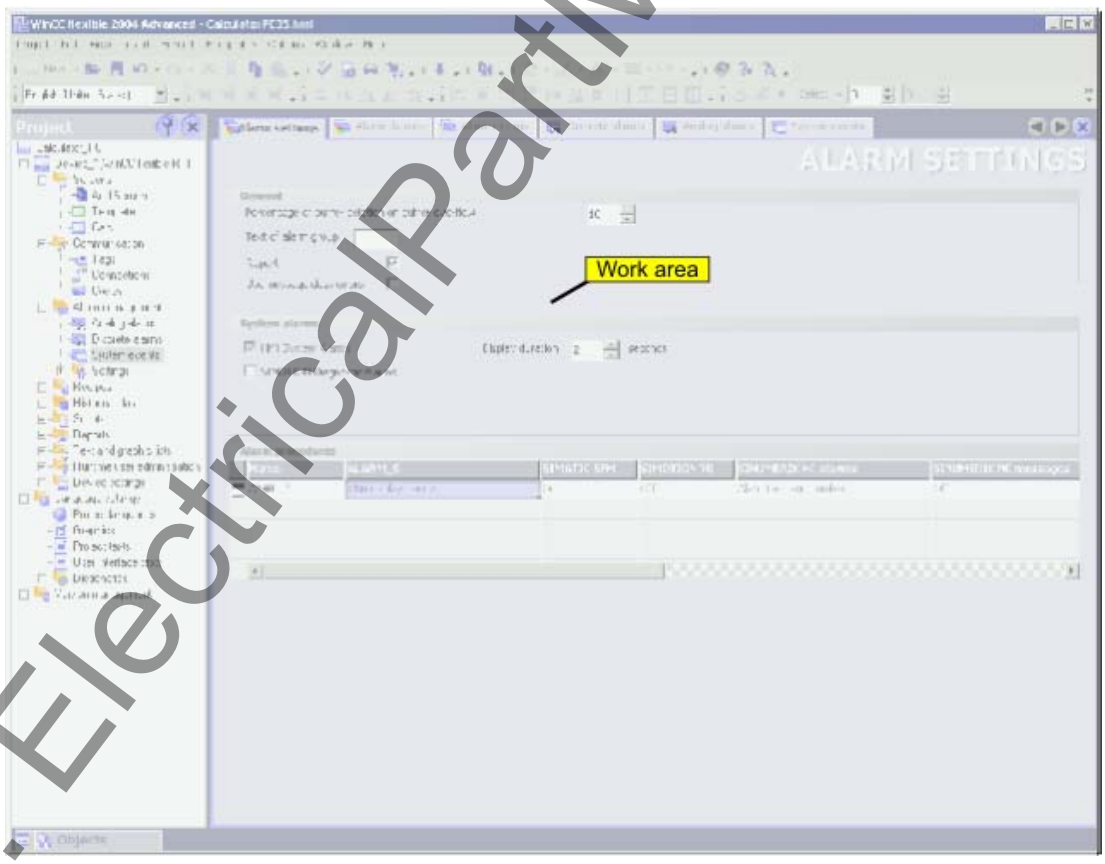
Introduction

In principle, the WinCC flexible alarm system is functional using the default settings. You only need to change these default settings if you want to adapt the alarm system behavior to specific plant conditions.

Opening the basic settings

In the project view, double-click "Alarm settings" in the "Alarms > Settings" group.

Layout



Work area

You define the settings for the alarm system in the work area. In the "System alarms" area you select, for example, the types of system alarms to be displayed on the HMI device. In integrated operation further settings can be defined in the "Alarm procedures" area.

www.ElectricalPartManuals.com

6.3 Working with alarms

6.3.1 Reporting alarms

Introduction

Configure a report in WinCC flexible with which you can output the alarms from the alarm buffer or an alarm log.

Output data of an alarm report

In order to report the alarms from the alarm buffer or an alarm log, insert the "Print alarm" object from the toolbox view into a report. Select the object in order to have the properties displayed in the property view. Configure the data selection for the report in the property view.

The following data can be output in the report:

- Current alarms from the alarm buffer
- Alarm from an alarm log

Specify the alarm classes which you want to output for the selected source. The following selections are possible:

- Error
- Operation
- Control

Specify the sequence of the alarms for the output.

The following selections are possible:

- Oldest message first
- Most recent message first

In order to output the alarms of a certain period, connect the "Display beginning" and "Display end" fields with tags. The tags can be supplied in runtime with the date and time for the first or the last alarm of the period.

6.3.2 Integrating alarms with the alarm numbering procedure

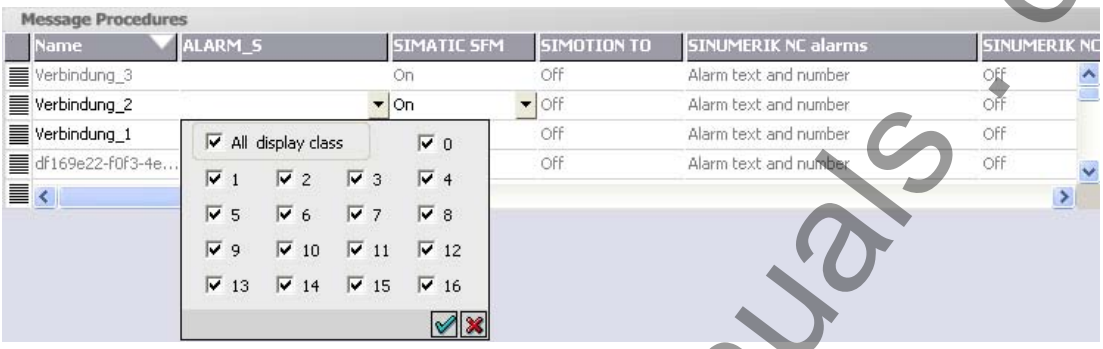
Configuring in SIMATIC STEP 7

- ALARM_S and ALARM_D are alarm numbering procedures. Alarm numbers are assigned automatically during STEP 7 configuration. These numbers are used to uniquely assign alarm messages.

  During alarm configuration in STEP 7, the stored alarms and attributes are placed in the STEP 7 configuration data. WinCC flexible automatically imports the required data and transfers them later to the HMI device.

www.ElectricalPartManuals.com

In WinCC flexible, you can filter the display of ALARM_S alarms via display classes. In the project view select "Alarms ▸ Settings" and double-click the "Alarm settings". The existing connections are displayed in the "Alarm Procedures" area.



In the row of the required connection, select the field in the "ALARM_S Display Classes" column and open the selection dialog by pressing the selection button. Select the display class you want. Close the selection dialog by pressing the button.

In the "SFM Alarms" column of a link, specify whether system errors should be displayed. For more information, consult the STEP 7 documentation.

Alarm class layout

The ALARM_S and ALARM_D alarms are assigned to particular alarm classes in STEP 7. To edit the display options for these alarm classes, select "Alarms ▸ Settings ▸ Alarm Classes." Open the context menu and select the "Open Editor" command. You can recognize alarm classes by the S7 prefix in the alarm class name.

You configure the display options for the alarm classes using the "Alarm Classes" editor.

www.ElectricalPartManuals.com

6.4 Alarm logging

6.4.1 Basic principles of alarm logging

Introduction

Alarms indicate fault states and operating states of a process in a project. They are generally triggered by the controller. Alarms are indicated in a screen display on the HMI device.

WinCC flexible lets you log alarms and document operational states and error states of the plant.

Archiving is not available on all devices.

Principle

You can configure alarm logging. The alarms to be logged are assigned to an alarm log via the alarm class. Every message belongs to a specific alarm class. When configuring an alarm class, enter the alarm log to be used.

You can save alarms from various alarm classes in a single log.

When you create a log, you specify the log properties and select the log behavior.

A log contains the following data:

- Date and time of alarm
- Alarm text
- Alarm number
- Alarm status
- Alarm class
- Alarm procedure
- Values in the tags contained in the alarm text
- Controller

Note

The alarm text and controller are only logged if this has been configured in the properties of a log.

www . ElectricalPartManuals . com

6.4.2 Alarm logging

Introduction

To log alarms, you group them in alarm classes. Each alarm class can be recorded in a separate log. Depending on the HMI device, you can select among several log types when making configuration settings. You specify the log behavior when configuring the log.

Log types

In WinCC flexible, you can select from the following log types:

1. Circular log
2. Segmented circular log
3. Log with level-dependent system alarm
4. Log with level-dependent execution of system functions

Alarms can be logged automatically or managed by an operator.

Storage media

Log data can be saved either in a file or in a database. Saved data can undergo additional processing in other programs, e.g. for analysis purposes.

Displaying log contents

You can display log contents on the HMI device. To do so, you must configure an alarm view.

6.4.3 "Alarm logs" editor

Introduction

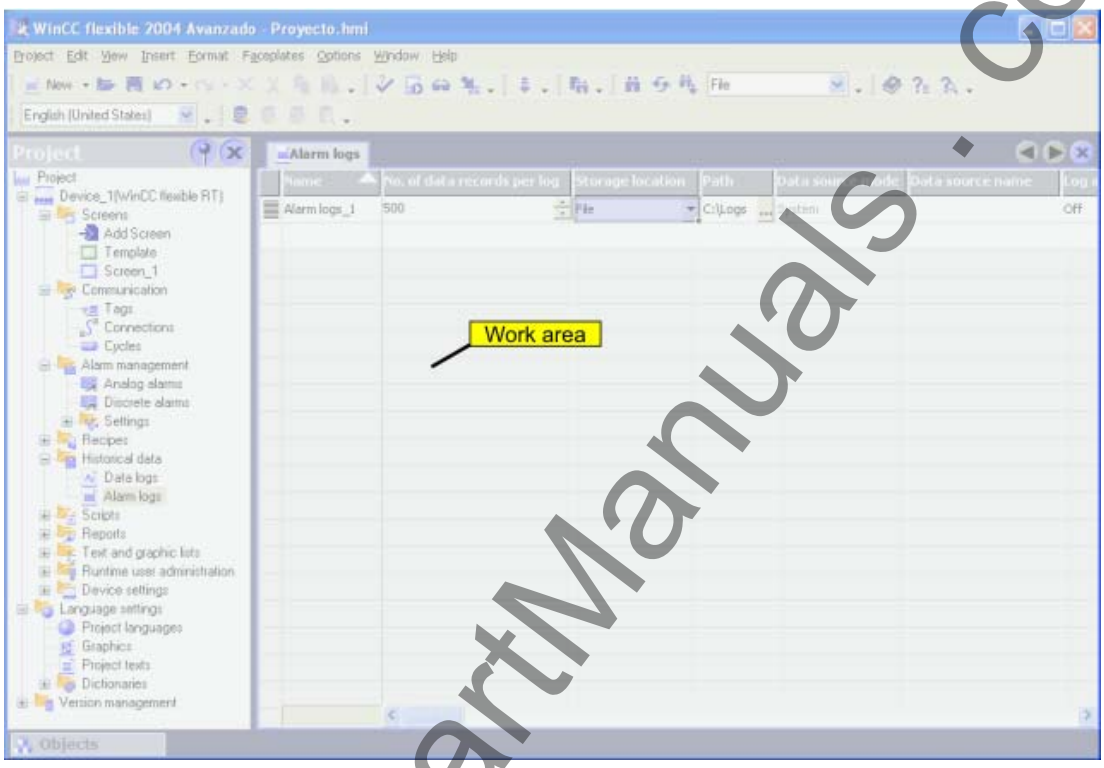
In the "Alarm logs" table editor, you configure alarm logs for logging alarm classes and define their properties.

Opening the "Alarm logs" editor

Double-click on "Alarm logs" in the project window in the "Log" group to open the "Alarm log" editor.

www.ElectricalPartManuals.com

Layout



Work area

All alarm logs are displayed in a table in the work area. You can edit the properties of the alarm logs in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

6.4.4 Basic settings for alarm logs

Introduction

The properties of an alarm log can be defined in the "Alarm log" editor or in the Properties window for logs.

General properties

- Name

  The name of the alarm log is freely selectable but the name must include at least one letter or one number.

www.ElectricalPartManuals.com

Note

The characters which can be used in the name of the data source depend on the storage location.

If the storage location "File" is used, the following characters may not be used: \ / * ? : " < > |

If the storage location "Database" is used, the following characters may not be used: a-z A-Z 0-9 _ @ # $
However, the characters _ @ # $ may not be used as the first character of the name.

- Memory location

  The alarm log may be stored in an ODBC database (only on a PC) or in a separate "*.csv" file. Select "File" or "Database" as the storage location correspondingly.

  Depending on the configuration of the HMI device, you can select the local hard disk of the PC or the storage card of the panel or, if present, a network drive as "path."

  If you have chosen an ODBC database as the storage location, you have the name suggested by the system (system-defined data source name) or enter one yourself (user-defined data source name).

- Size

  The size of the log is calculated from the number of data records and the rough size of an entry. The size of an entry is dependent, amongst other things, on whether the alarm text and associated tag values are to be logged with it.

Alarm log properties

- Start-up behavior

  Under Enable you can specify that logging starts when runtime is started. Enable the checkbox "Enable logging at runtime start."

  You can also control the behavior at runtime start in other ways. Enable "Reset log" if you want to overwrite previously logged data with the new data or "Append data to existing log" if you want to append new data to an existing log.

Note

System functions can be used to control the restart of a log during runtime.

- Logging method

  Here you can specify what should happen when the log is full. You can choose one of the following options:

  - Circular log: When the log is full, the oldest entry will be overwritten.
  - Segmented circular log: Multiple logs of the same size will be created and filled one after the other. When all logs are completely full, the oldest log is overwritten.
  - Display system message when: When a defined fill level is reached a system message is displayed.
  - Trigger event: The "Overflow" event is triggered as soon as the log is full.

www.ElectricalPartManuals.com

- Settings

  Define whether the alarm text and error location should be stored each time an alarm is logged. The alarm texts will be logged in the current runtime language.

- Comment

  Here you can enter descriptive text regarding the log.

6.4.5 Alarm logging

Introduction

In runtime, alarms can be stored in logs for later evaluation. When configuring the logging of alarms and alarm classes, the user defines the log in which the alarms are to be stored and whether just the alarm events are stored or the associated alarm texts and error location, too.

Principle

Several steps are involved in alarm logging:

- Creating and configuring alarm logs

  When creating an alarm log, the following must be defined:

  - General settings, e.g. name, size, storage location
  - Behavior at runtime start
  - Behavior when the log is full

- Configuring logging of the alarms in an alarm class

  An alarm log can be specified for each alarm class in which the alarm events are stored during runtime.

- Further processing logged alarms

  The logged alarms can be evaluated directly in your WinCC flexible project, e.g. in an alarm view, or via another application, e.g. Excel.

www . ElectricalPartManuals . com

6.4.6 Displaying logged alarms on screens

Introduction

During runtime, you can display logged alarms on the screens of the HMI device. During this process, alarms in an alarm class are downloaded from the log database and presented in an alarm view.

Principle

You must configure an alarm view to display logged alarms on the HMI device. When configuring an alarm view, specify the alarm class of the alarms to be displayed.

6.4.7 Structure of a *.csv file with alarms

Introduction

In the *.csv (Comma separated value) file format, table columns (name and value of entry) are separated by a semicolon. Each table row ends with a carriage return.

Example of a *.csv file

This example shows a file with logged alarms:

```
"Time_ms";"MsgProc";"StateAfter";"MsgClass";"MsgNumber";"Var1";...;"
Var8";"TimeString";"MsgText";"PLC"37986550590,27;1;1;3;110001;"";...
;"";"30.06.99 13:12:51";"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";
```

Structure of a log file in *.csv format

The following values are entered in the individual columns of a WinCC flexible log file:

Parameter	Description
Time_ms	Specify a time stamp as a decimal value (see below for conversion)
Msg_Proc	Alarm procedures: 0 = Unknown alarm procedure 1 = System alarm 2 = Alarm bit procedure (operating alarms) 3 = Alarm number procedure ALARM_S 4 = Diagnostic event 100 = Alarm bit procedure (fault alarms)
State after	Alarm event: 0 = Arrived/Departed 1 = Arrived 2 = Arrived/Acknowledged/Departed 3 = Arrived/Acknowledged 6 = Arrived/Departed/Acknowledged

www.ElectricalPartManuals.com

Parameter	Description
Msg_Class	Alarm class 0 = No alarm classes 1 = "Interuption" 2 = "Operation" 3 = "System" 64 ... = User configured alarm classes
Msg Number	Alarm number
Var1 to Var8	Alarm tag value as STRING
Time string	Time stamp as STRING, i.e., readable date format
Msg text	Alarm in a readable STRING
PLC	Alarm localization (relevant PLC)

Conversion of the time stamp decimal value

If the value needs to be processed using a different program, proceed as follows:

1. Divide Time_ms by 1,000,000.

   Example: 37986476928 : 1.000.000 = 37986,476928

2. The whole number portion (37986) is the date calculated from 31.12.1899.

   You can now convert the time stamp value to days in Excel by assigning a corresponding format from the "Date" group to the cells, which contain the time stamp.

   Result: 37986 results in 31.12.2003

3. The value after the comma (0,476928) indicates the time:
   - Multiply the value (0,476928) by 24 results in the hours (11,446272).
   - Multiply the remainder (0,446272) by 60 results in the minutes (26,77632).
   - Multiply the remainder (0,77632) by 60 results in the seconds (46,5792).

   Sum 11:26:46.579

   This conversion is supported by Microsoft Excel, for example.

www . ElectricalPartManuals . com

6.4.8 Accessing the ODBC log database directly

Introduction

The storage location of a log can be a database or a file.

The database is addressed by means of its "Data source name" (DSN). Select the database you would like to use in WinCC flexible in the Windows Start menu under Settings > Control panel > ODBC data sources.

To store log data, specify the "Data source name" (DSN) instead of a directory name when making your configuration settings. With the DSN, you are referencing the database and the storage location.

Application

The entire functional scope of the database is available for additional processing and evaluation of log data.

Principle

You create the data source that connects to the database on the same computer that contains the runtime software. You then specify the DSN configured here when you create a log in WinCC flexible.

Using the ODBC interface, you can access the database directly with other programs such as MS Access or MS SQL server.

With the "StartProgram" system function, you can also configure a program call (for MS Access, for example) on the HMI device. This does not interrupt the runtime program sequence.

www . ElectricalPartManuals . com

7 Working with a connection

7.1 Basics

7.1.1 Communication basics

Introduction

The term communication refers to the data exchange between two communication partners. The communication partners can be interconnected via direct cable connection or network.

Communication partners

A communication partner can be any node which is capable of communicating and exchanging data with other nodes on the network. In the WinCC flexible environment, those nodes may be CPUs and communication modules of the automation system, HMI devices and CPs in the PC.

Data transferred between the communication partners may serve different purposes. In WinCC flexible, those are:

- process control
- process data acquisition
- reporting states in a process
- process data logging

www . ElectricalPartManuals . com

7.1.2 Principles of communication

Introduction

WinCC flexible controls communication between the HMI and the PLC by means of tags and area pointers.

Communication using tags

In WinCC flexible, tags are centrally managed in the "Tag" editor. There are external and internal tags. External tags are used for communication, and represent the image of defined memory locations on the PLC. The HMI and the PLC both have read and write access to this memory location. Those read and write operations may cyclic or event-triggered.

In your configuration, create tags that point to specific PLC addresses. The HMI reads the value from the defined address, and then displays it. The operator may also enter values on the HMI which will be written to the relevant PLC address.

Communication using area pointers

Area pointers are used to exchange data of specific user data areas. Area pointers are parameter fields WinCC flexible uses in Runtime to obtain information about the location and size of data areas of the PLC. During communication, the PLC and the HMI alternately access those data areas for read and write operations. Based on the results of the evaluation of data stored in those areas, the controller and HMI trigger defined actions.

Area pointers used in WinCC flexible:

- PLC job
- Project ID
- Screen number
- Data record
- Date/time
- Date/time PLC
- Coordination

The availability of the various area pointers is determined by the HMI used.

Communication between WinCC flexible and automation systems

Industrial communication using WinCC flexible means that data are exchanged using tags and area pointers. To acquire the data, the HMI sends request frames to the automation system using a communication driver. The automation system (AS) returns the requested data to the HMI in a response frame.

www . ElectricalPartManuals . com

Communication drivers

A communication driver is a software component that establishes a connection between an automation system and an HMI, and thus allows the transfer of process values to the WinCC flexible tags. WinCC flexible supports the interconnection of different automation systems with various communication drivers.

Users can select the interface, the profile and the transmission speed for each specific communication partner.

Communication between HMIs

The SIMATIC HMI HTTP Protocol is available for the communication between HMIs. This protocol is a component of the "Sm@rtAccess" option. The Protocol can be used on PCs with WinCC flexible 2005 Runtime and on Panels as of the 270 series. For detailed information, refer to the SIMATIC HMI HTTP Protocol documentation.

Communication via uniform and manufacturer-independent interface

WinCC flexible provides a uniform and manufacturer-independent software interface using OPC (OLE for Process Control). This interface allows a standardized data exchange between applications for industry, office, and production. For detailed information, refer to the OPC documentation.

7.2 Elements and basic settings

7.2.1 Connections Editor

Introduction

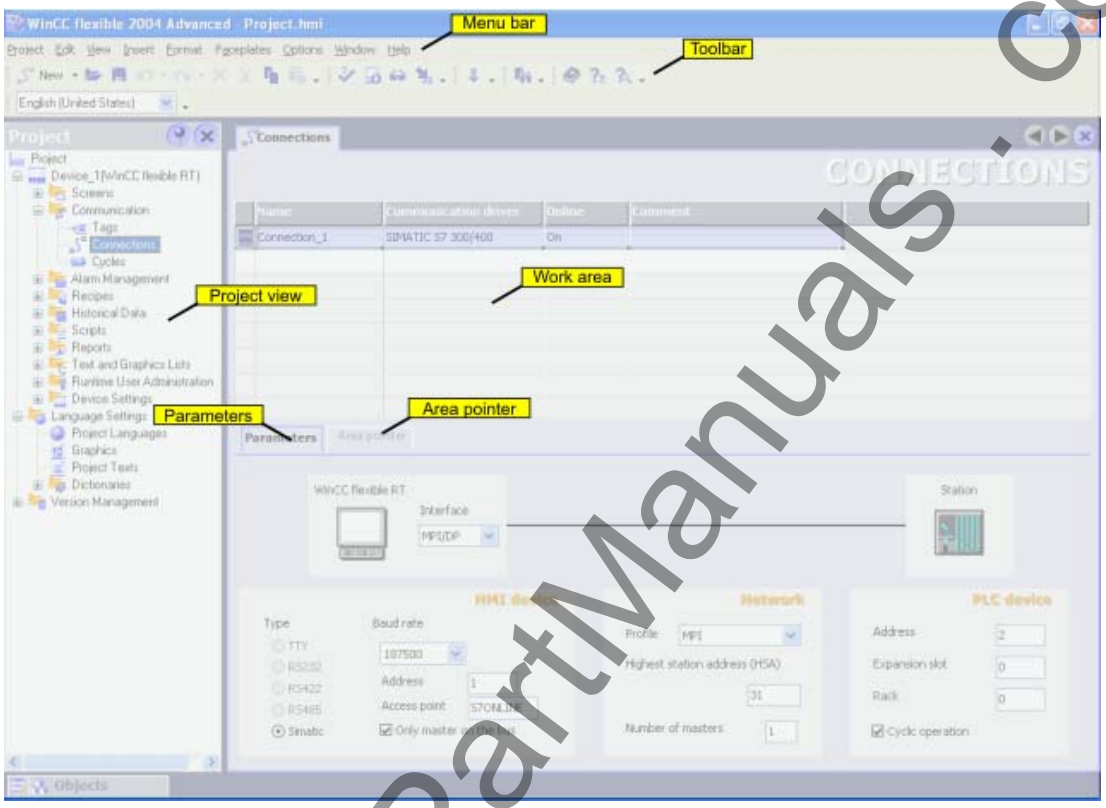
In the Connections editor, you create and configure connections.

Open

Select "Connections" from the project view, and then open the context menu. Select "New connection" from in this context menu. The new connection will be created and opened in the work area.

www . ElectricalPartManuals . com

Structure



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available keystroke is indicated next to the menu command.

Toolbars

The toolbars contain the most frequently used buttons.

Select "View > Toolbars" to show or hide the specific toolbars. The button of a toolbar can be used to show or hide specific buttons of this toolbar.

Work area

All connections are visualized in the work area in tabular format. You select the communication drivers from the table cells, and edit the relevant connection properties. To sort the table by its column entries, simply click the column header.

www . ElectricalPartManuals . com

"Parameters" tab

Using the "Parameters" tab in the work area, you configure the parameters of the communication driver you selected from the table. Select the settings for the HMI, the network and for the controller.

"Area pointer" tab

You configure the area pointers of the connections on the "Area pointer" tab in the work area.

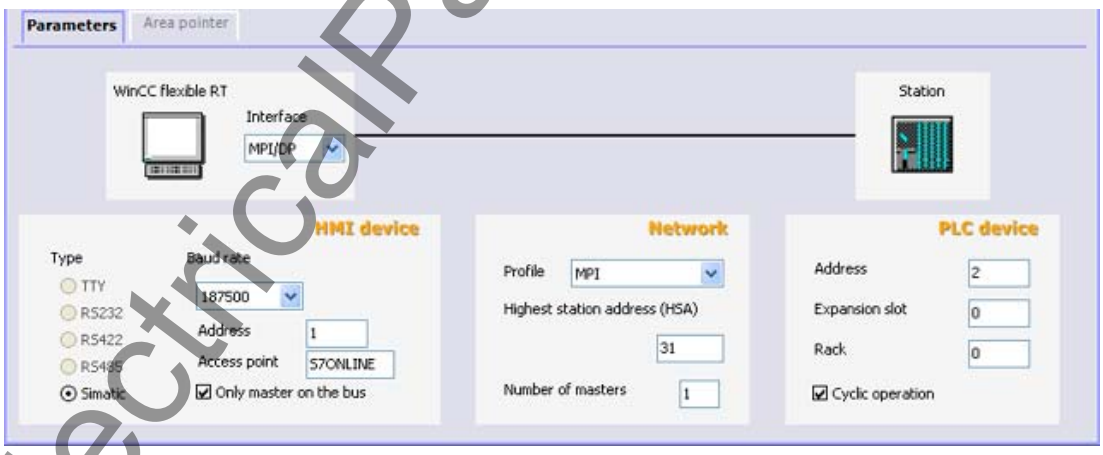
7.2.2 Parameters for connections

Introduction

Select the "Parameters" tab of the "Connections" editor to configure the properties of a connection between the HMI and the communication partner.

Structure

The communication partners are visualized schematically on the "Parameters" tab. This tab provides the "HMI device", "Network" and "Controller" areas where you can declare the parameters of the relevant interface used.



The system sets default parameters. Always ensure consistency on the network whenever you edit parameters. For detailed information on configurable parameters, refer to the description of the supported protocols

www . ElectricalPartManuals . com

7.2.3 Area pointers for connections

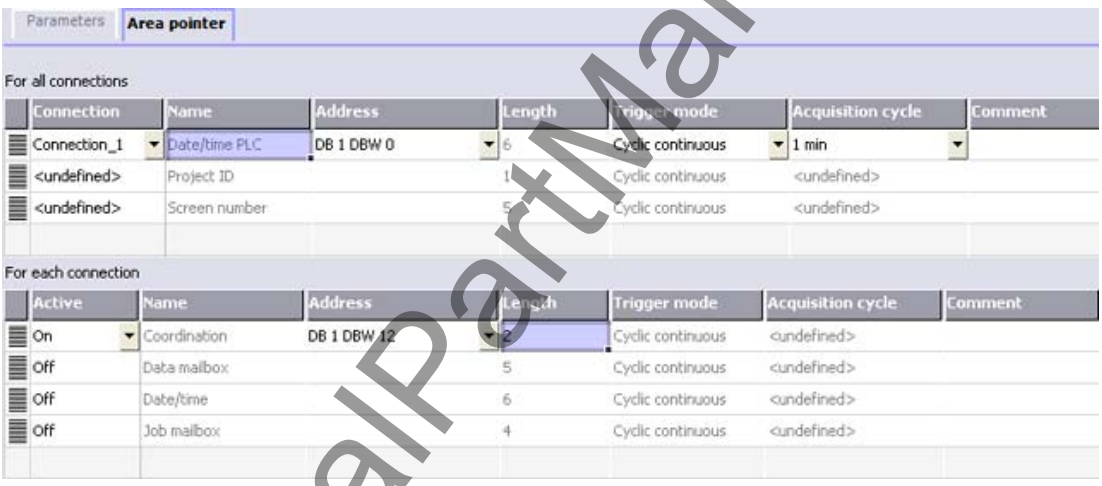
Introduction

Using the "Area pointer" tab of the "Connections" editor, you can configure the usage of the available area pointers.

Structure

The "Area pointer" tab contains two tables of area pointers. The "For all connections" table contains those area pointers which are created only once in the project and can be used for only one connection.

The "For each connection" table contains the area pointers you can set separately for each available connection.



The availability of the various area pointers is determined by the HMI used. For detailed information on area pointers and their configuration, refer to the description of the supported protocols

www . ElectricalPartManuals . com

Structure of a recipe management system 8

8.1 Basics

8.1.1 Basic principles of recipes

Introduction

Recipes are a collection of associated data, e.g. machine configuration or production data. You can transfer these data, for example, from the HMI device to the controller in a single step in order to change the production variant. If you have programmed directly at the machine, for example, you can transfer the data to your HMI device and write these to the recipe.

Principle

You create recipes with the associated data in the "Recipes" editor. In order to display and edit recipes on the HMI device, you configure either a recipe view or recipe screen in the process screen.

The following data entry options are available:

- Data entry during runtime

  If you have to change or adjust production data on a frequent basis, you can acquire the data directly on the HMI device or the machine itself during runtime. One example is the "Teach in" mode for assigning parameters to a machine. You can move movable components to their desired positions directly on the machine. You then transfer the acquired position data from the PLC to the HMI device and store them in the recipe.

- Data import during runtime

  If, for example, production data are stored on a server in a database, you can import the production data to the HMI device via a CSV file during runtime.

- Data entry during configuration

  If production data are already available or are fixed, you can enter or import them in the "Recipes" editor during recipe configuration.

www . ElectricalPartManuals . com

Examples of recipe applications

Recipes are used in the manufacturing industry and in machine building. These two examples illustrate typical applications that you can implement with the recipe functionality of the WinCC flexible engineering system:

- Machine parameter assignment

  One field of application for recipes is the assignment of machine parameters in the manufacturing industry. A machine cuts wooden boards of various sizes to specified dimensions and drills holes. Depending on the board size, the guide rails and drill must be moved to a new position. The required position data are stored as data records in a recipe. You reassign the machine parameters using "Teach in" mode if, for example, a new board size is to be processed. You transfer the new position data directly from the PLC to the HMI device and save it as a new data record.

- Batch production

  Batch production in the food processing industry represents another field of application for recipes. A filling station in a fruit juice plant produces juice, nectar, and fruit drinks in a variety of flavors. The ingredients are always the same, differing only in their mixing ratios. Each flavor corresponds to a recipe. Each mixing ratio corresponds to a data record. All of the required data for a mixing ratio can be transferred to the machine control at the touch of a button.

www . ElectricalPartManuals . com

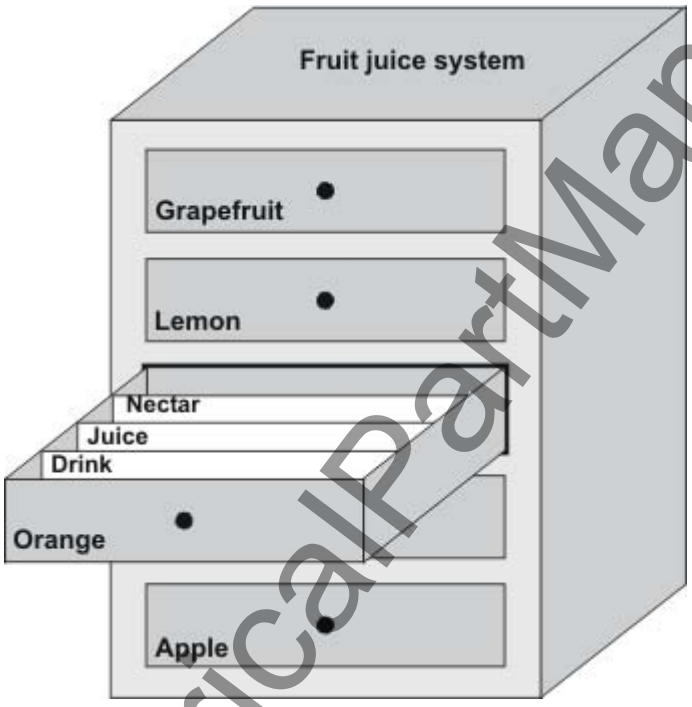
8.1.2 Structure of recipes

Introduction

A product often has several variants. For example, product variants can differ with respect to size or quality. This condition is accurately reflected in a recipe.

Principle

A recipe consists of recipe data records containing values. The structure of a recipe is explained using the example of a filing cabinet.



Each recipe represents a drawer of the file cabinet shown, and thus precisely one product. If the fruit juice mixing plant is producing orange, apple, and tropical fruit flavors, you would then configure one recipe for each flavor.

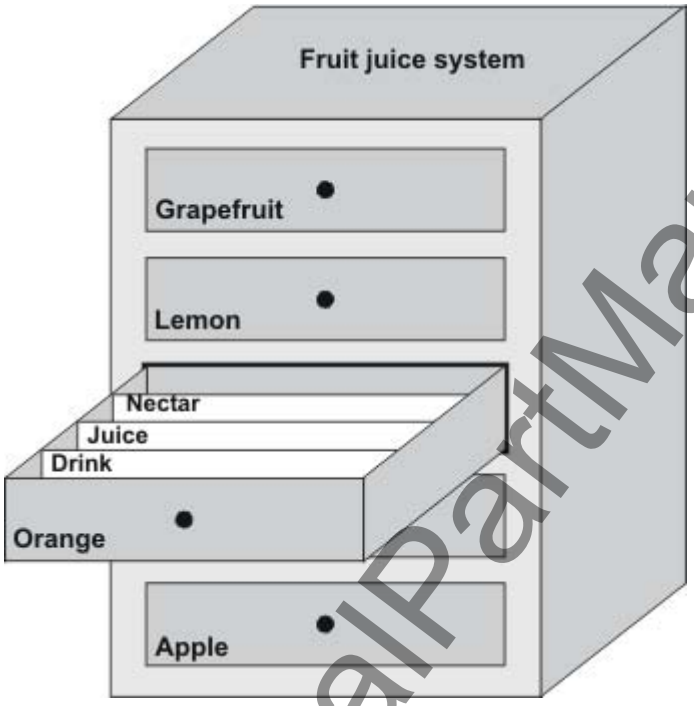
You define the recipe elements in the recipe. A recipe element consists of the display name and a tag. The display names are indicated in the recipe data records and on the HMI device in the recipe view. In Runtime, the appropriate tag value is read from the controller or transferred to the controller.

www . ElectricalPartManuals . com

8.1.3 Structure of recipe data records

Introduction

A recipe data record corresponds to a file card in an individual drawer and thus to a single product variant. If the fruit juice mixing plant is producing juice, nectar, and fruit drinks, you would then create a recipe data record in the recipe for each product variant. In this case, the product variants consist of the different mixing ratios for the ingredients.



A recipe data record is a set of values for the tags defined in the recipe. You enter the values in the input fields. You can enter the values either during configuration or during runtime on the HMI device or the machine.

Elements | Data records

Name	Display name	Numb...	Water	Concentrate	Sugar	Aroma
Beverage	Beverage	1	30	70	45	600
Nektar	Nektar	2	50	50	10	300
Juice	Juice	3	5	95	3	100

To produce a product, you transfer the appropriate recipe data record from the HMI device to the connected controller. The values in the recipe data record cannot be changed on the HMI device unless the configuring engineer has provided for this.

www.ElectricalPartManuals.com

Editing recipe data records

You can edit recipe data records during configuration or in runtime on the HMI device.

- During configuration, you can define recipes in the "Recipes" editor in the "Elements" tab. You can enter values in the recipe data records in the "Data records" tab.
- During runtime, you have the option of entering recipe data record values directly on the HMI device or importing them via a CSV file. You can also export the recipe data records to a CSV file.

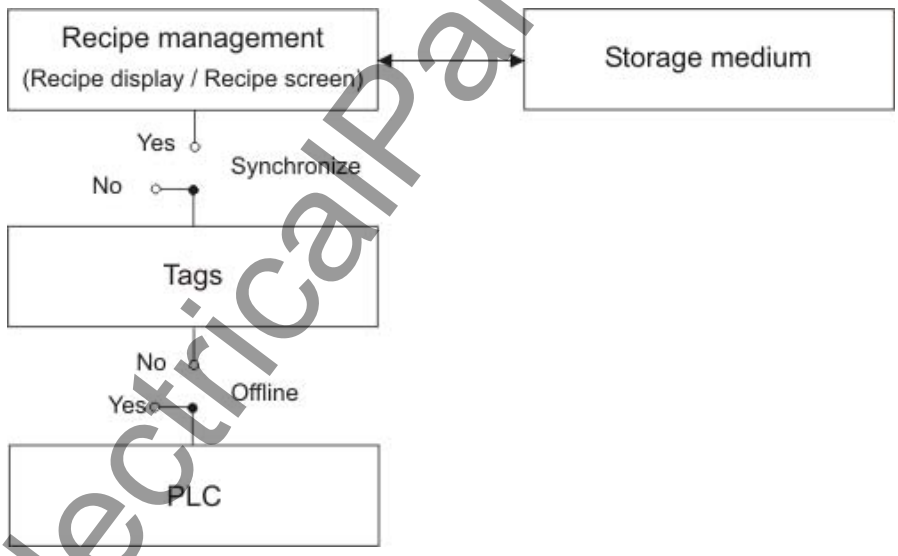
8.1.4 Configuration of recipes

Introduction

You configure recipes according to your intended application. To write a value to a recipe data record on your HMI device without disturbing the current process, you need configuration settings other than those required for assigning parameters to a machine.

Principle

In the configuration settings of a recipe, you specify the behavior of the tags you are using in the recipe. The figure below shows the basic differences when working with recipe data records.



These configuration settings are made under "Settings" in the property view:

www.ElectricalPartManuals.com

Configuration 1: Recipe without "Synchronize tags"

Data of a data record that has been read are only displayed and can only be edited in the recipe view. Using these same tags outside of the recipe view does not affect their values.

Configuration 2: Recipe with "Synchronize tags" and with "Tags offline"

The "Synchronize tags" option is used to specify that the data of a data record read from the controller or storage medium are to be written to or read from the tags you have configured for the recipe.

The "Offline" option ensures that the input data are written to the tags without being transferred directly to the controller.

Configuration 3: Recipe with "Synchronize tags" and without "Tags offline"

The "Synchronize tags" option is used to specify that the data of a data record read from the controller or storage medium are to be written to or read from the tags you have configured for the recipe.

The input or read data are transferred immediately to the controller:

Synchronization with the controller

In the case of synchronous transfer, both the controller and the HMI device set status bits in the shared data compartment. You can use this mechanism to prevent uncontrolled overwriting of data in either direction in your control program. You define the address range of the data compartment separately for each controller on the "Range pointer" tab in the "Connections" editor.

Applications for synchronous transfer of recipe data records:

- The controller is the "active partner" for the transfer of recipe data records.
- The controller evaluates the data containing the recipe number and name, as well as the recipe data record number and name.
- Triggering the transfer of data records by means of system function or PLC job, e.g. with the system functions "SetDataRecordToPLC" and "GetDataRecordFromPLC", or with the PLC jobs "Set_Data_Record_To_PLC" and "Get_Data_Record_From_PLC."

In order to synchronize transfer of data records between the HMI device and the controller, the following requirements must be met during configuration:

- The "Data mailbox" range pointer is located under "Range pointers" in the project view.
- The controller with which the HMI device synchronizes the data record transfer is specified in the recipe properties.

www . ElectricalPartManuals . com

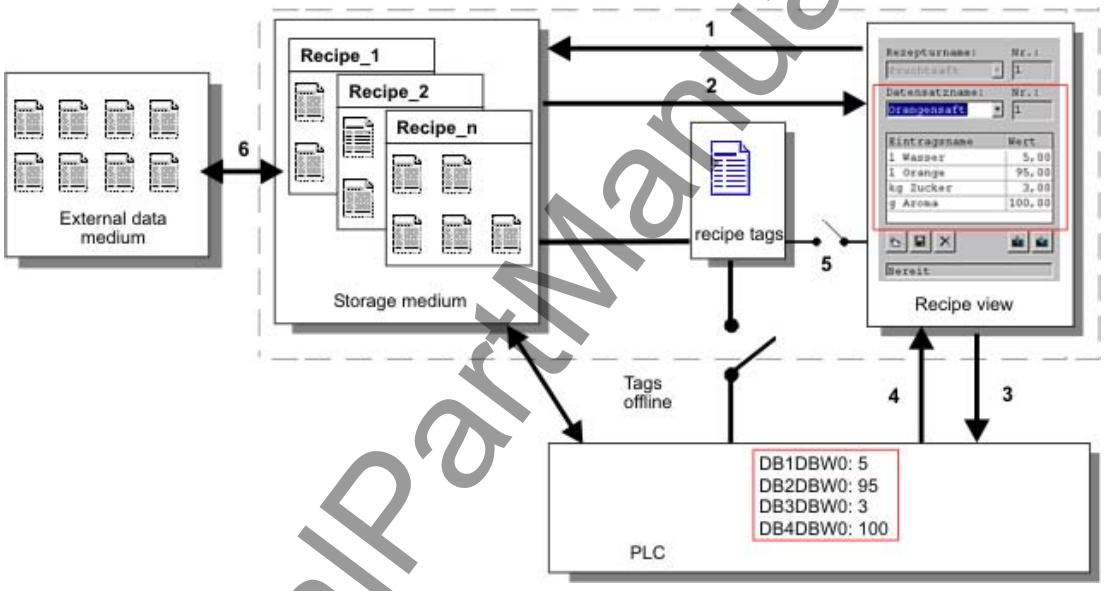
8.1.5 Transfer of recipe data records

Introduction

Recipe data records can be transferred in Runtime between external data storage media, e.g. a flash memory, an HMI device and a controller.

Principle

The figure below shows how recipe data records can be transferred. You configure the appropriate functionality for transferring data records in the recipe view. In a recipe screen, you use the system functions provided for this purpose.



The HMI device stores recipe data records on a storage medium such as a flash memory device or hard disk. You can edit a recipe data record in a recipe view or recipe screen on the HMI device display.

(1) Save: Values you change on the recipe view or recipe screen are written to the recipe data record on the storage medium by executing the "Save" function.

(2) Load: The "Load" function is used to update the values of recipe tags shown on the recipe screen with the values of the recipe data record of the storage medium. The function overwrites any values changed on the recipe screen. The "Load" function is executed for the Recipe view when the data record is selected again.

(3) Write to controller: The values deltas of the recipe view and recipe screen are downloaded to the PLC by calling the "Write to controller" function.

(4) Read from controller: Call the "Read from controller" function to update the indicated values of the recipe view and recipe screen with the controller values. The function overwrites any data changed on the recipe view or screen.

(5) Synchronization with control: In your configuration, you can decide to synchronize the values in the recipe view with the values of the recipe tags by setting the "Synchronization with control" function. After this synchronization, both the recipe tags and the recipe view contain the current updated values. When the "Variables offline" setting is disabled for the recipe, the current values are also applied in the controller.

(6) Import, Export: A data record can be exported to an external data carrier in order to process it in MS Excel, for example. The data record is there stored in *.csv format.

www.ElectricalPartManuals.com

8.2 Elements and basic settings

8.2.1 "Recipes" editor

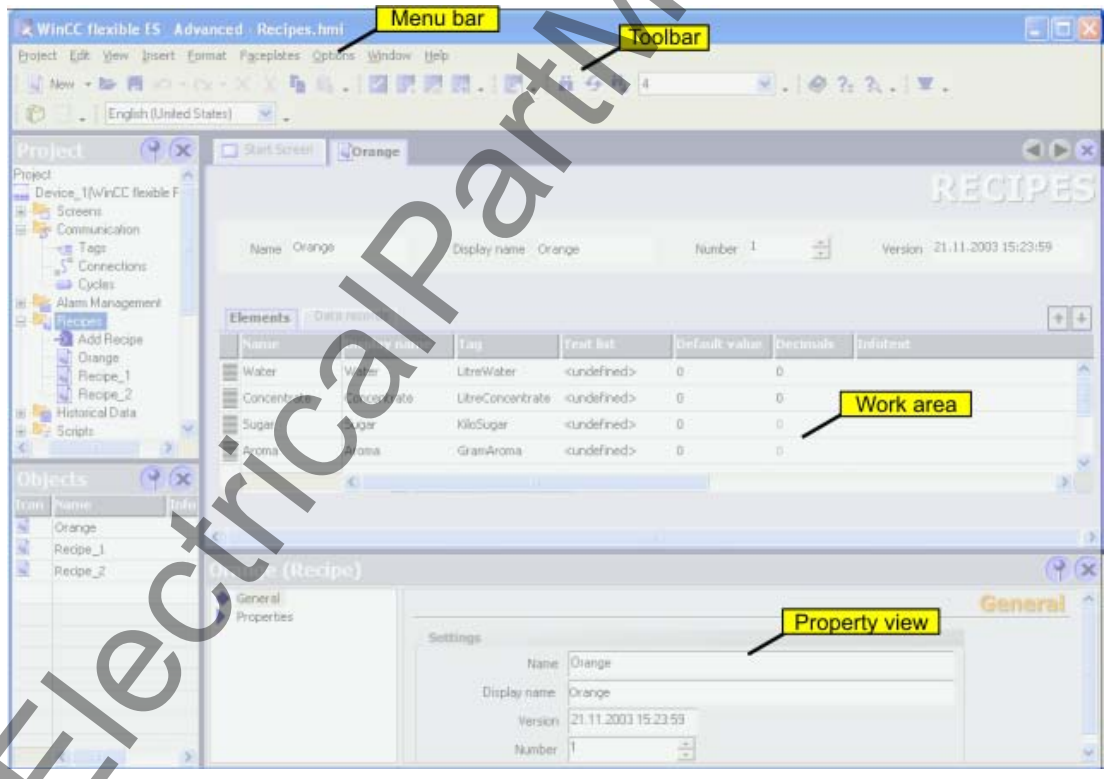
Introduction

You create, configure, and edit recipes in the "Recipes" editor. In addition, the "Recipes" editor can be used to enter values in existing recipe data records.

Opening the "Recipes" editor

Open the "Recipes" editor either by creating a new recipe or opening an existing recipe.

Structure of the "Recipes" editor



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Available keystroke combinations are indicated beside the respective menu command.

"Recipes" toolbar

The toolbars contain the most important commands from the menus.

www.ElectricalPartManuals.com

Work area

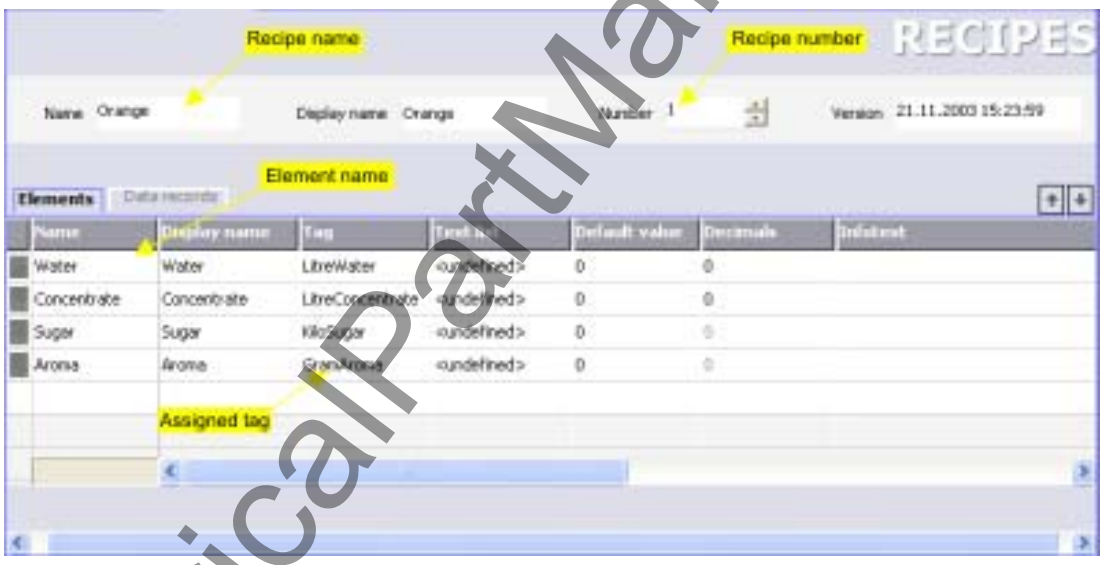
This area is used to create and edit recipe elements and recipe data records. A recipe is defined in the "Elements" tab. A recipe is defined in the "Elements" tab.

