

# SIEMENS

## SIMATIC HMI

### WinCC flexible 2004 Micro

#### User's Manual

Preface

Introduction to WinCC  
flexible Micro

1

WinCC flexible Engineering  
System

2

Working with tags

3

Creating screens

4

Creating an alarm system

5

Configuring a connection

6

Using global events

7

Structure of multi-lingual  
projects

8

Project documentation

9

Transfer

10

Configuration examples

11

Features

12

Order number 6AV6691-1AA01-0AB0

Edition 03/2004

A5E00280157-02

## Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety as well as to avoid property damage. 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.



### Danger

indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.



### Warning

indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.



### Caution

used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

### Caution

used without safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

### Notice

used without the safety alert symbol indicates a potential situation which, if not avoided, may result in an undesirable result or state.

When several danger levels apply, the notices of the highest level (lower number) are always displayed. If a notice refers to personal damages with the safety alert symbol, then another notice may be added warning of property damage.

## Qualified Personnel

The device/system may only be set up and operated in conjunction with this documentation. Only qualified personnel should be allowed to install and work on the equipment. Qualified persons are defined as persons who are authorized to commission, to earth, and to tag circuits, equipment and systems in accordance with established safety practices and standards.

## Intended Use

Please note the following:



### Warning

This device and its components may only be used for the applications described in the catalog or technical description, and only in connection with devices or components from other manufacturers approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up and installed correctly, and operated and maintained as recommended.

## Trademarks

All designations marked with ® are registered trademarks of Siemens AG. Other designations in this documentation might be trademarks which, if used by third parties for their purposes, might infringe upon the rights of the proprietors.

### ACopyright Siemens AG, 2004. All rights reserved

Reproduction, transmission or use of this document or its contents is not permitted without express written authority. 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 checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in the manual are reviewed regularly, and any necessary corrections will be included in subsequent editions. Suggestions for improvement are welcomed.

Siemens AG  
Automation and Drives Group  
P.O. Box 4848, D-90327 Nuremberg (Germany)

© Siemens AG 2004  
Technical data subject to change

# Preface

## Purpose of this manual

This user manual is part of the WinCC flexible Micro documentation. The manual provides you with a complete overview of configuring with WinCC flexible Micro. 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 Micro.

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

## Basic knowledge required

General knowledge in the field of automation engineering forms a basic requirement for working with this manual.

Users should be firm in handling PCs operating on a Windows 2000 or Windows XP platform.

## Where this manual is valid

This manual is valid for the WinCC flexible Micro 2004 software package.

## Its place in the information landscape

This manual is part of the SIMATIC HMI documentation. The information below provides you with an overview of the world of SIMATIC HMI.

### User manual

The following documents marked with (\*) are recommended for understanding the Micro Edition.

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

- WinCC flexible Migration:
  - describes how to migrate an existing ProTool project to WinCC flexible.
  - describes how to migrate an existing WinCC project to WinCC flexible.
  - describes how to migrate ProTool projects with an HMI migration from OP7 to OP 77B.
  - describes how to migrate ProTool projects with HMI migration from graphic devices to Windows CE devices.
- Communication:
  - Communication part 1 describes the integration of your HMI device into control systems of the SIMATIC family.
  - Communication part 2 describes the integration of the HMI device into control systems of external suppliers.

### **Operating instructions**

- Operating instructions for the HMI devices TP 170micro/TP 170A/TP 170B/OP 170B. (\*)
- Operating instructions for other HMI devices.

### **Getting started**

- WinCC flexible for newcomers:
  - based on a practical example, this is a step-by-step introduction to the basics of programming screens, messages, recipes and screen navigation.
- WinCC flexible for advanced users:
  - based on a practical sample project, this is a step-by-step introduction to the basics of engineering log files, project reports, scripts, user management, multilingual projects and integration into STEP 7.
- WinCC flexible Options:
  - based on a practical sample project, this is a step-by-step introduction to the basics of engineering the options WinCC flexible Sm@rtServices, Sm@rtAccess and OPC Server.

### **Online availability**

The links below guide you directly to the technical documentation pool for SIMATIC products and systems in German, English, French, Italian and Spanish language.