Property view

This view is used to configure the recipe. For additional information on recipe settings, refer to "Recipe settings."

8.2.2 Recipe elements

Structure of the "Elements" tab



Elements in the "Elements" tab

The individual recipe elements are briefly described below.

Recipe name

The recipe name identifies the recipe uniquely within the project.

Display name

For example, the display name of the recipe appears during runtime in the recipe view and can be configured in several languages. It is possible to assign descriptive names or designations that relate directly to a product, such as "FruitJuice_Orange."

www.ElectricalPartManuals.com

Recipe number

The recipe number identifies the recipe uniquely within the project.

Version

The version identifies the date and time of the most recent change made to the recipe.

Element name

The element name identifies a recipe element uniquely within the recipe. It is possible to assign unique descriptive names or designations, such as axis designations, on a machine or ingredients such as "flavoring."

Assigned tag

Each recipe element is assigned a recipe tag in which the recipe data record value is stored during runtime.

Default value

The default value is used as the default entry when you create a new recipe data record.

Text List

Text is assigned to a value or value range in a text list. This text can then be displayed in an output field, for example.

Decimal places

This number defines exactly how many decimal places will be displayed for the recipe data record value during runtime.

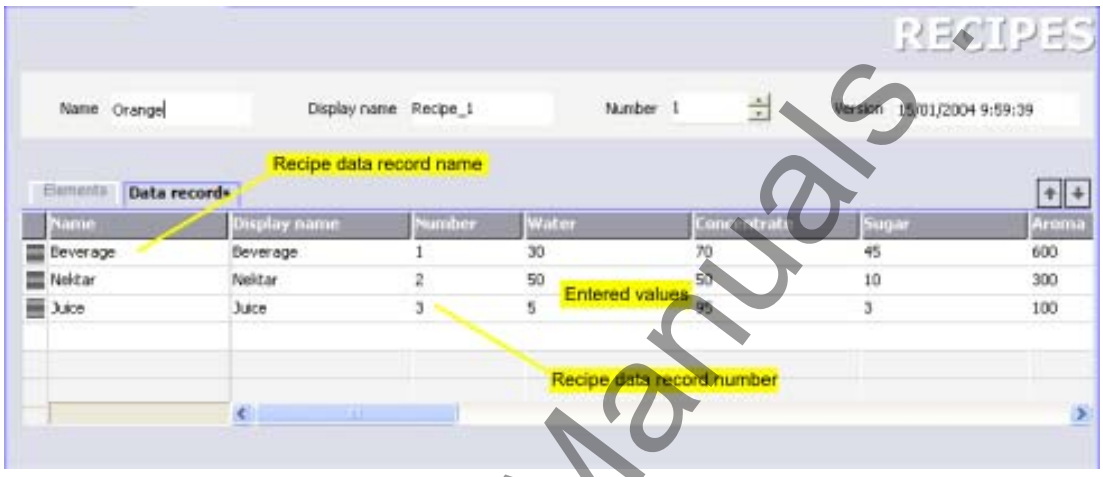
Infotext

Here you can enter a help message about the recipe element that will be displayed to the user during runtime.

www . ElectricalPartManuals . com

8.2.3 Recipe data records

Structure of the "Data records" tab



Elements in the "Data records" tab

The "Data records" tab includes the following elements:

Recipe data record name

The recipe data record name identifies the recipe data record uniquely within a recipe.

Display name

The display name of the recipe data record appears during runtime in the recipe view and can be configured in several languages. It is possible to assign descriptive names or designations that relate directly to a product, such as product numbers.

Recipe data record number

The recipe data record number identifies the recipe data record uniquely within a recipe.

Entered values

You can enter values in a recipe data record during configuration. When the project is transferred to the HMI device, the recipe data records are also transferred. If the HMI device already has data records, they are overwritten after a user prompt is displayed and based on the transfer settings.

Infotext

Here you can enter a help message about the recipe element that will be displayed to the user during runtime.

www . ElectricalPartManuals . com

8.2.4 Recipe settings

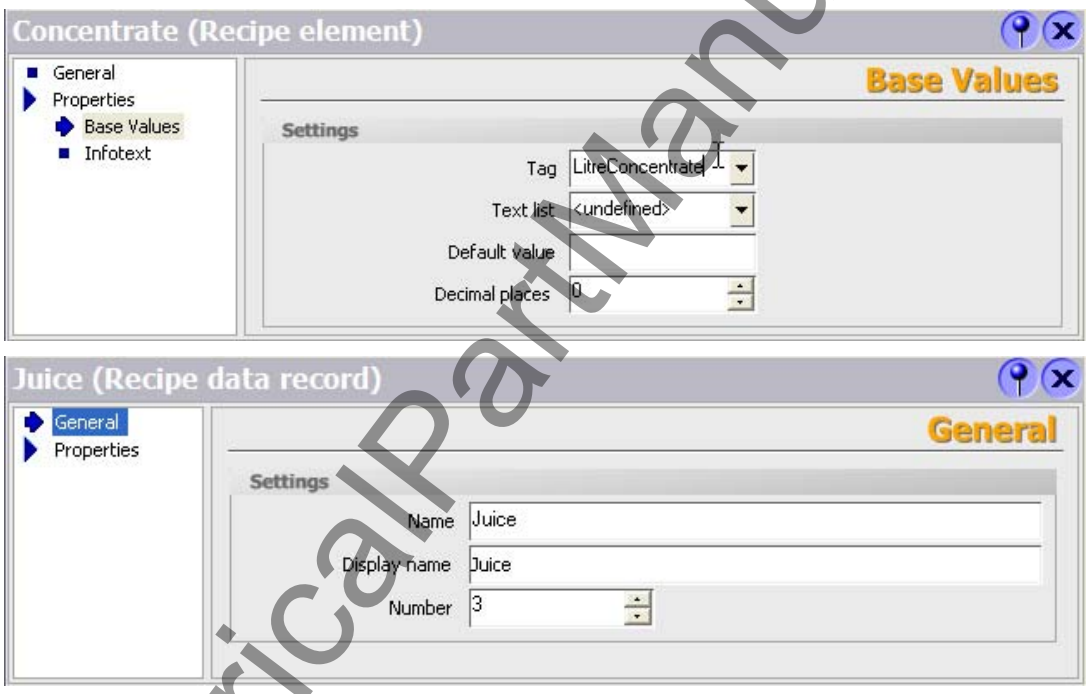
Introduction

You enter recipe settings for a recipe in the property view.

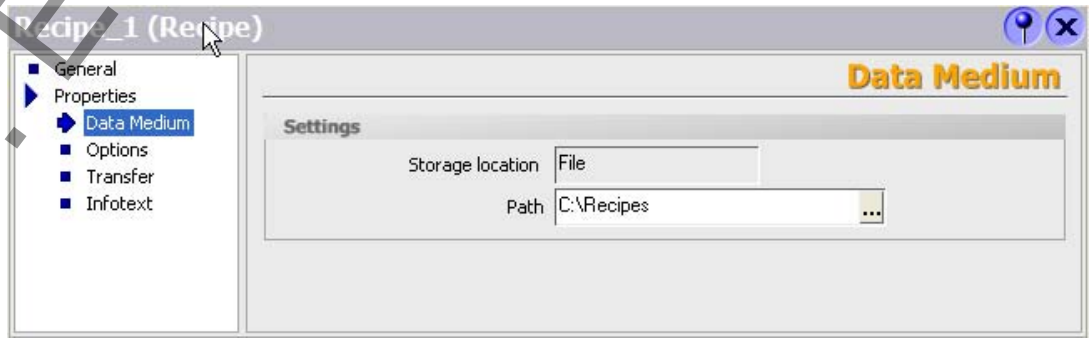
Principle

The display in the property view is dependent on the selection you made in the "Recipes" editor. If you are editing recipe elements or recipe data records on the "Elements" or "Data records" tab, you can also modify the contents in the property view.

In order to modify recipe settings, click the "Recipe name" or "Recipe number" field in the "Recipes" editor. The following settings can be modified in the property view:



For example, in "Data medium" under the "Properties" group, you define where the file containing the recipe data records will be saved. The selection capabilities are dependent on the operator panel used. Depending on the equipment of the operator panel, select the place where you want to save the Flash memory or an MMC memory card of the operator panel. When using WinCC flexible Runtime as an operator panel, save the file on the hard disk drive of the computer. Enter the path directly or navigate to the desired index of the data medium using the dialog.



www.ElectricalPartManuals.com

In the "Properties" group, you configure the behavior of the recipe during runtime in "Settings" and "Transfer." In "Settings," for example, you define whether values of the recipe tags should be transferred immediately to the PLC during runtime.

8.3 Viewing and editing recipes in Runtime

8.3.1 Viewing and editing recipes in Runtime

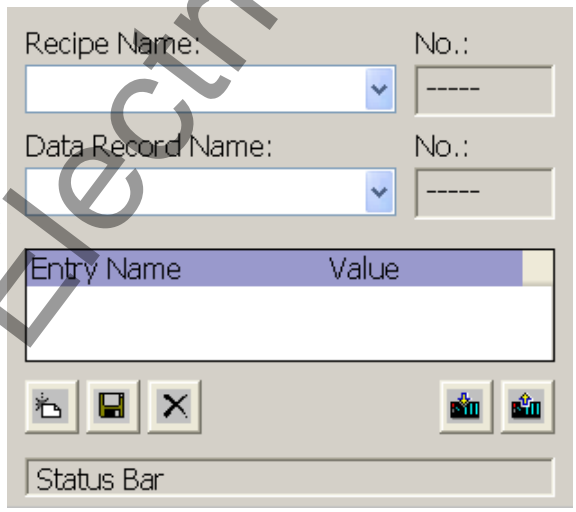
Introduction

The WinCC flexible ES offers you two configuration options of viewing and editing recipes and their corresponding data records in Runtime on the HMI device:

- Recipes view
- Recipe screen

Recipes view

The recipe view is a screen object that is configured in the "Screens" editor. For example, you can specify what operating function the recipe view will have in Runtime:

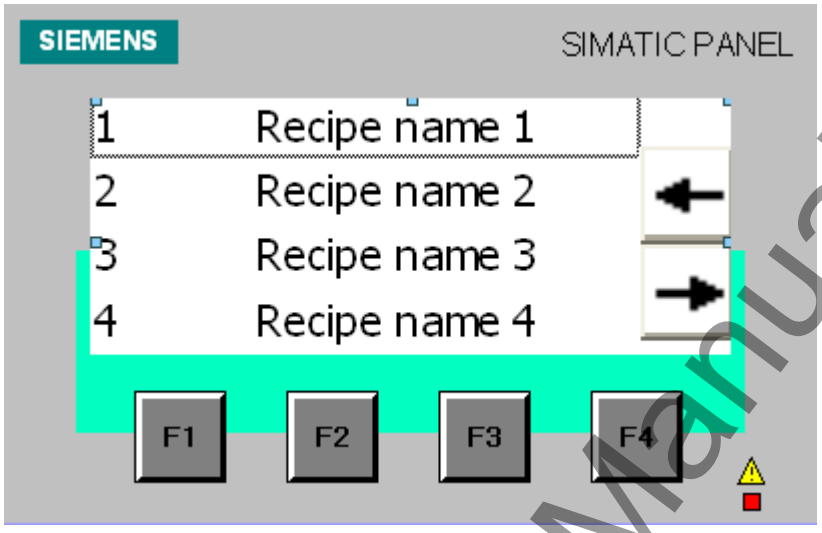


The recipe view shows recipe data records in tabular form. The Recipe view is particularly useful if data records are small in size or only a few values are to be modified.

www . ElectricalPartManuals . com

Simple recipe view

On HMI devices which have a display smaller than 6" (e.g. OP 77B), the simple Recipe view is used to display and edit recipes.



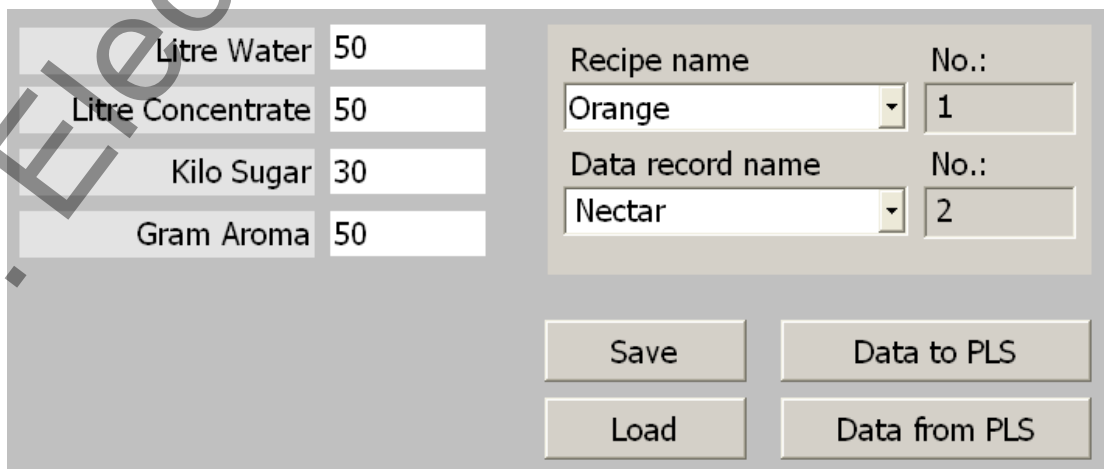
The simple recipe view consists of three areas:

- Recipe selection
- RecipeDataRecordSelection
- Recipe entries

In the simple recipe view, each area is shown separately on the HMI device. The simple recipe view always begins with the recipe selection.

Recipe screen

A recipe screen is a process screen with a customized input screen form that you create by setting up input/output fields and other screen objects in the "Screens" editor. This makes it possible for you to input parameter data in the context of machine visualization. The I/O fields for a recipe can be distributed over multiple recipe screens, which allows you a topical organization of recipe elements. The operating functions for the recipe screens must be configured explicitly in the process screens.

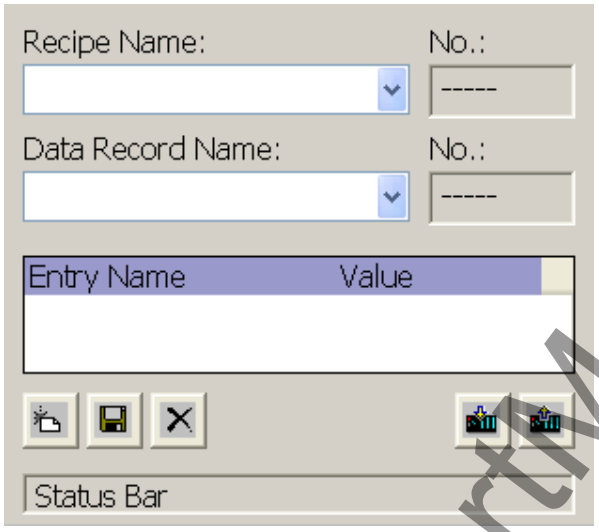


www.ElectricalPartManuals.com

8.3.2 Basic principles of the recipe view

Introduction

The recipe view is a screen object that you use to display and edit recipe data records during runtime.



Configuring

The functions of the recipe view can be configured.

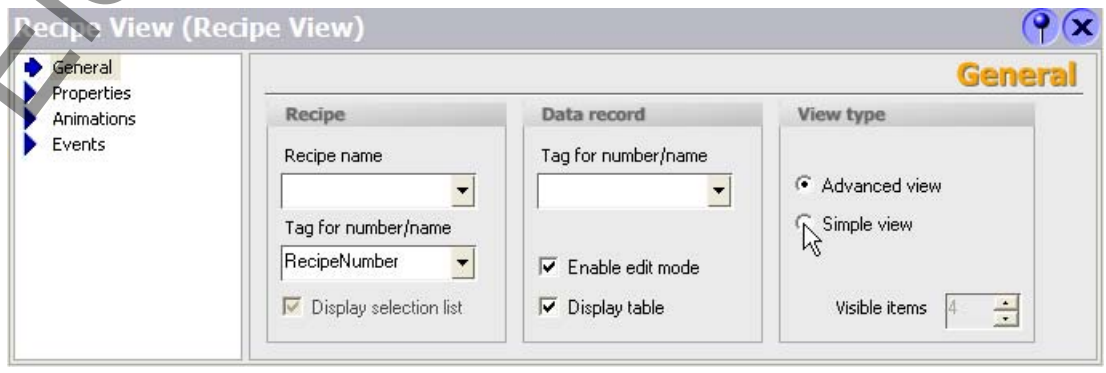
Moreover, you can specify whether available recipes and associated recipe data records can be both selected and modified in the recipe view or simply selected.

8.3.3 Basic principles of the simple recipe view

Introduction

On HMI devices which have a display smaller than 6" (e.g. OP 77B), the simple recipe view is used to display and edit recipes.

The recipe views can be used on all other HMI devices.



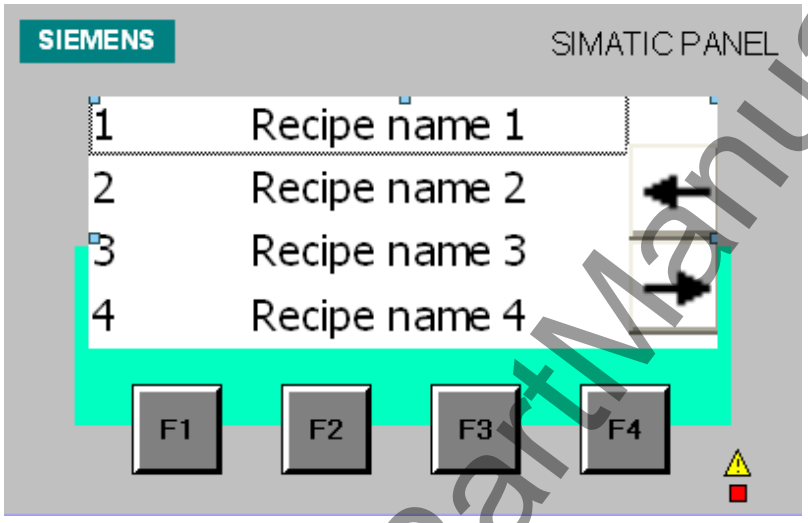
www . ElectricalPartManuals . com

Layout and function

The simple recipe view consists of three areas:

- Recipe selection
- RecipeDataRecordSelection
- Recipe entries

In the simple recipe view, each area is shown separately on the HMI device. The simple recipe view always begins with the recipe selection.

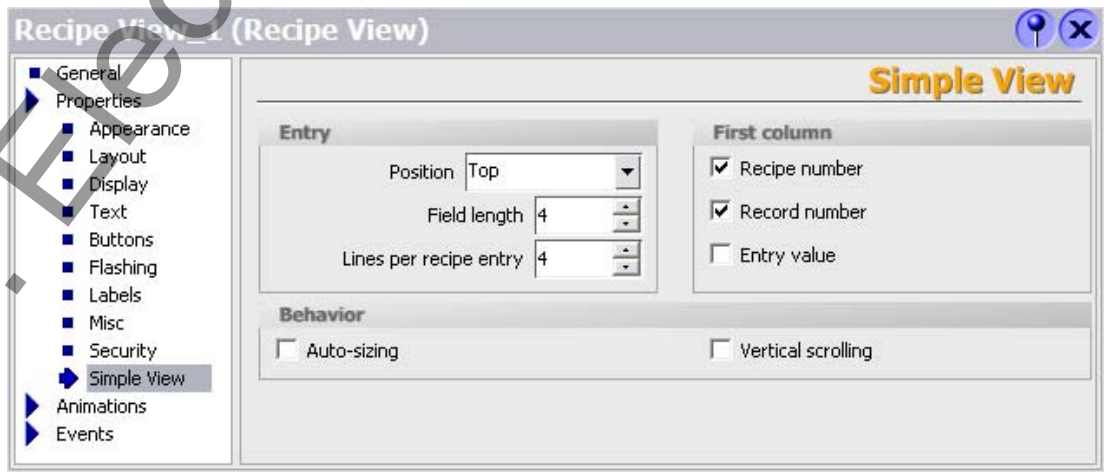


A command option selection can be called in for each display area by pressing the ⇨ button. The command selection lists those commands that are available in the current display area. Each command is assigned a number which you can use to select it directly from the list (without using the <Enter> key).

Configuration

To configure the simple recipe view, select "Recipe view" under "Enhanced objects" in the toolbox view.

The "Simple view" group is also available in the property view for configuring the simple recipe view.



www.ElectricalPartManuals.com

The following system functions can be configured on function keys of the HMI device for operator control of the simple recipe view:

- RecipeViewMenu: Opens the selection of commands
- RecipeViewOpen: Depending on the selection, the system shows either the recipe records or recipe entries
- RecipeViewBack: Returns to the previous display area

8.3.4 Operator control elements of the recipe view

Operator control elements of the recipe view

The following operator control elements can be configured in the recipe view:

Operator control element	Function
	Displays the configured operator notes of the given recipe view.
	Creates a new recipe record in the recipe that is displayed in the specified recipe view. The recipe record values are preset with the values that were specified as "Basic value" when the recipe was configured.
	Saves the recipe data record which is currently displayed in the recipe view. You specify the storage location during configuration in the "Properties" > "Data medium" group in the property view.
	Saves the recipe record currently being displayed in the recipe view under a new name. You specify the storage location during configuration in the "Properties" > "Data medium" group in the property view.
	Deletes the recipe record that is displayed in the recipe view from the data medium of the HMI device.
	Synchronizes the values of the recipe record that is currently displayed in the recipe view with the associated tags. During synchronization, only those values that have been modified in the recipe view are written to the associated tags. The values are then read from the tags and used to update the recipe view.
	Transfers the recipe record, which is currently displayed in the recipe view, to the connected PLC.
	Transfers the recipe record, which is currently loaded in the PLC, to the HMI device and displays it in the recipe view.

You can also configure system functions for operating buttons. This is the case, for example, if the buttons in the recipe view are not to be used or the HMI device does not have touch functionality.

Note

The operator control elements are shown as menu functions in the simple recipe view.

www . ElectricalPartManuals . com

8.3.5 Behavior of the recipe view in Runtime

Screen change

If you change to another screen and have not yet saved changes to the recipe data in the recipe view, you will be prompted to save the recipe data. The recipe name and the name of the recipe record are displayed to show which recipe data have not been saved yet.

If you change to a process screen that contains a recipe view with loaded recipe data, the recipe data will be automatically updated.

Operating the recipe view with softkeys

The Recipe view can be operated with function keys, e.g. when the HMI device does not have touch functionality. System functions allow you to assign functions such as "Save data record" to the function keys of the HMI device.

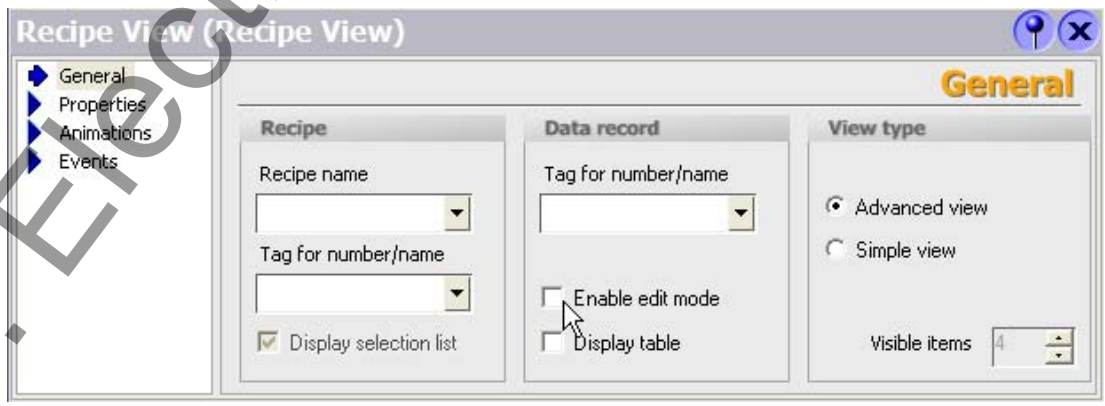
Display after import of recipe data

If you open the recipe view during the import of recipe data, only the recipe data that is already completely imported will be displayed. The recipe view is not automatically updated with a data import. In order to have a complete view of all recipe data, only open the recipe view after the system alarm informs you that the import of recipe data was successful. Alternatively, update the recipe view after successful completion of the import procedure.

8.3.6 Configuration options for the recipe view

Using the recipe view as a drop-down list

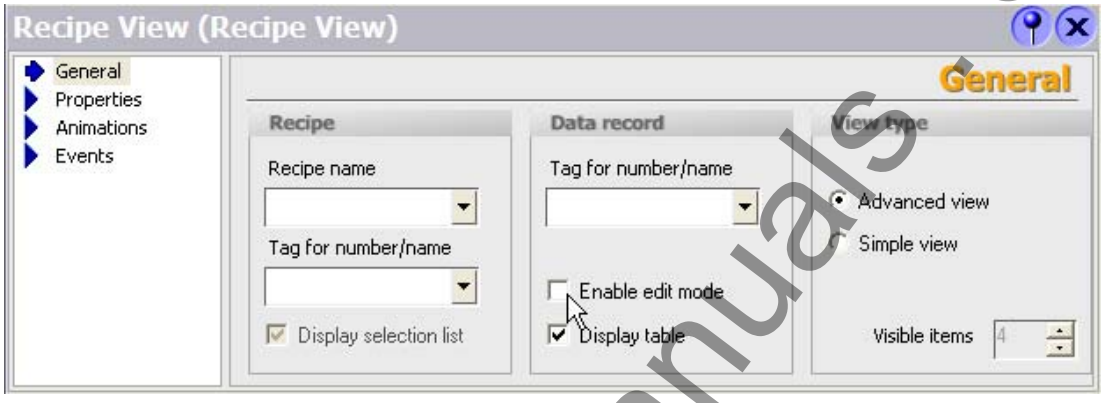
You can use the recipe view as a drop-down list for recipes or recipe data records (or both) in a recipe screen. To accomplish this, hide all operator control elements as well as the table for the recipe data records. The process screen then displays only two drop-down lists in which the recipe and recipe data record can be selected.



www.ElectricalPartManuals.com

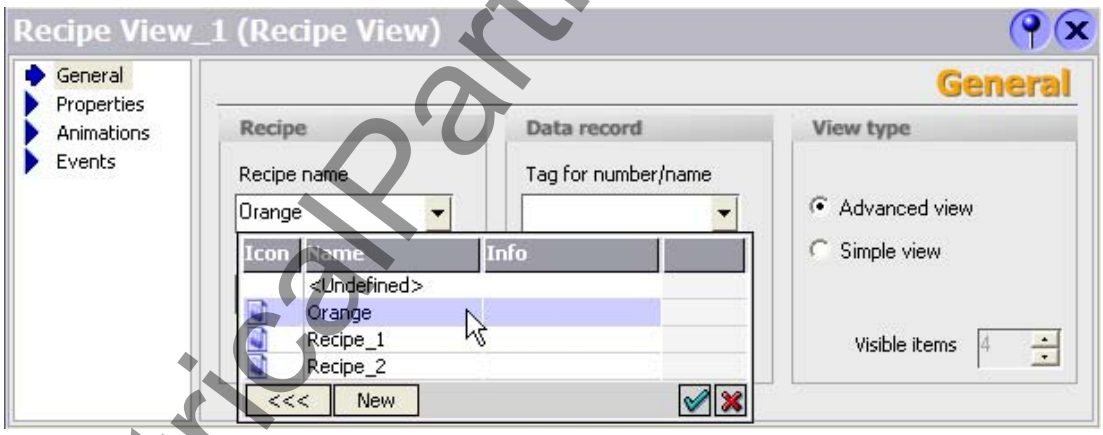
Displaying recipe data record values only

If you want to display recipe data in a recipe view for inspection only, you can prevent editing of the recipe data records. To do so, deactivate "Enable edit mode".



Displaying a particular recipe

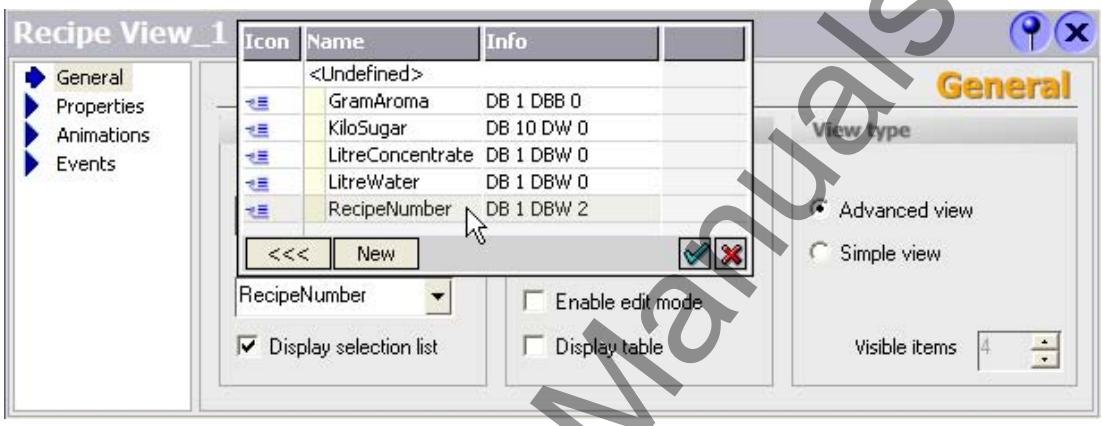
If you only want to authorize access to recipe data records of a particular recipe in a process screen, you can specify the recipe in the recipe view.



www . ElectricalPartManuals . com

Writing a recipe number or name and recipe data record number or name to a tag

Both the recipe and the recipe data record can each be linked to a tag in the recipe view. If you select a recipe or a recipe data record, its number or name is stored in the tag. Conversely, you can use the tag to select a recipe or recipe data record by entering the corresponding value. The tag type determines whether the name or the number is stored. If you want to store the name, you must specify a tag of type STRING. You can, for example, transfer the tag as a parameter for a system function.



8.3.7 Basic principles of the recipe screen

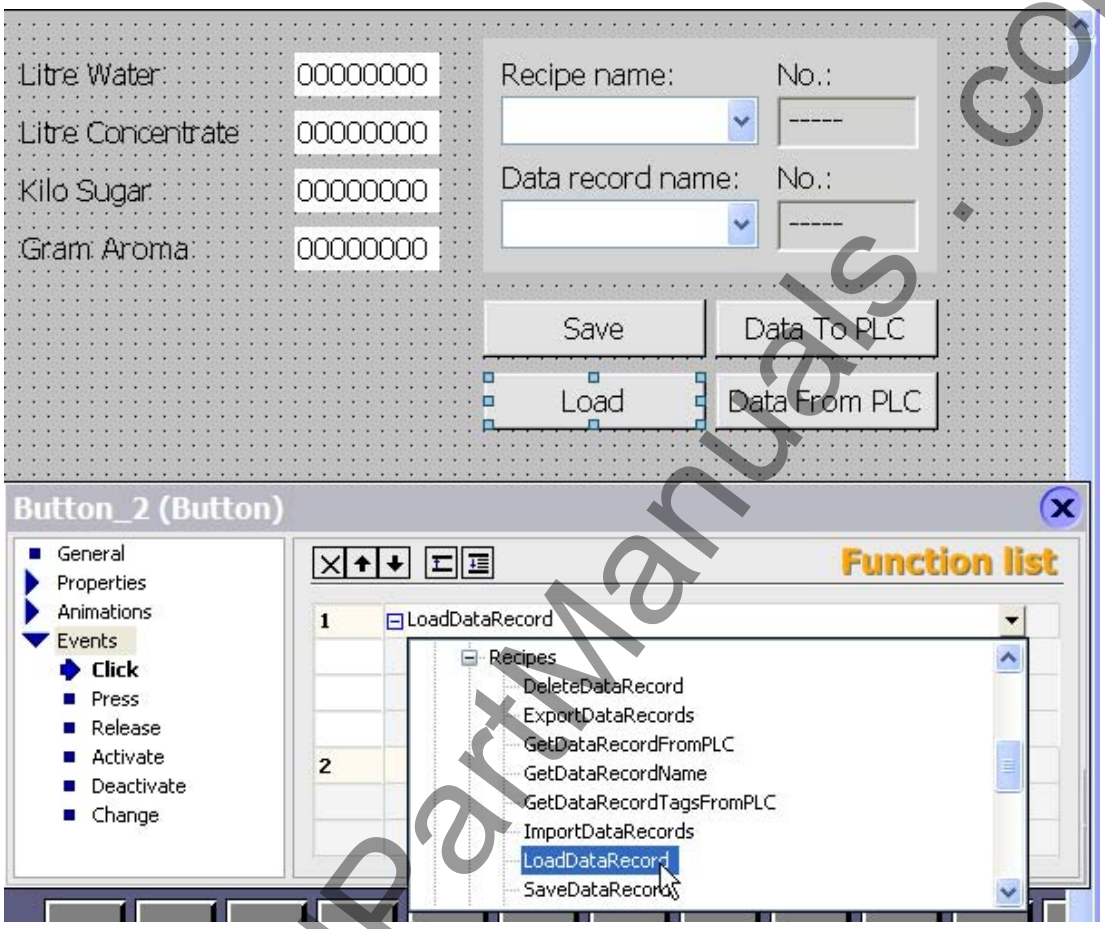
Introduction

A recipe screen is a process screen in which you configure a customized input mask in the "Screens" editor. The input mask is created from input/output fields and other screen objects. System functions are used to configure the recipe functionality, such as saving recipe data records.

Note

You can configure a recipe screen in the TP 170B and higher models.

www . ElectricalPartManuals . com



Principle

Configuration of a recipe screen offers you the opportunity for customization: You can spread large recipes over several process screens according to topic and display them vividly, using features such as graphical screen objects.

- Spreading recipes over several process screens according to topic

  You can spread recipe data records containing many entries over several process screens. For example, for each plant section you can configure a process screen containing the associated input masks for the recipe data records. The process for manufacturing table tops, for example, can be divided into process screens for in-feed, trimming, cutting, drilling, sanding, and packaging operations.

  Spreading recipes over several process screens is useful for HMI devices with small displays. For one thing, you can avoid having to scroll in tables during runtime.

- Visual machine simulation

  You can visually simulate your machine in a process screen using graphical screen objects. This enables you to display parameter assignment settings more vividly by placing input/output fields directly next to machine elements such as axes or guide rails. You can use this to produce a direct reference between the values and the machine.

www.ElectricalPartManuals.com

Configuration settings

You must select "Synchronize tags" in the property view in order to be able to enter the recipe data record values in the configured input/output fields outside the recipe view.

If the entered values are to be transferred immediately to the connected PLC during runtime, you must disable "Tags offline" in the property view.

Configure the "SetRecipeTags" system function if you want to enable and disable immediate transfer of entered values during runtime.

System functions

The following system functions are available for operator control of a recipe screen.

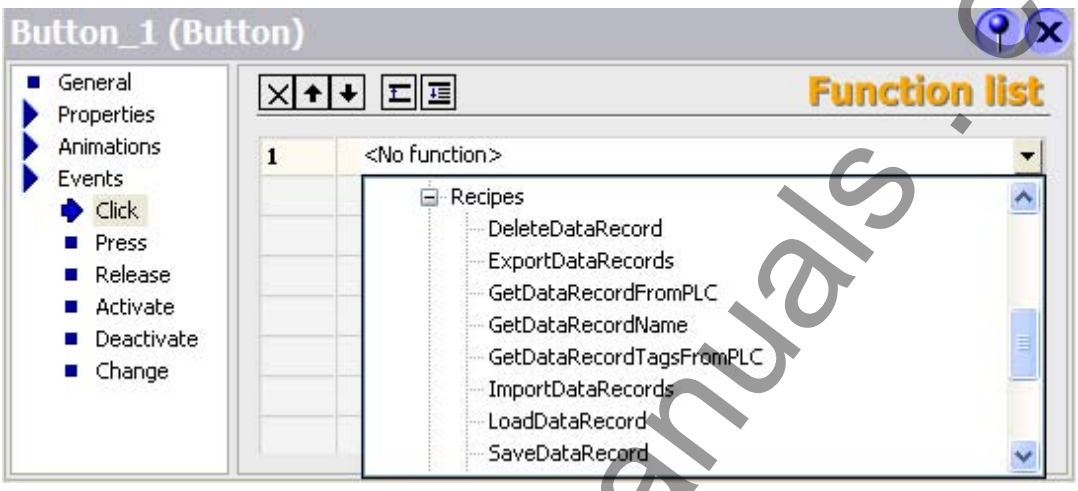
- ImportDataRecords
- ExportDataRecords
- LoadDataRecord
- SaveDataRecord
- SetDataRecordTagsToPLC
- GetDataRecordTagsFromPLC

The following system functions are available for operator control of the recipe view when it is being used in the recipe screen.

- RecipeViewSaveDataRecord
- RecipeViewSaveAsDataRecord
- RecipeViewSynchronizeDataRecordWithTags
- RecipeViewDeleteDataRecord
- RecipeViewNewDataRecord
- RecipeViewGetDataRecordFromPLC
- RecipeViewRenameDataRecord
- RecipeViewShowInfoText
- RecipeViewMenu (for simple recipe view only)
- RecipeViewOpen (for simple recipe view only)
- RecipeViewBack (for simple recipe view only)

www.ElectricalPartManuals.com

The system functions for loading, saving, and transferring recipe data records and recipes are located in the "Recipes" group.



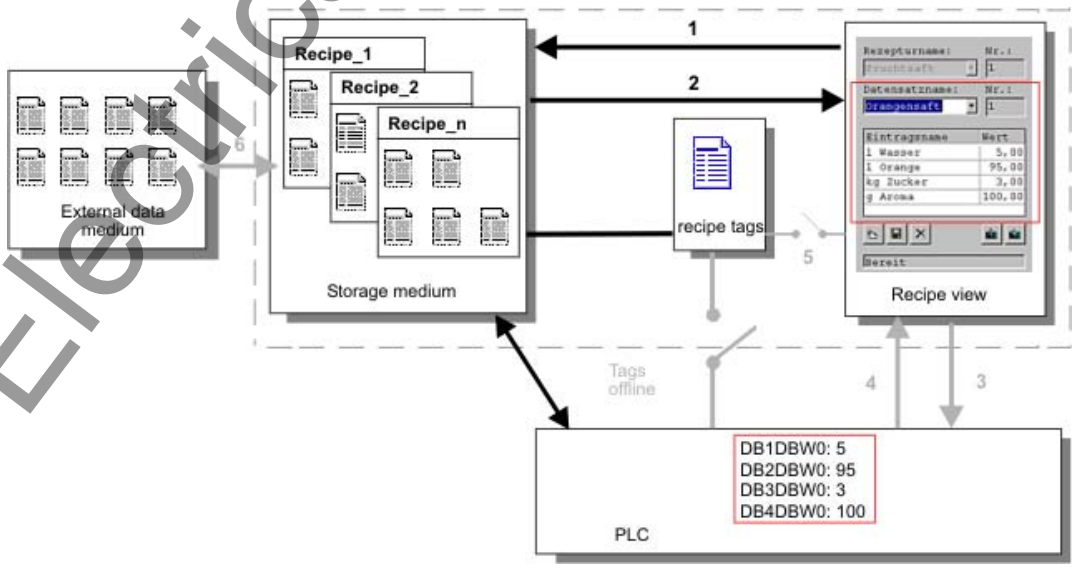
8.4 Scenarios

8.4.1 Scenario: Entering recipe data records in Runtime

Objective

You want to enter production data on the HMI device without disturbing the process that is currently underway. Therefore, the production data should not be transferred to the PLC.

Sequence



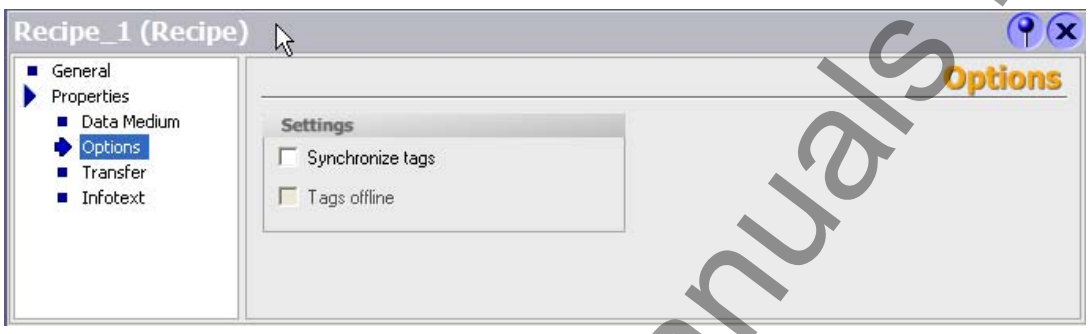
You enter the production data in the recipe view or the recipe screen, assign a recipe data record name, and save the new recipe data record on the storage medium of the HMI device.

www . ElectricalPartManuals . com

Configuration in WinCC flexible

You configure the recipe along with the associated tags.

Synchronization with the recipe tags is not necessary, because production data (tags) are not intended to be transferred to the PLC. Make the following settings for the recipe in the property view:



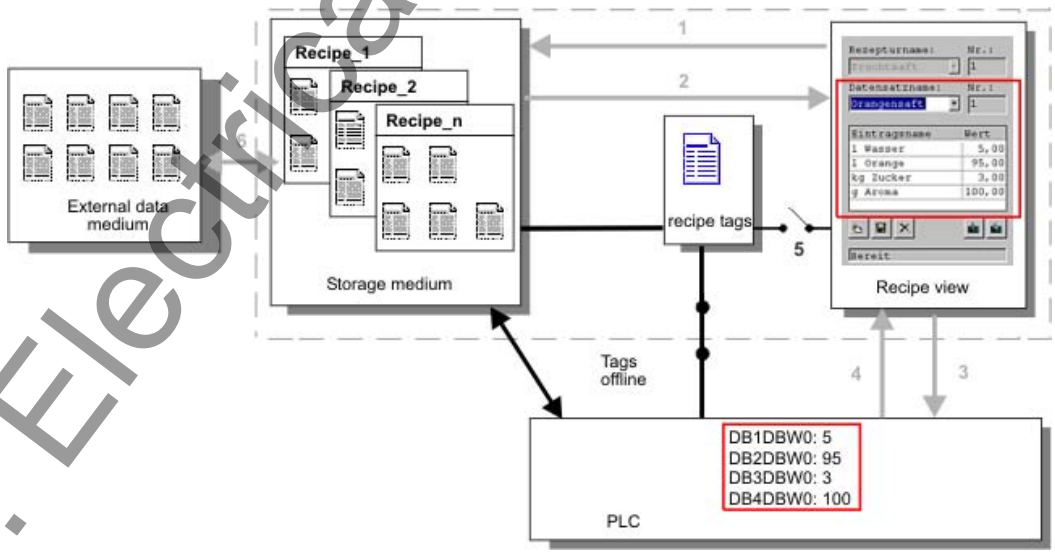
Depending on the extent of the recipe, you either configure a recipe view or create a recipe screen.

8.4.2 Scenario: Manual production sequence

Objective

The production data are to be requested by the PLC according to the work piece to be processed and displayed on the HMI device for inspection. You want to be able to correct the transferred production data online, if necessary.

Sequence



A reading device connected to the PLC reads a bar code on the work piece to be processed. The recipe data record names correspond to the respective bar code names. This will enable the PLC to load the necessary recipe data record from the storage medium of the HMI device. The recipe data record is displayed for inspection. Changes are transferred immediately to the PLC.

www . ElectricalPartManuals . com

Configuration in WinCC flexible

You configure the recipe along with the associated tags.

Production data are to be transferred to the PLC, so it is necessary to synchronize with the PLC to prevent the data from accidentally overwriting each other. The tags are to be transferred to the PLC. Make the following settings for the recipe in the property view:

Depending on the extent of the recipe, you either configure a recipe view or create a recipe screen.

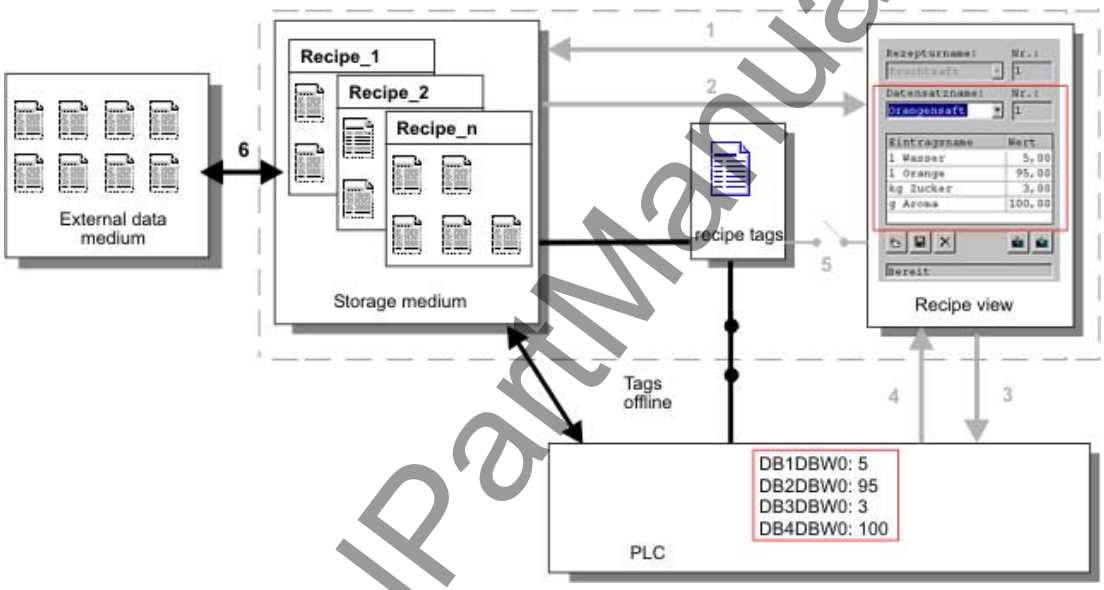
www . ElectricalPartManuals . com

8.4.3 Scenario: Automatic production sequence

Objective

You want production to be executed automatically. The production data should be transferred directly to the PLC either from the data storage medium in the HMI device or from an external data storage medium. The production data do not have to be displayed.

Sequence



Production can be controlled using one or more "Scripts", which transfer production data records automatically to the PLC. The sequence can be checked using the return values of the utilized functions.

Configuration in WinCC flexible

You can implement the automatic production sequence with available system functions. The "ImportDataRecords" system function loads data records from a CSV file to the data medium. The "SetDataRecordTagsToPLC" system function transfers a data record from the data storage medium to the PLC.

www . ElectricalPartManuals . com

9 Logging and displaying tags

9.1 Basics

9.1.1 Basic principles for data logging

Introduction

Data logging is used to capture, process and log process data from industrial equipment.

The collected process data can then be analyzed to extract important business and technical information regarding the operational state of the equipment.

Application of the data logging

You can use data logging to analyze faults and to document the process run. By analyzing data logs, you can extract the information necessary to allow you to optimize maintenance cycles, increase product quality and ensure that quality standards are met.

9.1.2 Trends

Introduction

A trend is a graphic representation of the values that a tag takes during runtime. In order to display trends, configure a trend view in a screen of your project.

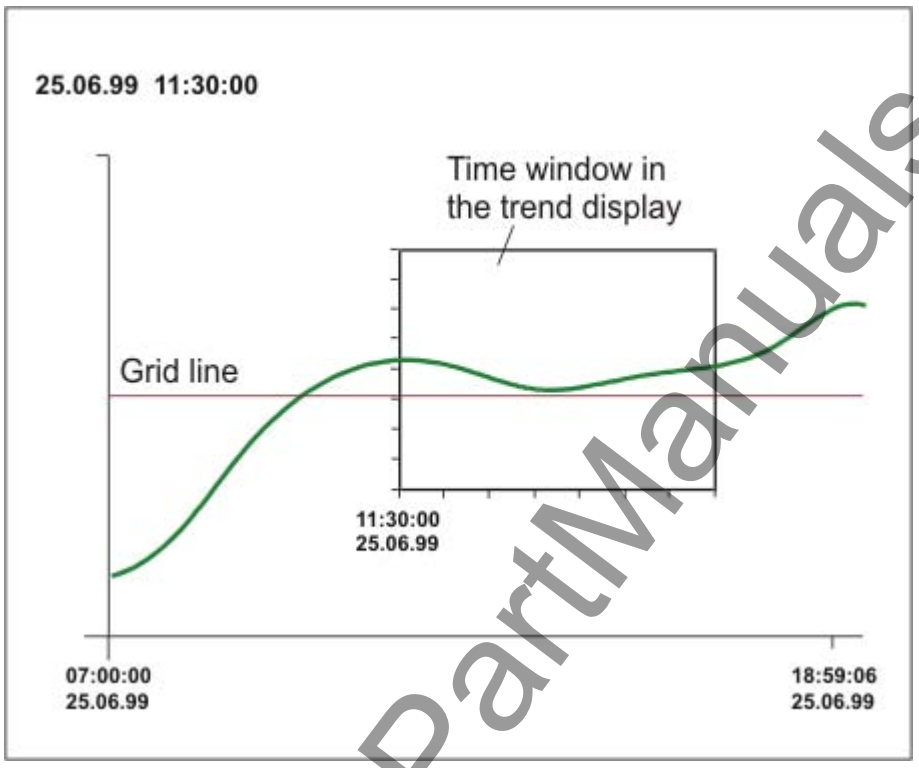
To configure the trend view, specify a trend type for the values to be displayed.

- Log: For displaying the logged values of a tag
- Realtime pulse triggered: For time-triggered display of values
- Realtime bit triggered: For event-triggered display of values
- History bit-triggered: For event-triggered display with buffered data acquistion

www . ElectricalPartManuals . com

Displaying logged values

The trend view shows the logged values within a definable time period. In runtime, the operator can shift the time period to view the desired information (logged data).



Pulse-triggered trends

The values to be displayed are determined individually with a definable time pattern. Pulse-triggered trends are suitable for representing continuous courses such as the changes in the operating temperature of a motor.

Bit-triggered trends

The values to be displayed are event-driven by setting a defined bit in "Trend transfer" tags. The bit is reset after reading has been completed. Bit-triggered trends are useful for displaying fast changing values such as the injection pressure for producing plastic parts.

Bit-triggered trends with buffered data acquistion

When you set buffered data acquistion, the values to be displayed are buffered in the PLC and read in bit-triggered as a block. These trends are suitable for displaying rapid changes when the course of the trend as a whole is interesting and not so much the individual values.

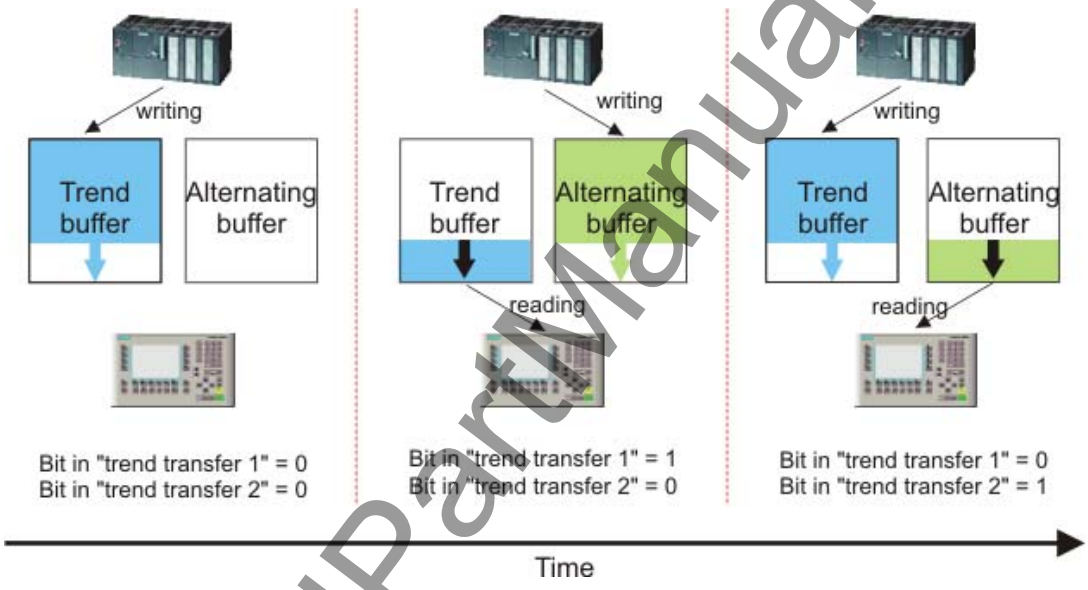
You configure a switch buffer in the PLC so it can continue to write the new values while the trend buffer is being read. The switch buffer ensures that the PLC does not overwrite values while the operator devices reads the values for the trend.

www.ElectricalPartManuals.com

The switch between the trend buffer and the switch buffer functions as follows:

Whenever the bit which is assigned to the trend is set in the "Trend transfer 1" tag , all the values are read simultaneously from the trend buffer and displayed as a trend at the operator device. The bit in "Trend Transfer 1" is reset after reading has been completed.

While the operator device is reading the tag values from the trend buffer, the PLC writes the new tag values into the switch buffer. When the bit which is assigned to the trend is set in the "Trend transfer 2" tag , all the trend values are read from the switch buffer and displayed at the operator device. While the operating device is reading the switch buffer, the PLC writes again to the trend buffer.



9.1.3 Data logging in WinCC flexible

Introduction

Data is information that is collected during the process and saved in the memory of one of the connected automation systems. This data reflects the state of the equipment, e.g. temperatures, fill levels or states (e.g. motor off). To work with the process variables, you must define tags in WinCC flexible.

In WinCC flexible, external tags are used to collect process values and to access a memory location in a connected automation system. Internal tags are not connected to any process and are only available to their respective HMI device.

Principle

The values from external and internal tags can be saved in data logs. You can individually specify the log in which each tag will be saved.

Data logging is controlled via cycles and events. Logging cycles are used to ensure continuous acquisition and storage of the tag values. In addition, data logging can also be triggered by events, e.g. when a value changes. These settings can be made for each tag individually.

In runtime, the tag values which are to be logged are captured, processed and stored in an ODBC database or a file.

www . ElectricalPartManuals . com

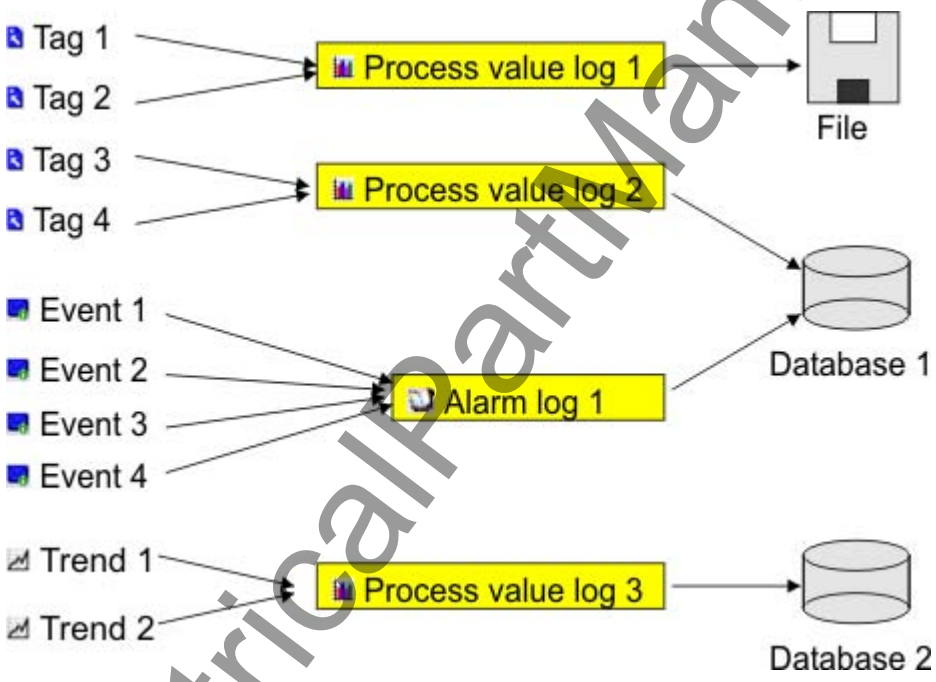
Log types

In WinCC flexible, you can select from the following log types:

- Circular log
- Segmented circular log
- Circular log which sends a system alarm message when it is full
- Circular log which executes system functions when it is full.

Storage media and location

The logged data will be stored in either an ODBC database (only on a PC) or a file.



Depending on the hardware configuration of the HMI device, the data may be logged locally (on the hard disk of a PC or on the storage card of a panel) or, if present, on a network drive.

Saved data can undergo additional processing in other programs, e.g. for analysis purposes.

Outputting the contents of a log

In runtime, you can output the logged tag values as trends in the process screens.

www . ElectricalPartManuals . com

9.2 Elements and basic settings

9.2.1 "Data Logs" editor

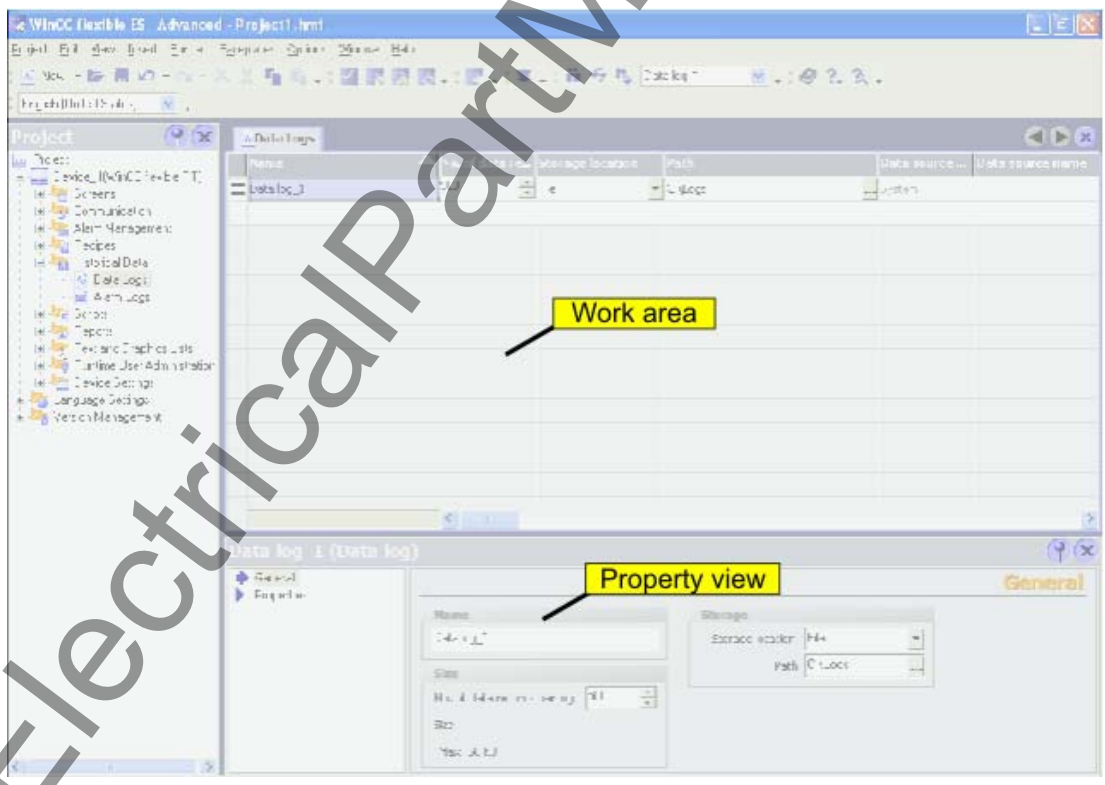
Introduction

To log process values, they must be assigned to a log. You can plan logs and specify their properties in the tabular "Data Log" editor.

Open

In the project window in the "Log" group, double-click on "Data logs" to open the "Data log" editor.

Layout



Work area

All the data logs are displayed in a table in the work area. You can edit the properties of the data logs in the table cells. You can sort the table according to the entries in a column by clicking on the column header.

You can show and hide the table columns. To do so, activate or deactivate the entries in the pop-up menu of the column table header.

www . ElectricalPartManuals . com

Property view

Here you configure data logs. The property view offers the same information and settings as the work area table.

9.2.2 Basic settings for data logs

Introduction

The properties of a data log can be defined in the "Data log" editor or in the Properties window for logs.

Structure of the property view

The Properties window has a tree structure on the left from which you can select all property categories. The fields for configuring the currently selected properties category are shown on the right of the Properties window.

You can set the following properties for data logs in the Properties window:

General properties

- Name

  The data log may be given any desired name, but the name must include at least one letter or number.

- Memory location

  The data log may be stored in an ODBC database (only on a PC) or in a separate "*.csv" file. Select "File" or "Database" as the storage location correspondingly.

  Depending on the configuration of the HMI device, you can select the local hard disk of the PC or the storage card of the panel or, if present, a network drive as "path."

  If you have chosen an ODBC database as the storage location, you can accept the name given by the system (system-defined data source name) or enter one yourself (user-defined data source name).

- Size

  The size of a log is calculated as follows:

  The number of items * the length of each tag value to be logged.

  In the Properties window, the minimum and maximum sizes of the log using the currently selected "Number of data records" are shown under the "Number of data records" input field. The maximum size of a is limited by the amount of storage on the HMI device.

www . ElectricalPartManuals . com

Settings for the log behavior

- Start-up behavior

  Under Enable you can specify that logging starts when runtime is started. Enable the checkbox "Enable logging at runtime start."

  You can also control the behavior at runtime start in other ways. Enable "Reset log" if you want to overwrite previously logged data with the new data or "Append data to existing log" if you want to append new data to an existing log.

  Note

  You can use system functions to control the Restart of a log during runtime.

- Logging method

  Here you can specify what should happen when the log is full. You can choose one of the following options:

  - Circular log: When the log is full, the oldest entry will be overwritten.
  - Segmented circular log: Multiple logs of the same size will be created and filled one after the other. When all logs are completely full, the oldest log is overwritten.
  - Display system message when: When a defined fill level is reached a system message is displayed.
  - Trigger event: The "Overflow" event is triggered as soon as the log is full.

- "Comment"

  Here you can enter descriptive text regarding the log.

Events

Here you can configure a function list which will be processed whenever an "Overflow" event is triggered by the overflow of the log.

www . ElectricalPartManuals . com

9.3 Logging tags

Introduction

In runtime, tag values can be stored in logs for later evaluation. For the logging of a tag, you must specify the log in which the values are to be stored, how often this should happen and whether only the tag values in a specific value range are to be saved.

Note

The main purpose of data logging is to log the values of external tags. However, you can also log the values of internal tags.

Principle

Several steps are involved in data logging:

- Creating and configuring data logs

  When creating a data log, you must define the following:

  - General settings, e.g. name, size, storage location
  - Behavior at runtime start
  - Behavior when the log is full

- Configuring the logging of tags

  You can specify a data log for every tag. This log records the values of the tags in runtime and other information, e.g. the time the value was logged.

  Furthermore, you can define when and how often the value of the tag should be logged. To perform the latter, you have the following options:

  - "On request":

    The tag values are logged by calling the "LogTag" system function.

  - "On change":

    The tag values are logged, as soon as the operator device detects a change of value in the tag.

  - "Cyclic continuous":

    The tag values are logged at regular intervals. In addition to the standard cycles available in WinCC flexible, you can add cycles of your own, which are based on the standard cycles.

    Furthermore, you can restrict the logging to those values that are within or outside of a tolerance band. In this manner, you can distribute tag values specifically to different logs for separate analysis later.

- Processing logged tag values further

  The logged process tag values can be evaluated directly in your WinCC flexible project, e.g. in a trend view, or with another application, e.g. Excel.

www.ElectricalPartManuals.com

9.4 Outputting logged data

9.4.1 Outputting tag values in screens

Introduction

In runtime you can output tag values in the screens of the operator device in the form of a trend. The data can be requested by the PLC from the current process or be loaded from a log database.

Displayed values

You have to configure a trend view in a screen so that tag values are displayed at the operator device. When configuring the trend view you specify which tag values are displayed:

- Current values from the PLC

  The trend can be continued either with individual values from the PLC (display in real-time) or with all the values which are stored in a buffer between two read processes from the PLC (display of an interval).

  The reading moment can be controlled by setting a bit or by means of a cycle.

- Logged tag values

  In runtime the trend view displays the values of a tag from a data log. The trend shows the logged values in a particular window in time. In runtime, the operator can shift the window in time to view the desired information (logged data).

9.4.2 The structure of a *.csv file with tags

Introduction

In the *.csv (Comma separated value) file format, table columns (name and value of entry) are separated by a semicolon. Each table row ends with a carriage return.

Example of a *.csv file

The example shows a file with logged tag values:

"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 AM";60.00;1;35886460358.73

"Var_107";"01.04.98 11:02:57 AM";59.00;1;35886460381.22

www . ElectricalPartManuals . com

Structure of a log file in *.csv format

The following values are entered in the individual columns of a WinCC flexible log file:

Parameter	Description
VarName	Name of the WinCC flexible tag
Time string	Time stamp as a STRING, e.g. readable date format
VarValue	Value of the tag
Validity	Validity: 1 = value is valid 0 = an error occurred (e.g. interrupted process connection)
Time_ms	Specify a time stamp as a decimal value (see below for conversion). Only needed to display the tag values in a trend.

Conversion of the time stamp decimal value

If the value needs to be processed using a different program, proceed as follows:

1. Divide Time_ms by 1,000,000.

   Example: : 36343476928:1 000 000 = 36343,476928

2. The whole number portion (36344) is the date calculated from 31.12.1899.

   Example: 36343 results in 02.07.1999

   You can now convert the time stamp value to days in Excel by assigning a corresponding format from the "Date" group to the cells, which contain the time stamp.

   Result: 37986 results in 31.12.2003

3. The value after the comma (0,476928) indicates the time:
   - Multiply the value (0,476928) by 24 results in the hours (11,446272).
   - Multiply the remainder (0,446272) by 60 results in the minutes (26,77632).
   - Multiply the remainder (0,77632) by 60 results in the seconds (46,5792).

   Sum 11:26:46.579

   This conversion is supported by Microsoft Excel, for example.

www . ElectricalPartManuals . com

9.4.3 Accessing the ODBC log database directly

Introduction

The storage location of a log can be a database or a file.

The database is addressed by means of its "Data source name" (DSN). Select the database you would like to use in WinCC flexible in the Windows Start menu under Settings > Control panel > ODBC data sources.

To store log data, specify the "Data source name" (DSN) instead of a directory name when making your configuration settings. With the DSN, you are referencing the database and the storage location.

Application

The entire functional scope of the database is available for additional processing and evaluation of log data.

Principle

You create the data source that connects to the database on the same computer that contains the runtime software. You then specify the DSN configured here when you create a log in WinCC flexible.

Using the ODBC interface, you can access the database directly with other programs such as MS Access or MS SQL server.

With the "StartProgram" system function, you can also configure a program call (for MS Access, for example) on the HMI device. This does not interrupt the runtime program sequence.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

10 Working with reports

10.1 Principles on the report system

Introduction

In WinCC flexible, reports are used to document process data and completed production cycles. You can report messages and recipe data in order to create shift reports, output batch data, or to document a manufacturing process for the acceptance test.

Overview

You can edit the report files in the graphic editor. In this editor, you configure the layout of the reports and determine the output data. You can add various objects for the output of data to a report file. Some of the toolbox objects are either available with restricted functionality or not at all. This depends on the HMI device you are configuring. Objects not available in the toolbox are grayed out and cannot be selected.

You can create separate report files for reporting different types of data. You can set the triggering of the output separately for each report file. You can choose to trigger the output of data at a specific time or in defined intervals, or by other events.

The modular structure of these features allows you to explicitly configure reports for different requirements.

Application Examples

At the end of a shift, create a shift report which contains the batch data and error events of the completed production.

You can create a report which outputs the recorded production data of a batch production.

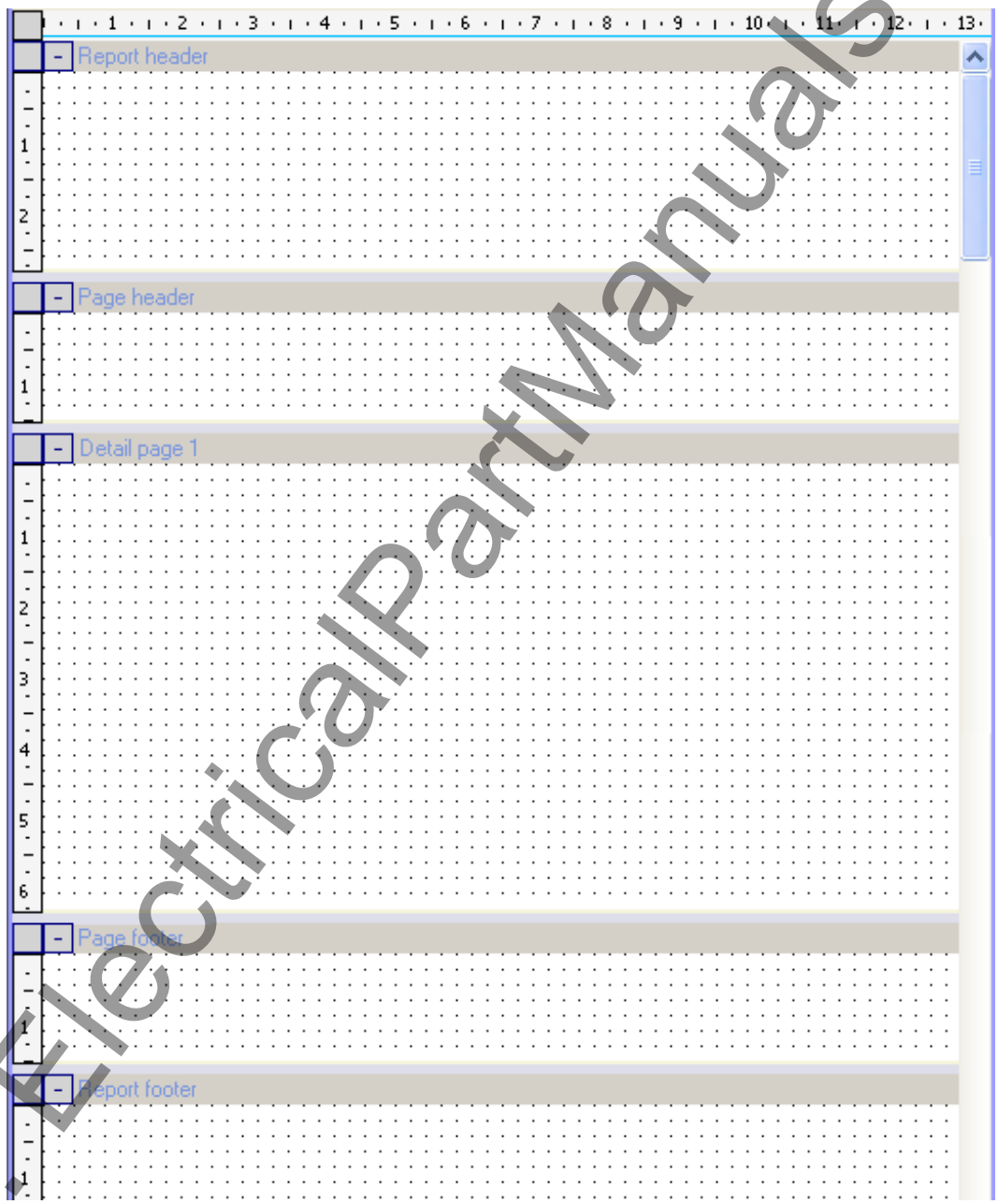
You can create a report which outputs messages of a certain class or type.

www . ElectricalPartManuals . com

10.2 Structure of reports

Structure of a report

The reports in WinCC flexible all have the same basic structure. They are subdivided into different sections as shown in the figure below.



www.ElectricalPartManuals.com

The individual sections are used to output different data and can contain general objects and specific report objects.

- Report header

  The report header serves as the cover sheet for a report. The report header is used to output the project title and general information on the project. The report header is output without page header and without page footer. The report header is output once at the start of a report.

- Report footer

  The report footer is used as the final page of a report. The report footer is used to output a summary of the report or other information which is required at the report end. The report footer is output without page header and without page footer. The report footer is output once at the end of a report.

- Page header

  The page header is output with every page of a report. The page header is used to output the date, time, title or other general information.

- Page footer

  The page footer is output with every page of a report. The page footer is used to output the page numbers, the total number of pages or other general information.

- Detail page

  The runtime data are output in the "Detail page" area. The objects for outputting the runtime data are inserted in the "Detail page" area. When you output the data, page breaks are added automatically depending on the amount of data. You can also insert several pages into a report in order to optically separate the configuration of various output objects.

The creation of a report is described in the "Creating a report" chapter.

10.3 Elements and basic settings

10.3.1 Protocols

Introduction

You can create and edit reports with the report editor.

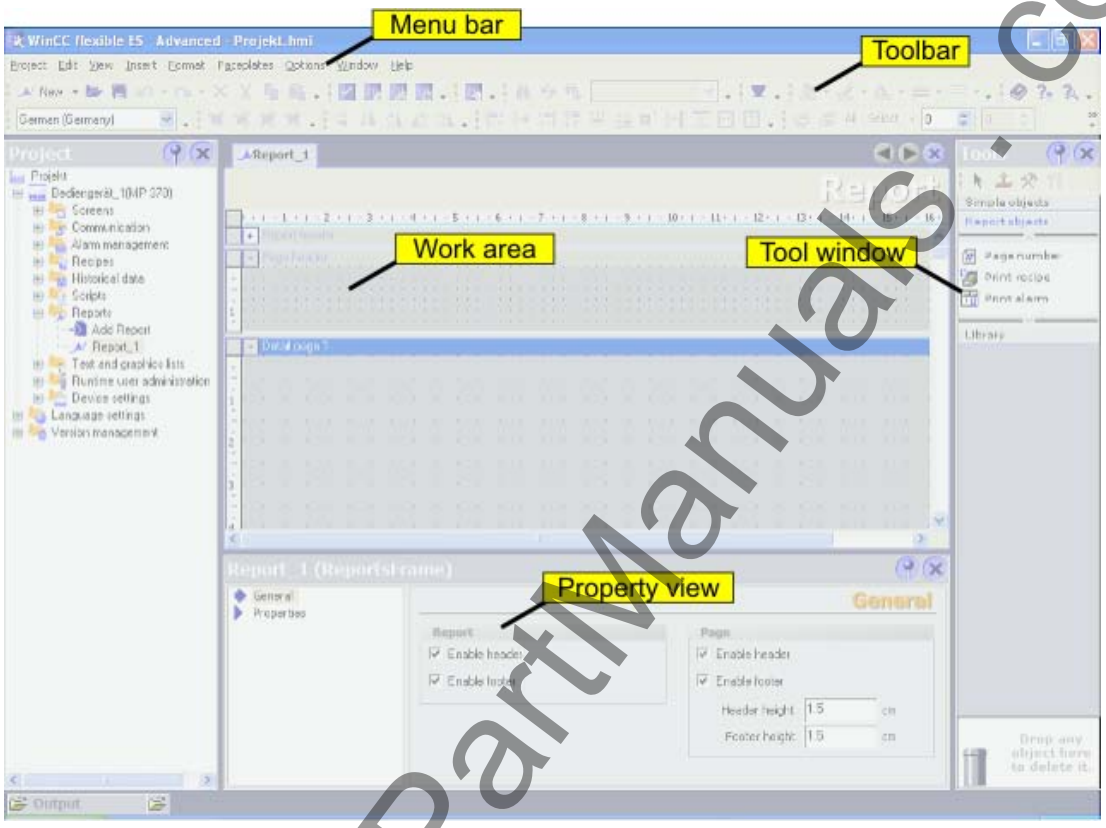
Open

Select the "Reports" entry in the project view and open the pop-up menu. Select the "New report" entry in the pop-up menu. A new report is created and opened in the work area.

In order to open an existing report, double-click in the object view on the desired report. The selected report is opened.

www.ElectricalPartManuals.com

Layout



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available shortcut keys are indicated next to the menu commands.

Toolbars

The toolbars contain the buttons most often used.

With menu "View ▸ Toolbars" you can display or hide the available toolbars. The ▾ button of a toolbar is used to display or hide the individual buttons of this toolbar.

Work area

You configure the reports in the work area.

Toolbox

The toolbox gives you access to the objects required to configure a report. The objects are inserted into the report using the drag-and-drop function.

www.ElectricalPartManuals.com

Property view

When an object is selected, you can edit the properties of the selected object in the "Property view."

When no object is selected, you can edit the properties of the active area of a report in the "Property view."

10.3.2 Using the toolbox view

Introduction

The toolbox view contains a selection of objects which you can insert into your reports in the "Simple objects" and "Report objects" groups.

If the view of a report is activated in the work area, the toolbox only displays those objects which can be used in the report. Some of the toolbox objects are either available with restricted functionality or not at all. This depends on the HMI device you are configuring. Objects not available in the toolbox are grayed out and cannot be selected.

Changing default properties

Default properties are preset for the various object types in the toolbox view. When you insert an object from the toolbox view into a report, the object takes over these default properties.

You can customize the default properties of an object type to suit the requirements of your project. When you change the default properties of an object type, the properties of objects which have already been inserted are retained. You should therefore adapt the default properties before you insert the objects.

The default properties of the objects are coupled to the user names under which you are logged on in the operating system.

In order to change the default properties, you open the pop-up menu of an object in the toolbox view. Select the command "Edit default properties." The "Properties" dialog box is displayed. Adapt the default properties of the object to the requirements of your project.

Displaying the toolbox view

With menu "View ▸ Toolbox view" you can display or hide the toolbox view.

www . ElectricalPartManuals . com

10.4 Working with reports

10.4.1 Creating a report

Introduction

When creating a report, you specify the individual sections and contents. Configure the contents of the following sections:

- Report header
- Page header
- Page
- Page footer
- Report footer

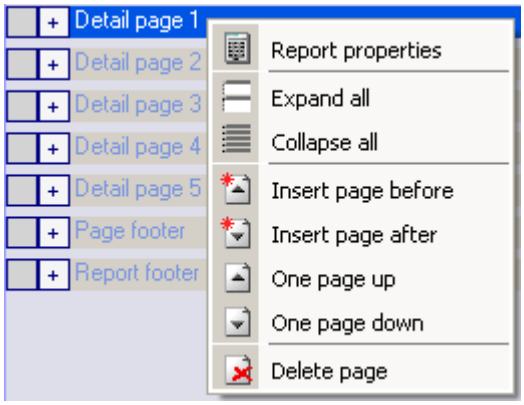
Configuration overview

Objects from the toolbox view are available for designing a report and configuring the data for the output. Some objects have a limited functional scope when used in a report compared to similar objects of the screen editor. An IO field can, for example, only serve as an output field.

When a report is created in the report editor, it is displayed as wysiwyg. The dynamic objects for outputting the data, for example "Print alarm" and "Print recipe", are exceptions. The configured height of these objects is irrelevant to the output format since the size of the dynamic objects depends of the existing amount of data. Page breaks are inserted consecutively on the pages depending on the amount of data. You can only insert one of these objects each into each page of a report. Objects which are positioned on the same side below one of these dynamic objects are not output. The "Print alarm" and "Print recipe" objects are inserted automatically with the width configured for the report. The width of the output follows the configured width of the report.

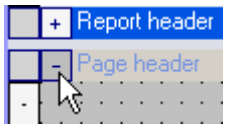
A new report always only contains one page. This page represents a page for the output. If required, you can insert further pages into the report. To do so, move the cursor onto the title bar of an existing page and open the pop-up menu with the right mouse button. The commands "Insert page before" and "Insert page after" are used to insert a new page before or after the existing page. The pages have a consecutive number assigned to them. A maximum of 10 pages is permitted per report. If you create more than 10 pages, the consecutive numbers of the superfluous pages are placed in pointed brackets (for example: Page <11>). The superfluous pages are not taken into consideration for the output. The "Delete page" command in the pop-up menu of a page is used to delete the selected page.

www . ElectricalPartManuals . com

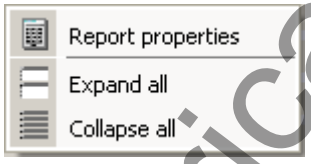


The sequence of the created pages can be changed subsequently. To do so, move the cursor onto the desired page and open the pop-up menu with right mouse button. Select the corresponding command "One page up" or "One page down" in the pop-up menu. The page is moved correspondingly. The consecutive numbering of the pages is retained. If, for example, Page 4 is moved "One page up" by means of the corresponding command, Pages 3 and 4 are swapped.

The individual report sections can be closed in order to obtain a better overview in the work area. In order to minimize or maximize, click on the node before the section designation.



You can also display or hide all the areas simultaneously. To do so, move the cursor onto the title bar of a report area and open the pop-up menu with the right mouse button. Select the corresponding command "Display all" or "Hide all" in the pop-up menu.



www . ElectricalPartManuals . com

10.4.2 Adapting the report properties

Introduction

You edit the output options and the format options for the report in the report properties. The following property groups are available:

- General
- Properties/Display

Requirements

- The report whose properties are to be changed must be open.
- The property view has to be open.

Editing the report properties

To do so, move the cursor on the title bar of a report area, for example on the title bar of the "Page header." Open the pop-up menu with the right button and select the command "Document properties." The report properties are displayed in the "Property view."

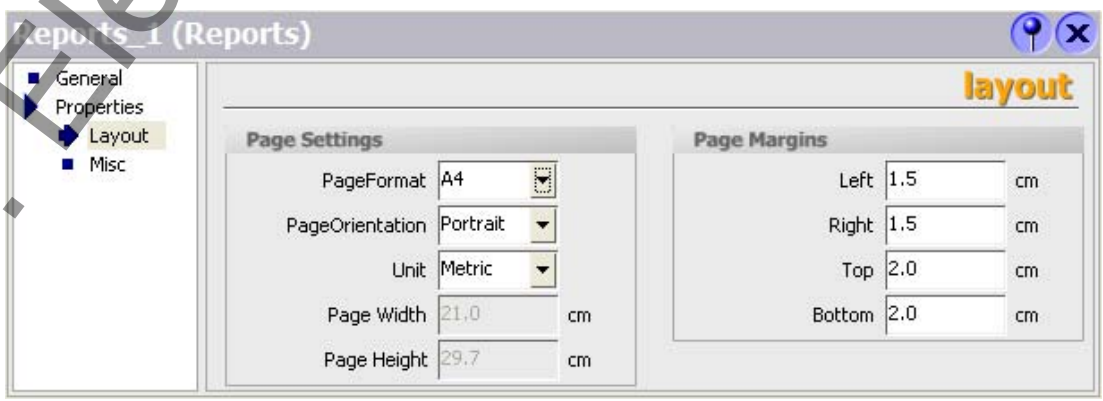
Activate the "General" properties group in the "Property view."

Activate or de-activate the output of the report header and report footer in the "Report" area.

Activate or de-activate the output of the page header and page footer in the "Page" area.

If the output of a report area is de-activated, this area is identified in the title bar by an "(X)."

Activate the "Properties/Display" properties group in the "Property view."



www.ElectricalPartManuals.com

Select the page format for the output in the "Page" field.

As an alternative, select the "User defined" format. You can then enter values for your own format in the "Width" and "Height" fields.

Select the portrait or landscape format in the "Page orientation" field.

Select the unit of measurement for setting the page size and page margins in the "Unit" field.

Set the size of the page margins by using the fields in the "Page margins" area. The page margins set may not be smaller than the page margins set at the printer.

10.4.3 Objects for report creation

Introduction

Objects can be either graphics elements for laying out your project report or dynamic elements for outputting data. The objects have limitations which depend on the configured HMI device. Please note the information in the object descriptions.

The objects are made available in the "Simple objects" object group in the toolbox view.

The special report objects are also available for creating reports. The special report objects are contained in the "Report objects" object group.

Simple objects

Symbol	Object	Description
	Line	The line is an open object. The line length and angle are defined by the height and width of the rectangle enclosing the object. The line ends can be represented as arrows or dots.
	Polyline	The polyline is an open object. The surface cannot be filled even if the starting and end points have the same coordinates. A polyline can have any number of corners. The corners are numbered in the sequence of their creation and can be changed or deleted individually. The line ends of a polyline can be indicated by arrows or dots, for example.
	Polygon	The polygon is a closed object which can be filled with a color or pattern. A polygon can have any number of corners. The corners are numbered in the sequence of their creation and can be changed or deleted individually.
	Ellipse	The ellipse is an enclosed object which can be filled with a color or pattern. You can customize the width and height of an ellipse in order to align it horizontally or vertically.
	Circle	The circle is an enclosed object which can be filled with a color or pattern. The circle diameter can be adjusted freely.
	Rectangle	The rectangle is an enclosed object which can be filled with a color or pattern. The height and width of a rectangle can be varied freely in order to allow a horizontal or vertical adjustment. The corners of a rounded rectangle can be rounded off as required.
	Text field	The field for static text is an enclosed object which can be filled with a color or pattern. The static test is entered in a text field of any size. You can enter single or multiple line text for all configured languages.

www . ElectricalPartManuals . com

Symbol	Object	Description
	"I/O box"	The IO field can only be used as an output field in a report. With an "IO field" you can output values with the following data formats: Binary, date, date-and-time, decimal, hexadecimal, string and time.
	"Date-time box"	The date and time are output in a report with the "Date-time" field. You can output the system time or connect a WinCC flexible tag through which the "Date-time" field is supplied with corresponding values.
	"Graphic I/O box"	The graphic IO field can only be used as an output field in a report. The field is used to select graphics from a graphics list. This allows you to display, for example, states of tags graphically. Example: Instead of the values 0 and 1, you can output one graphic each for a closed and an open valve.
	"Symbolic I/O box"	The drop-down list can only be used as an output field in a report. The field is used to select texts from a text list. This allows you to display, for example, states of tags in text form. Example: Instead of the values 0 and 1, you output "Motor OFF" and "Motor ON" for the state of a motor.
	"Graphic view"	The graphics object offers the possibility of inserting graphics which were created with other programs into a report. You can insert graphics or images with the following formats: "*.emf", "*.wmf", "*.dib" and "*.bmp." You define the size and the graphics object properties.

Note

Some of the toolbox objects are either available with restricted functionality or not at all. This depends on the HMI device you are configuring. Objects not available in the toolbox are grayed out and cannot be selected.

10.4.4 Use of report objects

Introduction

Special objects are available for reports under the "Report objects" section in the toolbar. These objects are intended exclusively for use in reports.

Overview of the objects

Symbol	Object	Short description
	Page number	Outputs the page number in a report. The object only has to be inserted once in a report, for example in the page footer.
	Print recipe	Outputs recipe data in a report.
	Print alarm	Outputs alarms in a report.

www.ElectricalPartManuals.com

10.5 Reporting alarms

10.5.1 Reporting alarms

Introduction

Configure a report in WinCC flexible with which you can output the alarms from the alarm buffer or an alarm log.

Output data of an alarm report

In order to report the alarms from the alarm buffer or an alarm log, insert the "Print alarm" object from the toolbox view into a report. Select the object in order to have the properties displayed in the property view. Configure the data selection for the report in the property view.

The following data can be output in the report:

- Current alarms from the alarm buffer
- Alarm from an alarm log

Specify the alarm classes which you want to output for the selected source. The following selections are possible:

- Error
- Operation
- Control

Specify the sequence of the alarms for the output.

The following selections are possible:

- Oldest message first
- Most recent message first

In order to output the alarms of a certain period, connect the "Display beginning" and "Display end" fields with tags. The tags can be supplied in runtime with the date and time for the first or the last alarm of the period.

www . ElectricalPartManuals . com

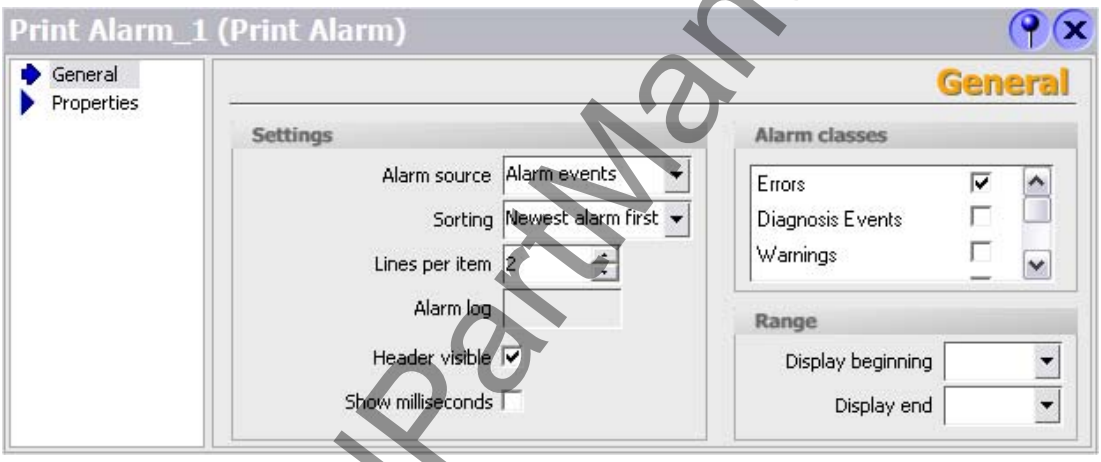
10.5.2 Editing output parameters for an alarm report

Introduction

You process the output parameters for an alarm protocol in the properties window The "Print alarm" object has to be inserted in a report in order to display the properties. The property view has to be open.

Output parameters of "Print alarm"

Select the "Print alarm" object in the work area. The properties of the object are displayed in the "Property view." Select the data source and configure the selection and the layout of the data for the output in the "General" category.



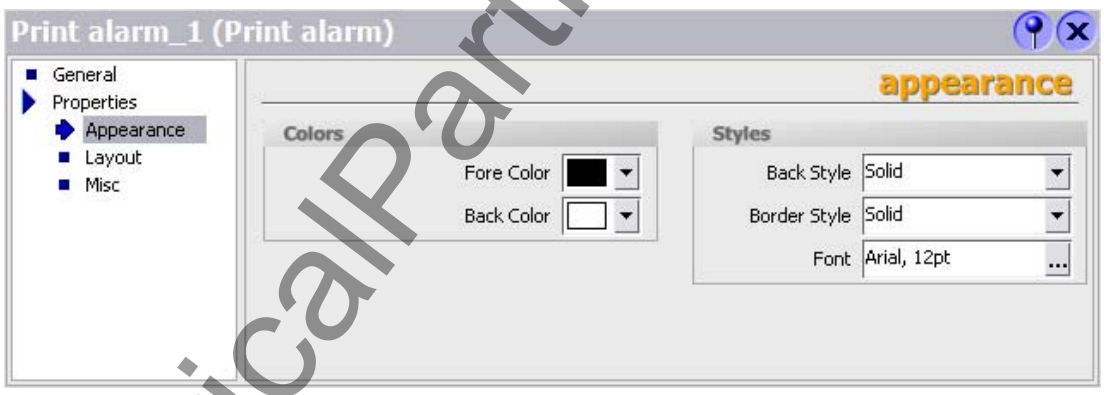
The following entries are available for outputting the alarms:

Attribute	Function	Requirements
"Source for alarms"	This is used to select the alarm source. You can select from the following options: • Alarm events • Alarm log	
"Sorting"	This is used to specify the sequence for the output. You can select from the following options: • Oldest message first • Most recent message first	
"Lines per entry"	This specifies the number of lines available per alarm. The required number of lines depends on the number and width of the selected columns in for the output as well as the font used and the paper format of the printer.	
"Page header visible"	Used to specify whether the table is to be output with column headers.	
"Alarm log"	This is used to select the alarm log for output.	An alarm log has to be selected as the alarm source in the "Source for alarms."

www.ElectricalPartManuals.com

Attribute	Function	Requirements
"Alarm classes"	Used to select the alarm classes for the output. You can select from the following options: • "Alarms" • "Alarm events" • "System alarms" • "S7 diagnostic events"	
"Display beginning"	Used to select the first alarm for outputting the alarms of a specific period. Connect the field via the selection list to a tag. Supply the tag in runtime with a start value, for example via an input field.	The tag must be of the type "Date and time."
"Display end"	Used to select the last alarm for outputting the alarms of a specific period. Connect the field via the selection list to a tag. Supply the tag in runtime with an end value, for example via an input field.	The tag must be of the type "Date and time."

Select the "Appearance" subcategory in the "Properties" category. Configure the foreground color, the background color, the style and the font settings.



Select the "Display" subcategory in the "Properties" category. Configure the position and size of the "Print alarm" object. Select the columns for the output in the report in the "Visible elements" area.

www . ElectricalPartManuals . com

The following columns can be output:

- "Alarm number"
- "Time"
- "Alarm status"
- "Alarm text"
- "Date"
- "Alarm class"
- "Acknowledgement group"
- "Diagnosable"
- "PLC"

Note

The height of the "Print alarm" object configured in the report is irrelevant to the output. Since a large amount of data can occur during the report output, the "Print alarm" object is extended dynamically so that all the data arising can be output. If the page length is exceeded, an automatic page break is carried out.

www . ElectricalPartManuals . com

10.6 Reporting recipes

10.6.1 Reporting recipes

Introduction

Configure a report of recipe records in WinCC flexible.

Output data of a recipe

In order to create a recipe report, insert the "Print recipe" object from the toolbox view into a report. Select the object in order to have the properties displayed in the property view. Configure the data selection for the report in the property view.

Specify the data selection in the "General" category in the property view. Select the recipe records for your report.

The following selections are possible:

- All the records of a recipe
- A record range of a recipe
- All the records of several or all of the recipes
- A record range of several or all of the recipes

When several recipes are selected, you can select only one range of consecutive recipes. The system uses the numbers of the recipes as an orientation. The same behavior also applies to a data record range.

Formatting for the output

In the property view, you specify in the "Properties/Display" category whether the data are to be output line-by-line or in table form. In the same category, select the record elements in the "Visible elements" area for output.

The following record elements can be output:

- "Recipe number"
- "Recipe name"
- "Data record number"
- "Data record name"
- "Tag name"
- "Tag type"
- "Element"

The report is output time-controlled or event-driven.

www.ElectricalPartManuals.com

10.6.2 Editing output parameters for a recipe report

Introduction

The output parameters for a recipe report are edited in the property view. The "Print recipe" object has to be inserted in a report in order to display the properties. The property view has to be open.

Output parameters of "Print recipe"

Select the "Print recipe" object in the work area. The properties of the object are displayed in the "Property view." Select the recipe data for the output in the report in the "General" category.

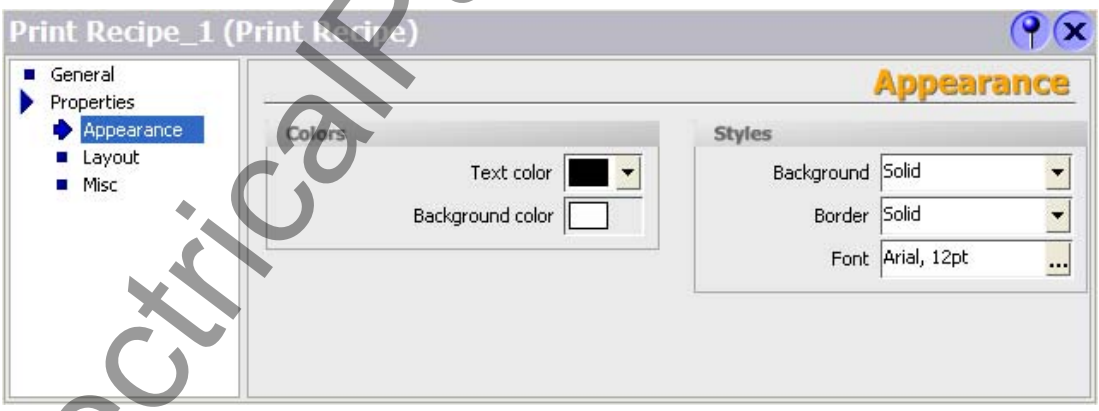
The following entries are available for the recipe selection:

Attribute	Function	Requirement
"Recipe selection"	Used to select the selection criterion for outputting the recipes. You can select from the following options: "All" "Name" "Number"	
"Recipe name"	This is used to select a recipe by its name. If you have already configured recipes, you can either enter the name of the recipe or select one from the object list.	The "Name" option has to be selected in the "Recipe selection" selection field.
"First recipe"	Used to select the first recipe number for outputting the records of several recipes. Enter a fixed starting number in the field or connect the field to a tag via the selection list. The tag can be given a start value dynamically during runtime.	The "Number" option has to be selected in the "Recipe selection" selection field.
"Last recipe"	Used to select the last recipe number for outputting the records of several recipes. Enter a fixed end number in the field or connect the field to a tag via the selection list. The tag can be given an end value dynamically during runtime.	The "Number" option has to be selected in the "Recipe selection" selection field.

www.ElectricalPartManuals.com

Attribute	Function	Requirement
"Data record selection"	Used to select the selection criterion for outputting the records of recipes. You can select from the following options: "All" "Name" "Number"	
"Data record name"	Used to select a record by its name.	The "Name" option has to be selected in the "Data record selection" selection field.
"First data record"	Used to select the first record number for outputting a record range of a recipe. Enter a fixed starting number in the field or connect the field to a tag via the selection list. The tag can be given a start value dynamically during runtime.	The "Number" option has to be selected in the "Data record selection" selection field.
"Last record"	Used to select the last record number for outputting a record range of a recipe. Enter a fixed end number in the field or connect the field to a tag via the selection list. The tag can be given an end value dynamically during runtime.	The "Number" option has to be selected in the "Data record selection" selection field.