- SIMATIC Guide Technical Documentation in German:  
[http://www.ad.siemens.de/simatic/portal/html\\_00/techdoku.htm](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](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 3
- Transferring a project to an HMI device – Chapter 10
- Features of WinCC flexible – Chapter 12

## Conventions

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

- "WinCC flexible 2004" designates 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 2004" is always used to distinguish other versions thereof. An edition name such as "WinCC flexible 2004 Micro" is always used to distinguish it from other editions.

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

Layout	Range of validity
"Add screen"	<ul style="list-style-type: none"><li>• Terminology of the user interface, e.g. dialog names, tabs, buttons, menu commands.</li><li>• Input required, e.g. limit values, tag values.</li><li>• Path specification</li></ul>
"File > Edit"	Operating sequence, e.g. menu / context menu commands.
<F1>, <Alt + P>	Keyboard operation

Please observe the following emphasized notes:

### Note

Notes containing important information about the product and its handling, or the relevant part of the documentation you should pay particular attention.

## Registered trademarks

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

Other names in this document could be trademarks, whose use by third parties for their own purposes may constitute an infringement of the rights of the trademark owners.

## Further Support

### Representatives and offices

If you have any further questions relating to the products described in this manual, please contact your partner at you local SIEMENS office.

You contact partner is found under:

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

### Training Center

To familiarize you with automation systems, we offer a variety of courses. Please contact your regional training center, or the central Training Center in D-90327 Nuremberg.

Phone: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

### Service & Support on the Internet

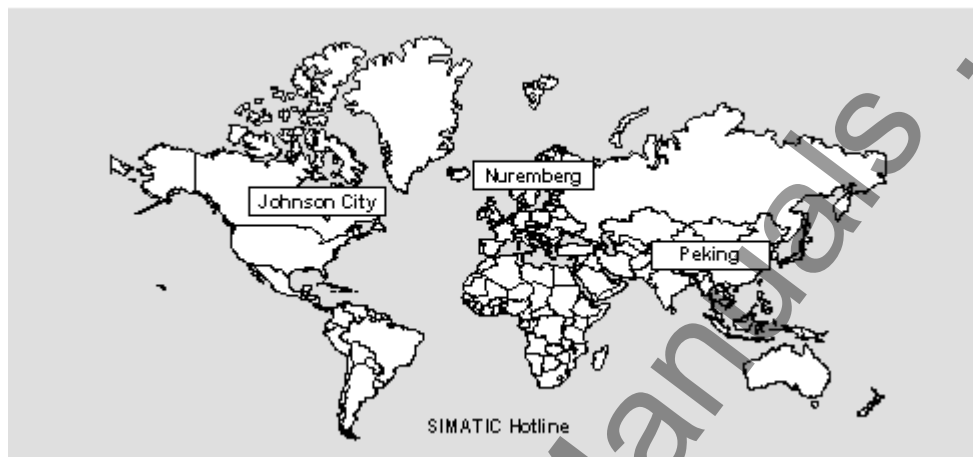
The online services of SIEMENS Service & Support also offer you a comprehensive database relating to SIMATIC products

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

- The newsletter keeps you up-to-date with information relating to your products.
- Our Service & Support search engine provides you with abundant documentation.
- A bulletin board where users and experts exchange their knowledge worldwide.
- Current product information, FAQs and downloads.
- You local partner for Automation & Drives.
- Information about on-site services, repairs, spare parts and lots more is found on our "Services" pages.

## A&D Technical Support and Authorization

Available 24 hours worldwide:



<b>Worldwide (Nuremberg)</b> <b>Technical Support</b> (Free Contact) Local time: Mo.–Fr. 7:00 to 17:00 Telephone: +49 (0) 180 5050-222 Fax: +49 (0) 180 5050-223 E-Mail: adsupport@siemens.com GMT: +1:00	<b>Worldwide (Nuremberg)</b> <b>Technical Support</b> (charged, with Automation Value Card only) Local time: 0:00 to 24:00, 365 days Telephone: +49 (911) 895 -7777 Fax: +49 (911) 895-7001 E-mail: – GMT: +1:00	
<b>Europe/Africa (Nuremberg)</b> <b>Authorization</b> Local time: Mo.–Fr. 7:00 to 17:00 Telephone: +49 (911) 895 -7200 Fax: +49 (911) 895-7201 E-Mail: adauthorisierung@siemens.com GMT: +1:00	<b>America (Johnson City)</b> <b>Technical Support and Authorization</b> Local time: Mo.–Fr. 8:00 to 19:00 Telephone: +1 423 461-2522 Fax: +1 423 461-2289 E-Mail: simatic.hotline@sea.siemens.com GMT: –5:00	<b>Asia/Pacific (Beijing)</b> <b>Technical Support and Authorization</b> Local time: Mo.–Fr. 8:30 to 17:30 Telephone: +86 (10) 64 75 75 75 Fax: +86 (10) 64 74 74 74 E-mail: adsupport.asia@siemens.com GMT: +8:00