Select the "Appearance" subcategory in the "Properties" category. Configure the foreground color, the background color, the style and the font settings.

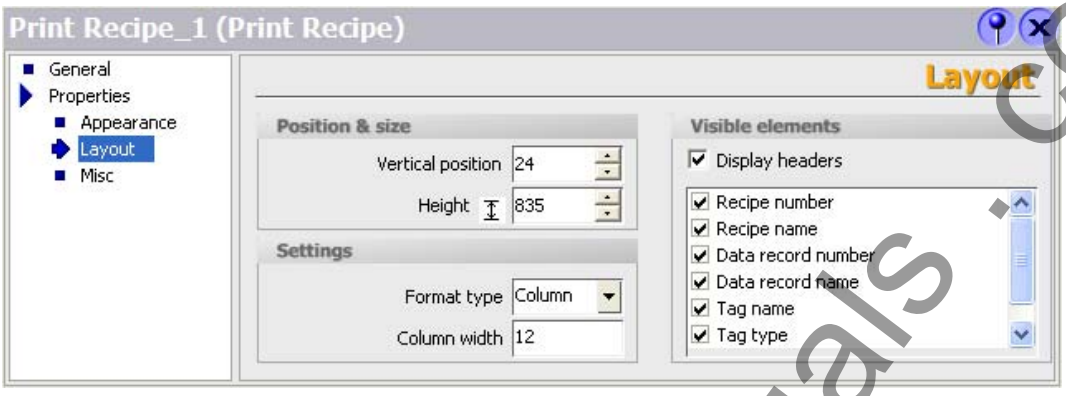


Select the "Display" subcategory in the "Properties" category. Configure the position and the size of the "Print recipe" object in the "Position and size" area. You can also use the mouse to change the size and position of the "Print recipe" object in the work area.

Select the output form as table or line-by-line output in the "Settings" area. In case of table form output, specify the number of characters for the width of the columns in the "Column width" field. The set width affects all columns of the table.

Select the record elements for the output in the report in the "Visible elements" area.

www . ElectricalPartManuals . com



Note

The height of the "Print recipe" object configured in the report is irrelevant to the output. Since a large amount of data can occur during the report output, the "Print recipe" object is extended dynamically so that all the data arising can be output. If the page length is exceeded, an automatic page break is carried out.

10.7 Outputting a report

Introduction

WinCC flexible offers the following options for outputting a report:

Time-controlled output, for example:

- Non-recurring, time-controlled output
- Output repeated at intervals

Event-controlled output, for example:

- Through a change in the tag value
- Through activating a configured button in a WinCC flexible screen
- Overflow of a log
- Through a WinCC flexible script

Configuration of the output

Time-controlled output is configured through the scheduler. The report output can furthermore be controlled by system events which are made available by the scheduler.

Event-controlled output of an object is configured directly at a tag, a button in the WinCC flexible screen, or at a log.

Note

Te output goes to the default printer with Windows-based HMI devices.

The printer is set in the Control Panel on the HMI device for Windows-CE-based HMI devices. A network printer must be accessable via the printer name. In other words, the printer must be connected to the network via a DNS server. Addressing of a network printer via the IP address is not possible at Windows-CE-based HMI devices.

www.ElectricalPartManuals.com

11 User administration

11.1 Field of application of the user administration

Principle

The access security system controls user access to data and functions in runtime to prevent unauthorized operations. Safety-relevant operations are already limited to specified user groups when a project is being created. For this purpose, users and user groups are set up and have characteristic access rights, the authorizations, assigned to them. Required operation authorizations are configured for objects. For example, operators only have access to specific function keys. Commissioning engineers, on the other hand, have unlimited access during runtime.

Definition

Users, user groups and authorizations are administered centrally in the user administration.

The user administration controls access to data and functions during runtime. For this purpose, users and user groups are created, administered and transferred to the HMI device in the engineering system. In Runtime you manage the users and passwords by using the "User view".

Example of an application

You create and configure an access protection in order to protect operating elements, such as input fields and function keys, against unauthorized operation. Only specified persons or operator groups can change parameters and settings and call functions.

Caution

Access protection does not protect against incorrect operations. You are responsible for ensuring that only correspondingly trained and authorized personnel design, commission, operate and maintain, etc. plants and machines.

Access protection is not suitable for defining work routines and monitoring their observance.

---

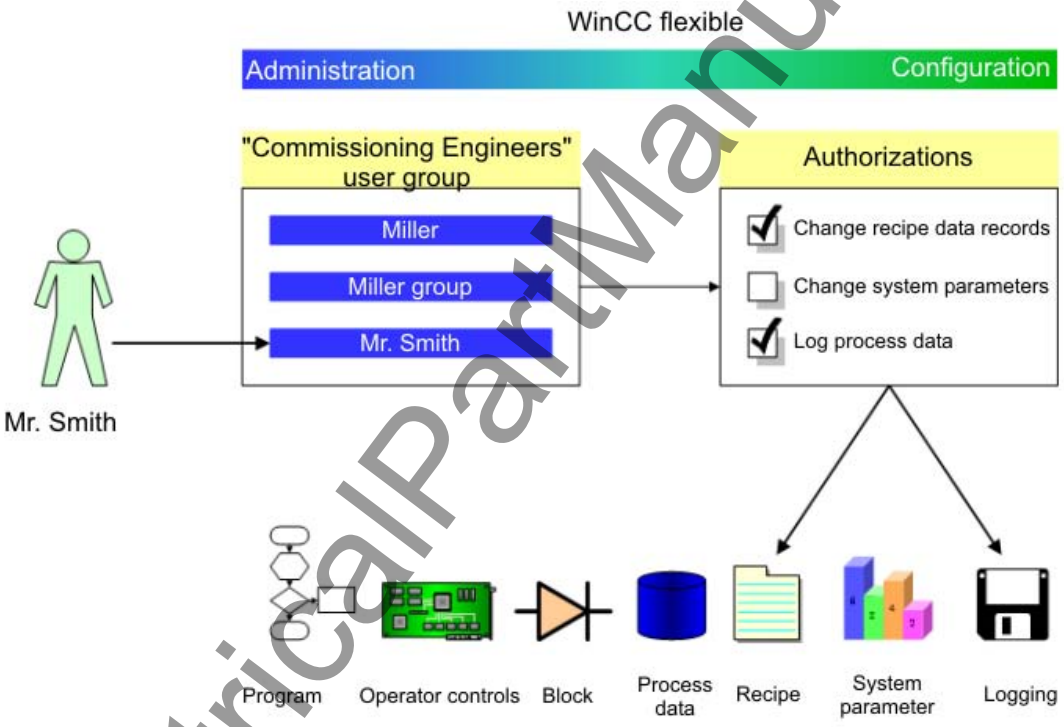
www . ElectricalPartManuals .com

11.2 Structure of the user administration

Introduction

In case of a project in manufacturing engineering, the environment at the equipment manufacturer has to be differentiated from the environment at the end customer as plant operator. The equipment manufacturer allows the user, for example Mr. Foreman, a specific access to the project. However, a user Foreman does not exist at the end customer.

Similar difficulties arise, for example, when different projects on a plant are to be integrated into one project in process engineering. In order to integrate the projects you have to be able to access the data of each individual project without restriction during commissioning.



Authorizations are therefore not assigned directly to users in the user administration, but rather to user groups. The user Foreman is then, for example, assigned to the "Operator" user group and receives its authorizations. Authorizations do not have to be assigned individually to each user, only to the user group.

In a different environment, for example at the customer, there are other users. The authorizations and user groups of the project, however, remain unchanged. Only the users are re-assigned to the user groups, for example "Operator."

The user administration separates the administration of the users from the configuration of the authorizations. This ensures flexibility at the access protection.

www.ElectricalPartManuals.com

11.3 Elements and basic settings

11.3.1 "Groups" user administration

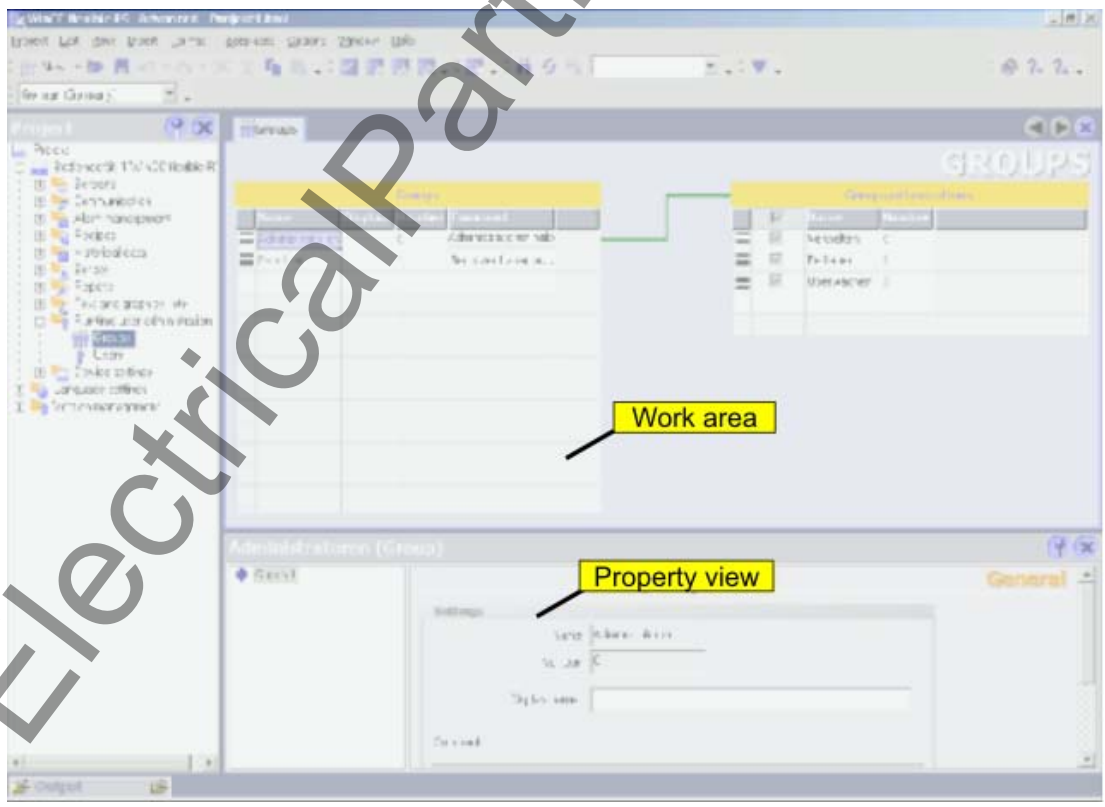
Introduction

In user administration you administer users and user groups in order to control access to data and functions in runtime. The user administration is divided into the administration of the users and the administration of the user groups. This section described the administration of the user groups.

Open

You open the administration of the user groups in the project window by double-clicking on "Groups."

Structure



Work area

The "Groups" work area shows the existing user groups and their authorizations.

www . ElectricalPartManuals . com

Property view

When a user group or an authorization is selected, you can edit the designation and the comment in the "General" group.

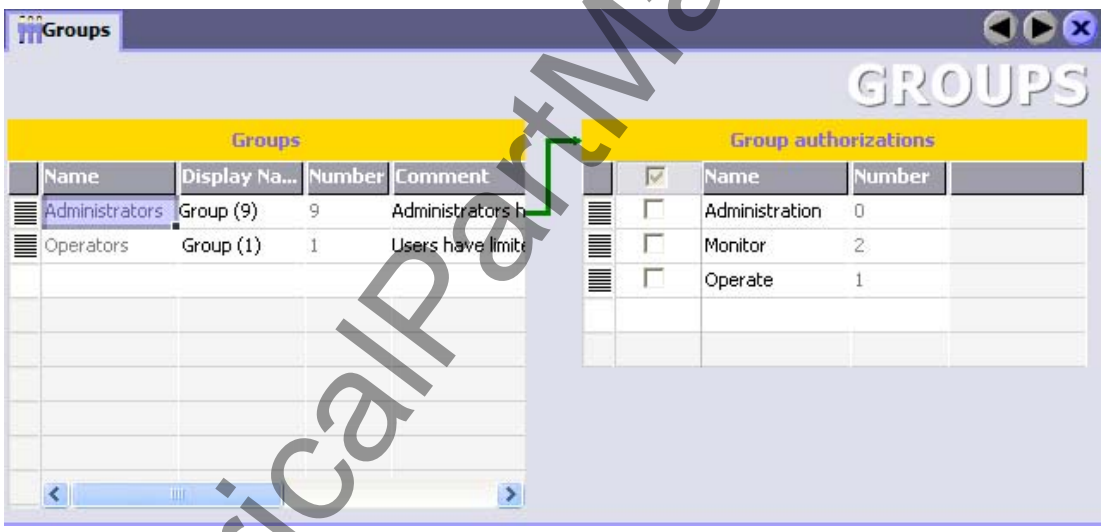
11.3.2 User groups work area

Introduction

The "Groups" work area shows a table of the user groups and their authorizations. You administer the user groups and assign authorizations to them.

Principle

The work area consists of the "Groups" and "Group Authorizations" tables.



The "Groups" table shows the existing user groups. When you select a user group in this table, the "Group Authorizations" table shows the authorizations which were assigned to the user group.

The number of the user group and of the authorization is assigned by the user administration. The designations and descriptions are assigned by you.

www . ElectricalPartManuals . com

11.3.3 "Users" user administration

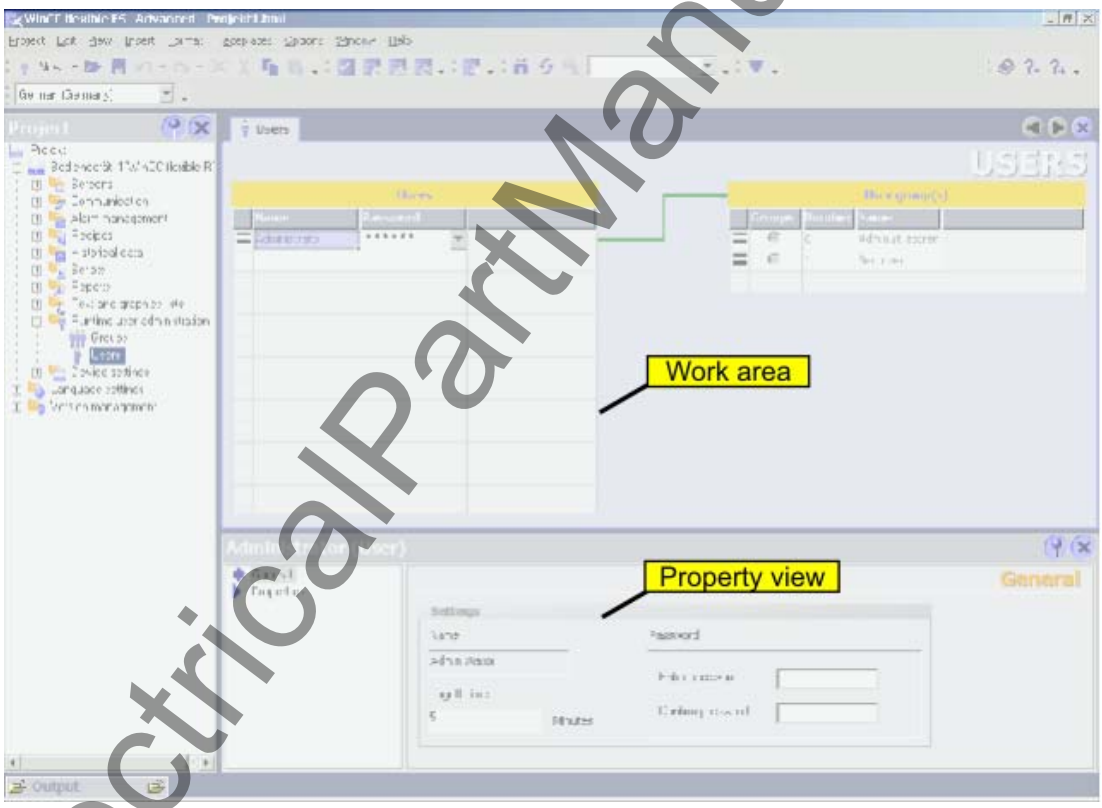
Introduction

In user administration you administer users and user groups in order to control access to data and functions in runtime. The user administration is divided into the administration of the users and the administration of the user groups. This section describes the administration of the users.

Open

You open the administration of the users in the project window by double-clicking on "Users."

Structure



Work area

The "Users" work area shows the existing users and the user groups to which they are assigned.

Note

- A user can only be assigned to one user group.

www.ElectricalPartManuals.com

Property view

When a user has been selected, edit the password and the time after which the user is logged off automatically in the "General" group.

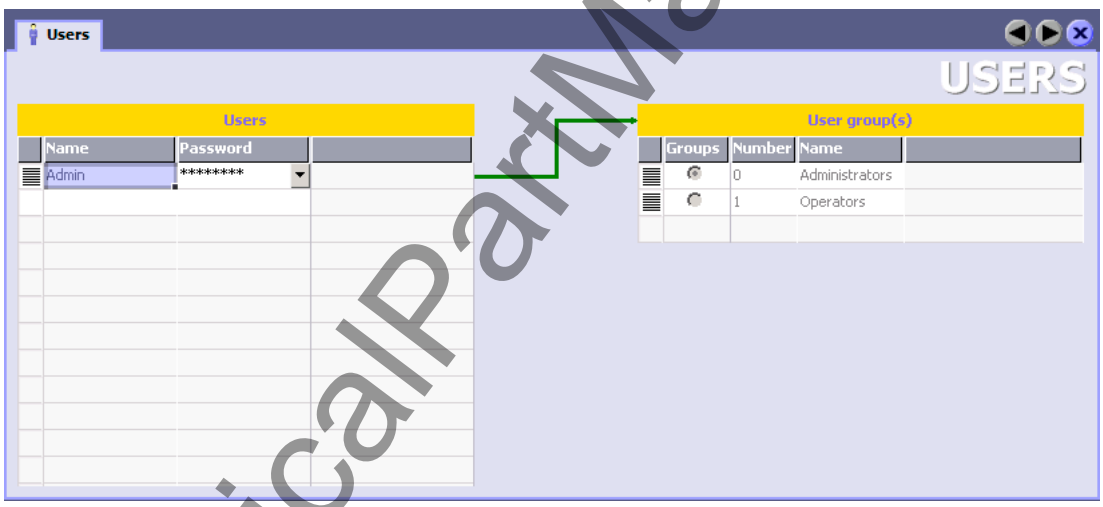
11.3.4 Users work area

Introduction

The "Users" work area lists the users and user groups in table form. You administrate the users and assign them to a user group.

Principle

The work area consists of the "Users" and "Groups of the user" tables.



The "Users" table shows the existing users. When you select a user in this table, the "Groups of the user" table displays the user group to which the user is assigned.

www . ElectricalPartManuals . com

11.4 Working with the user administration

11.4.1 Users in Runtime

Principle

In the engineering system you create users and user groups and allocate authorizations to them. You configure objects with authorizations. After transfer to the HMI device, all objects which were configured with an authorization are protected against unauthorized access in runtime.

User view

When you configure a user view in the engineering system, you can administer users in the user view following transfer to the HMI device.

Caution

Changes in the user view are effective immediately in runtime. Changes in runtime are not updated in the engineering system. When the users and user groups are transferred from the engineering system to the HMI device, all the changes in the user view are overwritten after a user prompt and based on the transfer settings.

Some HMI devices do not support the user view. These HMI devices only support the functions "Log on" and "Log off". The only user "Administrator" is logged on and logged off. The "Administrator" is assigned to the only user group "Administrators."

Exporting and importing user data

The users and passwords existing at an HMI device are exported and imported to a different operator panel by means of a system function. This ensures that the user administrations of the different HMI devices have the same status.

Note

Once you have exported the user data with WinCC flexible 2004, you can then import this file into WinCC flexible 2005. WinCC flexible 2005 is downward compatible to WinCC flexible 2004.

Once you have exported the user data with WinCC flexible 2005, you cannot then import this file into WinCC flexible 2004.

www . ElectricalPartManuals . com

11.4.2 Access security

Introduction

You configure an authorization at an object in order to protect it against access. All logged-on users who have this authorization can access the object. When a user does not have authorization to operate an object, the logon dialog is displayed automatically.

Note

Several system functions are available under "User administration" so that user, password and user group can be edited, for example, in the control system.

www . ElectricalPartManuals . com

12 System functions and runtime scripting

12.1 Basics

12.1.1 System functions and runtime scripting

Introduction

WinCC flexible provides predefined system functions for common configuration task. You can use them to perform many tasks in Runtime and need no programming skills to do so.

You can use Runtime scripting to solve more complex problems. Runtime scripting has a programming interface which can be used to access part of project data in runtime. The use of runtime scripting is aimed at project planners with knowledge of Visual Basic (VB) and Visual Basic Script (VBS).

Use of system functions

System functions provide support if you wish to assign a function to an operator control element:

- Set a bit in the PLC
- Change the value of a tag
- Start logging

System functions can be configured in function lists and scripts.

Use of runtime scripting

Runtime Scripting is available from OP 270/TP 270 and therefore also from WinCC flexible Standard. VBScript is supported as a programming language. The use of runtime scripting allows flexibility in the realization of configurations. Create scripts with runtime when extra functionality in runtime is needed, e.g.:

- Conversion of values

  You can use scripts to convert values between different measurement units, e.g. temperatures.

- Automation of production sequences

  A script can control a production sequence by transferring production data to a PLC. Using the return values, you can check the status and initiate the appropriate measures, if necessary.

www.ElectricalPartManuals.com

Scripts

You can save your own VB script code in a script. You can use the script just like a system function in the project. You have access to the tags of the project and the runtime object model of WinCC flexible in the script. In addition, you can use all standard VBS functions in the script. You can call other scripts and system functions in the script.

Execution of system functions and scripts

System functions and scripts are executed in runtime after the onset of a configured event (e.g. a mouse click on a button).

Recursion level

The recursion level in scripts is limited by the stack size of the HMI device. In Runtime, an unrestricted number of recursions leads to a system error message. Therefore, please limit the number of recursions in a script.

12.1.2 System functions

Introduction

System functions are pre-defined functions you can use to implement many tasks in runtime even without having any programming knowledge, e.g.:

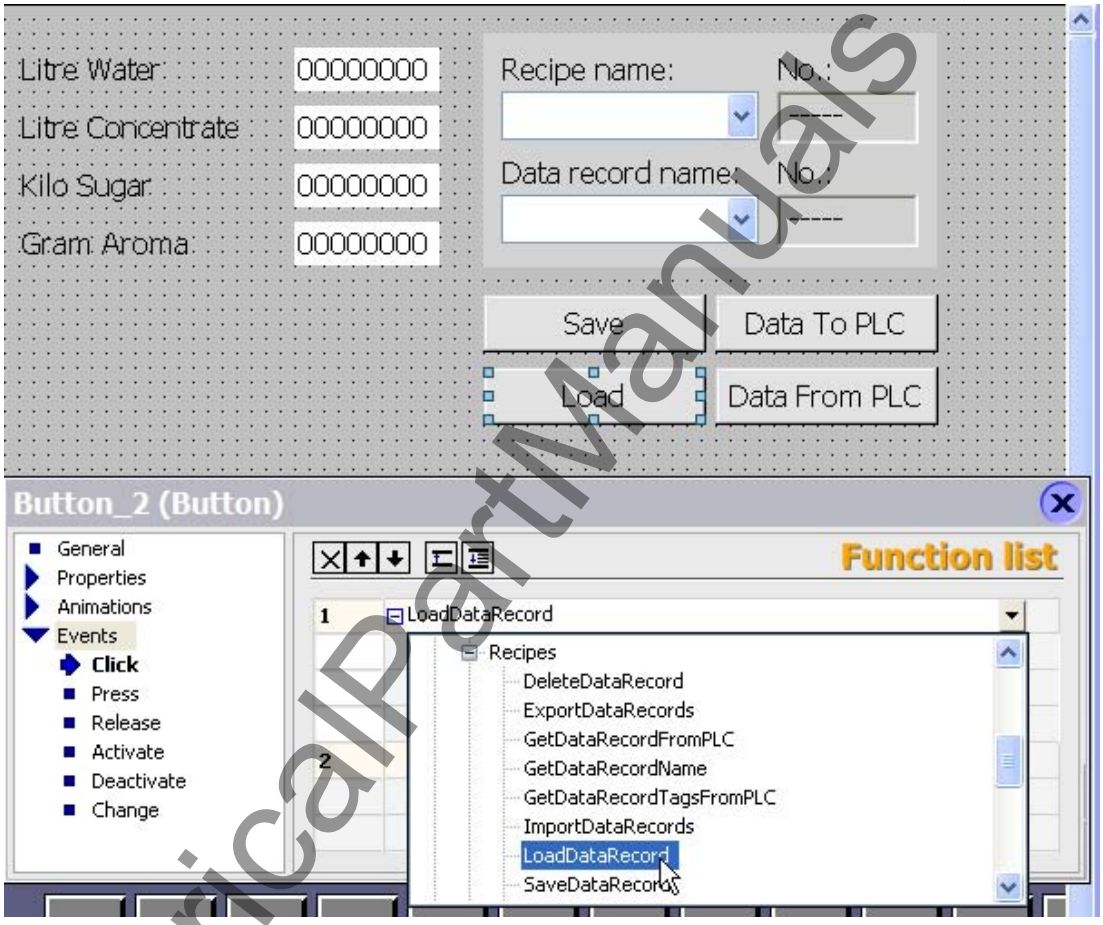
- Calculations, for instance the increasing of a tag value to a specific or variable amount.
- Log functions, for instance starting a process value log.
- Settings, for instance changing the PLC or setting a bit in the PLC.
- Messages, for instance after change of user.

www . ElectricalPartManuals . com

Application

You can use system functions in a function list or in a script. You cannot change system functions, since system functions are pre-defined functions.

When configuring a function list, select the system functions from a selection list that is sorted by categories:



When you want to use a system function in the script, you can select it from a selection list. You can call up the selection list in the script with <Ctrl+Space>.

Language dependency

The names of the system functions are dependent on the set project language. The functionality can then be recognized immediately by the project planner.

Exception: When calling up system functions in a script, please use the English name for the system function. You can find the English name of the system function in the system function reference.

Availability

In WinCC flexible you can only configure functionalities which are supported by the selected HMI device. Therefore, in a function list you can configure only system functions which are supported by the selected operating unit. If you use a project for several operating units, the system functions which are not supported by a operating unit are marked in color.

www . ElectricalPartManuals . com

12.1.3 Use of system functions

Introduction

In runtime a function list will be carried out when the configured event has taken place. The operator can trigger an event, for instance by pressing a function key on the operating unit. An event can also be triggered by the system, for instance if a process value falls below a limit value.

Applications

You can configure system functions on all the objects that are able to react to an event. You can use system functions directly in function lists and scripts and thereby control the course.

- Function List

  System functions are processed sequentially in a function list, that is, from the first to the last system function. In order to avoid waiting times, system functions with a longer running time (for instance file operations) are processed simultaneously. For instance, a subsequent system function can already be performed even though the previous system function has not yet been completed.

  An example for the configuring of a function list can be found under "Example: Changing the operating mode on the HMI device with the current display.

- Script

  In a script you are able to use system functions in connection with orders and requirements in the code. This way, you can execute a script depending on a specific system state. In addition, return values of system functions can be evaluated, for example. Depending on the return value, test functions can be carried out, for example, which in turn affect the course of the script.

12.1.4 Scripts

Introduction

You program VB script codes in a script. You can use finished scripts in the project just like a system function. When creating a script, you determine its type and define transfer parameters. Scripts of the type "Function" have a return value. "Sub" type scripts are referred to as "procedures" and have no return value.

Properties of scripts

You can call up other scripts and system functions in a script. You can access the runtime objects of WinCC flexible by using the runtime object model. When calling up a system function, please use the English name of the system function. You can use the full scale of language from Microsoft VBScript in scripts. Functions and methods for user interaction are excluded, for instance "MsgBox."

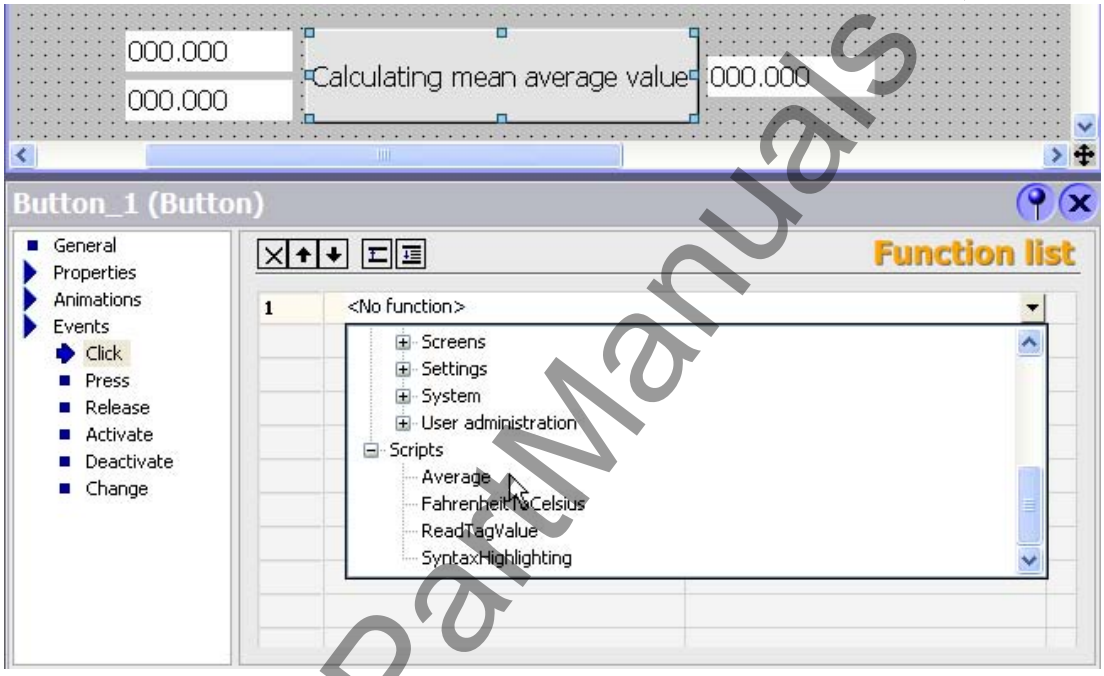
If you use system functions in a script which are not available on the set operating unit, you will receive a warning message. In addition, the respective system function in the script will be underlined with a wavy blue line.

www.ElectricalPartManuals.com

Organization of scripts

Scripts are stored in the project database. The available scripts are listed in the project view under scripts.

If you want to use a script in a function list, you can find the scripts in the selection list under scripts.



12.1.5 Use of scripts

Principle

Scripts provide more flexibility by using control elements of a programming language.

Using scripts in runtime you can implement individual solutions in a project, for instance:

- Configuring an advanced functions list

  You can use a script just like a function list by calling up system functions and other scripts in the script.

  You can execute system functions and scripts in the script dependent on conditions, or have them repeated. You then add the script to a functions list.

- Programming new functions

  Scripts are available in the entire project. You can use scripts just as you would system functions. You can define delivery parameters and return values for these scripts. You can use scripts, e.g. to convert values.

www.ElectricalPartManuals.com

12.2 Working with function lists

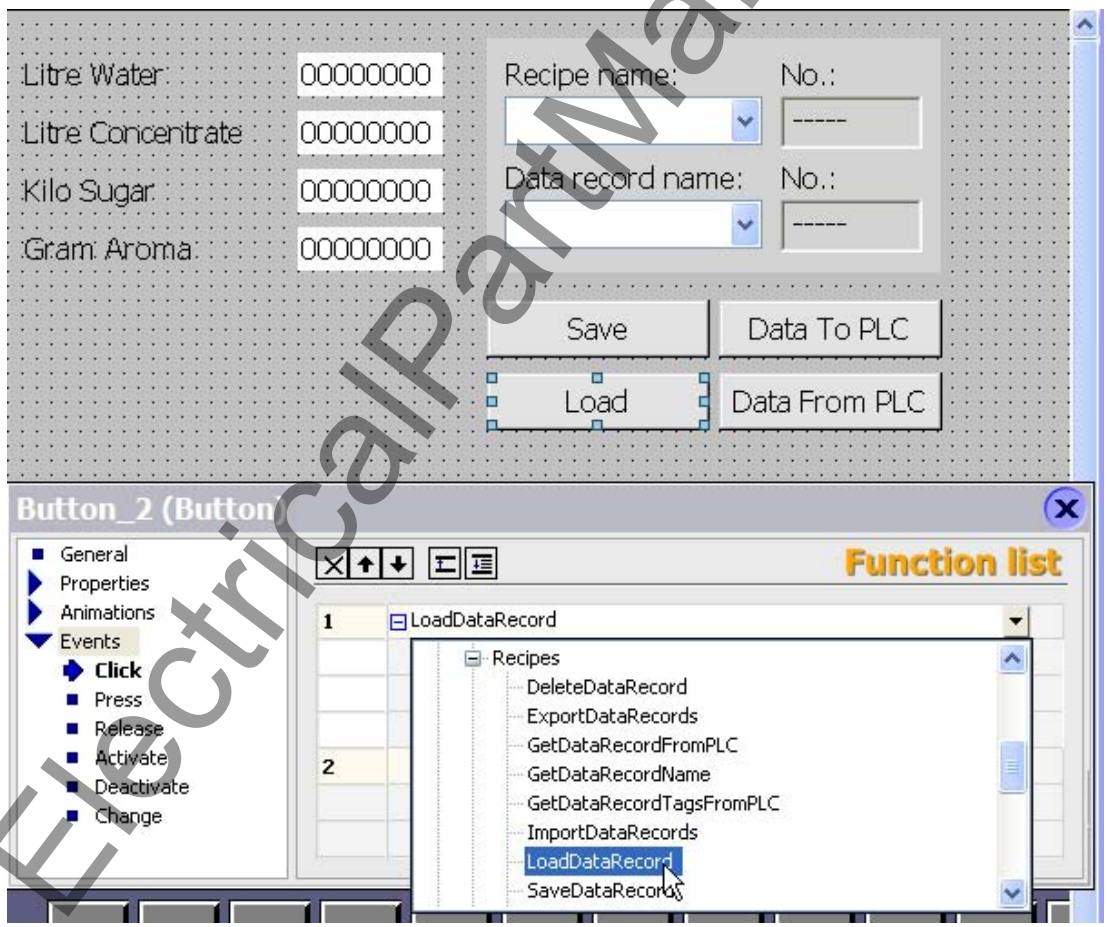
12.2.1 Basic principles of the functions list

Introduction

When the configured event occurs, several system functions and scripts can be performed with the function list.

Principle

The function list is configured for an event of an object, e.g. a screen object or a tag. The events which are available depend on the selected object and the HMI device.



Events occur only when the project is in runtime. Events are, for example:

- Value changes of a tag
- Pressing of a button
- Activation of runtime

You can configure a function list precisely on every event.

www.ElectricalPartManuals.com

Note

The choice of configurable system functions in a function list is dependent on the HMI device chosen.

12.2.2 Properties of a function list

HMI device dependency

You can use a project for different HMI devices. When you change the HMI device in a project, all system functions and scripts which are not supported by the selected HMI device are marked in yellow. The system functions which are not supported will also not be performed in runtime.

Status information

During configuration the project data is tested in the background. A status information returns in each function list the status of the respective system functions and scripts.

The status information has the following meaning:

- Orange: Function list is not performed in runtime because at least one system function or a script has not been supplied completely with parameters.
- Yellow: Function list is performed in runtime. However, the function list contains at least one system function or script which is not supported by the HMI device (e.g. due to the change of device type).

Completion of system functions and scripts

System functions and scripts in a function list are processed in runtime sequentially from top to bottom. In order to avoid waiting times, system functions with a longer running time (for instance file operations) are processed simultaneously. For instance, a subsequent system function can already be performed even though the previous system function has not yet been completed.

To avoid programming sequential and conditional procedures, use a script with loops, conditional statements and cancellation requirements.

www . ElectricalPartManuals . com

12.3 Elements and basic settings

12.3.1 Scripts

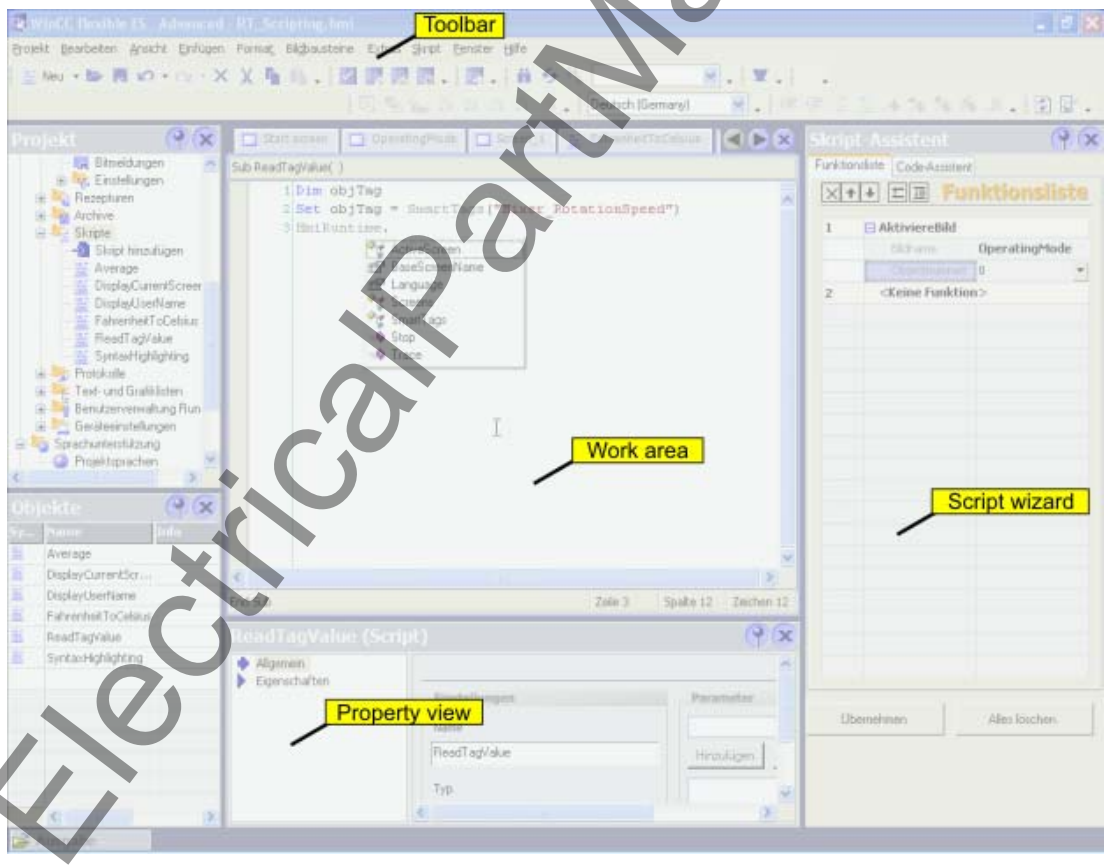
Introduction

Create and edit scripts in the script editor.

Open

The script editor opens automatically a new script is created or an existing script is opened.

Layout



Menu bar

The menu bar contains all commands required for operating WinCC flexible. Any available shortcut keys are indicated next to the menu commands.

www . ElectricalPartManuals . com

"Scripts" toolbar

The commands for synchronizing objects an tags as well as for checking script syntax are located in the "Script" toolbar.

"Advanced Edit" toolbar

The commands for working with bookmarks, for moving code in and out, for commenting code and for jumping to a certain line of code are found in the "Advanced Edit" toolbar.

"IntelliSense" toolbar

The commands for displaying selection lists, e.g. all objects of the object model, available system functions or VBS constants, are found in the "IntelliSense" toolbar.

Work area

Create and edit scripts in the work area. The creation of scripts is supported by syntax emphasis and IntelliSense.

Property view

Configure the script in the property view. You determine whether the script is a procedure or a function. Furthermore you can declare parameters for the script.

"Script Wizard"

In the "Script Wizard" system functions and scripts can be set up with assigned parameters just as in a function list. The filed system functions and scripts can also be transferred to the active script from the "Script Wizard." In this way, you only need to perform the parameter assignment once.

If system functions or scripts have already been configured on an event, these may be transferred to the "Script Wizard" using copy and paste. Only system functions which are allowed in a script may be filed in the "Script Wizard." When you transfer system functions which cannot be used in a script using copy and paste, these system functions will be marked.

www . ElectricalPartManuals . com

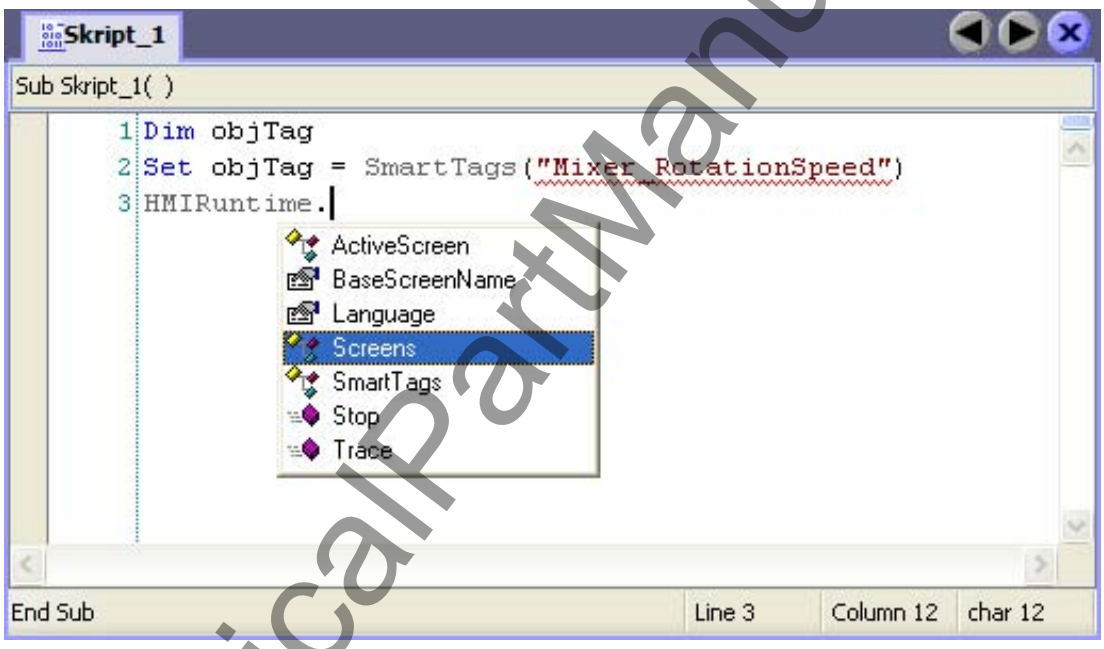
12.3.2 Properties of the "Script" editor

Introduction

The script editor supports you during programming with functionalities such as IntelliSense, emphasized syntax, and others. For example, references to tags can be created with the drag-and-drop function.

IntelliSense

When you access objects, methods or properties of the VBS object models, you are supported by IntelliSense:

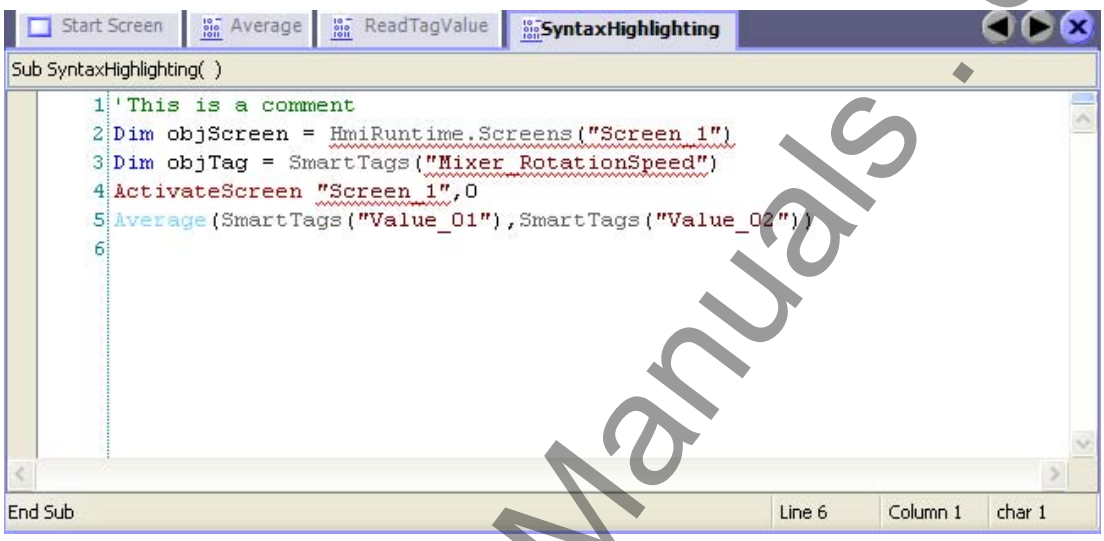


The methods and properties which the given object possesses can be selected from the selection list.

www . ElectricalPartManuals . com

Syntax emphasis

In the script editor, keywords are emphasized by different colors. Objects which the script editor recognizes are displayed in bold. Unknown words are underlined with a red wavy line:



The table shows the pre-set colors for the most important keywords.

Color	Meaning	Example
Blue	Keyword (VBS)	Dim, If, Then
Gray	Keyword (object model)	HMI runtime
Cyan	Script	Fahrenheit to Celsius
Brown	System function	IncreaseValue
Red	Tag	Tag_1
Green	Comment	'This is a comment

Synchronization of objects

When opening the script, instances of objects (e.g. tags) are automatically synchronized with the configuration data. If a tag has been renamed in the "Tag" editor, e.g., this change also affects the script. When change is made and the affected script is open, the renamed object will be underlined with a blue wavy line. The synchronization can then be performed manually in the script editor.

Object list

Using the key sequence <Alt+Right>, the object list can be called up in which all available objects are displayed depending on the context. The object lists can be called up during the assignment of parameters, e.g., or when using listings.

Example: You want to reference an existing process screen by means of the screen list. Enter HmiRuntime.Screens in the script editor and then call up the object list with <Alt+Right>. All of the process screens available in the project are listed there:

Select the desired process screen and take on the selection with <Return>.

www.ElectricalPartManuals.com

Drag-and-drop

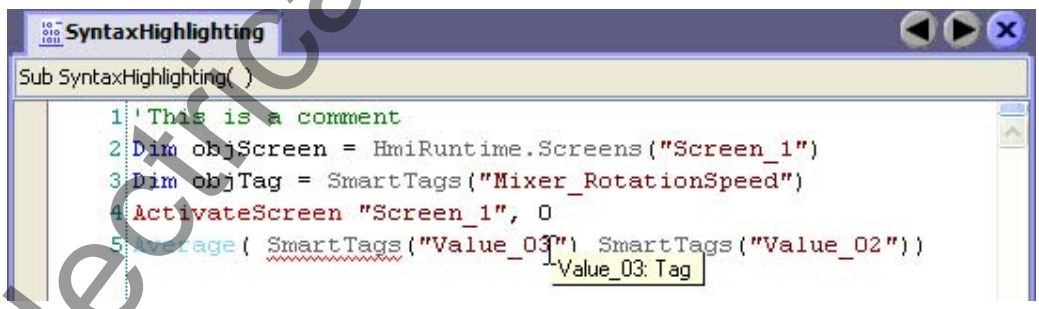
If a tag is required in the script, it can be pulled from the object window.

Help functions

During programming you will be shown automatically short descriptions of the necessary parameters for the methods and system functions. In addition, the following help functions are available in the script editor:

- Tooltip

  Unknown or incorrectly written keywords will be underlined with a wavy line. When you move the mouse over a keyword, Tooltip appears:



  For known keywords, Tooltip shows the type of keyword.

- ParameterInfo

  The ParameterInfo offers information concerning the syntax and the parameters of a system function or a VBS standard function.

- Context sensitive help

  The context sensitive help offers information concerning system functions, VBScript language elements, objects, etc.

  If information about an object, a method or a property is needed, move the mouse pointer over the corresponding keyword and press <F1>. This allows you to reach the corresponding reference description in the online help.

www.ElectricalPartManuals.com

12.4 Creating scripts

12.4.1 Access to tags

Introduction

In the script you have access to external and internal tags which you set up in the project. The value of a tag can be read or changed in runtime. Furthermore, you can set up local tags as a counter or as buffer storage in the script.

Project tags

If the tag name in the project corresponds to the VBS name conventions, the tag can be used directly in the script:

```
'VBS_Example_03
If BeltDriveOilTemperature > 100 Then [instruction]
```

If the tag name in the project does not correspond to the VBS name conventions, then the tag must be referenced by means of the "Smart tags" list. In the following example, the tag name contains the & sign, which is not allowed according to VBS name conventions:

```
'VBS_Example_04
Dim objTag
SetobjTag = SmartTags("Test&Trial")
```

The VBS name conventions are found in the help for VBS in the information system.

Local tags

Local tags can be defined in the script using the Dim statement. Local tags can be used only within the script. Therefore, they do not appear in the "Tags" editor.

For example, in the script a local tag is used as counter in a For statement.

```
'VBS_Example_05Dim intCountFor intCount = 1 To 10[Instruction]Next
```

Note

You have to use a local tag if you need a tag for a "For statement." Project tags are not allowed within a "For statement."

www . ElectricalPartManuals . com

12.4.2 Call up of scripts and system functions in the scripts

Principle

System functions and other scripts can be called up in a script.

Call up a system function or a script without return value ("Sub") as follows:

```
<Function name> [Parameter1], [Parameter2], [...]
```

A system function or a script with return value ("Function") is called up by means of assignment to an expression:

```
<Expression> = <Function name> ([Parameter1, Parameter2,...
[Parameter N])
```

If you do not want to evaluate the return value, use the call up as you would for a system function or a script without return value.

Particularities when calling up system functions

You can insert system functions and scripts into the script from the "Script Wizard." The system functions are displayed in the currently configured language in the "Script Wizard."

When calling up a system function in the script, always use the English name of the system function:

```
SetValue Tag1, 64
```

You can find the English name of the system function in the system function reference under "Syntax." The set project language is not taken into consideration.

The following rules apply to the parameter delivery for system functions:

- Constants.

  If you use a constant as a parameter, then the parameter type must correspond to one of the three data types: Integer, double or string. The available constants are shown in a selection list when assigning parameters. The usual VBS conventions apply to constants.

- Tags

  Independent of the spelling, tags are always delivered as "Call by reference." When the tag to be delivered corresponds to the VBS name conventions, the tag name can be delivered without the keyword smart tags:

  ```
  SetValue Tag1, 64
  ```

  or

  ```
  SetValue SmartTags ("Tag1"), 64
  ```

- References to objects, e.g. process screens, connections and logs

  An object reference is delivered as parameter in quotation marks:

  ActivateScreen "MainScreen", 0

Particularities when calling up scripts

When calling up a script, parameters are delivered as "Call by Reference." When you pass a tag as a parameter, for example, the value assignments in the script have an immediate effect on the value of the tag.

www.ElectricalPartManuals.com

Operator device dependency in the script

The code of the script is dependent on the selected operator device. If system functions are used in the script which are not supported by the selected operator device, an error message is received in the output window.

12.4.3 Access to objects

Introduction

The objects of the runtime object model with the accompanying properties and methods are available to you in the script.

The object properties can be read and changed in runtime.

Referencing objects

In the script, reference objects by means of the accompanying list. To identify the object, use its name or the position number within the list.

The first object in the screen "Main screen" is referenced with the following statement:

```
'VBS_Example_01
Dim objObject
'Change to Screen "MainScreen"
HMIRuntime.BaseScreenName = "MainScreen"
Set objObject = HMIRuntime.Screens(1).ScreenItems(1)
```

An object is referenced by means of its name and an object property is changed with the following statement: In order to do this, the object must be set up with this name in the screen.

```
'VBS_Example_02
Dim objCircle
HMIRuntime.BaseScreenName = "MainScreen"
Set objCircle = HMIRuntime.Screens(1).ScreenItems("Circle_01")
objCircle.BackColor = vbGreen
```

www . ElectricalPartManuals . com

12.4.4 Synchronization of tags and objects

Introduction

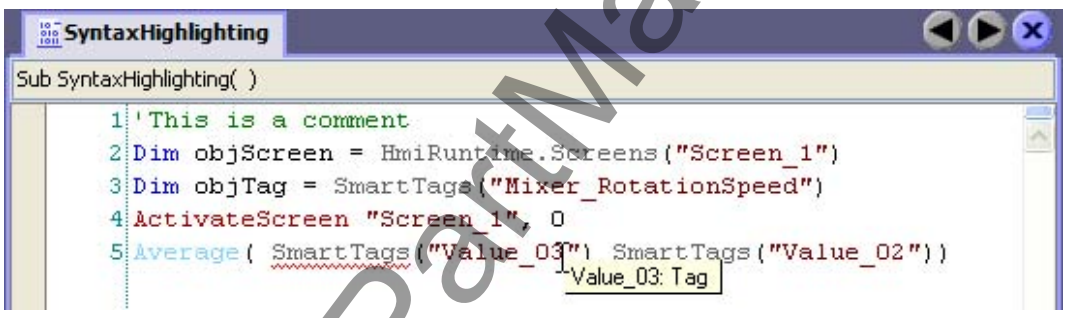
When an object name is changed in WinCC flexible, the change affects the entire project. Such changes are identified as "Synchronizing" in the script.

Application example

In the tag editor you defined the tag "Oil Temperature" which you want to use in a script. During the configuration rename this tag "OilTemperatureMotor1" in the tag editor.

- Script was open during the renaming:

  The old tag name is underlined by a blue wavy line in the script. When you move the mouse pointer over the tag name, Tooltip appears. When you click on the button "Synchronize", the tag is renamed in the script:



- Script was closed during the renaming

  When the script is reopened, the tag is automatically synchronized.

12.5 Debugging

12.5.1 Debugging Scripts

Introduction

Debugging allows you to test your scripts in runtime for logical programming errors. For example, you can test whether the proper values were delivered to the tags, and whether cancellation terms are realized correctly.

To debug your scrips use exclusively the "Microsoft Script Debugger" or the "Microsoft Script Editor" supplied with Microsoft Office XP.

www.ElectricalPartManuals.com

Error types

The following error types are distinguished when debugging:

- Runtime error

  A runtime error occurs when you try to perform an invalid of incorrect instruction, e.g. when a tag is not defined.

  In order to intercept runtime errors, you can use the instruction "On error resume next" in the script. This instruction causes a successive instruction to be carried out after a runtime error. You can check the error code with the error object in the next line. In order to stop the processing of runtime errors in the script, use the instruction "On error go to 0." Additional information about error processing is found in the Microsoft VBS help in the information system.

- Logical error

  A logical error occurs when the event you are expecting does not take place, e.g. because a condition was checked incorrectly. In order to resolve logical errors, go through the script step by step in order to identify the part of the script which does not function.

12.5.2 Integrating the debugger

Installing a Script Debugger for WinCC flexible

A script debugger must be installed in order to search for errors in scripts with WinCC flexible.

The following script debuggers have been tested and released:

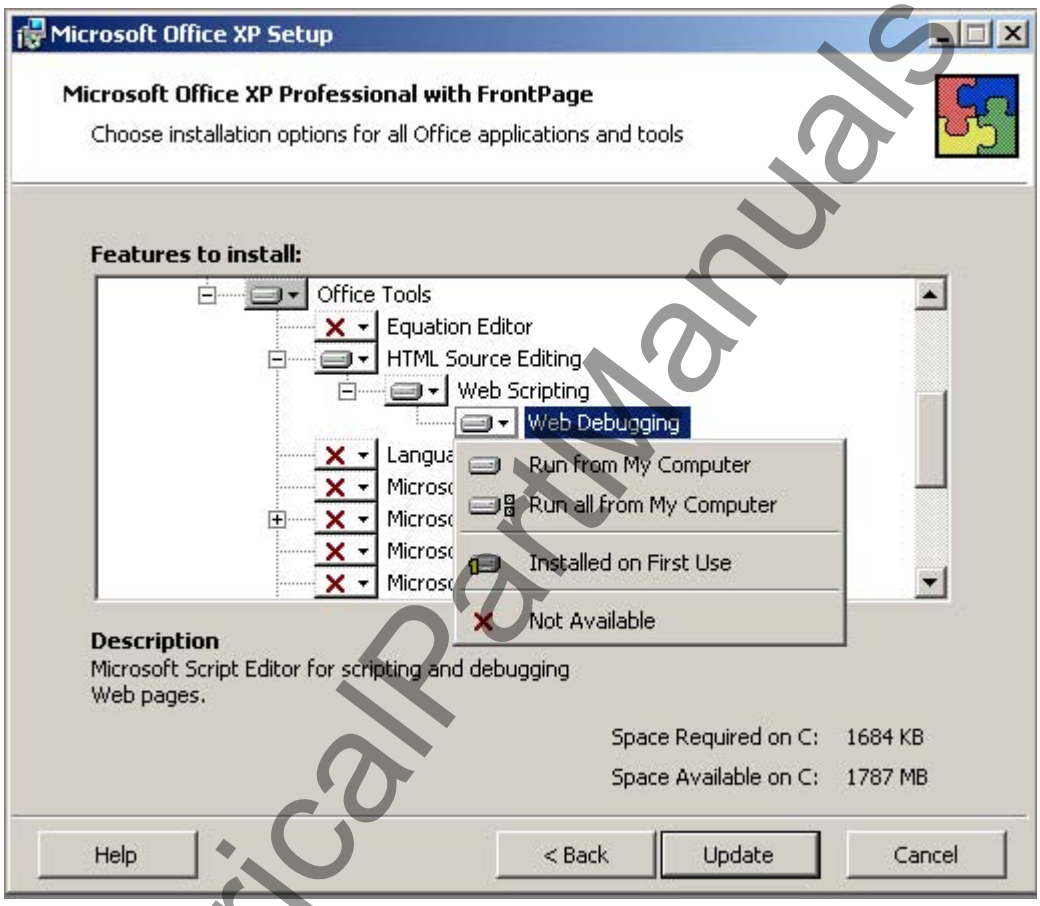
- Microsoft Script Editor by Office XP
- Microsoft Script Debugger

An installed script debugger is either started automatically when a runtime occurs in a script or manually with the command "Start runtime system with script debugger".

www . ElectricalPartManuals . com

Microsoft Script Editor

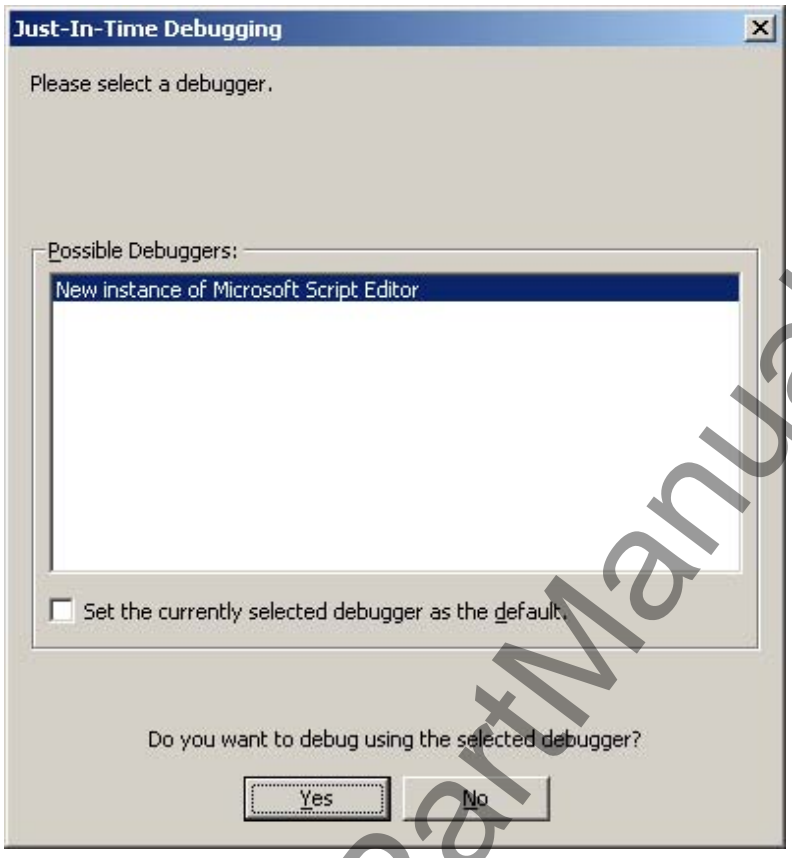
The Microsoft Office XP component "Microsoft Script Editor" contains such a script debugger. If the default settings were used to setup Microsoft Office, the "Microsoft Script Editor" component was set for ("Installed on First Use"). If you wish to explicitly install this component, you must specify it in the Microsoft Office setup. Click on "Web Debugging" in the component selection window and select the option "Run from My Computer".



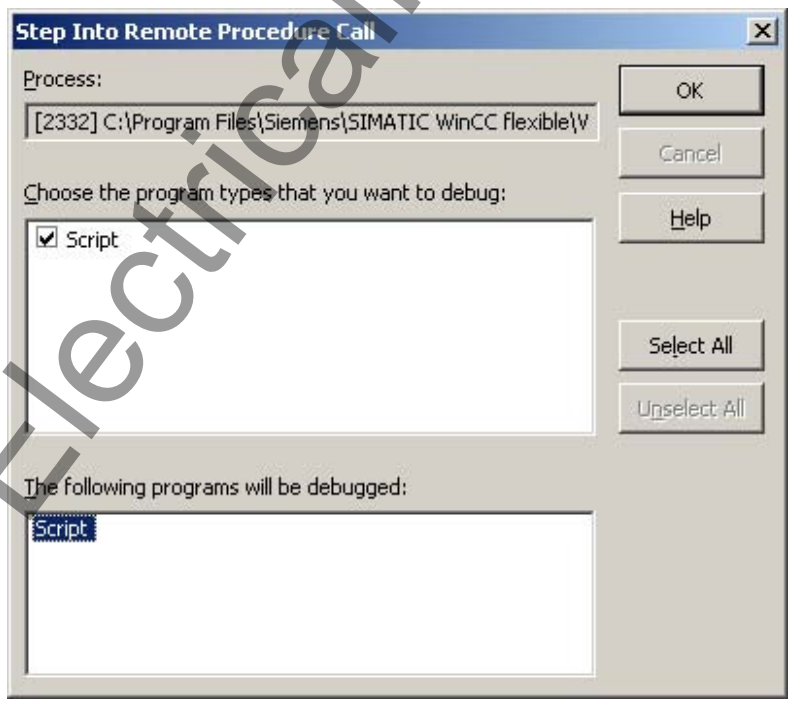
If a project is activated in WinCC flexible with the command "Start runtime system with script debugger", a dialog with a list of available script debuggers appears when the first script is run.

Other installed script debuggers such as "Microsoft Visual Interdev" or "Microsoft Visual Studio .NET" may appear in the list. Select "Microsoft Script Editor" and confirm your selection by clicking "Yes".

www.ElectricalPartManuals.com



Activate the "Script" program object in the "Step Into Remote Procedure Call" dialog and confirm your selection with "OK".

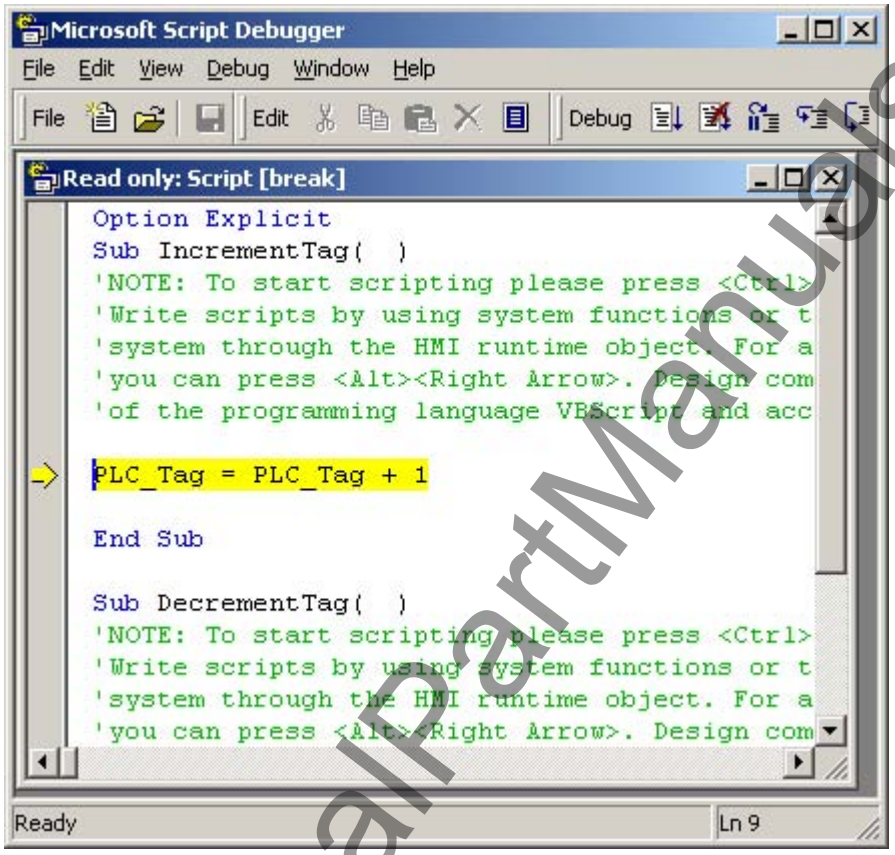


The "Microsoft Script Editor" is started and operation is stopped at the first line of the script.

www . ElectricalPartManuals . com

Microsoft Script Debugger

If no script debugger is available, you can download the "Microsoft Script Debugger" (scd10en.exe) for free from Microsoft (www.microsoft.com). It will be started automatically in WinCC flexible once it is installed.



Note

The "Microsoft Script Debugger" is not supported when another script debugger system is available on your computer!

Script Debugger Does Not Start When Runtime Starts

If you have installed a script debugger but it does not start with the command "Start Runtime with Script Debugger", check the following entry in the system registry:

1. The following entry must be present for "Just in Time Debugging" to be activated:
   [HKEY_CURRENT_USER\Software\Microsoft\Windows Script\Settings]
   "JITDebug"=dword:00000001

   Any change you make only take effect after restarting the computer.
2. To disable debugging of the Internet Explorer, the following entry must be present:
   [HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main]
   "Disable Script Debugger" = "yes"

www.ElectricalPartManuals.com

12.6 Runtime behavior of functions in runtime

12.6.1 Completion of the function list in runtime

Principle

In runtime a function list is completed from top to bottom. A distinction is made between synchronous completion and asynchronous completion, so that no waiting periods ensue during completion. The distinction is made by the system by evaluating the different runtimes of the system functions. Scripts are always processed synchronously independent of the runtime. If a system function returns an error status, the completion of the function list is cancelled.

Synchronous completion

During synchronous completion, the system functions in a function list are performed one after another. The previous system function must be finished before the next system function can be performed.

Asynchronous completion

System functions, which perform file operations such as storing and reading, have a longer runtime than system functions which, for example, set a tag value.

Therefore, system functions with longer runtimes are performed asynchronously. While a system function writes to a storage medium, e.g. a recipe record, the next system function is already being performed. Due to the parallel completion of system functions, waiting periods at the HMI device are avoided.

12.6.2 Processing of scripts in runtime

Principle

Only one script at a time can be performed in runtime. If several scripts are waiting to be edited, the scripts are lined up in a queue and completed one after another.

Note

A loop in a script therefore blocks the execution of other scripts in the queue even if the scripts are triggered asynchronously.

WinCC flexible supports a maximum nesting depth of eight scripts. Please note that the nesting depth is not checked.

www.ElectricalPartManuals.com

Note

If a script is configured for the "Shutdown" event, only those functions may be used in the script which are specified as configurable objects in the reference of the "Shutdown" system function.

Ensure that the ending of the runtime is not interferred with by the execution of the script.

12.6.3 Delivery and return of values

Delivery of a value

When calling up a script, parameters are delivered according to the principle "Call by Reference". When you pass a tag as a parameter, for example, the value assignments in the script have an immediate effect on the value of the tag.

You don't have to set up any parameters for local tags in the script, but rather, you can use the parameters directly.

Example: The system function "SetValue(Y, X)" assigns the value "5" to the tag "IndexTag":

SetValue IndexTag, 5

Return of a value

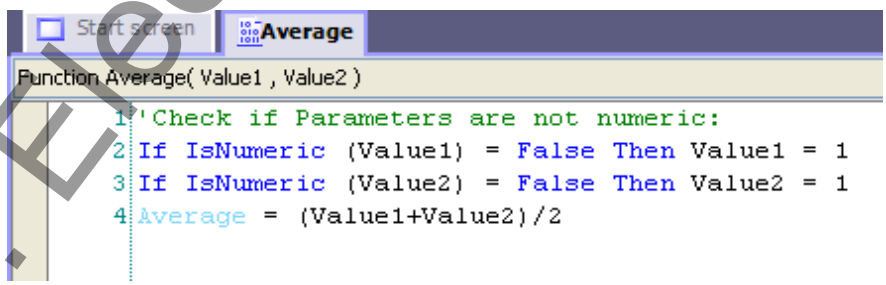
Return values can return the result of a calculation (e.g. average value of two numbers). But a return value can also give information about whether an instruction was performed correctly.

Therefore, the system functions which perform file operations such as "Delete" also have return values.

Note

The return value of a system function can only be assigned an external or internal tag.

In order for a script to return a value, you must have chosen the type "Function" for the script. In the script you assign the return value to the name of the script:



In order to create an average value from two numbers, call up the "Average" function and deliver the values to be processed to a tag, for example:

```
Average Value = Average 4. 6
```

You can then output the average value in an output field.

www.ElectricalPartManuals.com

12.6.4 Changing of object properties in runtime with VBS

Introduction

You can access object properties of screen objects and tags in runtime with VBS. When you change values of object properties with VBS, this has no effect on the project data.

Changing object properties

When you change an object property of a screen element with VBS in runtime, this change remains effective only as long as the screen is active. As soon as you change the screen or reload the screen, the configured object properties are displayed.

Language switching

When you change the language in runtime, the foreign language labels are loaded from the configuration data. In case you changed text with VBS, this text is then overwritten.

12.6.5 HMI device dependent system functions in the script

Principle

If you use system functions in a script which are not available on the set operating unit, you will receive a warning message. In addition, the respective system function in the script will be underlined with a wavy blue line.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

13 Structure of Multilingual Projects

13.1 Languages in WinCC flexible

13.1.1 Working with multiple languages

Multi-language configuration in WinCC flexible

You can configure your projects in multiple languages using WinCC flexible. There are various reasons for creating a project in multiple languages:

- You would like to use a project in more than one country. The project is created in multiple languages; when the HMI device is commissioned, only the language spoken by the operators at the respective site is transferred to the HMI device.
- You would like to provide multiple languages to the various operators in a plant. The project is created in multiple languages because the service personnel do not speak the same language as the operators. Example: An HMI device is used in China, but the service personnel understand only English.

Translating project texts

With WinCC flexible, you can directly enter project texts in several languages in various editors, for example in the "Screens" editor or the "Project texts" editor. In addition, WinCC flexible provides options for exporting and importing your configuration for translation purposes. This is particularly advantageous if you configure projects containing a large amount of text and want to have it translated.

Language management and translation in WinCC flexible

The following areas of the project view are used to manage languages and translate texts in WinCC flexible.

Area	Short description
Project languages	Management of project languages, editing language, and reference language.
Languages and fonts	Management of runtime languages and fonts used on the HMI device.
Project texts	Central management of configured texts in all project languages.
Graphic browser	Management of graphics and their language-dependent variants.
Dictionaries	Management of system dictionary and user dictionaries

www.ElectricalPartManuals.com

13.1.2 WinCC flexible terminology

Language principles in WinCC flexible

Multi-language capability is implemented on various language levels in WinCC flexible.

User interface language and project languages

Two language levels are differentiated in WinCC flexible:

- User interface language

  During configuration, text is displayed in the WinCC flexible menus and dialog boxes in the user interface language. You select the user interface language to be used when you install WinCC flexible. You can change the user interface language with the menu command "Options ▸ Settings."

- Project languages

  Project languages are used to create a project in multiple languages.

The two language levels are completely independent of one another. For example, you can create English projects at any time using a German user interface and vice versa.

Project languages

The following project languages have been released for WinCC flexible:

- Chinese (PRC)
- Chinese (Taiwan)
- Danish
- German
- English
- Finnish
- Flemish
- French
- Greek
- Italian
- Korean
- Norwegian
- Polish
- Portuguese
- Russian
- Swedish
- Spanish
- Czech
- Turkish

www.ElectricalPartManuals.com

- Hungarian
- Japanese

You can generally also configure in any language available in Windows. However, restrictions may apply when some languages are used for configuration, such as:

- The HMI does not support right-to-left languages such as Hebrew or Arabic.
- Language-specific fonts are not available.
- Non-editable texts stored in WinCC flexible are displayed in English.

The following languages are differentiated within the project languages.

- Reference language

  The reference language is the language that you use to configure the project initially.

  During configuration, you select one of the project languages as the reference language. You use the reference language as a template for translations. All of the texts for the project are first created in the reference language and then translated. While you are translating the texts, you can have them displayed simultaneously in the reference language.

- Editing language

  You create the translations of the texts in the editing language.

  Once you have created your project in the reference language, you can translate the texts into the remaining project languages. For this purpose, you select one of the project languages as the editing language and edit the texts for this language. You can change the editing language at any time.

  Note

  When switching the project languages, the assignment to the keys on the keyboard also changes. In the case of some languages (e.g. Spanish), switching the keyboard assignment is not possible due to the operating system. In this case, the keyboard assignment is switched to English.

- Runtime languages

  Runtime languages are those project languages that are transferred to the HMI device. You decide which project languages to transfer to the HMI device depending on your project requirements.

  You must provide appropriate operator control elements so that the operator can switch between languages during runtime.

www . ElectricalPartManuals . com

13.2 Language Settings

13.2.1 Language settings in the operating system

Introduction

The operating system settings on the configuration computer influence the language management of WinCC flexible in the following areas:

- Selection of project languages
- Regional format of dates, times, currency, and numbers
- Displaying ASCII characters

Language settings in the Operating System

A language is not available as a project language unless it is installed in the operating system.

- Settings in Windows 2000:

  You can select the languages you want to install subsequently from the list of "Language settings for the system," which is located on the "General" tab in "Start > Settings > Control Panel > Regional Options."

- Settings in Windows XP:

  You can call the "Regional and Language Options" dialog using the control panel icon of the same name in "Start > Settings > Control Panel > Date, Time, Language, and Regional Options." Afterwards, you can install your choice of languages on the "Languages" tab.

The Input Method Editor (IME) is available in Windows for configuring Asian texts. Without this editor, you can display Asian text but not edit it. For more information on the Input Method Editor, refer to the documentation for Windows.

If language-dependent project texts, such as alarm texts, should be displayed in the simulator in Asian characters, the operating system must be switched to the respective language.

Regional format of dates, times, currency, and numbers

WinCC flexible specifies a fixed date and time format in the Date - Time field for the selected project language and runtime language.

In order for dates, times, and numbers to be presented correctly in the selected editing language, this language must be set in the Regional Options on the Control Panel.

Displaying ASCII characters

With text output fields, the display of ASCII characters as of 128 depends on the set language and the operating system being used.

If the same special characters are to be displayed on different computers, then the computers must use the same operating system and country settings.

www.ElectricalPartManuals.com

13.2.2 Operating system settings for Asian languages

Settings on Western operating systems

If you want to enter Asian characters, you must activate the support for this language in the operating system. To do this, open the Control Panel and select "Regional and Language Options". On the "Languages" tab, activate the check box "Install files for East Asian languages". Then click on "Details" under "Text Services and Input Languages". The dialog "Text Services and Input Languages" is opened. On the "Settings" tab add the required default input language under the "Installed Services".

For entering Asian characters on Western operating systems, the "Input Method Editor" must also be installed. To install, under Windows XP in the Control Panel open "Regional and Language Options" ► Languages ► Details". In the "Text Services and Input Languages" add the necessary default input language under "Installed Services". Under Windows 2000 you´ll find the default input language in the Control Panel under "Regional Options ► Input".

To enter Asian characters when configuring, switch to the Asian entry method in the "Input Method Editor".

Settings on Asian operating systems

If you are configuring on an Asian operating system, to enter ASCII characters, e.g. for object names, you must switch to the English default input language. As the English default input language is included in the basic installation of the operating system, you do not need to install an additional input locale.

13.2.3 "Project Languages" editor

Introduction

You select the languages for creating your project in the "Project Languages" Editor:

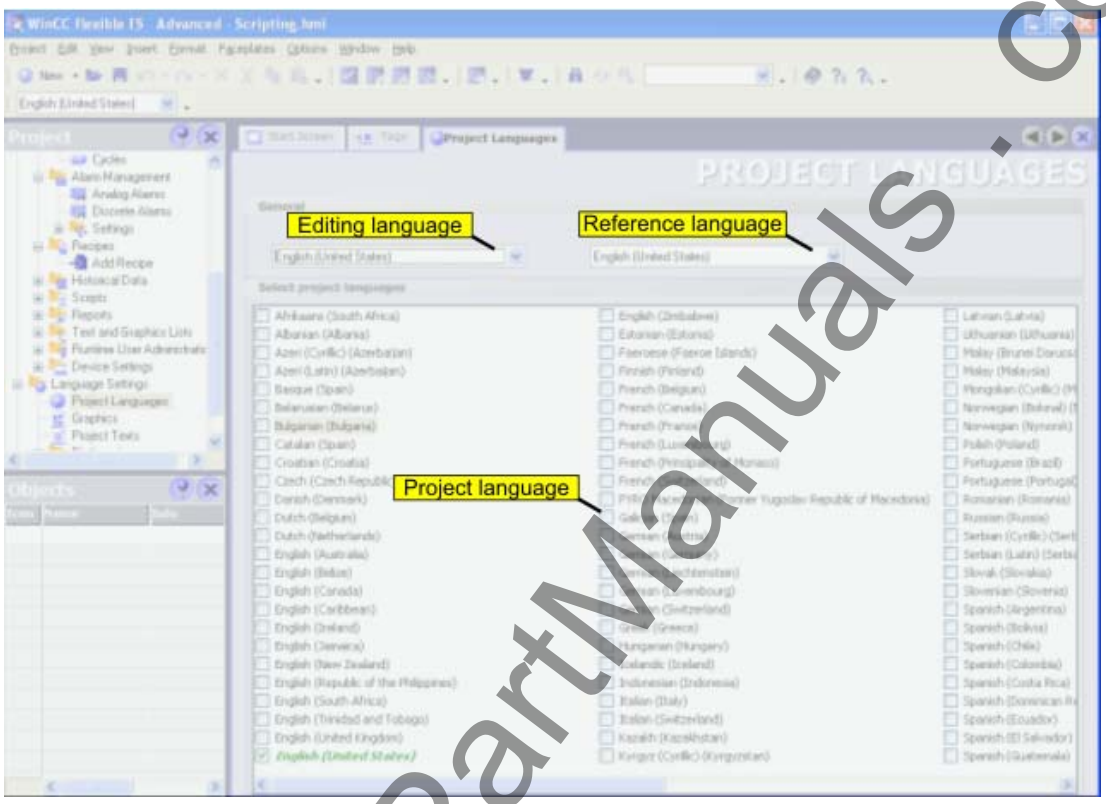
- The project languages for creating your project
- The reference language in which you configure the project initially.
- The editing language in which you translate the text.

Open

To open the "Project Languages" editor, double-click "Project Languages" in the "Localize" group in the "Project view."

www . ElectricalPartManuals . com

Structure



Project languages

Here you enable the project languages for creating your project.

Reference language

Here you select the reference language from the project languages. The languages displayed are limited to those that you enabled in the list of available languages.

Editing language

Here you select the editing language from the project languages. The languages displayed are limited to those that you enabled as project languages in the list of available languages.

www . ElectricalPartManuals . com

13.3 Creating a project in multiple languages

13.3.1 Creating a project in multiple languages

Translation methods

You first create a project for one language only and then test it.

Afterwards, you configure all other languages required. The following options are available:

- Direct translation of project texts in the editors used to create the individual objects.
- Central translation of project texts in the "Project texts" editor.
- Export of project texts, translation in another program, and import of translated texts to WinCC flexible.

Use dictionaries to speed up translation and maintain consistency of translated texts.

Recommended workflow

1. Set a language you are familiar with as the reference language in the "Project languages" editor.

   At the start of configuration, the reference language should correspond to the editing language.
2. Create the project in this language. The reference language text is used as the source language for translation.
3. In the "Project languages" editor, set one of the other project languages as the editing language.
4. Translate all project texts into this language. You can perform the translation either directly in the individual editors or in the centralized "Project texts" editor where all project texts and their points of use are displayed.

   As an alternative,, you can export the texts to a *.csv file, have them translated, and then import the translated texts back into your project.
5. If necessary, adapt the graphics in the project to reflect the editing language or country-specific factors.
6. Repeat steps 3 to 5 for all other project languages.

Result

The project can now be compiled and transferred to the HMI device. Specify which runtime languages are to be available on the HMI device in the transfer settings.

www . ElectricalPartManuals . com

13.3.2 Specific features of Asian and Eastern languages in the engineering system

Introduction

When configuring for Asian languages or in an Asian language some specific features should be observed. These specific features must also be observed for other languages that have complex characters.

Basic principles of configuration

In order to ensure the full functionalitiy of a project, when configuring in WinCC flexible some elements should include no complex characters. The restrictions apply for the following elements:

- Object names
- Alarm texts

The object names in WinCC flexible are unique names and therefore are not translated when switching between languages. As the object names are used and processed functionally, they are subject to some restrictions. The object names may not include any special characters, umlauts or complex characters. Affected object names are, for example, project names, tag names, screen names etc.

If you want to log alarms with alarm texts, then you may not use Asian Runtime language. When using an Asian Runtime language you cannot log alarm text, even if the alarm text itself is written, for example, in English. The restriction applies merely to logging, it is still possible to view and output in Runtime. Russian and other 1 Byte languages are not affected by this restriction.

When using Sm@rtAcess and Sm@rtService only those characters that are known on the HMI device can be used.

User administration

No Asian or other complex characters may be used for user names and passwords.

Project documentation

You can optimize the appearance of the printout by selecting the respective Asian or Eastern font from the Configuration Dialog for the project documentation.

Integration in STEP 7

Asian-language projects integrated in STEP 7 must be started via WinCC flexible. If you start integrated Asian projects via STEP 7, you will receive error messages and display errors.

www.ElectricalPartManuals.com

13.3.3 Translating project texts in the editor

Introduction

As a general rule, if you are creating a project for multiple languages, all texts are configured initially in a language you are familiar with. You use this language as the reference language for translation.

Define the project languages in the "Project languages" editor. Select the reference language and the respective editing language to which it should be translated, from the project languages.

Editors with language-dependent objects

The following editors contain language-dependent objects:

- Screens
- Protocols
- Analog alarms
- Discrete alarms
- System alarms
- Recipes
- Text lists
- Graphics lists

Switching the editing language in the WinCC flexible editors

You can switch the editing language by means of the "Localize" toolbar. The editing language applies to all editors.

Reference texts

As a general rule, if you are creating a project for multiple languages, all texts are configured initially in a language known to you. This language then serves as the reference language.

If you then switch the editing language to enter texts in another language, all of the text fields are empty.

WinCC flexible offers a convenient reference text function so that you have a template for translation. You can display the reference text window containing texts in the reference language in dialogs and editors.

www . ElectricalPartManuals . com

13.3.4 "Project texts" editor

"Project texts" editor

You have access to all texts of a project in the "Project texts" editor.

Examples:

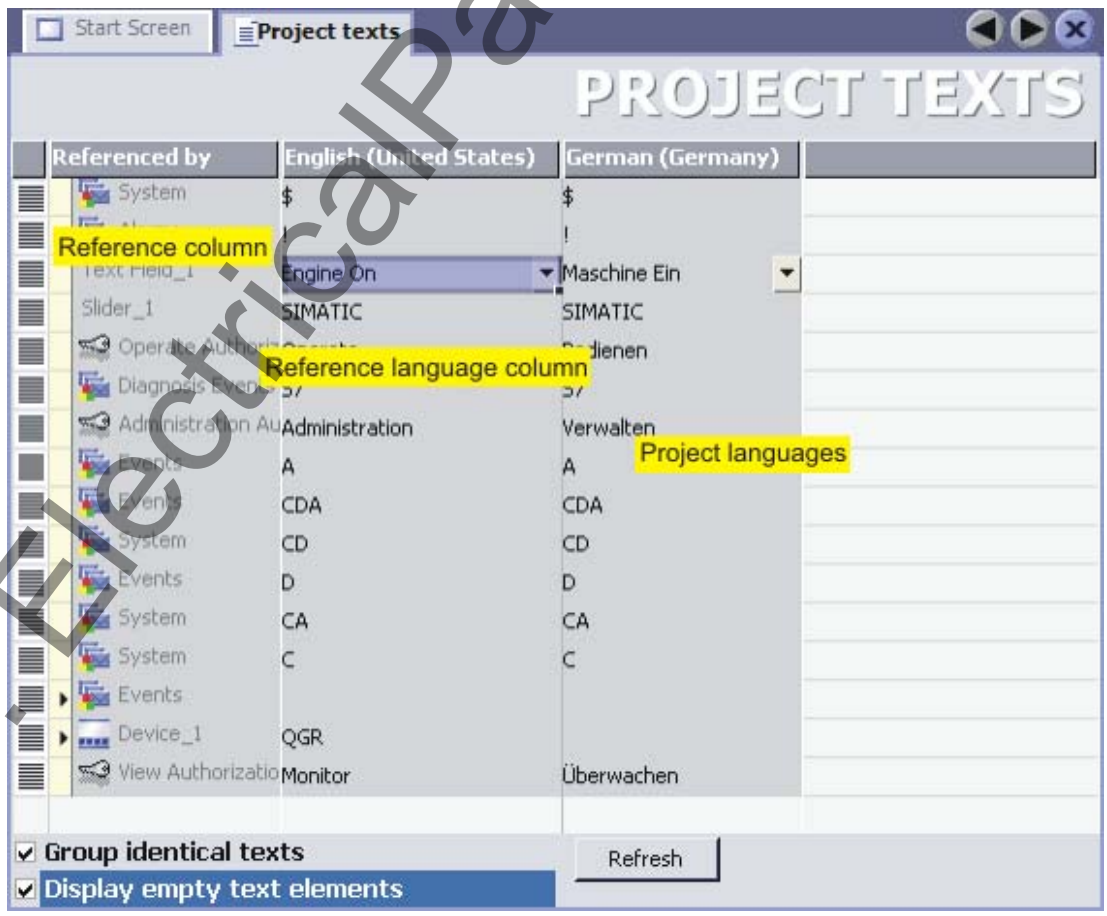
- Texts in screens
- Alarms
- Comments
- Operator notes
- Texts of recipes

Central text display

Opening the "Project texts" editor

To open the "Project texts" editor, double-click "Project texts" in the "Localize" group in the project view.

Structure of the "Project texts" editor



www.ElectricalPartManuals.com

Elements in the "Project texts" editor

In the "Project texts" editor, a single column is created for each project language you have set. The text of a configured object is represented in all languages in a single row in the table. The rows are divided into the following columns.

- Reference column

  The "Referenced by" column indicates the editor where the texts originated.

- Reference language column

  The second column displays the texts in the reference language.

- Project languages

  The remaining columns display the texts in the other project languages provided the texts have already been translated.

Translation methods

You can translate the configured texts as follows:

- Internal translation of texts directly in the "Project texts" editor.

  This method is recommended for texts with little to be translated.

- External translation of texts using the export and import function.

  This method is recommended for large amounts of text to be translated or when there are many project languages.

The "Project texts" editor is linked to the other editors. Texts introduced here are automatically available in other editors as appropriate. You can also jump directly from the "Project texts" editor to the point of use of the object to be translated.

13.3.5 Exchanging texts with translators

Introduction

External translation of project texts is advantageous when there is a large amount of text in multiple languages. You can use the export function to transfer project texts to external translators. You can then use the import function to reintegrate the translated texts back into your project.

Scope of export and import

- Export and import of all project texts

  If you want to send all texts for translation (in the case of a new project, for example), you export all texts from the project to a *.csv file for the translator. You then import the texts following translation. The translated texts are automatically assigned to the correct point of use in the project.

  If you have made changes to project texts in WinCC flexible in the meantime, the modified texts are not overwritten during the import.

www . ElectricalPartManuals . com

- Export and import of texts in a particular editor

  You can limit the export and import of texts to those in a particular editor in WinCC flexible.

- Export and import of new texts for partially translated projects

  If you have inserted new texts in a previously translated project, you can selectively export the texts that have not yet been translated. This minimizes the translation effort required.

13.4 Working with dictionaries

13.4.1 Working with dictionaries

System dictionary and user dictionary

Various dictionaries assist you when translating projects in WinCC flexible.

- System dictionary

  The system dictionary included in WinCC flexible contains commonly used process automation terminology and the corresponding translations. The system dictionary can be viewed but not modified.

  All rights for the source documents lie with the "Landesinstitut für Erziehung und Unterricht (LEU)", Rotebühlstraße 131, 70197 Stuttgart, Germany,
  Tel. +49 711 6642-235, Fax +49 711 6642-203
  "http://www.schule-bw.de/unterricht/faecher/englisch/tech_english/tech_woerterb"

  In as far as nothing is specified to the contrary and in as far as rights of other parties are not affected, the distribution of these documents as a whole or in part, in electronic and printed form, is desired under the condition that the source (Landesbildungs-Server Baden-Württemberg) and the URL are named.

  A commercial distribution of the documents is expressly prohibited without the previous written permission of the LEU.

- User dictionaries

  You store translations of terms that occur repeatedly in your project texts in a user dictionary. In a user dictionary you can directly enter terms or adopt project texts from the editors.

  WinCC flexible enables the use of several user dictionaries. These are physically managed within a file and can be integrated in new projects.

Auto translate function

When the "Auto translate" function in the "Project texts" editor is enabled, all dictionaries are searched for the terms to be translated. Terms that are found are entered as suggested translations in the "Project texts" editor. You can then accept or modify the suggested translations.

www.ElectricalPartManuals.com

13.4.2 "System dictionary" editor

Introduction

Terms in the system dictionary are managed in the "System dictionary" editor. You can view and sort the system dictionary in this editor, but you cannot make changes.

Opening the "System dictionary" editor

To open the "System dictionary" editor, double-click "Dictionaries > System dictionary" in the "Localize" group in the project view.

Structure of the "System dictionary" editor

French (France)	German (German...	English (United Stat...	Italian (Italy)	Spanish (International S...
Couleur de bord...	3D-Rahmenfarbe oben	3D Border Color	Colore del bordo 3D	Color superior de borde 3D
Largeur de cadr...	3D-Rahmenbreite	3D Border Weight	Larghezza della cornic...	Ancho de borde 3D
Couleur d'ombre...	3D-Rahmenfarbe un...	3D Shadow Color	Colore d'ombreggiatu...	Color inferior de borde 3D
Bargraphe tridim...	3D-Balken	3D-Bar Graph	Barra 3D	Barra 3D
abréviation de li...	ZULI	abbreviation of assignm...	abbreviazione di lista ...	abreviatura de lista de asign...
annulation	Abbruch	abort	interruzione	cancelar
a propos de	Info	about	Informazio...	...nación
droit a l' accès	Zugriffsrechte	access authorization	diritti di accesso	derecho de acceso
touche d'accès r...	Zugriffstaste	access key	tasto di scelta	tecla de acceso
acquitté	quittiert	acknowledged	acquisito	acusado
acquittement	Quittierung	acknowledgement	acquisizione	acuse
acquittement obl...	Quittierpflicht	acknowledgement requi...	obbligo di acquisizione	acuse obligatorio
variable d'acquit...	Quittiervariable	acknowledgement tag	variabile di acquisizione	variable de acuse
concept d'acquit...	Quittierphilosophie	acknowledgement theory	concetto di acquisizione	concepto de acuse
groupe d'acquitt...	Quittiergruppen	acknowledgment group	gruppo di acquisizione	grupo de acuse
bit d'acquittement	Quittierbit	acknowledgement bit	bit di acquisizione	bit de acuse
cycle d'acquisition	Erfassungszyklus	acquisition cycle	ciclo di rilevamento	ciclo de registro
action	Aktion	action	azione	acción
active	aktiv	active	attiva	activa
apparu	gekommen	active	In arrivo	llegado
valeur de mesure	Istwert	actual value	valore istantaneo	valor actual
acyclique	azyklisch	acyclic	aciclico	acíclico
Adaptation de bor...	Rahmenanpassung	Adapt Border	Adatta bordo	Cambio de borde
Adapter vue	Bild anpassen	Adapt Picture	Adatta immagine	Adaptar imagen
Adapter taille de...	Fenster anpassen	Adapt Size	Adatta dimensioni	Adaptar ventana
ajouter	Hinzufügen	add	aggiungi	agregar
adresse	Adresse	address	indirizzo	dirección

Work area

The languages are displayed in a table in the work area. A separate column is created for each language. Each table row contains a process automation term and its translations.

In order to find the translation for a particular term quickly, you can sort the table alphabetically according to the entries in a column. To accomplish this, click the header of the appropriate column.

www.ElectricalPartManuals.com

13.4.3 "User dictionary" editor

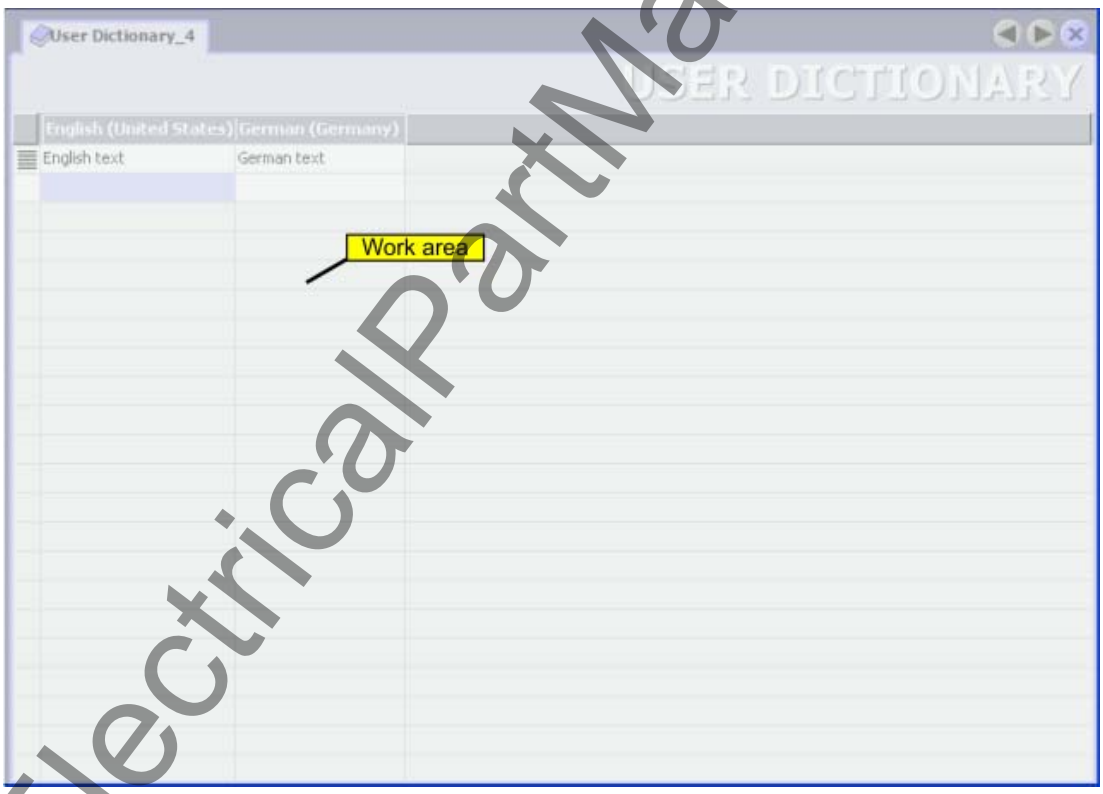
Introduction

You can view and edit the terms of a user dictionary in the "User dictionary" editor.

Opening the "User dictionary" editor

User dictionaries are displayed in the "Localize > Dictionaries > User dictionaries" group in the project view. To open the "User dictionary" editor, double-click the name of a user dictionary.