Information is available in German and English language

www.ElectricalPartManuals.com



# Table of contents

	<b>Preface .....</b>	<b>i</b>
<b>1</b>	<b>Introduction to WinCC flexible Micro .....</b>	<b>1-1</b>
1.1	Components of WinCC flexible .....	1-1
1.2	Configuration support.....	1-2
<b>2</b>	<b>WinCC flexible Engineering System .....</b>	<b>2-1</b>
2.1	Basic principles on the programming interface.....	2-1
2.2	Program user interface of WinCC flexible.....	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 window .....	2-6
2.2.5	Properties window.....	2-7
2.2.6	Output window .....	2-8
2.3	Working with the mouse.....	2-9
2.4	Keyboard operation.....	2-10
2.5	Working with WinCC flexible.....	2-10
2.5.1	Displaying help.....	2-10
2.5.2	Editor properties.....	2-11
2.5.3	Opening an editor .....	2-12
2.5.4	Editing several projects with WinCC flexible.....	2-14
2.5.5	Object list .....	2-14
2.5.6	Function list.....	2-15
<b>3</b>	<b>Working with tags .....</b>	<b>3-1</b>
3.1	Basic principles .....	3-1
3.1.1	External tags .....	3-1
3.1.2	Internal tags .....	3-2
3.2	Elements and basic settings .....	3-2
3.2.1	"Tags" editor.....	3-2
3.2.2	Basic settings for tags and arrays.....	3-4
3.3	Working with tags.....	3-6
3.3.1	Properties of a tag.....	3-6
3.3.2	External tags for communication with the PLC .....	3-7
3.3.3	Tag limit values .....	3-8
3.3.4	Updating the tag value in runtime .....	3-8
3.4	The basics of arrays.....	3-9
3.5	The basics of cycles.....	3-10

<b>4</b>	<b>Creating screens.....</b>	<b>4-1</b>
4.1	Basic principles .....	4-1
4.1.1	Screen basics.....	4-1
4.1.2	"Screens" editor.....	4-3
4.1.3	Procedures .....	4-5
4.1.4	Managing screens.....	4-5
4.2	Objects of the "Screens" editor .....	4-6
4.2.1	Overview of objects.....	4-6
4.2.2	Object groups.....	4-8
4.3	The advantage of layers.....	4-8
4.4	Security in Runtime .....	4-9
<b>5</b>	<b>Creating an alarm system .....</b>	<b>5-1</b>
5.1	Basic principles .....	5-1
5.1.1	Displaying process and system alarms.....	5-1
5.1.2	User-defined alarms .....	5-1
5.1.2.1	Available alarm procedures.....	5-1
5.1.2.2	Acknowledging alarms .....	5-2
5.1.2.3	Alarm classes.....	5-2
5.1.3	Displaying alarms.....	5-3
5.1.3.1	Displaying alarms on the HMI device.....	5-3
5.1.3.2	System functions for alarm editing.....	5-4
5.2	Elements and basic settings .....	5-5
5.2.1	Alarm components and properties.....	5-5
5.2.2	Editors for Configuring Alarms .....	5-6
5.2.2.1	Basic principles of editors .....	5-6
5.2.2.2	"Discrete alarms" editor.....	5-6
5.2.2.3	"Alarm classes" editor .....	5-7
5.2.3	Basic settings for the alarm system .....	5-9
<b>6</b>	<b>Configuring a connection .....</b>	<b>6-1</b>
6.1	"Connections" editor.....	6-1
<b>7</b>	<b>Using global events .....</b>	<b>7-1</b>
7.1	Use cases for global triggers.....	7-1
7.2	Working with events .....	7-1
7.3	Elements of global triggers.....	7-2
7.3.1	Work area of the global triggers.....	7-2
<b>8</b>	<b>Structure of multi-lingual projects.....</b>	<b>8-1</b>
8.1	Language terminology in WinCC flexible .....	8-1
8.2	Multilingual configuration.....	8-3
8.3	Language settings.....	8-4
8.3.1	Language settings in the operating system .....	8-4
8.3.2	"Project languages" editor.....	8-5
8.4	Languages during runtime .....	8-6

<b>9</b>	<b>Project documentation .....</b>	<b>9-1</b>
9.1	Project documentation .....	9-1
9.2	Selecting the objects for the project documentation .....	9-1
<b>10</b>	<b>Transfer .....</b>	<b>10-1</b>
10.1	Basic principles of the transfer operation.....	10-1
10.2	Transfer settings .....	10-2
10.3	Managing files on the HMI device.....	10-3
10.3.1	ProSave .....	10-3
10.3.2	Data backup for the HMI device.....	10-4
10.3.3	Updating the operating system .....	10-5
<b>11</b>	<b>Configuration examples .....</b>	<b>11-1</b>
11.1	Create screen.....	11-1
11.2	Creating external tags .....	11-4
11.3	Configuring an alarm view .....	11-6
11.4	Configuring an alarm window.....	11-8
11.5	Configuring discrete alarms .....	11-10
11.6	Using the system function "AlarmViewEditAlarm" .....	11-11
<b>12</b>	<b>Features .....</b>	<b>12-1</b>
12.1	General technical data .....	12-1
12.1.1	Released operating systems.....	12-1
12.1.2	Legal characters .....	12-1
12.2	System limits .....	12-2

www.ElectricalPartManuals.com

## 1

## WinCC flexible Micro

You can upgrade to larger editions from the "Micro" edition. Projects created with the "Micro" edition can also be edited in the more comprehensive editions of WinCC flexible.

WinCC flexible is available in the following editions:

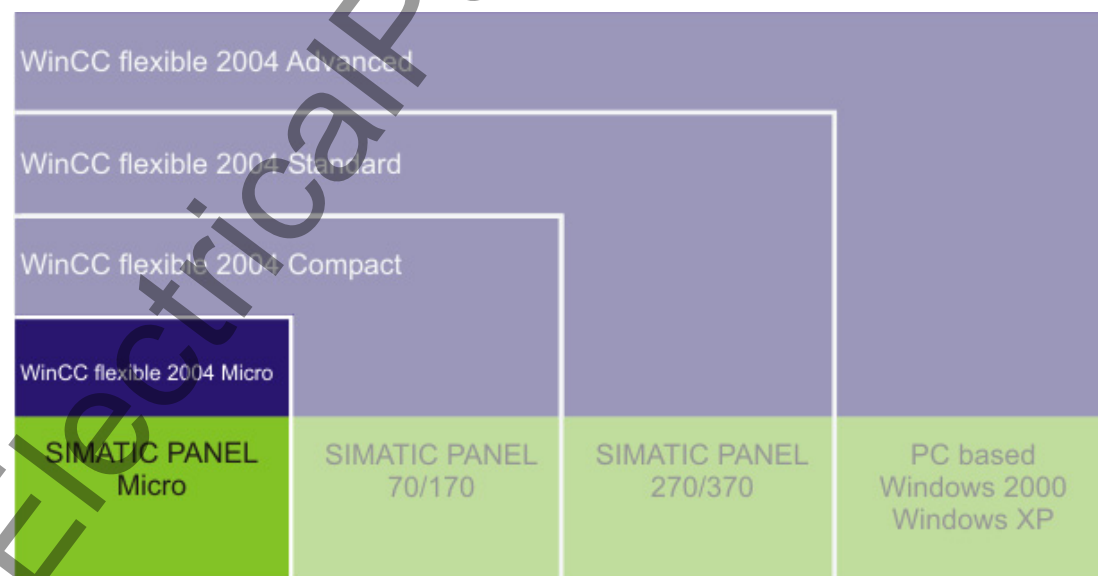


Figure 1-1 The WinCC flexible Editions

## WinCC flexible Engineering System

WinCC flexible offers a range of scalable engineering systems which are optimally adapted to the respective tasks involved in configuring a variety of HMI devices and controllers. Each edition supports a wider range of HMI devices and functions, whereby the "Standard" edition can be used to configure HMI devices from the "Compact" edition.