Structure of the "User dictionary" editor



Work area

The text of the user dictionary is shown in tablular form in the work area. When creating a user dictionary, you select the languages the user dictionary should include. The table in the editor contains one column for each of these languages. Each table row contains a term in the reference language and its translations provided you have entered these translations in the user dictionary.

www.ElectricalPartManuals.com

13.5 Use of language-dependent graphics

13.5.1 Use of language-dependent graphics

Language-dependent screen versions

You use the "Graphics" editor to import graphics into your project and manage their language-dependent versions. The graphics can then be linked to the process screens of the project in the "Screens" editor. If you create a project in multiple languages, different graphics may be required for the various project languages due to the following:

- The graphics contain text.
- Cultural factors play a role in the graphics.

In both cases, you must produce language-dependent versions of the graphics.

Basic procedure

1. First, configure all process screens for one language in the "Screens" editor.
2. Create a version of the graphics for each project language in a graphics program.
3. In the "Graphics" editor, import the language-dependent graphics into the project.

Result

The version of each graphic for the current editing language setting is displayed in the "Screens" editor. The version of each graphic for the current runtime language setting is displayed during runtime.

13.5.2 "Graphics" editor

Introduction

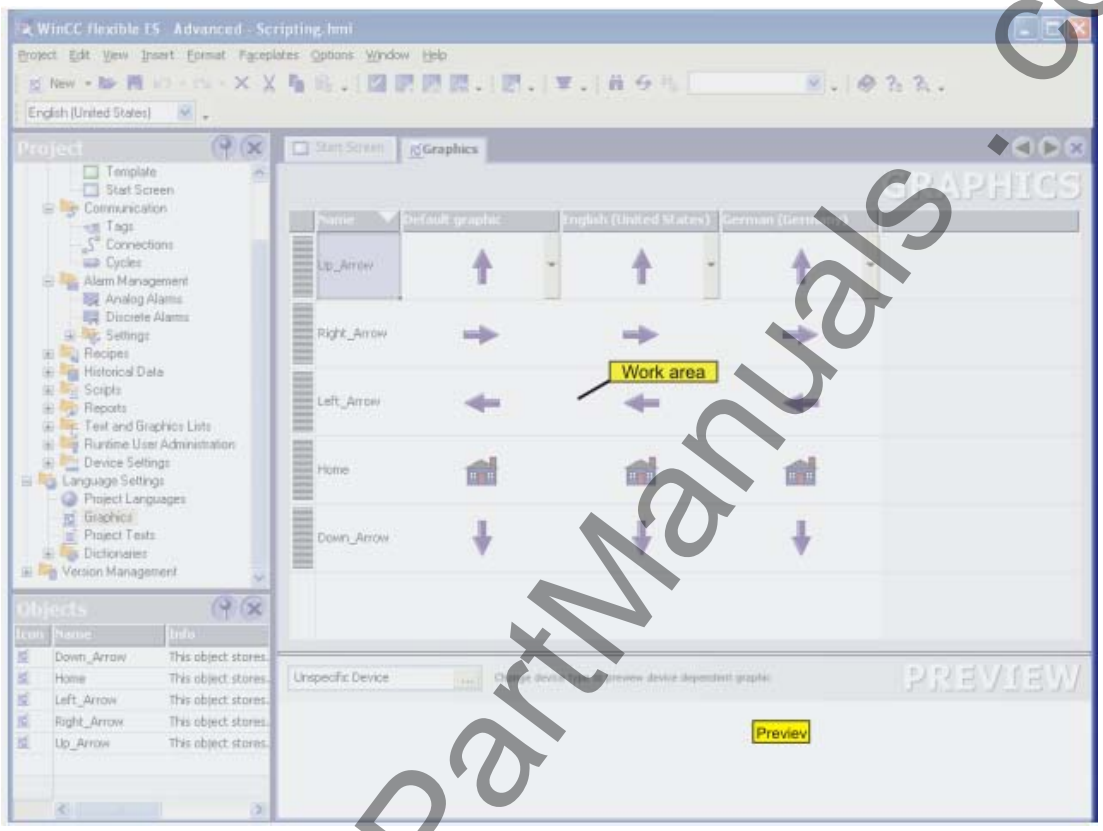
You manage the configured graphic objects in multiple language versions in the "Graphics" editor.

Opening the "Graphics" editor

To open the "Graphics" editor, double-click the "Graphics" editor entry in the "Localize" group in the project view.

www . ElectricalPartManuals . com

Structure of the "Graphics" editor



Work area

Here all of the configured graphic objects are displayed in a table. A separate table column is created for each project language, which contains the versions of the graphics for that language.

In addition, you can specify a default graphic for each graphic to be displayed whenever a language-specific graphic for a project language does not exist.

Preview

Here you can preview the graphic displays on various HMI devices.

www . ElectricalPartManuals . com

13.6 Languages in Runtime

13.6.1 Languages in Runtime

Using multiple runtime languages

You can decide which project languages are to be used as runtime languages on a particular HMI device. The number of languages that can be available simultaneously on the HMI device is dependent on the device. To enable the operator to switch between languages during runtime, you must configure a corresponding operator control element.

When runtime starts, the project is displayed according to the most recent language setting. When runtime starts the first time, the language with the lowest number in the "Order for language setting" is displayed.

Setting runtime languages during configuration

In the "Languages and Fonts" editor you can specifiy:

- The project languages to be available as runtime languages for the respective HMI device
- The order in which the languages are to be switched

13.6.2 Configuring language switching

Introduction

If multiple Runtime languages are available on the HMI device, you must configure language switching. This is necessary to enable the operator to switch between the various Runtime languages.

Methods for language switching

You can configure the following methods for language switching:

- Direct language selection

  Each language is set by means of a separate button. In this case, you create a button for each Runtime language.
- Language switching

  The operator toggles the languages by means of a single button.

Regardless of the method used, the button names must be translated into each of the languages used. You can also configure an output field that displays the current language setting.

www . ElectricalPartManuals . com

13.6.3 Specific features of Asian and Eastern languages in Runtime

Introduction

When configuring for Asian languages some specific features should be observed for operation in Runtime.

Memory requirement for Asian character sets

The memory requirement is of course greater when using Asian languages. Therefore, watch out for any error messages when compiling.

Inputting Eastern and Asian characters (not ANSI)

The input of Eastern and Asian characters is not possible on PC-based HMI devices.

Interpretation of Asian characters

When using Sm@rtAcess and Sm@rtService only those characters that are known on the HMI device can be used. To be able to use Asian characters, these must be configured in the engineering system. Additionally configured characters require additional memory on the HMI device. Please observe the size of the available memory on the HMI device.

Configurable character sets

With the 270 series HMI devices and with MP 370, alongside the default European and Asian character sets, only configurable European character sets can be used. Additional, configurable character sets for Asian languages cannot be used at present.

www . ElectricalPartManuals . com

14 Project documentation

14.1 Basics

14.1.1 Project documentation

Introduction

Project documentation serves to print the configuration data of a WinCC flexible project, e.g. a table containing the tags used and their parameters.

Application

You can output configuration data in a project report. You can output project reports for:

- A complete WinCC flexible project
- A component of WinCC flexible
- A single or multiple objects

The selection of the output data depends on the objects or components selected. The composition of the data depends on the selected output format, "Compact" or "Complete" and is carried out when the project report is generated by the system.

If you output the configuration data or several or all the components of WinCC flexible, a separate chapter is output for each component. A separate chapter is also output for each WinCC flexible screen in view of the possible data quantity.

You can open the project report in a preview before you start the output. The preview allows you to verify the project report before you output it.

Output media

Project reports can be output to:

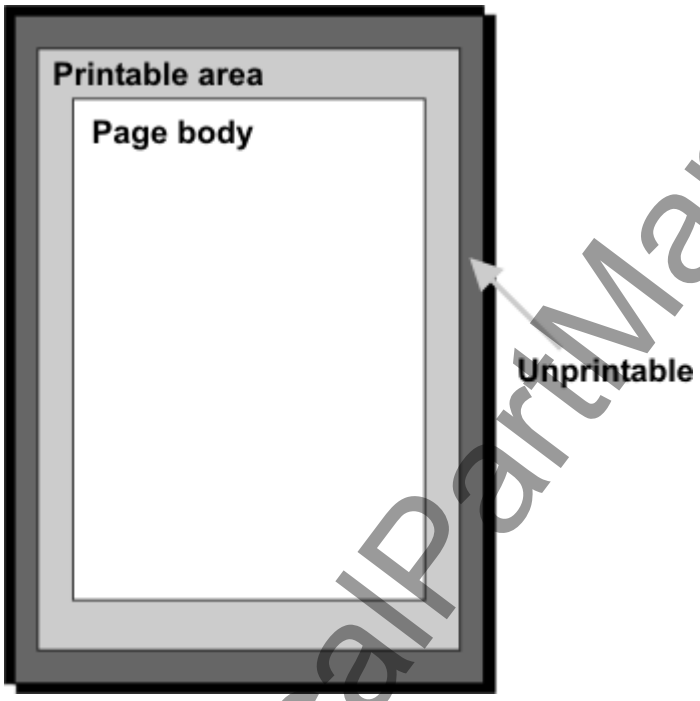
- A printer
- A file
- The screen

www.ElectricalPartManuals.com

14.1.2 Structure of a layout

Introduction

A layout for the project documentation consists of a cover sheet and a formal contents sheet which is filled dynamically with the configuration data. If the data for the output fill several pages, page breaks are inserted automatically by the system. Layouts are divided into various areas. The page area displays the entire layout surface. The print margins can be specified for this area. The printable area consists of the header, footer and the page body.



The header and footers are output on every page of a project report. The header and footer are not output on the cover sheet.

Layout of the cover sheet

General information about the project can be output on the cover sheet. The cover sheet contains predefined fields in which you can enter the corresponding information by means of a dialog box. The following information can be output on the cover sheet:

- Project name
- Company name
- Department name
- Author name
- Company logo
- Project logo

www.ElectricalPartManuals.com

Layout of the contents pages

The configuration data are output on the contents pages. The following elements are output in a project report:

Line	Contents
Title	Designation of the selected components for the project report
Name	Designation of the objects, the attributes and the output WinCC flexible screens.
Array	Output configured attribute values of objects.

The lines listed in the table are repeated for all the objects contained in the project report. Two formats are available for the output.

In "Complete" format the data are output in two columns. In "Complete" format all the attributes of an object are output in the report.

In "Compact" format the data are output in a five-column table. In "Compact" format the five most important attributes of an object are output. The five attributes to be output are preset in the system. The selection of these attributes cannot be modified.

The output format is selected in the "Print project documentation" dialog box. Select the "Compact" or "Complete" format on the "Contents" tab in the "Documentation of the properties" area.

14.2 Using layouts

14.2.1 Using layouts

Introduction

The "Print project documentation" dialog box is used to edit the layouts. This dialog box is used to create new layouts and to duplicate and/or delete existing layouts.

Overview

WinCC flexible provides a ready-made layout as the basis for a project report. The ready-made layout with its settings is always used if you use the  command button to create a new layout. From the ready-made layout, WinCC flexible generates the "Default layout" during installation. It is used to output project reports via the "Print selection" function.

The layouts for the project reports are stored centrally in WinCC flexible and are therefore available in all projects for all users. Configure the common properties of a layout for a WinCC flexible project, e.g. author, company name, project name, header, footer, display used and the settings for the output. Duplicate this template several times and specify different configuration data for the output in each of these templates. For example, create a separate project report for each WinCC flexible component.

A predefined style is available in order to lay out a project report. The style can be modified as required. The style is not saved with the individual layouts. A change in the style therefore affects all the existing layouts.

www.ElectricalPartManuals.com

Commands for layout editing

The following commands are available in the "Print project documentation" dialog box to edit the layouts.

Button	Pop-up menu command	Hotkeys
	New	<CTRL+SHIFT+N>
	Duplicating	<CTRL+SHIFT+D>
	Delete	<DEL>
	Rename	F2
	Print	<CTRL+SHIFT+P>
	Preview	<CTRL+SHIFT+V>
	Export	<CTRL+SHIFT+E>

14.2.2 Editing a layout for the project documentation

Introduction

A layout is opened by using the "Print project documentation" dialog box. The "Print project documentation" dialog box is accessed via:

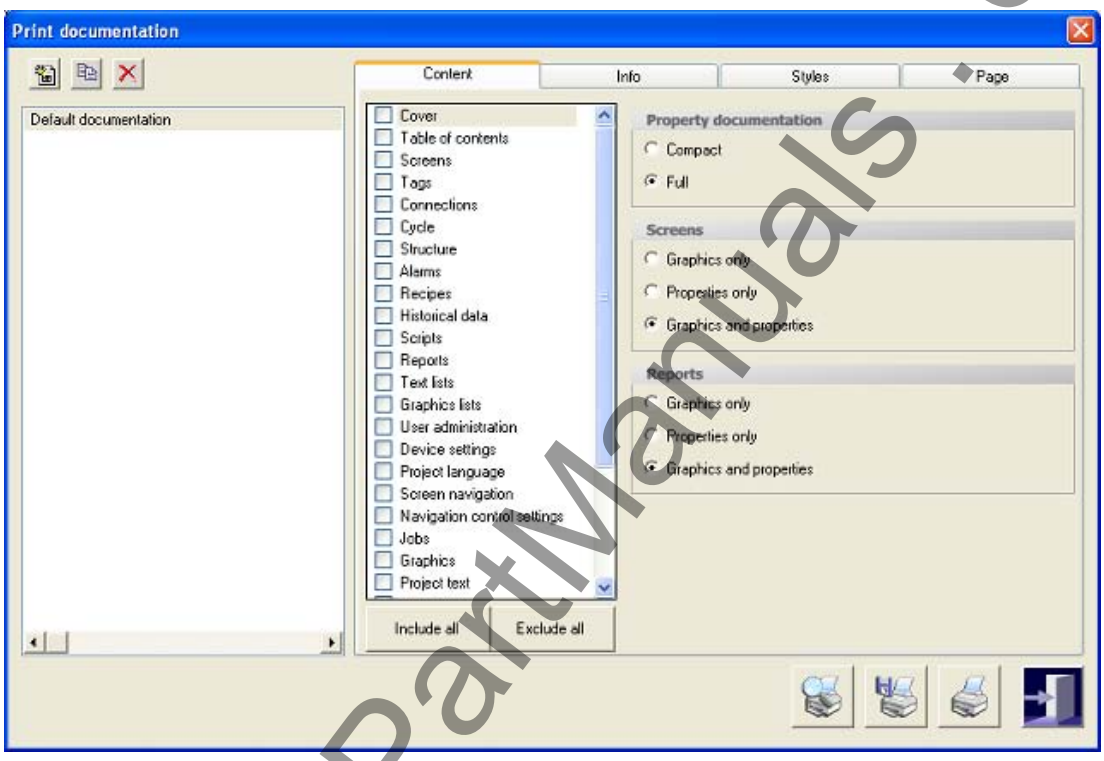
- The "Project ▸ Print project documentation..." menu command
- The  command button in the "Print" toolbar.

Open the properties of the layout to be edited by using the mouse to select the layout.

www . ElectricalPartManuals . com

Editing possibilities

The properties of the selected layout are displayed in the "Print project documentation" dialog box.



The following table shows the available categories and the editing possibilities.

Tab	Editing possibilities
"Contents"	Is used to select the data for the output.
"Info"	Is used to enter the contents for the cover sheet and the contents for the header and footer.
"Style"	Is used to configure the style.
"Page"	Is used to configure the paper format, the page orientation, the page margins and the height of the header and footer.

www . ElectricalPartManuals . com

14.3 Creating a project report

14.3.1 Selecting the data for a project report

Introduction

The selection of the data depends on the selected output format in the layout used. All the attributes of all the configured objects of a WinCC flexible component are output in the "Complete" output format. Five attributes each specified by the system are output in the project report for each object in the "Compact" output format.

Overview

Select the WinCC flexible components for the output on the "Contents" tab in the "Print project documentation" dialog box. Select the output format "Complete" or "Compact" in the "Documentation of the properties" area. In the WinCC flexible components "Screens" and "Reports" you can limit the output of the data. The following options are available:

- "Only graphics"
- "Only properties"
- "Graphics and properties"

14.3.2 Outputting of data of selected objects

Introduction

WinCC flexible offers you the possibility to output the configuration data of individual objects. The output can also be carried out for several selected objects.

Overview

Select the objects for the data output in the project view or in the object view.

The output of the configuration data of a selected object is always performed with the "Default Layout". The desired output options have to be set in this layout. Selection of another layout is valid for the documentation of individual object data.

The configuration data of the selected objects are opened in the preview. Outputting to a printer can be started from the preview. The button can be used to copy the data to the clipboard for further use.

www . ElectricalPartManuals . com

14.3.3 Selecting Objects for the Project Documentation

Introduction

WinCC flexible offers various options for outputting the configuration data of individual or multiple objects of a WinCC flexible component. Start the output using the:

- Main menu
- the toolbar
- the context menu of selected objects

Selecting the objects

Activate the object view and select the desired WinCC flexible component in the project view. The existing objects of the WinCC flexible component are displayed in the object view. Use the mouse to select one or more objects whose data you want to output in the object view.

You can also open the node of a WinCC flexible component in the project view. The existing objects of the WinCC flexible component are displayed. Select one or more objects using the mouse.

Outputting the data

WinCC flexible provides several options for the output of data. After the object selection you start the output via:

-  button.
- The "Print selection" command in the pop-up menu of the selection.
- "Project ▸ Print Selection" command in the menu bar

The configuration data are inserted into the "Default layout" and opened in the preview window.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

15 Planning jobs

15.1 Field of application of the scheduler

Definition

In the scheduler, you link system functions or scripts to an event. For example, you link the SendEMail system function to the "Runtime stop" event so that an e-mail is always sent to a particular recipient at the end of operation.

A task therefore exists: When the event occurs, the linked function is called. An e-mail is sent when runtime ends.

Application example

The scheduler is used to execute event-controlled jobs automatically. For example, you use a job to automate the following:

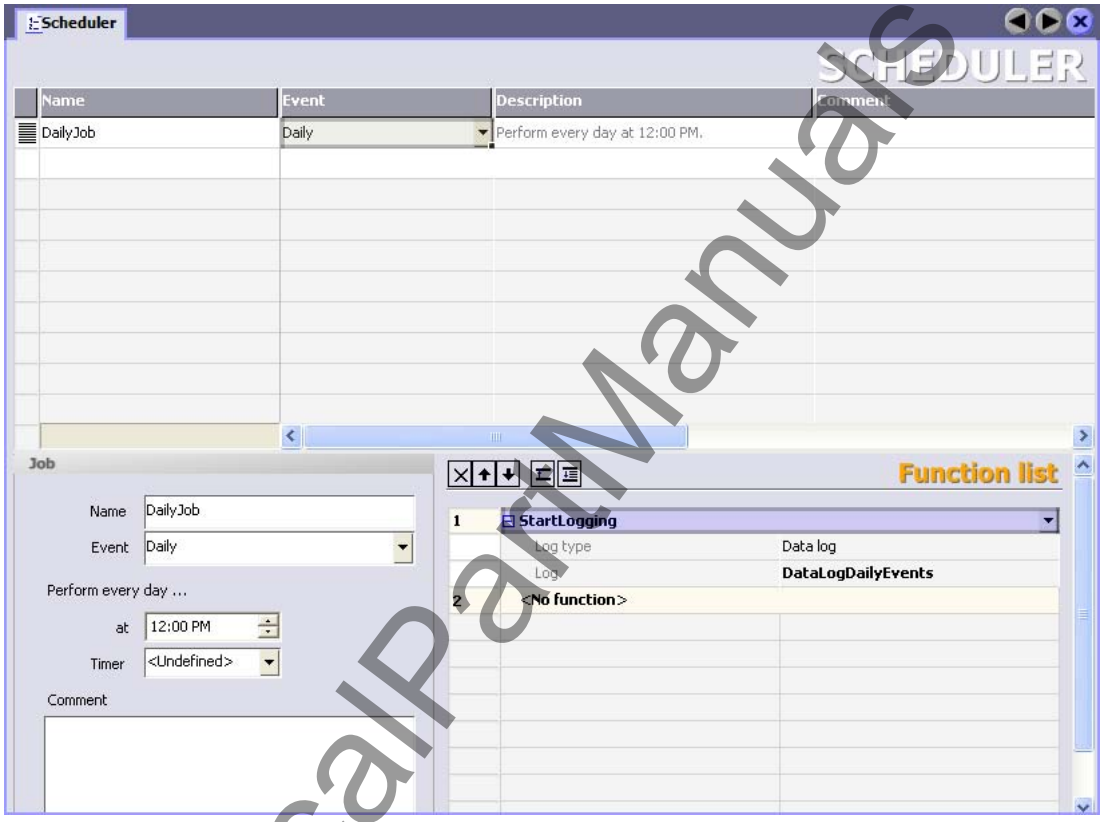
- Regular swap out of log data
- Printout of an alarm report when an alarm buffer overflow occurs
- Printout of a report at shift end

www . ElectricalPartManuals .com

15.2 Working with jobs and events

Introduction

A job consists of a triggering event and a "function list".



Definition

The scheduler differentiates between time-based events and system events. A time-based event occurs at a particular time, for example, "Starting daily at 12:00". Examples of system events are "Runtime stop" and "Change user".

The event occurs either cyclically, for example "Starting every day at 12:00", or acyclically, for example "Change user".

Notice

The events available depend on the HMI device. Not every HMI device supports all events.

The "function list" contains a system function or a script in each line.

www.ElectricalPartManuals.com

Job sequence

When the event occurs, the scheduler starts the jobs associated with the event. The jobs are executed consecutively. A job is executed by executing the function list line-by-line.

In the case of a system event, only one job per HMI device can be configured and executed.

Note

If many jobs are executed within short intervals, time delays can occur. In the case of a cyclic event, make sure that all of the jobs are executed before the next event occurs.

Timer for time-based events

To make dynamic changes to the configured start time during runtime for daily, annual, or one-time events, select an internal tag as a timer. The value of the tag determines the start time for the job during runtime.

Notice

The tag must be of the "DateTime" type.

www . ElectricalPartManuals . com

15.3 Elements

15.3.1 Scheduler

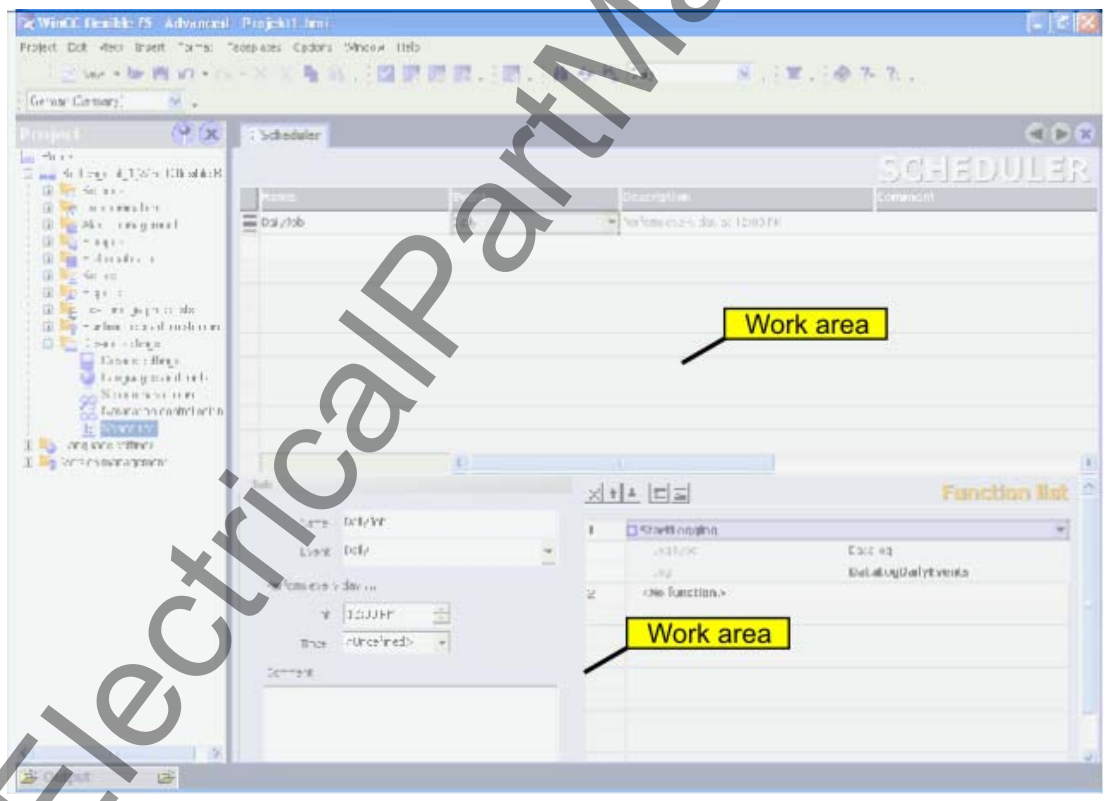
Introduction

In the scheduler, you plan a job by configuring a function list for an event.

Open

Double-click on "Scheduler" to open it in the project view.

Layout



Work area

The work area shows the planned jobs.

www.ElectricalPartManuals.com

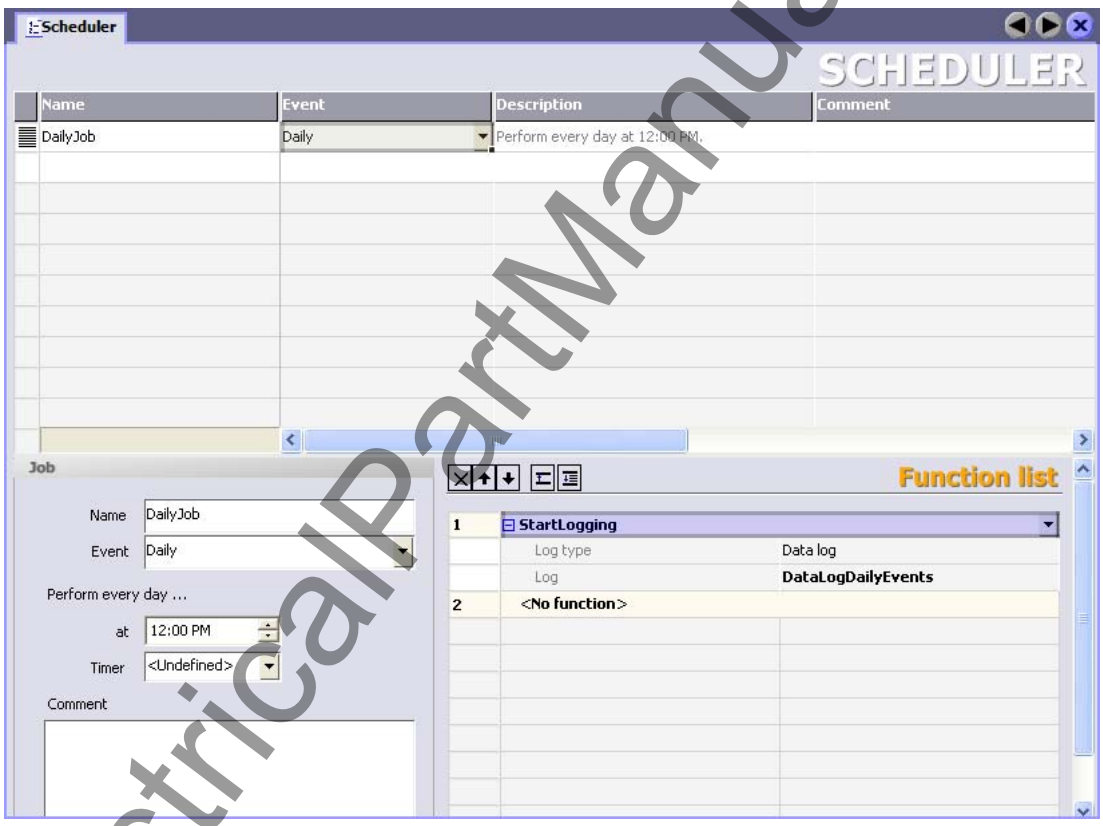
15.3.2 Work area of the "Scheduler" editor

Introduction

The work area shows the planned jobs, which consist of the triggering event and the function list.

Layout

The work area consists of the table of jobs, the properties, and the function list.



The table of jobs shows the job, the triggering event, and additional information. You assign the label and a comment and select the event. The scheduler compiles a description of the job.

The properties also show the job along with the triggering event. The time-based event is specified in the properties.

In the function list you configure the functions or scripts to be executed in the job.

Note

The compiled description provides a written summary of the job including the timing for the job. You can obtain more detailed information using the tooltip function. by moving the mouse pointer over the selected element in the user interface.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

16 Managing project versions

16.1 Applications for project versioning

Principle

A project version is a copy of a project that is saved at a defined storage location by version management. Each project version always represents a specific project status. You can go back to an older version or compare versions with one another.

Application example

You can use project versions in the following situations:

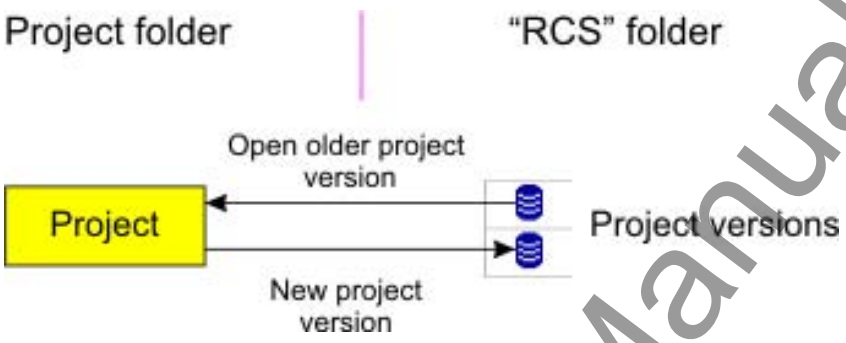
1. To archive accepted reference versions: You can revert to an older version if necessary.
2. To improve an older project version. For example, a customer may wish to correct an error in an older project version which is already being used in runtime. However, the project has been further developed in the meantime. The error is corrected in the older project version. The current project version remains unaffected at this time.
3. Assigning versions to different project states on different HMI devices: When an HMI device fails, you can always transfer the suitable project version to it.
4. Versioning alternative or experimental project configurations separately: Test versions, various device and plant types or special models of a machine.
5. Backing up data to a different medium. Loss of data, e.g. due to defective mass storage, is avoided. The most recent project version of the components for operating the plant is especially affected by this.

www . ElectricalPartManuals . com

16.2 Basics of version management

Definition

Version management distinguishes between the project itself and the project version. A project is a file in the project folder that you can edit in WinCC flexible. A project version is a file at the "RCS" location that is saved there by the version management.



Copies are exchanged between the project folder and the "RCS" location in both directions. When a new project version is created, a copy of the project is saved in a file at the "RCS" location. When an older project version is edited, a local copy is created in the project folder.

Note

Project versions can be distinguished by sequential version numbers. The version numbers are assigned automatically in order to avoid conflicts that might occur in the branches.

Introduction

You are continually developing your project. Changes are being added to changes step-by-step. If you are regularly versioning your project, sequential project versions are being created. All project versions with whole number such as 1, 2, 3 etc., form the trunk of the development.

Several branches may exist as well. The branches, for example, 2.1.1, 2.1.2, 2.1.3, are created when you edit the older project version 2 and version it regularly.

www.ElectricalPartManuals.com

16.3 Trunk

Principle

You create a new project version to record the current state of the project. The new project version is a copy of the current project. The first project version is assigned version number "1".

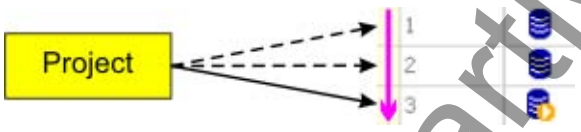
Notice

Once a project version is saved in the version management, it can no longer be changed. Changes are always passed along to the next project version.

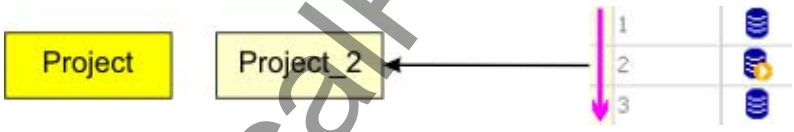
You are continually developing your project in WinCC flexible. Changes you are now making are based on the project state with the version 1. Version 1 is the current version.

The continuing development reaches a new milestone. You create a new project version to record the current state of the project. Version 2 is the current version.

When the next project version is created, the project status is saved as version 3.



Older project version

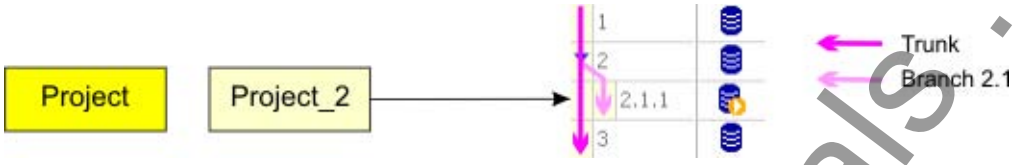


To access an older project state, open the respective project version, for example, version 2. A copy of version 2 is then created in the project folder with the name "Project_2" and opened in WinCC flexible. You can now edit the older project state in "Project_2". The changes are based on version 2. Version 2 is the current version.

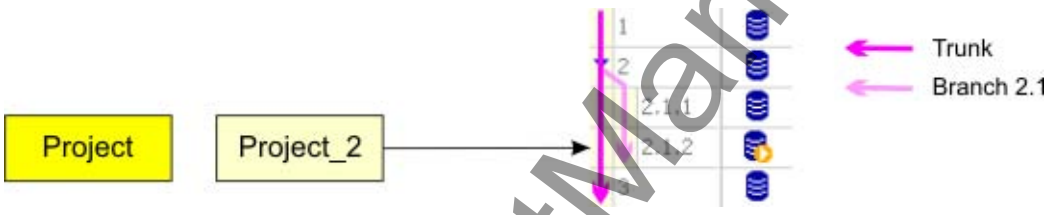
www . ElectricalPartManuals . com

16.4 Branch

Principle



In order to record the project status of "Project_2" in the version management, create a new project version of "Project_2". Since version 3 already exists, the new project version is saved as version 2.1.1. Version 2.1.1 is the current version. The next version is 2.1.2. Now there is a branch based on version 2 in addition to the trunk development.

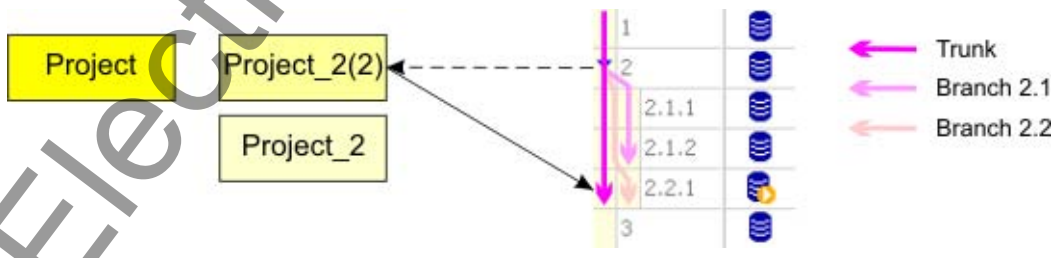


To continue the development of this branch, you always open "Project_2" in the project folder. When you create another new version of "Project_2", it will be saved as version 2.1.2. Version 2.1.1 is now the current version. Branch 2.1 is formed by all project versions, 2.1.1, 2.1.2, 2.1.3 etc.

Note

A branch always begins with a version, for example, version 2. Several branches may arise from the same version. This is why a branch is always assigned an extended version number, for example, 2.**1**. A project version of a branch always requires a two-number extension for its version number, for example, 2.**1**.**2**.

Additional branches



However, when you open version 2 in the version management again, "Project_2" is not overwritten in the project folder. Instead, "Project_2(2)" is saved. In "Project_2(2)" you can edit the state in version 2 once again.

A new branch numbered 2.2 is created when you now create a new project version. Branch 2.1 already exists. The new project version is saved as version 2.2.1.

However, in order to continue to work with branch 2.1, you can open the highest number of this branch, for example version 2.1.2 in the version management. As an alternative, you can open the most recently edited "Project_2" in the project folder.

www.ElectricalPartManuals.com

16.5 Elements

16.5.1 Version management

Introduction

The version management shows the project versions that have been created from the current project. You can create a new project version, open an older project version and compare project versions.

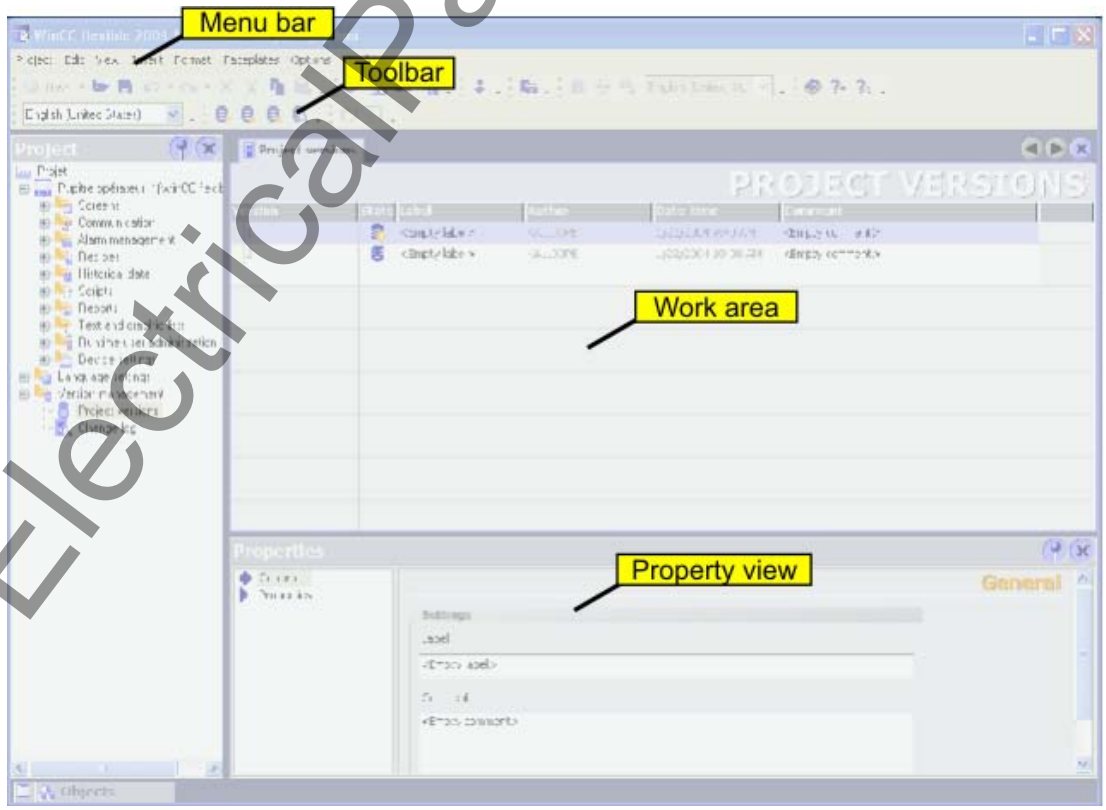
Open

You open the "Project Versions" editor in the Project view by double-clicking on "Project versions" .

Note

If you have moved the location of the project versions in another project, enter the new location in the "Reset path for project versions" dialog that subsequently appears.

Layout



www . ElectricalPartManuals . com

Menu bar

You can start the functions for version management from the menu bar.

Toolbar

You can start the functions for version management in the "Project versions" toolbar. The toolbar is displayed by default. The toolbar can be displayed or hidden from the context menu of the toolbar.

You can also access the version management functions in the context menu of the work area.

Work area

The work area displays the project versions that have been created.

Property view

When a project version is selected, its name and the related comment can be edited in the Property view. The current and the next version numbers are assigned by the version management.

16.5.2 Version Management Work Area

Introduction

The work area shows a table of the project versions that you have created from the current project. You can create new project versions, open older ones and compare two project versions.

Note

The work area always displays all versions of the project, even when you open an older project version.

www . ElectricalPartManuals . com

Layout

The work area contains the Tree view of the project versions. The work area displays the trunk of the project versions as the top-most level. A project version marked with the ⊞ symbol represents the beginning of a branch. You can open the view of a branch similar to a folder in the Windows Explorer.

The icon always shows you the project version on which the current project is based.

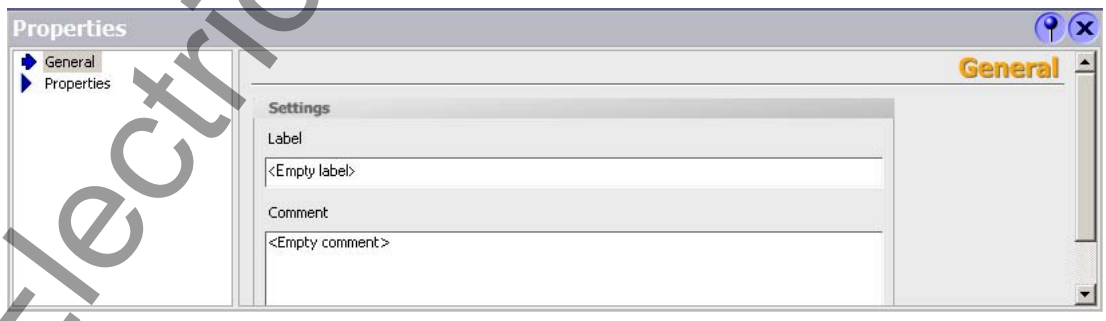
This project version is the current version.

The "Author", "Date/Time", "Version" and "Status" information is assigned by the version management. You can enter the comment and the description. The information is saved at the storage location of the project versions. When you change the comment or the name, the information is immediately updated.

16.5.3 Property view

Introduction

The Properties view shows the "Name" and the comment of the selected project version in the "General" group. The Properties group shows the current and the next version of the project under "Project versions".



Name

- The name can be used to identify different project versions over many projects, for example, "Released 01/15/2001": If you use the same name for a project version in a different project, you make it clear that these two project versions belong together, both sharing a common status of "Released 01/15/2001".

www.ElectricalPartManuals.com

Current version and next version

The information listed under "Current version number" and "Next version number" relate to the current project. "Current version number" shows the version number of the project version on which the current project is based. "Next version number" displays the version number that will be assigned to the next version when you create a new project version.

16.6 Working with project versions

16.6.1 Comparing versions

Principle

The current project with its current status is compared to a project version. This can be the same project version on which the current project is based or a project version with a lower or higher version number.

The project version is opened and compared with the current project status. The results of the comparison are displayed in the "Compare versions" table similar to the change log.

Introduction

The "Compare versions" table shows all the objects that have been created, deleted or modified.

Created: The object exists in the current project but not in the project version.

Deleted: The object exists in the project version but not in the current project.

Changed: The object exists in the current project and in the project version. However, the object properties are different.

When you select a changed object in the "Compare versions" table and this object has been changed in the current project, a second table is displayed. The second table shows which properties were changed for the selected object.

www . ElectricalPartManuals . com

17 Logging changes

17.1 Applications for the change log

Definition

The change log documents all changes made in a project in a continuous table. The table contains the changed objects and object properties.

Application example

1. Certain industrial sectors have a special interest in complete and authentic verification of the entire lifecycle of a product and the production conditions. The evidence of who did what, when, where and why is archived. It can then be documented even years later. One example is the pharmaceutical industry.
2. The American authority FDA (Food and Drug Administration), for example, is responsible for specifying the regulations for food and drugs.

   Not only do the FDA and various technical inspection companies require documentation of project changes, but many other industrial sectors and products are subject to this documentation regime.
3. Engineering businesses process customer orders in their project configuration. The customers often demand changes that go beyond the framework of the contract. The change log helps to document these changes and creates a basis for calculating the additional costs.

www . ElectricalPartManuals . com

17.2 Change log of a project

Principle

Every project has its own change log. Project changes are recorded as long as the change log is enabled.

Recorded project changes

The following project changes are recorded:

- Creating a new change log in WinCC
- Enabling the change log
- Creating an object
- Deleting an object
- Renaming an object
- Changing an object
- Copying an object
- Moving an object
- Saving a project
- Renaming a project
- Changing comments in the change log
- Disabling the change log

The following project changes are recorded only for a project with version management:

- Creating a new project version
- Opening an old project version
- Moving project versions to another storage location

www . ElectricalPartManuals . com

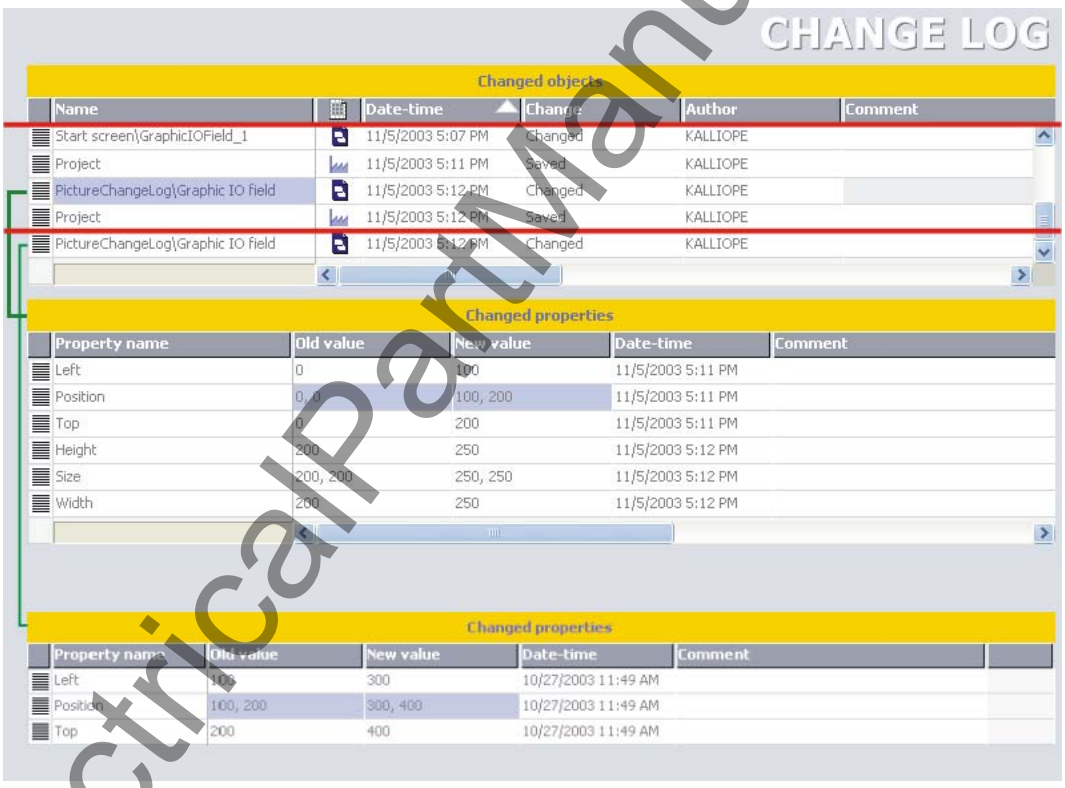
17.3 Change log of a project session

Principle

The "Changed objects" table lists each changed object on a separate line. The "Changed properties" table lists the changed object properties in detail.

Numerous project configuration changes are summarized as a single change.

- In the "Changed objects" table, all project changes to an individual object are summarized on the same line.
- In the corresponding "Changed properties" table, all project configuration changes to a single object properties are summarized on the same line.



www . ElectricalPartManuals . com

Example

When you enable the change log for an object for the first time and then change "GraphicIOField_First", for example, the changed object will be inserted in a new line in the "Changed objects" table.

In the "Changed properties" table the changed object property "Position ", for example, is inserted as the first line and the old value "0", for example, and the new value "100", for example is entered.

Each additional change to the object is recorded in the "Changed properties" table.

- For example, if you change the same object property such as "Position" again, the existing line is changed to the new value, for example, "200."
- When you change another object property for the first time, for example, "Size", the changed object property is inserted as a new line at the bottom of the table and the new and old value is entered.

Note

In a screen the object properties such as "Height" and "Width" are summarized under the object property "Size." A change to the height is recorded as a change to the "Size."

New change section

A project session consist of one or more change sections. All changes to one object are summarized in a line within a change section. A change section is limited by the following actions:

- Open project
- Creating a new project version
- Save project
- Enables the change log

A new change section is then started. A change section ends with the following actions:

- Close project
- Opening an older project version
- Save project again
- Disables the change log

Example

When you open the project, a new project is created. When you change an object the first time, for example "GraphicIOField_First", the changed object will be inserted in a new line in the "Changed objects" table.

Each additional change to the same "GraphicIOField_First" object is registered in the existing line.

When you save the project, a new change section begins. The next change to the same "GraphicIOField_First" object is now registered in a new line. All further changes to the same object are registered in the same line until a new change section begins.

www.ElectricalPartManuals.com

17.4 Change log of a project under version management

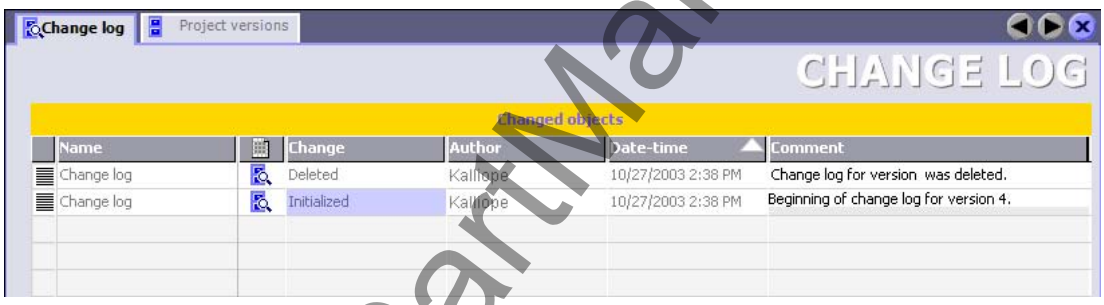
Introduction

The following section describes the special circumstances for a change log when you are using version management for your project. Each project version has its own change log with additional entries.

Creating a new project version

When a new project version is created, the change log is saved along with the current project in the version management. The change log is added as the last entry, "Project Saved." The change log is then deleted in WinCC flexible.

A new change log is created with the new project version.



Opening an older project version

The change log is opened along with the project when an older project version is opened.

All changes to the project version are recorded as long as the change log is enabled.

www . ElectricalPartManuals . com

17.5 Elements

17.5.1 Change log

Introduction

The change log displays configuration changes made in your project. You can see who, changed which objects and object properties, when changes were made, and the corresponding comments.

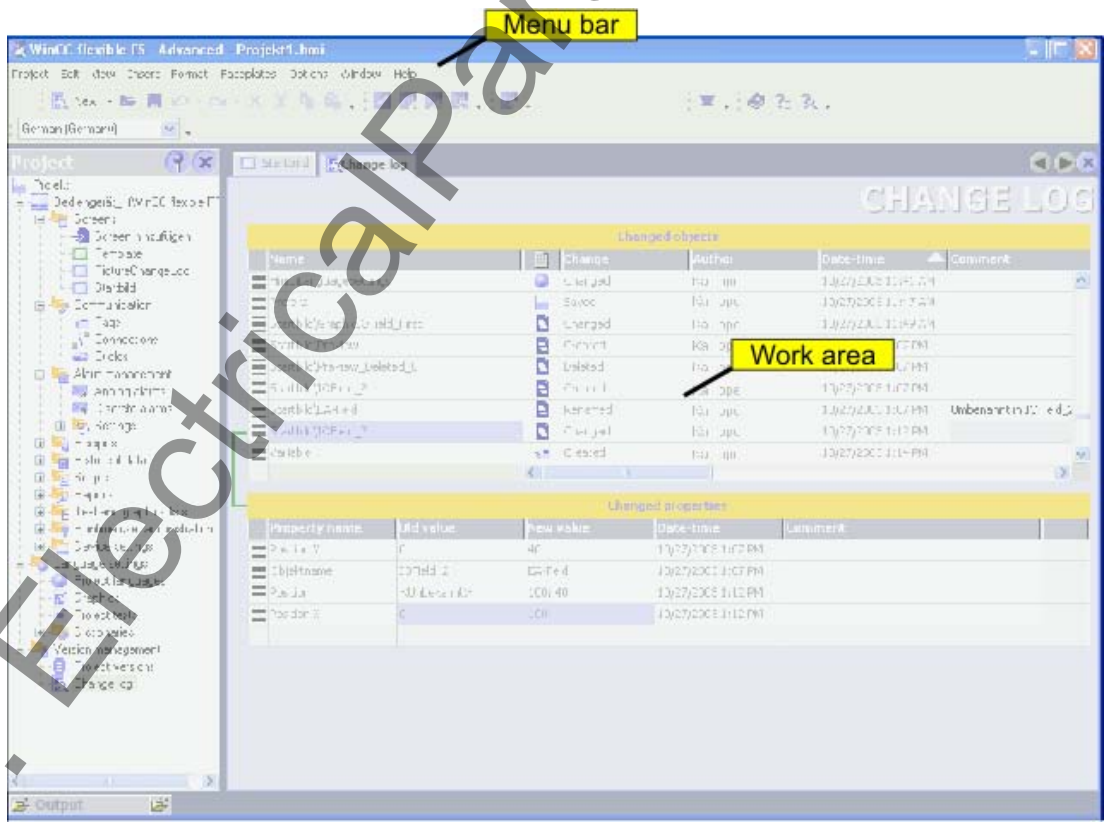
Open

You open the change log in the project view by double-clicking on "Version management\Change log."

Note

To open the change log of an older project version, first open the required project version in version management.

Layout



Menu bar

Under the menu item "Options > Version management" you can enable and disable the change log.

www.ElectricalPartManuals.com

Toolbar

You can open the change log from the "Project versions" toolbar.

Work area

The work area displays the modified objects and object properties.

17.5.2 Change log work area

Introduction

The work area displays the modified objects and object properties in the form of a table.

Layout

The work area consists of the "Changed objects" and "Changed properties" tables.

The "Changed objects" table shows all the objects that have been created, modified, or deleted. If you select a changed object in this table, the "Changed properties" table opens. The "Changed properties" table shows which object properties of the selected object were changed. The object selected in the "Changed objects" table and the "Changed properties" table are connected by a line.

www.ElectricalPartManuals.com

www . ElectricalPartManuals . com

18 Transfer

18.1 Basics

18.1.1 Basic Principles of the Transfer Operation

Transfer

A transfer operation refers to the transfer of a complete project file to the HMI devices where the project is to run.

After you have completed a configuration process, check the consistency of the project by using the menu "Project > Compiler > Check Consistency". After completing the consistency check, the system generates a compiled project file. This project file has the same name assigned to it as the project, however with the extension "*.fwx". Transfer the compiled project file to the configured HMI devices.

The HMI devices must be connected to the configuration computer to transfer the project data. If the HMI device is a PC, it is also possible to perform the transfer operation using data media such as diskettes.

Basic procedure

1. Enter the transfer settings for the individual HMI devices in your WinCC flexible project.
2. Enter the transfer mode on the HMI device where the project is to be transferred.
3. Transfer the compiled project file from the configuration computer to the HMI devices. The project file is transferred to all HMI devices for which the respective check box is selected in the transfer settings.

www . ElectricalPartManuals . com

Transfer mode

The HMI device must be in "transfer mode" for the transfer operation. Depending on the type of HMI device, transfer mode is enabled as follows:

- Windows CE systems

  The HMI device starts up automatically in transfer mode when the device is commissioned the first time.

  The HMI device switches automatically to transfer mode at the start of each additional transfer operation if this transfer option is enabled on the configuration menu of the HMI device.

  If not, restart the HMI device and call the transfer applet on the Start menu, or configure the "Change Operating Mode" system function in your project.

- PCs

  If the HMI device is a PC that does not yet contain a project, you must enable the transfer mode in the "RT Loader" manually before the first transfer operation.

Refer to your product manual for more detailed instructions on setting the transfer mode on the HMI device.

HMI device version

When transfering a project onto the operator device, the system checks if the configured operating system version corresponds with the version on the HMI device. If the system finds different versions the transfer is aborted and a message is displayed. You have following possibilities if the version of the operating system in the WinCC flexible project and on the HMI device are different:

- Update the operating system on the HMI device

  You can find further information in chapter "Operating System Transfer".

or

- Select the corresponding HMI device version in the WinCC flexible project.

  You can find further information in chapter "Project HMI device dependency".

18.1.2 Transfer settings

Introduction

You can enter transfer settings individually for each HMI device of your project. The transfer settings include communication settings and the HMI device selection for the transfer operation.

The "Transfer settings" dialog allows you to enter only those settings that are actually available for the selected HMI device.

www.ElectricalPartManuals.com

Selecting the HMI device for the transfer operation

When a transfer operation is performed, the compiled project file is transferred to all HMI devices of the project for which the respective check box is selected in the transfer settings on the configuration computer.

The relevant check box must be selected in the transfer settings on the configuration computer even if you use the context menu of the HMI device to start the transfer operation for this particular HMI device only.

Transfer modes

Depending on the HMI device, you can use one or more of the following transfer modes:

- Direct connection (USB cable (host-to-host), serial cable)

  Transfer takes place by means of a serial cable or a USB cable connecting the configuration computer and HMI device.

  Note

  Always select the highest possible transmission rate for a transfer operation by means of a serial cable. At lower transmission rates, it can easily take hours for the quantity of data to be transferred.

- Ethernet network connection

  The configuration computer and HMI device are located in a network or are connected point-to-point. The transfer operation between the configuration computer and the HMI device takes place by means of an Ethernet connection.

- MPI/PROFIBUS DP

  The configuration computer and HMI device are in an MPI network or PROFIBUS DP network. The corresponding protocol is used for the transfer operation.

- Http

  The http protocol is used for the transfer operation, for example, via the Internet or an Intranet.

The transfer mode setting for an HMI device is also applied if the HMI device is selected in the project view and one of the commands on the "Project > Transfer" menu is selected (for example, in the case of a back transfer operation or when the operating system is updated on the HMI device).

Transfer destination

On Windows CE HMI devices, you can store the compiled project file to the Flash memory or RAM of the HMI device.

www . ElectricalPartManuals . com

Delta transfer on Windows CE devices

To save time in the transfer, only delta transfers can be performed on Windows CE HMI devices. In the case of a delta transfer, only project data that has changed relative to the data on the HMI device is transferred.

During a delta transfer, it is possible to transfer data to the RAM memory. This is advisable if a new configuration is to be tested without loss of the old configuration. After a shutdown/restart of the HMI device, the configuration transferred to the RAM is lost and the configuration stored in the Flash memory is again applicable.

"Delta transfer" is the default setting for Windows CE HMI devices. You can change this default setting in the transfer settings to force the entire project to be transferred. It may be necessary to transfer the entire project, for example, if an executable project file no longer exists on the HMI device due to a malfunction or inconsistency after the delta transfer.

Note

If the HMI device is a PC, the complete data file is always transferred.

Upload

When transferring, you can transfer the compressed source data file along with the compiled project file to the HMI device. The compressed source data file is stored on the HMI device with the same name as the project but with the extension *.pdz added.

If necessary, you can back transfer this source data file onto any configuration computer. Thus, you can analyze and continue processing the original project on a computer other than the original configuration computer at a later time.

Notice

The source data file can be stored on the HMI device for back transfer purposes only if sufficient memory is available externally on the HMI device.

Overwriting the password list and recipes

When the compiled project file is transferred, the password list and recipes present on the HMI device are overwritten by the corresponding configuration data. Consequently, the option exists to create recipes and passwords as part of the project, which are then available on each HMI device to which the project has been transferred. During transfer, the compressed recipe data is transferred to the HMI device. When the transfer has ended, runtime starts on the HMI device and decompresses the recipe data. It then imports this into the project. After import, a system alarm is generated. Prior to concluding import you must not export any recipe data. You can only start an export or import of recipe data on the HMI device once the system alarm for a successful import / export has been issued.

To prevent overwriting existing passwords and recipes, clear the respective check box. Another option for retaining the existing password list and recipes is to first back them up from the HMI device. Once the transfer operation has been performed, the password list and recipes can then be restored from the backup.

www.ElectricalPartManuals.com

18.1.3 Back transfer of projects

Introduction

When transferring, you can transfer the compressed source data file along with the compiled project file to the HMI device. This source data file is required for the project to be back transferred from the HMI device to a configuration computer.

Use for back transfer

Normally, only the executable project is transferred to the HMI device during a transfer operation. The original project data remain on the configuration device and are thus available to develop the project further in future or for error analysis.

However, on Windows CE devices with an external storage medium and on PCs, you can store not only the compiled project file but also the compressed source data file for the project. This data file can be used at a later time to recover the project from the HMI device or device by back transferring the source data file to a configuration computer.

Advantage:

The back transfer operation enables you to subsequently perform analyses and make changes to an existing project even if the original configuration device is not available or the source file (*.pdf) for the project is no longer available on the configuration device.

Note

You can also use WinCC flexible to transfer the source data file of a ProTool V6.0 project back from the HMI device onto a configuration computer. You can then perform a migration of the ProTool project to a WinCC flexible project.

The source data of a ProTool project which was configured for an operating device not supported by WinCC flexible must be transferred back to a configuration computer with ProTool. Save the ProTool project. Then execute a migration using WinCC flexible.

Requirements for back transfer

- The source data file can only be transferred to the HMI device as part of the transfer operation for the compiled project file. The source data file is transferred along with the compiled project file to the HMI device if the "Enable back transfer" check box is selected in the transfer settings for the respective HMI device.
- There must be sufficient memory available on the HMI device to store the compressed source data file. If the source data file for the back transfer operation is provided by a Windows CE device, this device must have an external memory card. If the HMI device does not have a memory card or if there is insufficient memory space, the transfer is terminated. However, the compiled project file is transferred in its entirety beforehand so that runtime can be started with the transferred project data.

  If the source data of a large project should be stored for back transfer and an Ethernet connection is available to the operating device, you can select a network drive as the storage location rather than the memory card of the operating device. This avoids problems with the storage location.

www . ElectricalPartManuals . com

- If there is no project opened in WinCC flexible, you must select the HMI device on which the source data file for the back transfer operation is located and the loading method in the "Communication settings" dialog prior to carrying out the back transfer operation.

  If a project is open in WinCC flexible, the back transfer operation takes place from each selected HMI device. In this case, the transfer mode selected for this HMI device in the "Transfer Settings" dialog in WinCC flexible is applied.

Transfer and back transfer

When a source file is included in the transfer operation, the project is compressed from the source format (*.pdf) and transferred as a *.pdz file to the external storage medium of the HMI device or directly to the PC.

In the case of a back transfer operation, the *.pdz file is saved on the configuration computer. If a project was open in WinCC flexible during the back transfer, you are prompted to save and close it. Then, the project back transferred is decompressed and opened in WinCC flexible. When saving the project, you must assign a name for the back transferred project.

Caution

WinCC flexible cannot check whether the source data file on the operating unit actually belongs to the project running on the device. If you have performed a transfer operation in the interim that did not include the source data file, old project data may still be on the HMI device. Under certain circumstances, the data will then no longer match the project that is currently running.

Note

Use the back transfer process preferably for small and medium sized configurations in order to keep transfer times as short as possible.

You have the following options when there are numerous project files: Transfer the project file as a compressed *.arj file onto a CF card, for example, using the backup function of the project manager.

18.2 Managing Files on the HMI Device

18.2.1 ProSave

Introduction

The ProSave service tool is supplied with WinCC flexible. The functionality of ProSave is integrated into the WinCC flexible user interface on the programming device. ProSave can also be installed as a stand-alone program on a computer where WinCC flexible is not installed ("stand-alone operation").

www.ElectricalPartManuals.com

Functional scope

ProSave provides all of the functions needed to transfer files to the HMI device.

- Data backup and restoration of backed-up data
- Operating system update for Windows CE-based devices
- Transferring authorizations
- Installing and uninstalling drivers and options as well as information on installed options and options that can be installed on an HMI device
- Communication settings

Integrated operation on the configuration computer

ProSave is installed on the configuration computer as part of a standard WinCC flexible installation. The complete functional scope of ProSave is integrated within WinCC flexible on the "Project > Transfer" menu.

Stand-alone operation on a computer

ProSave can also be installed on a computer from the WinCC flexible CD and used without WinCC flexible being installed (for example, for service purposes).

When replacing a device, for example, you can use ProSave to back up a project from the original HMI device and restore it on a replacement device without having WinCC flexible installed.

If you are using ProSave outside of WinCC flexible, you have the option to change the user interface language. To select a language, use the "Language" menu command in ProSave. ProSave must be restarted for the language switch to take effect.

18.2.2 Backup of HMI data

Introduction

The data on an HMI device should be backed up at regular intervals.

A data backup allows you to quickly resume operation after a system failure or when a device was replaced. The backup data are simply transferred to the new HMI, and thus restore the original state.

Data backup using WinCC flexible or ProSave

You can use WinCC flexible and a programming device which is connected to the HMI to backup and restore all HMI data.

If the computer does not have WinCC flexible installed, you have the convenient option of using ProSave to perform a centralized backup.

www.ElectricalPartManuals.com

Scope of data backup

The backup and restore operation depends on the type of HMI device and can include the following project data:

- Complete backup (depending on the HMI device: Runtime, firmware, operating system image, configuration, recipes, passwords, and settings, but not authorizations)
- Recipes only
- Passwords only

A backup file with the extension *.psb is generated when you backup the HMI data.

The backup can be made to any memory medium, such as a data server, if an appropriate connection exists between the HMI device and the memory medium.

Note

Only use the restore function for project data on operating devices configured using the same configuration software.

If, for example, WinCC flexible recipe data is restored on a device configured using ProTool, the Flash memory can no longer be read. Delete the Flash memory, in this case, and transfer the ProTool project again.

Note

Note the following when performing a complete data file backup and restore operation for Windows CE devices:

Authorizations are not backed up!

When a complete data restoration is carried out, all of the data that were previously on the device, including authorizations and the operating system, are irrevocably deleted.

If an interruption occurred while data was being restored, you must first reload the operating system onto the HMI device via the serial port using the "bootstrap" mechanism before resuming the data restoration.

All installed options are also backed up, but not the associated authorizations. Basically, all the data for the option that are still available after "POWER OFF" are backed up.

Note

If at all possible, you should use USB or Ethernet to back up and restore data because use of these interfaces will result in the shortest transfer times.

Note

For Windows CE devices, a direct data backup can be performed from the device to an external storage medium, namely a CF card or PC card. For additional information, refer to the relevant operating instructions.

www.ElectricalPartManuals.com

18.2.3 Updating the operating system

Introduction

If the operating system version on a Windows CE device is not compatible with the configuration, the transfer of the configuration is cancelled. A message appears indicating the operating system must be updated.

Updating the operating system

Note

The operating system can only be updated on Windows CE-based devices

You can use WinCC flexible and a computer connected to the HMI to update the HMI operating system.

If the computer does not have WinCC flexible installed, you have the option of using ProSave to update the operating system of the HMI device.

Notice

Do not use a serial connection to transfer the operating system. The transfer operation can take up to an hour if a serial cable is used.

When an operating system is updated, all of the data on the destination device, including authorizations, are deleted. Therefore, use the "Authorizations" function to transfer the authorizations back onto the license diskette beforehand.

If you want to continue to use any user data (such as passwords and recipes) stored in the internal Flash memory once the operating system is updated, you must export the data to an external data memory beforehand and reload them onto the HMI device following the update.

"Bootstrapping"

If the operating system update was terminated prematurely, an operating system will no longer be available on the HMI device. A "bootstrap" operation is then the only option available for loading an operating system.

When an operating system is updated, the communication between the configuration computer and the HMI device takes place by means of the operating system of the HMI device. During a "bootstrap" operation, however, the configuration computer communicates with the boot loader of the HMI device. In this case, communication is only possible by means of a serial connection. The bootstrap operation can take time.

- Once the "bootstrap" operation has been started in WinCC flexible, power on the the HMI device must be switched off and on again (booted), so that the HMI device can communicate by means of the boot loader.

www.ElectricalPartManuals.com

18.2.4 Transferring license keys

Transferring license keys

The installation of some WinCC flexible runtime options requires license keys provided on a license key disk. You transfer the required license keys to the HMI device during installation by means of the configuration computer.

If necessary, you can transfer the license keys back onto the license key diskette, for example, to use the license keys on another HMI device.

Caution

To prevent irrevocable loss of license keys, you must transfer the license keys for an HMI device back onto the license key diskette in the following cases:

- Before updating the operating system on a Windows CE HMI device
- Before restoring a complete data file from the backup

License keys on an HMI device are not backed up as part of a backup operation.

18.2.5 Installation of options

Available options

You can install additional options supplied with WinCC flexible on HMI devices at a later time. Likewise, you can install options purchased separately from WinCC flexible at a later time.

The HMI device type determines which options can be installed.

For an overview of available options, refer to "Introduction to WinCC flexible."

www . ElectricalPartManuals . com

19 Integration of WinCC flexible in STEP 7

19.1 Basics

19.1.1 Basic principles of integration in STEP 7

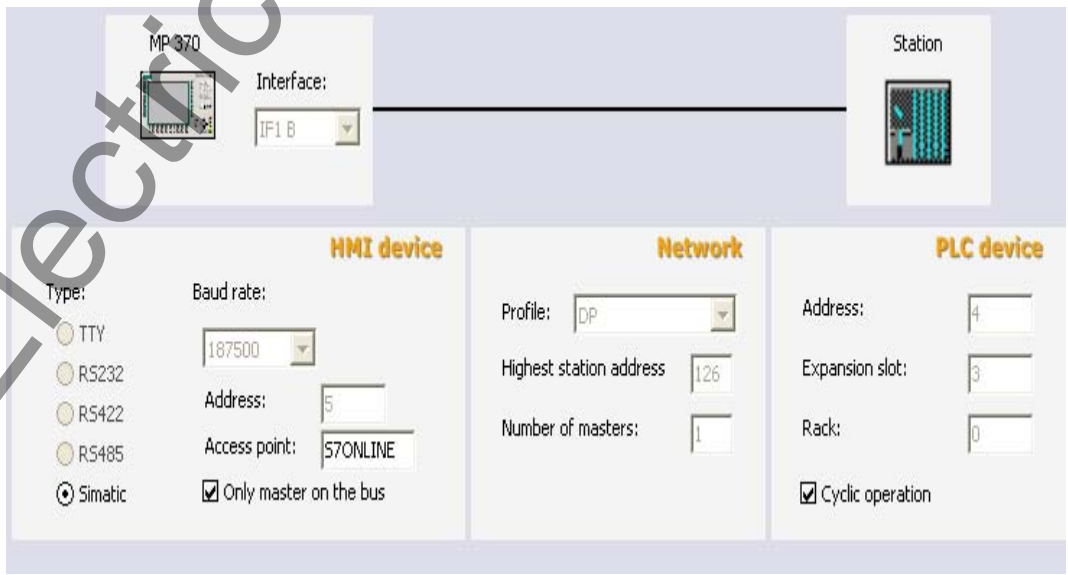
Introduction

If you are using a SIMATIC controller and have installed the STEP 7 configuration software on your system, you can integrate WinCC flexible in STEP 7.

Advantages of integrating in STEP 7

During integrated configuration, you access STEP 7 configuration data that you created when you configured the controller with STEP 7. This gives you the following advantages:

- You can use the SIMATIC Manager as a central point for creating, processing, and managing SIMATIC controllers and WinCC flexible projects.
- The communication parameters of the PLC are preassigned when the WinCC flexible project is created. When a change takes place in STEP 7, the communication parameters are updated in WinCC flexible.



Connection parameters created by the system during STEP 7 integration: Network parameters and partner parameters are preassigned

www . ElectricalPartManuals . com

- When configuring tags and area pointers, you can access the STEP 7 symbols directly in WinCC flexible. In WinCC flexible, simply select the STEP 7 symbol to which you would like to link a tag. Symbol changes made in STEP 7 are updated in WinCC.
- You assign a symbolic name once in STEP 7 and use it in STEP 7 and WinCC flexible.
- ALARM_S and ALARM_D alarms configured in STEP 7 are supported in WinCC flexible and can be output on the HMI device.
- You can create a WinCC flexible project without integration in STEP 7 and integrate it in STEP 7 at a later time.
- You can remove an integrated project from STEP 7 and use it as a standalone project.
- In a STEP 7 multiproject, communication connections can be configured across projects.

Installation requirements

A specific installation sequence must be followed to integrate WinCC flexible in STEP 7. You must first install the STEP 7 software, and then WinCC flexible. When installing WinCC flexible, it detects an existing STEP 7 installation and automatically installs the support for integration in STEP 7.

For user-guided installation, the "Integration in STEP 7" option must be activated.

If WinCC flexible is already installed and STEP 7 is installed subsequently, WinCC flexible must be uninstalled and reinstalled once the STEP 7 installation is complete.

19.1.2 Working with the SIMATIC Manager

Introduction

When you are working with WinCC flexible integrated in STEP 7, you can use the SIMATIC Manager for your WinCC flexible projects. In STEP 7 projects, the SIMATIC Manager is the central point for managing your projects, including your WinCC flexible projects. The SIMATIC Manager enables you to access the configuration of your automation systems and the configuration of the operator control and monitoring layer.

Requirement

WinCC flexible is integrated in SIMATIC STEP 7.

Working with the SIMATIC Manager

In integrated projects, the SIMATIC Manager provides the following options:

- Create an HMI or PC station with WinCC flexible Runtime
- Insert WinCC flexible objects
- Create WinCC flexible folders
- Open WinCC flexible projects
- Compile and transfer WinCC flexible projects
- Start WinCC flexible Runtime

www.ElectricalPartManuals.com

- Export and import texts for translation
- Specify language settings
- Copy or overwrite WinCC flexible projects
- Archive and retrieve WinCC flexible projects within the framework of STEP 7 projects

19.1.3 Working with HW Config

Introduction

The HW Config editor is provided in STEP 7 for configuring and assigning parameters to the hardware. Use drag-and-drop operations to assemble the required hardware. A catalog is provided for selecting the hardware components. During configuration, a configuration table with the address parameters is automatically created. During subsequent editing in STEP 7 or WinCC flexible, the system accesses this configuration table and accepts the prepared parameters.

Using HW Config

You use HW Config to create the hardware configuration for new stations or to add required modules to existing stations. HW Config provides a catalog with the available modules and preconfigured components and stations. HW Config checks the usability of the objects you wish to insert. Thus, unusable or illegal objects cannot be inserted. Edit the properties of an inserted object directly in HW Config. Open the context menu of the object and select "Object Properties". Edit the object properties directly in the dialog that appears.

For example, you can create a PC station in the SIMATIC Manager. Open the station for configuration in HW Config. Insert a WinCC flexible runtime application. Select a communication interface and insert it. Edit the settings for the communication interface in HW Config. The WinCC flexible runtime application will not open via HW Config. To open it, use SIMATIC Manager.

For more information, consult the HW Config documentation.

19.1.4 Configuring connections

Introduction

Data exchange between WinCC flexible and the automation layer requires connections for communication to take place. In integrated projects, you can create connections with the following applications:

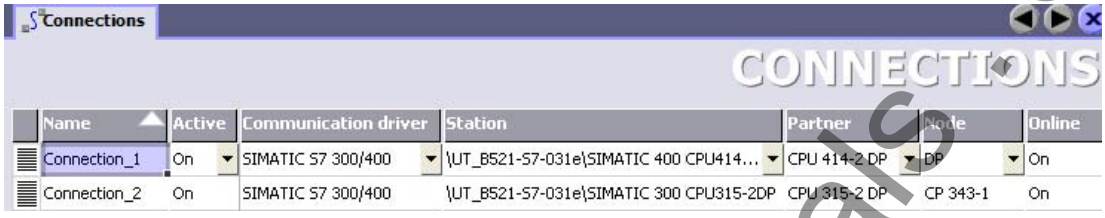
- WinCC flexible
- NetPro

This configuration can be made with either WinCC flexible or NetPro.

www.ElectricalPartManuals.com

Using WinCC flexible

You can create new connections or edit existing ones. In integrated projects, the "Station," "Partner," and "Nodes" columns are also provided in the editor for connection configuration.



When creating a connection, select the station, partner, and connection node from selection lists. The required connection parameters are automatically accepted in STEP 7. The project has to be saved after configuration has been completed. Connections which you configure in WinCC flexible are not transferred to NetPro and can only be edited with WinCC flexible.

Using NetPro

NetPro is recommended for use with larger projects. In NetPro, you configure the connections on a graphically supported interface. When you start up NetPro, the devices and subnets in the STEP 7 project will be displayed. NetPro has a catalog of network objects that you can use to insert additional devices or subnets. In integrated projects, this catalog also includes the SIMATIC HMI station object. You insert objects from the catalog in the work area of NetPro using a drag-and-drop operation. Drag and drop individual stations to connect them to the subnets. Use Properties dialog boxes to configure the connection parameters of the nodes and subnets. You then save the configuration in NetPro to update the data management in the WinCC flexible project. Connections which you configure with NetPro can only be read in WinCC flexible. In WinCC flexible you can only rename the connection, enter a comment for the connection and set the connection "Online". Editing of the connection itself is carried out exclusively with NetPro.

Note

Subnet properties, such as the data transmission rate, are set automatically in STEP 7 for all nodes in a subnet. If you create or modify the subnet properties yourself, you must ensure that these settings are applied for each node in the subnet. You can find more information on this topic in the NetPro documentation.

Note

If a new HMI station is set up in STEP 7, the MPI/DP nodes are set to MPI and Address 1 by the system. If the HMI station is not networked and the HMI station should be networked via a different substation type, the connection parameters must be changed in NetPro or in the HW configuration.

www . ElectricalPartManuals .com

19.1.5 Working with objects

Introduction

Perform the following steps to create an integrated WinCC flexible project:

- Create a new HMI station in the SIMATIC Manager
- Integrate a WinCC flexible project in STEP 7

Creating an HMI station

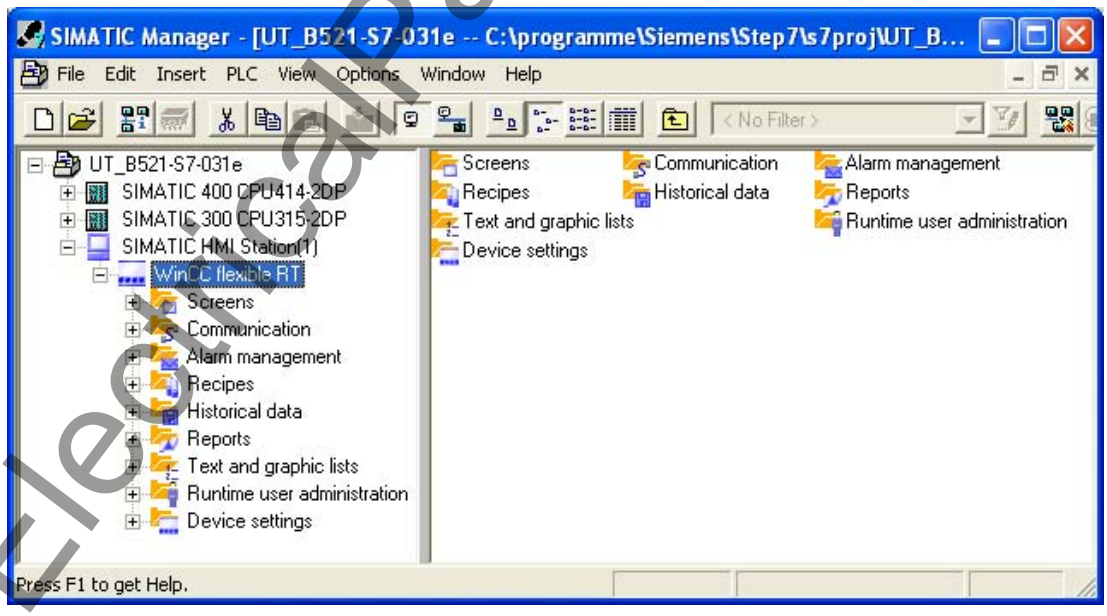
Creating an HMI station in the SIMATIC Manager basically creates a new WinCC flexible project.

Inserting multiple HMI devices in a WinCC flexible project

If you require multiple HMI devices in a WinCC flexible project, you must insert the HMI devices in the project in WinCC flexible.

Inserting WinCC flexible objects

Once a WinCC flexible project has been integrated in STEP 7, the project is displayed in the project window of the SIMATIC Manager. A WinCC flexible project is displayed in the project window of the SIMATIC Manager in the same way as in the project window of WinCC flexible. If you select a WinCC flexible element in the project window, the objects of the WinCC flexible project are displayed in the work area.



From here, you can open existing projects or create new ones. If you create or open a WinCC flexible object in the SIMATIC Manager, WinCC flexible is automatically started for editing the object.

Select, for example, the "Screens" element and create a WinCC flexible screen directly in the SIMATIC Manager. The new screen will be created and opened immediately for editing in WinCC flexible.

www . ElectricalPartManuals . com

Representing WinCC flexible objects

Global project elements that cannot be edited in the SIMATIC Manager are not displayed; examples of such elements include version management and language settings.

Data that you edit in WinCC flexible with a table editor are displayed as symbols in the SIMATIC Manager. Opening such symbols via the SIMATIC Manager causes WinCC flexible to open for editing the data. For example, if you select the "Tag" element, a symbol for all WinCC flexible tags will be displayed in the work area of the SIMATIC Manager. The individual WinCC flexible tags are not displayed in the SIMATIC Manager. If you create a new tag in the SIMATIC Manager, it will be created in WinCC flexible and opened for editing in WinCC flexible.

For more information about STEP 7, consult the SIMATIC Manager documentation.

19.2 Configuring communication settings

19.2.1 Configuring communication settings with routing

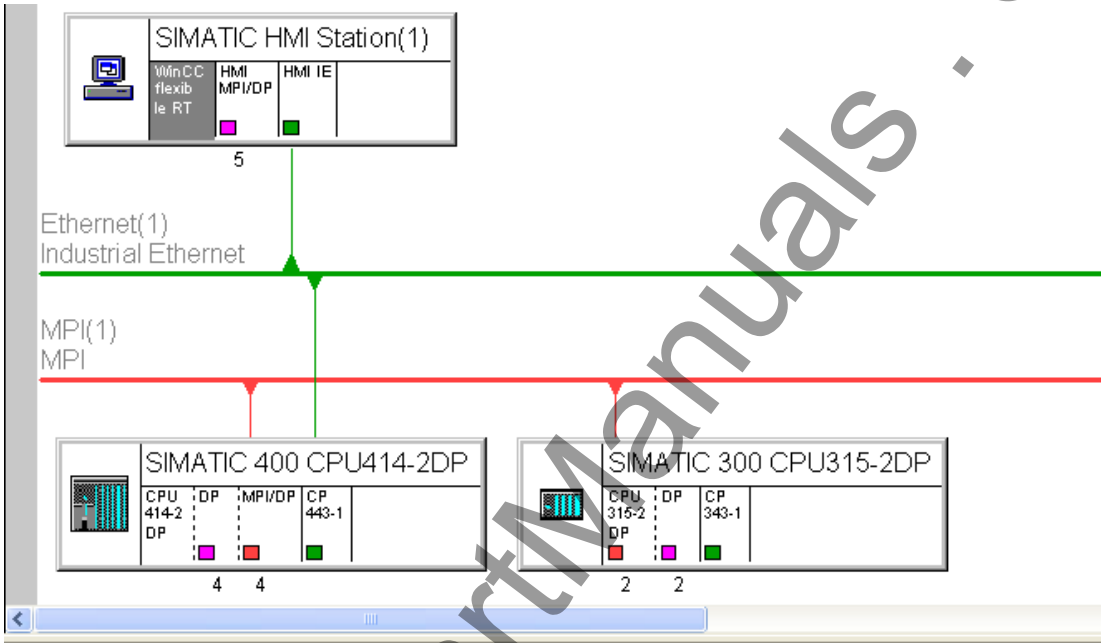
Introduction

If all stations in an automation system are not connected to the same bus (subnetwork), these stations cannot be accessed directly online. To establish a connection to these devices, a router must be interposed. In this case, a SIMATIC station can also act as a router if it has appropriate interfaces with the difference subnetworks. Modules with communication capability (CPUs or CPs) used to establish gateways between the subnetworks must have routing capability.

www . ElectricalPartManuals . com

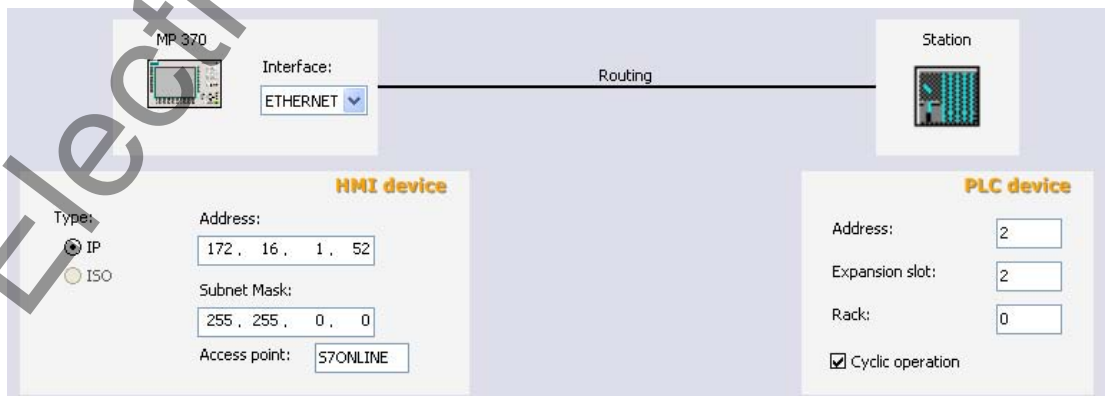
Routing connection

To create a routing connection, all communication partners must be configured and loaded in a STEP 7 project.



Local ID	Partner ID	Partner	Type	Active connection partner	Subnet
S7 connection_1		SIMATIC 400 CPU414-2DP / CPU 414-2 DP	S7 connection	Yes	Ethernet(1) [IE]
S7 connection_2		SIMATIC 300 CPU315-2DP / CPU 315-2 DP	S7 connection	Yes	MPI(1) [MPI]
S7 connection_3		SIMATIC 300 CPU315-2DP / CPU 315-2 DP	S7 connection	Yes	MPI(1) [MPI]

In the figure above, a routing connection is established between the SIMATIC HMI station(1) and the SIMATIC 300 automation device. The SIMATIC 400 automation device is acting as a router. In integrated projects, this type of routing connection can be established directly. This is done by setting up a connection in the SIMATIC HMI station and directly selecting the SIMATIC 300 automation device as the connection partner. The routing connection is detected automatically by the system. The connection is displayed as a routing connection in the connection properties in WinCC flexible.



A routing connection between a SIMATIC HMI station and an automation device can only be created in an integrated project.

www . ElectricalPartManuals . com

19.2.2 Project transfer via S7 routing

Introduction

With WinCC flexible 2005, you can load a WinCC flexible project from a configuration computer to an HMI device over different subnetworks. To establish a connection between different subnetworks, a router must be interposed. In this case, a SIMATIC station can act as a router if it has appropriate interfaces to the different subnetworks. Modules with communication capability (CPUs or CPs) used to establish gateways between the subnetworks must have routing capability.

To transfer a project, the WinCC flexible Engineering Station must be connected to an MPI bus or to a PROFIBUS. The HMI device to which the project is to be transferred must also be connected to an MPI bus or to a PROFIBUS.

The routing connection for the transfer is independent of the connection configuration between the HMI device and the automation device in your WinCC flexible project. The connection described in this chapter is only for the transfer of a WinCC flexible project to an HMI device over a routing connection.

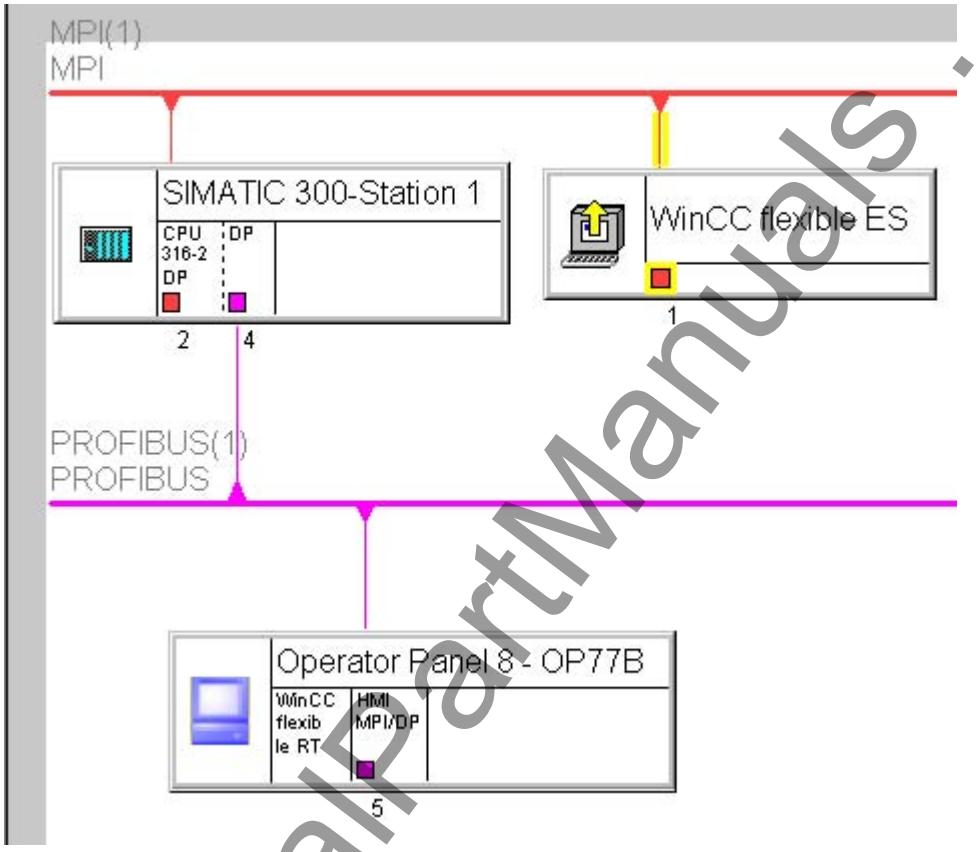
Note

Please see the technical documentation for the respective component to determine whether a component can be routed. Alternatively, open the object properties for the component in NetPro or in HW Config. The "General" tab contains a short description of the properties.

www . ElectricalPartManuals . com

Routing connection for the transfer operation

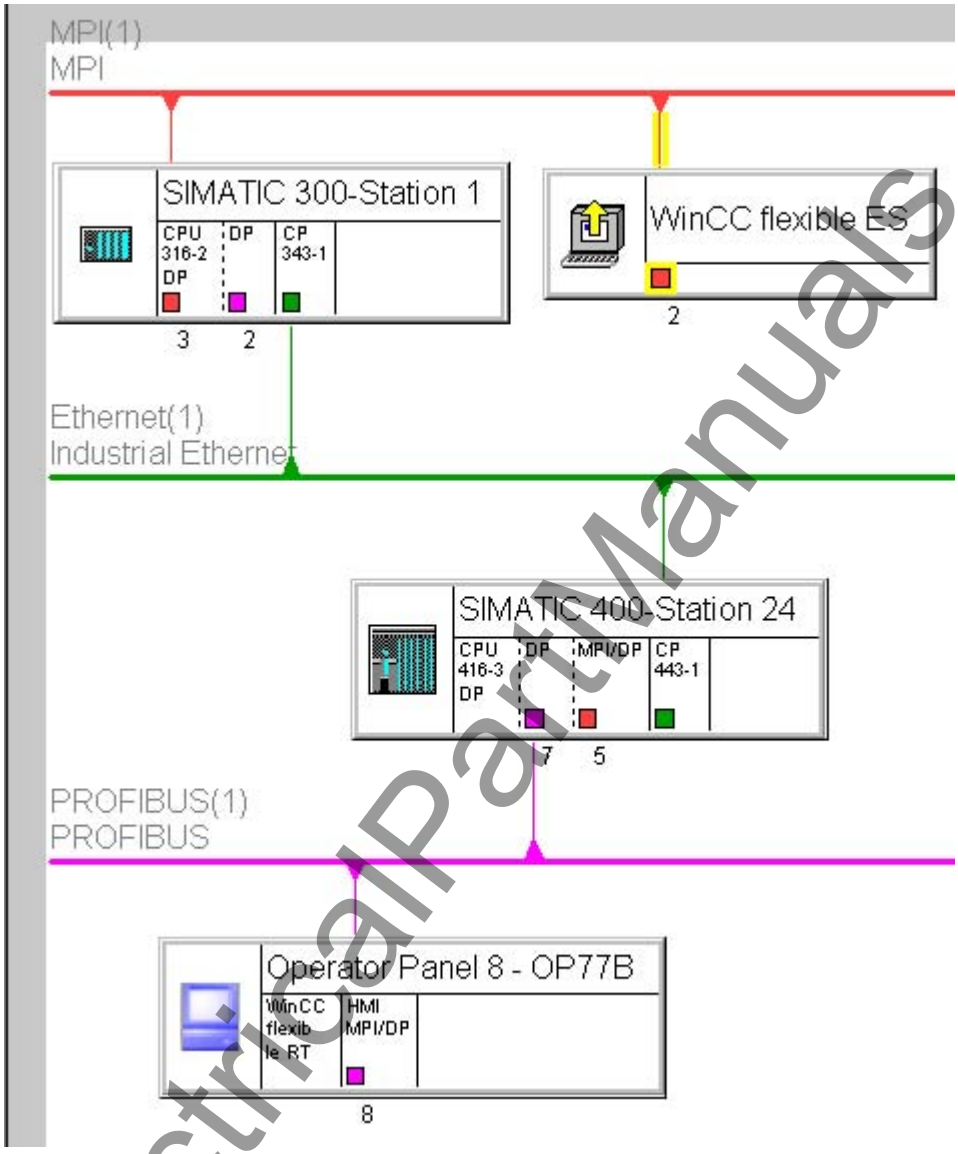
To create a routing connection, all stations must be configured and loaded within a STEP 7 project. The target device cannot be initialized over the routing connection.



In the above figure, a routing connection was established between the "WinCC flexible ES" WinCC flexible Engineering Station and the "Operator Panel 8 - OP77B" HMI device. The "SIMATIC 300-Station 1" automation device functions as the router. You configure the transfer connection between the concerned devices with NetPro. The interface of the configuration computer must be assigned. The association is indicated by the yellow connection line to the subnetwork and the yellow arrow in the symbol of the station. After the configuration in NetPro, save and recompile the project.

www . ElectricalPartManuals . com

A routing connection for the transfer can also be established over several routing partners.



Requirements for routing over several stations:

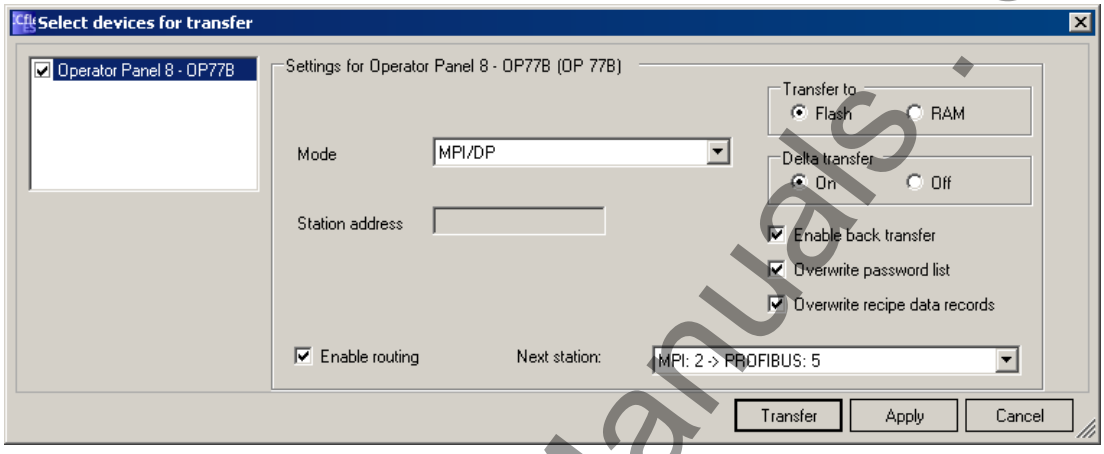
- The WinCC flexible Engineering Station must be connected to an MPI bus or to a PROFIBUS.
- The HMI device to which the transfer operation should take place must be connected to an MPI bus or to a PROFIBUS.

Interposed routing partners can also be interconnected via another type of bus.

www . ElectricalPartManuals . com

Starting the transfer in WinCC flexible

After configuration in STEP 7 is complete, open the HMI station in WinCC flexible. To trigger the transfer, select the "Project ► Transfer ► Transfer Settings" menu command.



"MPI/DP" must be set in the "Mode" field.

The "Enable routing" box must be checked.

The "Next station" field shows the bus type of the next and last connection and the network address of the next routing partner and the target device. Any potential intermediate routing partners are not shown here.

If you click the "Transfer" button, the transfer will start right away.

The routing settings are only made available, if you set the bus type to "MPI/DP" under Mode. If the settings for the routing are not displayed, the system cannot identify a persistent routing connection. Check the settings and the network addresses of the concerned stations. The configured parameters must match the configuration of the stations on the system. Rebuild the WinCC flexible project via "Project ► Compiler ► Rebuild all" to update the connection data from STEP 7 and WinCC flexible.

A routing connection for the transfer can only be established in an integrated project.

Note

The routing transfer to PC-based HMI devices with enabled station manager is not possible.

www . ElectricalPartManuals . com

19.3 Tag configuration

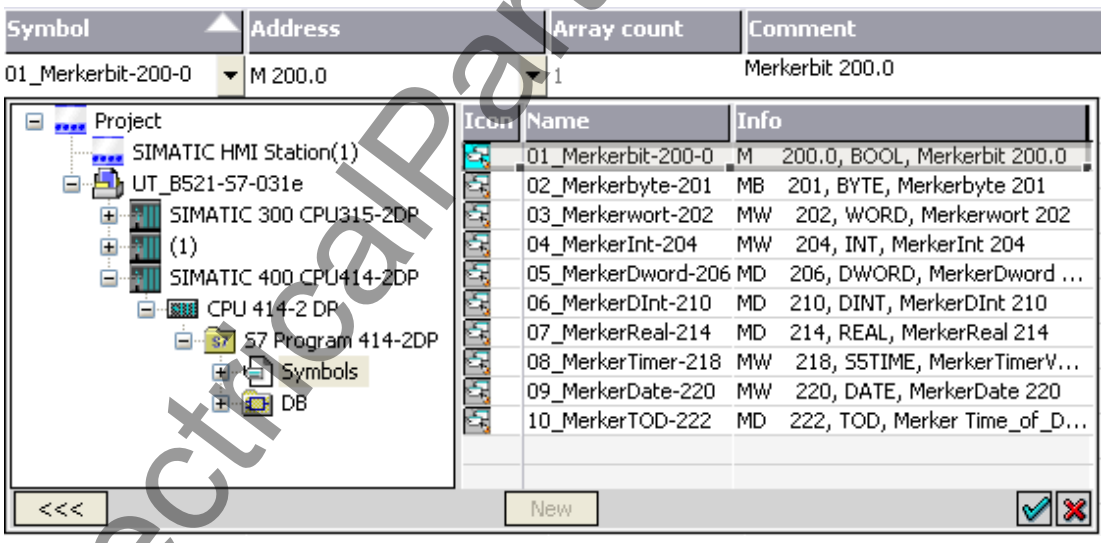
19.3.1 Configuring tags with the Tag editor

Introduction

To simplify editing, the absolute addresses of operands have symbolic names (symbols) in STEP 7. These symbols and their associations are listed in a symbol table. The symbol selection also enables direct access to symbols within data blocks (DB). In integrated projects, WinCC flexible tags are connected directly to the symbols from the STEP 7 project. The associated operands are automatically assumed.

Accepting tags from STEP 7

To accept tags from STEP 7, open the tag editor in WinCC flexible. A "Symbol" column is added to the Tag editor in integrated projects. Insert a new tag in the tag editor. Position the mouse pointer over the field in the Symbol column and click to display the selection button. Press the selection button to open the Selection dialog and navigate to the S7 program in the required controller. Select the required symbol from the symbol list or from a data block.



Click on the command button. The symbolic name from STEP 7 will be accepted as the tag name. The relevant data from the symbol table or data blocks will be integrated in the WinCC flexible tag.

The tag names, transferred from STEP 7 to the WinCC flexible project, are generated from the components of the general STEP 7 symbol. The tag name "Motor.Speed" is derived from "Motor.Speed" for example.

For unique identification, an index starting with "1" is assigned to identical tags. Non-supported characters in a tag name are replaced by an underscore ("_").

www.ElectricalPartManuals.com

Transferring an array from STEP 7

If you are using a SIMATIC S7 300, SIMATIC S7 400, or a SIMOTION controller, you can accept entire arrays from STEP 7 in addition to tags.

If you are using the SIMATIC 300/400 control protocol and you would like to accept arrays in WinCC flexible, proceed as follows:

1. Create a new tag in WinCC flexible.
2. Position the mouse pointer and click in the "Symbol" field of this tag; press the button that appears to open the selection dialog.
3. Navigate to the required controller and select the array you want to accept. A tag group corresponding to the number of array elements will be created.

Changing a connection

When you make changes to a connection, e.g., by changing a node, a program, or a station, the symbol association of a tag is not lost. The tag association is automatically reassigned to the STEP 7 symbol.

If a tag can no longer be assigned because the address or symbol does not exist, you have the following options:

- Save the association
  The tag will be labeled defective. The tag in question must be connected manually.
- Separate tag from symbol
  This tag will no longer be compared automatically with the STEP 7 symbol.

19.3.2 Connecting a tag via the application point

Introduction

Connections between WinCC flexible objects and operands in the control layer are configured simply by selecting the symbols in the connected controller.

Accepting tags from STEP 7

All WinCC flexible objects that can be connected to a tag can be used to accept tags via the application point. For example, when you make an IO field dynamic, you open the selection dialog for the tag in the Properties window of the IO field. Navigate to the S7 program in the required controller. Select the required symbol from the symbol list or from a data block. Click on the command button. The system automatically creates a WinCC flexible tag and connects it to the associated operand in STEP 7.

The symbolic name from STEP 7 will be accepted as the tag name. The relevant data from the symbol table or data blocks will be integrated in the WinCC flexible tag.

The tag names, transferred from STEP 7 to the WinCC flexible project, are generated from the components of the general STEP 7 symbol. The tag name "Motor_Speed" is derived from "Motor.Speed" for example.

For unique identification, an index starting with "1" is assigned to identical tags. Non-supported characters in a tag name are replaced by an underscore ("_").

www.ElectricalPartManuals.com

Changing a connection

When you make changes to a connection, e.g., by changing a node, a program, or a station, the symbol association of a tag is not lost. The tag association is automatically reassigned to the STEP 7 symbol.

If a tag can no longer be assigned because the address or symbol does not exist, you have the following options:

- Save the association
  The tag will be labeled defective. The tag in question must be connected manually.
- Separate tag from symbol
  This tag will no longer be compared automatically with the STEP 7 symbol.

19.4 Configuring alarms

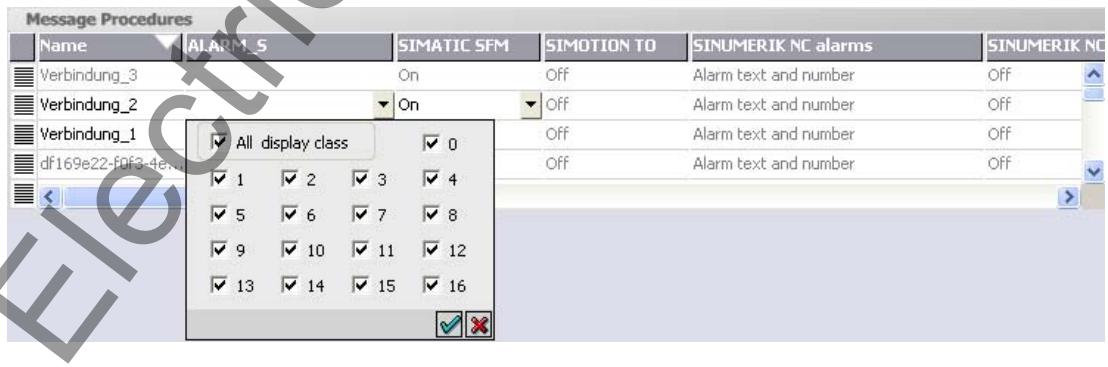
19.4.1 Integrating alarms with the alarm numbering procedure

Configuring in SIMATIC STEP 7

ALARM_S and ALARM_D are alarm numbering procedures. Alarm numbers are assigned automatically during STEP 7 configuration. These numbers are used to uniquely assign alarm messages.

During alarm configuration in STEP 7, the stored alarms and attributes are placed in the STEP 7 configuration data. WinCC flexible automatically imports the required data and transfers them later to the HMI device.

In WinCC flexible, you can filter the display of ALARM_S alarms via display classes. In the project view select "Alarms ▸ Settings" and double-click the "Alarm settings". The existing connections are displayed in the "Alarm Procedures" area.



In the row of the required connection, select the field in the "ALARM_S Display Classes" column and open the selection dialog by pressing the selection button. Select the display class you want. Close the selection dialog by pressing the button.

In the "SFM Alarms" column of a link, specify whether system errors should be displayed. For more information, consult the STEP 7 documentation.

www.ElectricalPartManuals.com

Alarm class layout

The ALARM_S and ALARM_D alarms are assigned to particular alarm classes in STEP 7. To edit the display options for these alarm classes, select "Alarms ▸ Settings ▸ Alarm Classes." Open the context menu and select the "Open Editor" command. You can recognize alarm classes by the S7 prefix in the alarm class name.

You configure the display options for the alarm classes using the "Alarm Classes" editor.

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

20 Appendix

20.1 Open Source Software

Open Source Software

WinCC flexible contains the Open Source Software, among others.

Name	Copyright / Owner of License	
Xerxes	Under Apache Software License, Version 1.1	"http://www.apache.org/"

20.2 Performance features

20.2.1 General Technical Data

20.2.1.1 Released operating systems

Released operating systems

Operating systems released for WinCC flexible:

- Windows 2000 Professional SP4[1)]
- Windows XP Professional SP1, SP2[1)]

[1)] For multilingual configurations, use the MUI (Multilingual User Interface) version of the operating system. Visit the Microsoft homepage at "http://www.microsoft.com/"

Note

For information on the installed Windows version, refer to the "General" tab in "Start ▸ Settings ▸ Control Panel ▸ System."

www.ElectricalPartManuals.com

20.2.1.2 Released databases

Released databases

Logging with WinCC flexible Runtime is released for the following databases:

- MS SQL Server: MS SQL Server 2000 was tested
- MS Access: MS Access 2000 and MS Access XP were tested
- MS Data Engine: MS Data Engine 2000 and MS Data Engine XP were tested

20.2.1.3 Further software versions supported

Further software versions required

Software versions supported in WinCC flexible:

Software	Version
STEP 7	Version 5.3 SP2
SIMOTION Scout	V3.2 SP1
SIMATIC NET	11/2003 (V6.2)

20.2.1.4 Recommended printers

Recommended printers

The current list of recommended printers for operator control devices can be found on the Internet at: "http://www4.ad.siemens.de/view/cs/en/11376409"

20.2.1.5 Legal characters

Introduction

WinCC flexible supports the full ASCII character set. We advise users, however, to refrain from using localized special characters. You should in particular refrain from using these special characters in the names of objects used in scripts.

Illegal characters

The following characters are not allowed: '

www.ElectricalPartManuals.com

20.2.1.6 Memory requirement of recipes

Calculation of memory requirements

The memory space required by each recipe (in KB) is derived from the sum of D1 + D2 + D3.

Valid is:

- D1 = (number of entries x 9 + M + 4):1024

  Applies to M:

  M = Accumulated length of all variable names = Sum of characters in all tags used in the entries.
- D2 = [(number of data records x 12) + 4]:1024
- D3 = [number of data records x (data record length + N) + 4]:1024

  Applies to N:

  Data record name < 13 characters: N = 12

  Data record name > 12 characters: N = 40

D1, D2 and D3 are rounded to the next higher number.

Memory requirements for using arrays

The memory required by each recipe (in KB) is derived from the sum of D1 + D2 + D3.

Valid is:

- D1 = (number of entries x 9 + M + 4):1024

  Each element of the tag array used counts as a single entry.

  Applies to M:

  M = (length of the array tag name + K) x number of array elements

  Applies to K:

  K = 3: 2 to 9 elements in the array

  K = 4: 10 to 99 elements in the array

  K = 5: 100 to 999 elements in the array

  K = 6: 1000 to 9999 elements in the array

  K = 7: 10000 to 12000 elements in the array
- D2 = [(number of data records x 12) + 4] : 1024
- D3 = [number of data records x (data record length + N) + 4] : 1024

  Applies to N:

  Data record name < 13 characters: N = 12

  Data record name > 12 characters: N = 40

D1, D2 and D3 are rounded to the next higher number.

www . ElectricalPartManuals . com

Note

If you use both tags and arrays in a recipe, you have to add the results of both formulas to calculate the total memory required.

20.2.2 System limits

20.2.2.1 System limits

Introduction

The following tables of system limitations provide assistance in estimating whether a specific project is still within the system limitations of a specific HMI device.

The maximum values specified are not additive, i.e. 4000 alarms can be configured if no further objects are used. However, please note that simultaneous output of 4000 alarms and 300 screens, each with 40 screen objects, is not possible.

In addition to the limitations specified, allowances must be made for restrictions in configuration memory resources.

Overview

Micro Panels

	OP 73micro	TP 170micro	TP 177micro
Tags			
Number of tags in the project	500	250	250
Number of PowerTags	--	--	--
Number of elements per array	50	100	100
Number of local tags	--	--	--
Alarms			
Number of alarm classes	32	32	32
Number of discrete alarms	250	500	500
Number of analog alarms	--	--	--
Character length of an alarm	80	80	80
Number of process values per alarm	8	8	8
Size of the alarm buffer	100	128	128
Number of queued alarm events	30	16	32
Screens			
Number of screens	250	250	250
Number of fields per screen	20	20	20
Number of tags per screen	20	20	20
Number of complex objects per screen	5	5	5

www.ElectricalPartManuals.com

	OP 73micro	TP 170micro	TP 177micro
Recipes			
Number of recipes	--	--	--
Number of elements per recipe	--	--	--
User data length in bytes per data record	--	--	--
Number of data records per recipe	--	--	--
Number of recipe elements in the project	--	--	--
Reserved memory for data records in the internal Flash	--	--	--
Logs			
Number of logs	--	--	--
Number of entries per log file (including all log segments)	--	--	--
Number of log segments	--	--	--
Cyclical trigger for tag logging	--	--	--
Trends			
Number of trends	--	--	--
Text lists and graphic lists			
Number of graphic lists	--	--	--
Number of text lists	150	--	150
Total number of lists	150		150
Number of entries per text or graphic list	30	--	30
Number of graphic objects	250	500	500
Number of text elements	1000	500	500
Scripts			
Number of scripts	--	--	--
Communication			
Number of connections	1	1	1
Number of connections based on the "SIMATIC HMI http Protocol"	--	--	--
Maximum number of connected Sm@rtClients (including a service client)	--	--	--
Help system			
Number of characters in a help text	320	--	320
Languages			
Number of runtime languages	5	5	5
Scheduler			
Tasks	--	--	--
User Administration			
User groups	1	2	1
Authorizations	--	32	--
Passwords	1	1	1
Project			
Size of the project file "*.fwx"	128 KB	256 KB	256 KB

www.ElectricalPartManuals.com

Mobile Panels

	Mobile Panel 170	Mobile Panel 177
Tags		
Number of tags in the project	1000	1000
Number of PowerTags	--	-
Number of elements per array	1000	1000
Number of local tags	500	500
Alarms		
Number of alarm classes	32	32
Number of discrete alarms	2000	2000
Number of analog alarms	50	50
Character length of an alarm	80	80
Number of process values per alarm	8	8
Size of the alarm buffer	256	256
Number of queued alarm events	64	64
Screens		
Number of screens	500	500
Number of fields per screen	50	50
Number of tags per screen	50	50
Number of complex objects per screen	5	5
Recipes		
Number of recipes	100	100
Number of elements per recipe	200	200
User data length in bytes per data record	800	800
Number of data records per recipe	200	200
Number of recipe elements in the project	--	--
Reserved memory for data records in the internal Flash	32 KB	32 kB
Logs		
Number of logs	--	--
Number of entries per log file (including all log segments)	--	--
Number of log segments	--	--
Cyclical trigger for tag logging	--	--
Trends		
Number of trends	50	50
Text lists and graphic lists		
Number of graphic lists	100	100
Number of text lists	300	300
Total number of lists	300	300
Number of entries per text or graphic list	30	30
Number of graphic objects	1000	1000
Number of text elements	2500	2500

www.ElectricalPartManuals.com

	Mobile Panel 170	Mobile Panel 177
Scripts		
Number of scripts	--	--
Communication		
Number of connections	4	4
Number of connections based on the "SIMATIC HMI http Protocol"	--	4
Maximum number of connected Sm@rtClients (including a service client)	--	2
Help system		
Number of characters in a help text	320	320
Languages		
Number of runtime languages	5	5
Scheduler		
Tasks	10	10
User administration		
User groups	50	50
Authorizations	32	32
Passwords	50	50
Project		
Size of the project file "*.fwx"	768 KB	2 MB

Panels

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	270 series
Tags								
Number of tags in the project	1000	1000	1000	500	1000	500	1000	2048
Number of PowerTags	--	--	--	--	--	--	-	--
Number of elements per array	50	100	--	100	1000	250	1000	1000
Number of local tags	--	--	250	250	500	--	500	1000
Alarms								
Number of alarm classes	32	32	32	32	32	32	32	32
Number of discrete alarms	500	1000	1000	1000	2000	1000	2000	4000
Number of analog alarms	--	--	50	--	50	--	50	200
Character length of an alarm	80	80	80	80	80	80	80	80
Number of process values per alarm	8	8	8	8	8	8	8	8

www.ElectricalPartManuals.com

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	270 series
Alarms								
Size of the alarm buffer	150	256	256	128	256	256	256	512
Number of queued alarm events	50	64	64	16	64	64	64	250
Screens								
Number of screens	500	500	500	250	500	250	500	500
Number of fields per screen	20	30	30	20	50	30	50	200
Number of tags per screen	20	30	30	20	50	30	50	200
Number of complex objects per screen	5	5	5	5	5	5	5	10
Recipes								
Number of recipes	--	--	100	--	100	--	100	300
Number of elements per recipe	--	--	200	--	200	--	200	1000
User data length in bytes per data record	--	--	800	--	800	--	800	4000
Number of data records per recipe	--	--	200	--	200	--	200	500
Number of recipe elements in the project	--	--	--	--	--	--	--	--
Reserved memory for data records in the internal Flash	--	--	32 KB	--	32 KB	--	32 kB	64 KB
Logs								
Number of logs	--	--	--	--	--	--	--	20
Number of entries per log file (including all log segments)	--	--	--	--	--	--	--	10000
Number of log segments	--	--	--	--	--	--	--	400
Cyclical trigger for tag logging	--	--	--		--	--	--	1 s
Trends								
Number of trends	--	--	--	--	50	--	50	300

www . ElectricalPartManuals . com

	OP 73	OP 77A	OP 77B	TP 170A	TP 170B OP 170B	TP 177A	TP 177B OP 177B	270 series
Text lists and graphic lists								
Number of graphic lists	--	--	--	--	100	--	100	400
Number of text lists	150	300	300	--	300	300	300	500
Total number of lists	150	300	300	--	300	300	300	500
Number of entries per text or graphic list	30	30	30	--	30	30	30	256
Number of graphic objects	500	1000	1000	1000	1000	1000	1000	1000
Number of text elements	2500	2500	2500	1000	2500	1000	2500	10000
Scripts								
Number of scripts	--	--	--	--	--	--	--	50
Communication								
Number of connections	2	4	4	4	4	4	4	6
Number of connections based on the "SIMATIC HMI http Protocol"	--	--	--	--	--	--	4	8
Maximum number of connected Sm@rtClients (including a service client)	--	--	--	--	--	--	2	6": 3 10": 2
Help system								
Number of characters in a help text	320	320	320	--	320	320	320	320
Languages								
Number of runtime languages	5	5	5	5	5	5	5	5
Scheduler								
Tasks	--	--	10	--	10	--	10	48
User administration								
User groups	25	50	50	2	50	50	50	50
Authorizations	32	32	32	32	32	32	32	32
Passwords	25	50	50	1	50	50	50	50
Project								
Size of the project file "*.fwx"	256 KB	320 kB	1 MB	320 KB	768 KB	6": 512 KB 10": 1024 KB	2 MB	2 MB

www.ElectricalPartManuals.com

Multi Panels

	270 series	370 series
Tags		
Number of tags in the project	2048	2048
Number of PowerTags	--	--
Number of elements per array	1000	1000
Number of local tags	1000	2000
Alarms		
Number of alarm classes	32	32
Number of discrete alarms	4000	4000
Number of analog alarms	200	200
Character length of an alarm	80	80
Number of process values per alarm	8	8
Size of the alarm buffer	512	1024
Number of queued alarm events	250	500
Screens		
Number of screens	500	500
Number of fields per screen	200	400
Number of tags per screen	200	400
Number of complex objects per screen	10	20
Recipes		
Number of recipes	300	500
Number of elements per recipe	1000	1000
User data length in bytes per data record	4000	4000
Number of data records per recipe	500	1000
Number of recipe elements in the project	--	--
Reserved memory for data records in the internal Flash	64 KB	128 KB
Logs		
Number of logs	20	50
Number of entries per log file (including all log segments)	10000	50000
Number of log segments	400	400
Cyclical trigger for tag logging	1 s	1 s
Trends		
Number of trends	300	400
Text lists and graphic lists		
Number of graphic lists	400	500
Number of text lists	500	500
Total number of lists	500	500
Number of entries per text or graphic list	256	256
Number of graphic objects	1000	2000
Number of text elements	10000	30000

www.ElectricalPartManuals.com

	270 series	370 series
Scripts		
Number of scripts	50	100
Communication		
Number of connections	6	6
Number of connections based on the "SIMATIC HMI http Protocol"	8	8
Maximum number of connected Sm@rtClients (including a service client)	6": max. 3 10": Max. 2	12": max. 3 15": Max. 2
Help system		
Number of characters in a help text	320	320
Languages		
Number of runtime languages	5	5
Scheduler		
Tasks	48	48
User Administration		
User groups	50	50
Authorizations	32	32
Passwords	50	50
Project		
Size of the project file "*.fwx"	4 MB	7 MB

WinCC flexible Runtime

	WinCC flexible Runtime
Tags	
Number of tags in the project	2048
Number of PowerTags	128 –2048
Number of elements per array	1600
Number of local tags	2000
Alarms	
Number of alarm classes	32
Number of discrete alarms	4000
Number of analog alarms	500
Length of an alarm	80
Number of process values per alarm	8
Size of the alarm buffer	1024
Number of queued alarm events	500

www.ElectricalPartManuals.com

	WinCC flexible Runtime
Screens	
Number of screens	500
Number of fields per screen	400
Number of tags per screen	400
Number of complex objects per screen	40
Recipes	
Number of recipes	999
Number of elements per recipe	2000
User data length in bytes per data record	8000
Number of data records per recipe	5000
Number of recipe elements in the project	--
Reserved memory for data records in the internal Flash	--
Logs	
Number of logs	100
Number of entries per log file (including all log segments)	500000
Number of log segments	400
Cyclical trigger for tag logging	1 s
Trends	
Number of trends	800
Text lists and graphic lists	
Number of graphic lists	500
Number of text lists	500
Total number of lists	500
Number of entries per text or graphic list	3500
Number of graphic objects	2000
Number of text elements	30000
Scripts	
Number of scripts	200
Communication	
Number of connections	8
Number of connections based on the "SIMATIC HMI http Protocol"	16
Maximum number of connected Sm@rtClients (including a service client)	5
Help system	
Number of characters in a help text	320
Languages	
Number of runtime languages	16
Scheduler	
Tasks	48

www.ElectricalPartManuals.com

	WinCC flexible Runtime
User Administration	
User groups	50
Authorizations	32
Passwords	100
Project	
Size of the project file "*.fwx"	

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com

Index

A

Accepting tags
  from STEP 7, 19-13
Accepting tags from STEP 7, 19-12
Access
  Tag, 12-13
Access to runtime object model, 12-15
accessing
  Script in the script, 12-14
  System function in the script, 12-14
Acknowledgement tag
  reading, 6-9
  write, 6-9
Acknowledging Alarms, 6-3
Acquisition cycle
  Tag, 4-8
  Tags, 4-10, 4-15
Actions
  in the Object View, 3-15
Addressing
  Multiplexing, 4-13
  Tag, indirect addressing, 4-13
Alarm
  Acknowledging by the PLC, 6-9
  Basic settings, 6-16
  Component, 6-8
  Editor, 6-9
  Property, 6-8
Alarm class, 6-3, 6-19
  Alarm classes editor, 6-14
Alarm classes editor, 6-14
Alarm group
  Alarm groups editor, 6-15
Alarm groups editor, 6-15
Alarm indicator, 5-8, 6-5
Alarm line, 6-5
Alarm log
  Basic principles, 6-19
  Displaying alarms, 6-24
Alarm logging, 6-19
  Log behavior, 6-20
  Storage media, 6-20
Alarm number, 6-8
Alarm number procedure, 6-2
Alarm procedure, 6-2
Alarm report
  Configure, 10-12
Alarm status, 6-2
  Layout, 6-4
Alarm text, 6-8
Alarm view, 5-8, 6-5, 6-20
Alarm window, 6-5
ALARM_D alarms
  Configuring in STEP 7, 6-17, 19-14
ALARM_S alarms
  Configuring in STEP 7, 6-17, 19-14
  Filtering a display, 6-17, 19-14
Alarms, 6-1
  Acknowledging, 6-3
  Basic principles, 6-1
  Display on the HMI device, 6-5
  Events, 6-7
  Logging, 6-6, 6-23
  Output, 6-24
  Printing, 6-6
  Reporting, 6-6, 6-17, 10-11
  System alarms, 6-4
  System functions, 6-6
Analog alarm procedure, 6-2
Analog alarms
  Analog alarms editor, 6-12
Analog alarms editor, 6-12
Application
  Of report objects, 10-10
  Project documentation, 14-1
Application example, 15-1
Application Examples
  for reports, 10-1
Area pointer
  in the Connections editor, 7-6
Array, 4-13
  Indirect addressing, 4-13
Array transfer from STEP 7, 19-13
Asian characters
  Input on the HMI device, 13-18
  Interpretation, 13-18
  Memory requirements, 13-18

www.ElectricalPartManuals.com

Asian languages
Configuring, 13-8
Asian operating system, 13-5
Assigning
a function key, 5-13
Audit Trail, 17-1
Automatic synchronization, 12-16
Automatic translating, 13-12
Automation
Automatic alarm dispatch, 1-11
Concepts, 1-8
Control with one HMI device, 1-8
Controller with several HMI devices, 1-8
Distributed HMI, 1-12
HMI System with centralized functions, 1-9
Mobile units, 1-10
Remote access, 1-10
Single-user System, 1-8

B

Backing up
HMI data, 18-7
Bar, 5-8
Basic settings
Alarm, 6-16
Data log, 9-6
Bootstrapping
Operating system, 18-9
Branch, 16-4
Bulk data processing, 1-16
Advantages, 1-16
Button, 5-8

C

Change
A connection, 19-13, 19-14
Change log, 17-1, 17-2, 17-3, 17-5
Changing, 2-23
Default properties, 10-5
Object property, 12-23
character sets
configurable, 13-18
Circle, 5-8, 10-9
Circular log, 6-20
Clock, 5-8
Collapse, 2-14
Windows, 2-14
Combining, 2-14
Windows, 2-14
Communication
between PLC and tag, 4-7
using area pointers, 7-2
using tags, 7-2
Communication drivers, 7-3
Component parts
of a project, 3-2
Configuration, 8-5
Move, 1-17
Of recipes, 8-5
Recipe, 8-12
Screen change, 1-18
configuring
Custom HMI device, 1-14
Editing objects simultaneously, 1-16
HMI device independent, 1-15
Movement, 1-17
Movement path, 1-17
Solution-oriented Concepts, 1-13
Target system dependency, 1-14
Target system independent, 1-15
Configuring
Asian languages, 13-8
Recipes view, 8-15
Toolbar, 2-4
Connecting tags
Via application points, 19-13
Via the Tag editor, 19-12
Connection
Changing, 19-13, 19-14
Connection data
Format for import, 4-18
Consistency test, 3-20
Context menu, 2-15
accessing, 2-15
Controller
Linking tags, 4-7
Cover sheet
Layout, 14-2
Creating, 17-5
Create new project version, 17-5
HMI station, 19-5
Creating a routing connection, 19-7
Cross-reference, 3-18
Editor, 3-18
Working with, 3-18
csv file
Example, 6-24, 9-9
Layout, 6-24, 9-9
CSV file, 8-5

www.ElectricalPartManuals.com

D

Data
  Global project, 3-5
  HMI device-specific, 3-5
Data backup
  HMI device, 18-7
Data log
  Basic settings, 9-6
  Outputting logged data, 9-9
  Tags, 4-10, 9-8
Data logging, 9-1, 9-3
  Acquisition cycle, 9-3
  Application, 9-1
  Logging cycle, 9-3
  Storage media, 9-4
Data logs
  Editor, 9-5
Data logs editor, 9-5
  Open, 9-5
Data mailbox
  For recipes, 8-6
Data record name, 8-11
Data record number, 8-11
Data selection
  For project report, 14-6
Data type
  External tag, 4-8
  Internal tag, 4-2
Date/time field, 5-8, 10-9
Debugger, 12-16
  Error types, 12-17
Debugging, 12-16
Default properties
  Changing, 10-5
Defining a display class
  For ALARM_S alarms, 6-17, 19-14
Delta download, 18-4
Design
  Protocol, 10-2
Device-based dependency, 2-18
  Principle, 2-18
Discrete alarm procedure, 6-2
Discrete alarms
  Discrete alarms editor, 6-11
Discrete alarms editor, 6-11
Dispatching Alarms
  Automatic, 1-11
  via e-mail, 1-11
Distributed HMI, 1-12
Docking, 2-13
  Toolbar, 2-13
  Windows, 2-13
Documentation in WinCC flexible, 3-20
Drag-and-drop, 2-15, 12-12
dynamic operation, 5-12

E

Eastern characters
  Input on the HMI device, 13-18
Editing, 8-5
  Layout, 14-3, 14-4
  Projects, 3-10
  Recipe data record in WinCC flexible, 8-5
  Recipe record, 8-5
  Report properties, 10-8
Editing connections
  With NetPro, 19-4
  With WinCC flexible, 19-4
Editing language, 13-3
Editing objects
  In the SIMATIC Manager, 19-5
Editing possibilities
  Project report, 14-5
Editor, 2-23
  Alarm logs, 6-20
  Brief description of the editors, 3-10
  Closing, 2-24
  Configuring alarms, 6-9
  Connections, 7-3
  Cross-reference, 3-18
  Graphical, 2-20, 3-11
  Graphics, 13-15
  opening, 2-21
  Possible WinCC flexible editors, 3-2
  Project documentation, 3-20
  Project languages, 13-5
  Project texts, 13-10
  Properties, 2-20
  Recipe data records, 8-11
  Recipes, 8-8
  Screens, 3-10
  Script, 12-8
  System dictionary, 13-13
  Tabular, 2-20
  Tabular editors, 3-10
  User dictionary, 13-14
  User dictionary, 13-14
  With language-dependent objects, 13-9
Editor, 2-20
Ellipse, 5-8, 10-9
Engineering support, 1-13
  Overview, 1-13

www.ElectricalPartManuals.com

Event, 15-2, 15-3
Time-based event, 15-3
Export, 11-7
Project texts, 13-11
External tag, 12-13

F

Faceplate, 3-18
Application, 5-17
Properties, 5-17
Reusing instances, 5-17
Fault
Logical error, 12-17
Runtime error, 12-17
Field of application, 11-1, 15-1, 16-1, 17-1
Logging changes, 17-1
Managing project versions, 16-1
Planning jobs, 15-1
User administration, 11-1
Function
Dependency on the type of HMI device, 3-3
Function key, 5-2
Function List, 12-6
Asynchronous completion, 12-21
Completion, 12-7
Completion in runtime, 12-21
HMI device dependency, 12-7
Property, 12-7
Script, 12-7
Status information, 12-7
Synchronous completion, 12-21
System function, 12-7
Functional scope
ProSave, 18-7

G

Gauge, 5-8
Global assignment
of a function key, 5-13
Graphic I/O box, 5-8, 10-9
Graphic object, 5-8
Graphic view, 5-8, 10-9
Graphics
Structure of editor, 13-15

H

Help, 2-25
Displaying, 2-25
Help function, 12-12
HMI device
configuration with several HMI devices, 3-2
Data backup, 18-7
Project with several HMI devices, 3-5
Remote access (concept), 1-10
Restoring data, 18-7
Selection, 3-3
synchronized, 1-12
Using a project for several HMI devices, 3-6
Version, 3-22, 18-2
HMI devices
Inserting multiple, 19-5
HMI station
Creating, 19-5
HMI system
Tasks, 1-1
HTML browser, 5-8

I

Import, 11-7
of variables, 4-17
Project texts, 13-11
Index tag, 4-13
Indirect addressing, 4-13
Initializing
Password, 18-4
Recipe, 18-4
Installing
Options, 18-10
Instance
Reusing, 5-17
Integrated configuration, 1-3
Integrated projects
Using HW Config, 19-3
Integration in SIMOTION SCOUT, 1-19
Integration in STEP 7, 19-1
IntelliSense, 12-10
Internal tag, 12-13
Introduction
Project documentation, 14-1
IO field, 5-8, 10-9

www . ElectricalPartManuals . com

L

Language dependency
System function, 12-14
Language support
Toolbar, 13-9
Language switching
ProSave, 18-7
Language switching, 13-17
In Runtime, 12-23
Language-dependent format, 13-4
Languages
In various editors, 13-9
Layout
Cover sheet, 14-2
Editing, 14-3, 14-4
Project report, 14-2
Property, 14-3
Regional format of the date, time, currency, and numbers, 13-4
Library, 2-8, 3-17, 5-15
Global, 2-9
Project-related, 2-9
Library object, 5-15
License
for options, 1-7
for WinCC flexible ES, 1-6
for WinCC flexible Runtime, 1-7
License Key
Transferring to HMI device, 18-10
License key diskette, 18-10
Licensing
order, 1-7
Limit range
Tags, 4-8
Line, 5-8, 10-9
Local assignment
of a function key, 5-13
Local tags, 12-13
Log contents
Displaying, 6-20, 9-4
Log database
Direct access with ODBC, 6-26, 9-11
Log type, 6-20
Circular log, 9-4
Level-dependent, 9-4
Segmented circular log, 9-4
Logging
Alarms, 6-23
Basic principles, 6-19
Circular log, 6-20, 9-4
Log types, 6-20, 9-4
Segmented circular log, 6-20, 9-4
Tag values, 9-1
Tags, 4-10, 9-8
Logging changes, 17-1, 17-6
Application, 17-1
Field of application, 17-1
Work area, 17-7
Logging cycle
Tags, 4-15
Logical error, 12-17

M

Managing project versions, 16-1, 16-5
Application example, 16-1
Field of application, 16-1
Property view, 16-7
Work area, 16-6
Manual synchronization, 12-16
Menu, 2-3
Command, 2-4
Migrating
WinCC or ProTool projects, 3-16
migration
WinCC flexible Edition, 1-4
Migration, 2-18
Principle, 2-18
Mobile Units
Use, 1-10
Mouse functions, 2-15
Movement path, 1-17
Multiple selection
And object groups, 5-12
Multiplexing, 4-13
Multi-user project, 3-3

N

Name, 16-7
Navigation, 5-5
Navigation arrow, 2-24
Navigation control, 5-7
navigation structure
Specifying, 1-18
Navigation Structure, 1-18
NetPro
Editing connections, 19-4

www.ElectricalPartManuals.com

O

Object, 10-9
Access, 12-15
Change property with VBS, 12-23
Reference, 12-15
Selecting for project report, 14-7
Synchronize in the script, 12-16
Object group, 5-12
Object selection
Outputting configuration data, 14-6
Object view
Properties, 2-11
Object view
Actions in, 3-15
Online help, 2-26
Displaying, 2-26
Open, 2-21
Data logs editor, 9-5
opening, 17-5
Editor, 2-21
Opening older project versions, 17-5
Recipes editor, 8-8
Script editor, 12-8
operating system
Asian language settings, 13-5
Settings on Western, 13-5
Operating system
Updating on HMI device, 18-9
Operation, 2-15
Hotkeys, 2-16
Mouse functions, 2-15
Operator control element
Editor-specific, 2-11
of a frame, 2-12
of a toolbar, 2-12
Placing editor-specific, 2-12
Recipes view, 8-17
Operator device dependency, 3-3
In the script, 12-15
Principle, 2-19
Option, 1-5
Licensing, 1-7
Options
Installing, 18-10
organizing
Script, 12-5
Output
Project report, 14-7
Output data of a recipe, 10-15
Output medium
Project report, 14-1
Output View, 2-9
Properties, 2-9
Overview
Report system, 10-1

P

Parameter delivery
Script, 12-14
System function, 12-14
Parameter delivery in runtime, 12-22
Password
Initializing, 18-4
Password view, 5-8
Performance features, 20-4
Picture
Steps in creating a, 5-4
Planning jobs, 15-1, 15-4
Application example, 15-1
Field of application, 15-1
Work area, 15-5
Polygon, 5-8, 10-9
Polyline, 5-8, 10-9
Print alarm
Configuring output parameters, 10-12
Print recipe
Configuring output parameters, 10-16
Procedures
to create screens, 5-4
Process screen:
Changing, 1-18
Processing
Scripts in runtime, 12-21
project, 2-17, 3-1, 17-2, 17-5
Editing, 3-10
For several HMI devices, 3-3
Functional scope, 2-19
Load, 2-18
Multilingual projects, 3-8
New, 2-18
Operator device dependency, 3-2
Several projects, 2-18
Testing, 3-20
Testing with the simulator, 3-20
under version management, 17-5
Use for several HMI devices, 3-6
with several HMI devices, 3-2
Working with, 2-17
Project
Migrating, 3-16

www.ElectricalPartManuals.com

project configuration changes, 17-2
recorded project changes, 17-2
Project data, 2-20
Updating, 2-20
Project documentation
Application, 14-1
Introduction, 14-1
Project language, 13-2
Editor, 13-5
Project languages
Editor, 13-5
Project library, 3-17, 5-15
Project navigation, 1-18
Project report, 14-1
Data selection, 14-6
Editing possibilities, 14-5
For individual objects, 14-6
Layout, 14-2
Layout contents page, 14-3
Output, 14-7
Output medium, 14-1
Outputting Compact, 14-3
Outputting Complete, 14-3
Selecting objects, 14-7
Project session, 17-3
Project tag, 12-13
Project text
Access to, 13-10
Project texts
Editor, 13-10
Translating externally, 13-11
Project version, 16-3, 16-8, 17-5
current version, 16-8
New project version, 17-5
next version, 16-8
Older project versions, 16-3, 17-5
Project versions, 16-1
Project View
Working with, 2-6
Project View, 2-6
HMI device dependent data, 3-5
Selection of HMI device types, 3-3
Properties
Faceplate, 5-17
Property
Alarm, 6-8
Function List, 12-7
Layout, 14-3
Script editor, 12-10
Tag, 4-6
Property view, 2-7
Properties, 2-7
Tag, 4-4
ProSave, 18-6
Language switching, 18-7
Protocol
Application Examples, 10-1
Design, 10-2
ProTool project
Migrating, 3-16

R

Read continuously
Tags, 4-10
Recipe, 8-1, 8-3, 8-4
Basic principles, 8-1, 8-3
Configuration, 8-12
Configuration options, 8-5
Configuration settings, 8-5
Data set, 8-4
Display in Runtime, 8-13
Initializing during a transfer operation, 18-4
Layout, 8-3
Output data for reporting, 10-15
Principle, 8-1
Settings, 8-12
Use, 8-2
recipe data
overwrite during transfer, 18-4
Recipe data record name, 8-11
Writing to a tag, 8-20
Recipe data record number, 8-11
Writing to a tag, 8-20
Recipe name
Writing to a tag, 8-20
Recipe number
Writing to a tag, 8-20
Recipe record, 8-4
Editing in WinCC flexible, 8-5
Layout, 8-3
Transfer, 8-7
Transfer options, 8-7
Recipe report
configuring, 10-16
Recipe screen, 8-14
applications, 8-21
Basic principles, 8-20
Overview, 8-14
Principle, 8-21
Spreading recipes over process screens according to topic, 8-21
Visual machine simulation, 8-21
Recipe settings, 8-12

www.ElectricalPartManuals.com

Recipes editor, 8-11
Data records, 8-11
Description, 8-8
Elements tab, 8-9
opening, 8-8
Structure, 8-8
Recipes view, 5-8, 8-13
Basic principles, 8-15
Behavior with screen change, 8-18
Configuring, 8-15
Displaying one recipe only, 8-19
Displaying values only, 8-19
Operation with function keys, 8-18
Operator control elements, 8-17
Overview, 8-13
Simple, 8-15
Simple, configuration, 8-16
Simple, display, 8-16
Using as a drop-down list, 8-18
Rectangle, 5-8, 10-9
Recursion level, 12-2
Reference
Object, 12-15
Reference language, 13-3
Reference text function, 13-9
Remote access, 1-10
Application possibilities, 1-11
Replacing, 3-19
Character string, 3-19
Object, 3-19
Report objects
Application, 10-10
Report properties
Editing, 10-8
Report system, 10-1
Overview, 10-1
Restoring
HMI data, 18-7
Restoring data
HMI device, 18-7
Return value, 12-22
Reusing
Faceplate, 5-17
Routing connection, 19-7
for the transfer, 19-9
Runtime
Change object property with VBS, 12-23
Completion of the function list, 12-21
Language switching, 12-23
Parameter delivery, 12-22
Processing scripts, 12-21
Task, 1-5
Runtime error, 12-17
Runtime language, 13-3, 13-17
Runtime scripting, 12-1
Application, 12-1
Runtime user administration, 11-1
Runtime User Administration, 11-1

S

Scalability, 1-14
scaling
Linear Scaling, 4-8
Scaling
Tag, linear scaling, 4-12
Scheduler, 15-1
Screen Editor
Layout, 5-3
Screen navigation, 1-18
Screen Navigation
Work area, 5-6
Screen Navigation\ editor, 5-5
Script, 12-2, 12-4
Call up in the script, 12-14
Debugging, 12-16
Help function, 12-12
In function list, 12-7
In the script, 12-14
Operator device dependency, 12-15
organizing, 12-5
Parameter delivery, 12-14
Particularity when calling up, 12-14
Processing in runtime, 12-21
Properties, 12-4
Recursion level, 12-2
Return value, 12-22
Use, 12-5
Use system function, 12-23
Script editor, 12-8
opening, 12-8
Properties, 12-10
Work area, 12-9
Script Wizard", 12-9
search, 3-19
Character string, 3-19
Object, 3-19
Segmented circular log, 6-20
Selecting
Objects for project report, 14-7
Sequence, 15-3
Task, 15-3
Setting
Alarm, 6-16
For transferring, 18-2
Languages in the operating system, 13-4

www.ElectricalPartManuals.com

Shared library, 3-17, 5-15
SIMATIC HMI
Definition, 1-1
Introduction, 1-1
Tasks, 1-1
WinCC flexible, 1-2
SIMATIC Manager
Inserting WinCC flexible objects, 19-5
Working with, 19-2
SIMATIC STEP 7, 3-7
SIMOTION SCOUT, 3-7
Simple objects, 10-9
Simple recipe view, 8-15
Layout, 8-16
Simulation, 3-20
Single-user project, 3-2
Slider control, 5-8
SmartClient display, 5-8
Softkey, 5-2, 5-13
Standard layout, 14-3
Start, 2-21
Start value
Tags, 4-9
starting
Editor, 2-21
Status force, 5-8
Status information
Function List, 12-7
STEP 7
Accepting tags in WinCC flexible, 19-12, 19-13
Array transfer to WinCC flexible, 19-13
STEP 7 integration
Advantages, 19-1
Requirements, 19-1
Structure, 11-2
User Administration, 11-2
Switch, 5-8
Switching
Between Runtime languages, 13-17
Symbol library, 5-8
Symbolic I/O field, 5-8, 10-9
Synchronization
With controller, 8-6
synchronizing
Of objects in the script, 12-16
Of tags in the script, 12-16
Synchronizing
Automatic, 12-16
By hand, 12-16
Manual, 12-16
Syntax emphasis, 12-11
System alarms, 6-4
System alarms editor, 6-13
System alarms editor, 6-13
System dictionary, 13-12
Editor, 13-13
Structure of editor, 13-13
System function, 8-7, 12-1, 12-2
Application, 12-3, 12-4
Call up in the script, 12-14
In function list, 12-4, 12-7
in script, 12-4
In the script, 12-14, 12-23
Language dependency, 12-3, 12-14
Parameter delivery, 12-14
Particularity when calling up, 12-14
to transfer of recipe data records, 8-7
Use, 12-4
System limits, 20-4

T

Tab, 2-23
tag
Internal tag, 4-2
Tag
Access with VBS, 12-13
Acquisition cycle, 4-8, 4-10, 4-15
Array, 4-13
Communication with a PLC, 4-7
Data log, 4-10, 9-8
External tag, 4-1
Index tag, 4-13
Indirect addressing, 4-13
Limit range, 4-8
linear scaling, 4-12
Linear Scaling, 4-8
Local, 12-13
Logging, 4-10, 9-8
Logging cycle, 4-15
Multiplexing, 4-13
Property, 4-6
Property view, 4-4
Read continuously, 4-10
Start value, 4-9
Synchronize in the script, 12-16
Tolerance band, 4-10, 9-8
Tag data
Format for import, 4-20
Tag editor, 4-2
Work area, 4-3, 6-11, 6-12, 6-13, 6-14, 6-15, 6-21, 9-5, 13-13, 13-14
Tag import
Data structure of connection data, 4-18
Format of tag data, 4-20
Procedure, 4-16

www.ElectricalPartManuals.com

Tag list
Indirect addressing, 4-13
Tag values
Output, 9-9
Tags
Importing into an HMI device, 4-17
Task, 15-2
Sequence, 15-3
Template, 5-13
Testing
project, 3-20
Text field, 5-8, 10-9
TIA, 1-19
Timer, 15-1, 15-3
Toggle, 2-23
Tolerance band
Tags, 4-10, 9-8
Toolbar, 2-4, 2-13
Configuring, 2-4
Docking, 2-13
Language support, 13-9
Positioning, 2-4
Toolbox, 5-3
Tooltip, 2-25
Totally Integrated Automation, 1-19
SIMOTION SCOUT, 1-19
Transfer, 8-7
Basic principles, 3-21, 18-1
Delta download, 18-4
Method, 18-3
Of recipe data records, 8-7
over routing connection, 19-9
Overwrite recipe data, 18-4
Transfer settings, 18-2
Upload, 18-4
Transfer mode
On the HMI device, 3-21, 18-2
Transferring
License key on HMI device, 18-10
Translating
Automatic, 13-12
Editors, 13-1
Workflow, 13-7
Translation
Of project texts externally, 13-11
Trend, 9-1
Trend view, 5-8, 9-1
Trunk, 16-3

U

Unit dependency, 3-2, 3-3
Update cycle, 4-15
Updating
Operating system on the HMI device, 18-9
Upload
From HMI device, 3-22, 18-5
Project file, 18-4
Use, 8-2
Faceplates, 1-15
Libraries, 1-16
Of recipes, 8-2
Script, 12-5
System function, 12-4
Text Libraries, 1-16
User administration, 11-1, 11-3, 11-5
Field of application, 11-1
Purpose, 11-1
Work area, 11-4, 11-6
User Administration, 11-1, 11-2
Structure, 11-2
User data, 11-7
Export, 11-7
Import, 11-7
User dictionary
Editor, 13-14
User dictionary, 13-12
Structure of editor, 13-14
User interface language, 13-2
User view, 11-7
User-dependency
Working environment, 2-26
Using HW Config
In integrated projects, 19-3

V

VBS
Change object properties, 12-23
Help function, 12-12
Version
HMI device, 3-22, 18-2
version comparison, 16-8
Version management, 16-1

www.ElectricalPartManuals.com

W

WinCC
Automation concepts, 1-8
Options, 1-2
WinCC flexible, 1-2
Application, 1-2
Customized setup of the configuration user interface, 1-15
Editing connections, 19-4
Edition, 2-17
Engineering support, 1-13
Engineering System, 1-4
Individual configuration, 2-26
Integrated configuration, 1-3
Migration to another edition, 1-4
Multilingual user interfaces, 3-8
Powerpack, 1-4
Runtime software, 1-5
Update, 1-4
without licensing, 1-7
Working with the, 2-17
WinCC flexible, 2-1
WinCC project
Migrating, 3-16
Windows, 2-13
Collapse, 2-14
Combining, 2-14
Docking, 2-13
Work area, 2-5
Logging changes, 17-7
Managing project versions, 16-6
of the screen editor, 5-3
Planning jobs, 15-5
Screen Navigation, 5-6
Script editor, 12-9
Tag editor, 4-3, 6-11, 6-12, 6-13, 6-14, 6-15, 6-21, 9-5, 13-13, 13-14
User administration, 11-4, 11-6
working
in the Object View, 3-15
with projects, 3-1
Working
with cross-reference, 3-18
Working environment
resetting, 2-26
User-dependency, 2-26
Working step, 2-20
Restoring, 2-20
Undo, 2-20

www . ElectricalPartManuals . com

www . ElectricalPartManuals . com