## 1.2 Configuration support

### Customized Setup of the Configuration User Interface

WinCC flexible allows you to customize the position and behavior of windows and toolbars. This provides the possibility of setting up the working environment according to individual 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 your working environment opens, it is identical to the configuration when last closed. Your configuration is also used when you open a project that was previously edited by a different project planner.

### Resetting the Working Environment

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

## Installation options for a multilingual configuration

During configuration in WinCC flexible, it is possible to switch the language of the configuration user interface, e.g. when several configuration planners of different nationality work with WinCC flexible. During the installation of WinCC flexible, you select the languages to be later available.

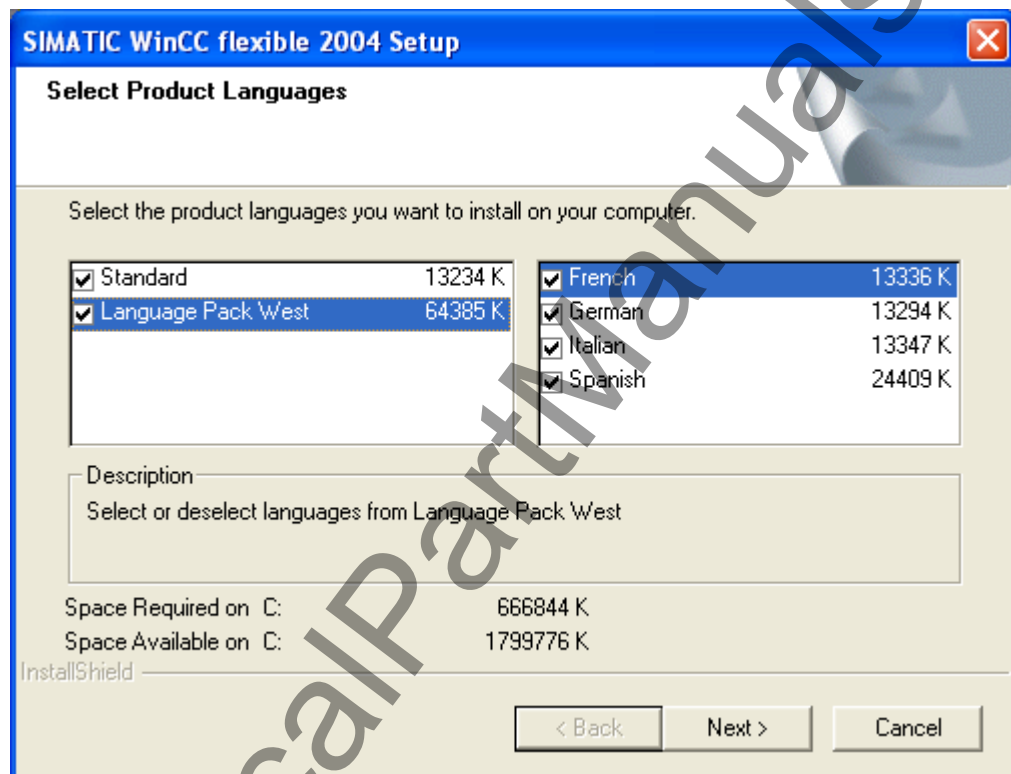


Figure 1-2 Language selection in the Setup of WinCC flexible

English is generally installed as the user interface language. You can select additional languages when you install WinCC flexible.

www.ElectricalPartManuals.com



# WinCC flexible Engineering System

## 2.1 Basic principles on the programming interface

### Principle

The HMI software WinCC flexible is designed for future-proof automation concepts at machine level with straightforward and highly efficient engineering features.

To start WinCC flexible, either click the desktop icon on the configuration computer or select it from the Windows Start menu.



Figure 2-1 Desktop icon

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.

## 2.2 Program user interface of WinCC flexible

### 2.2.1 WinCC flexible user interface elements

#### Introduction

The WinCC flexible working 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.

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.

### Note

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

## Elements of WinCC Flexible

WinCC flexible consists of the following elements:

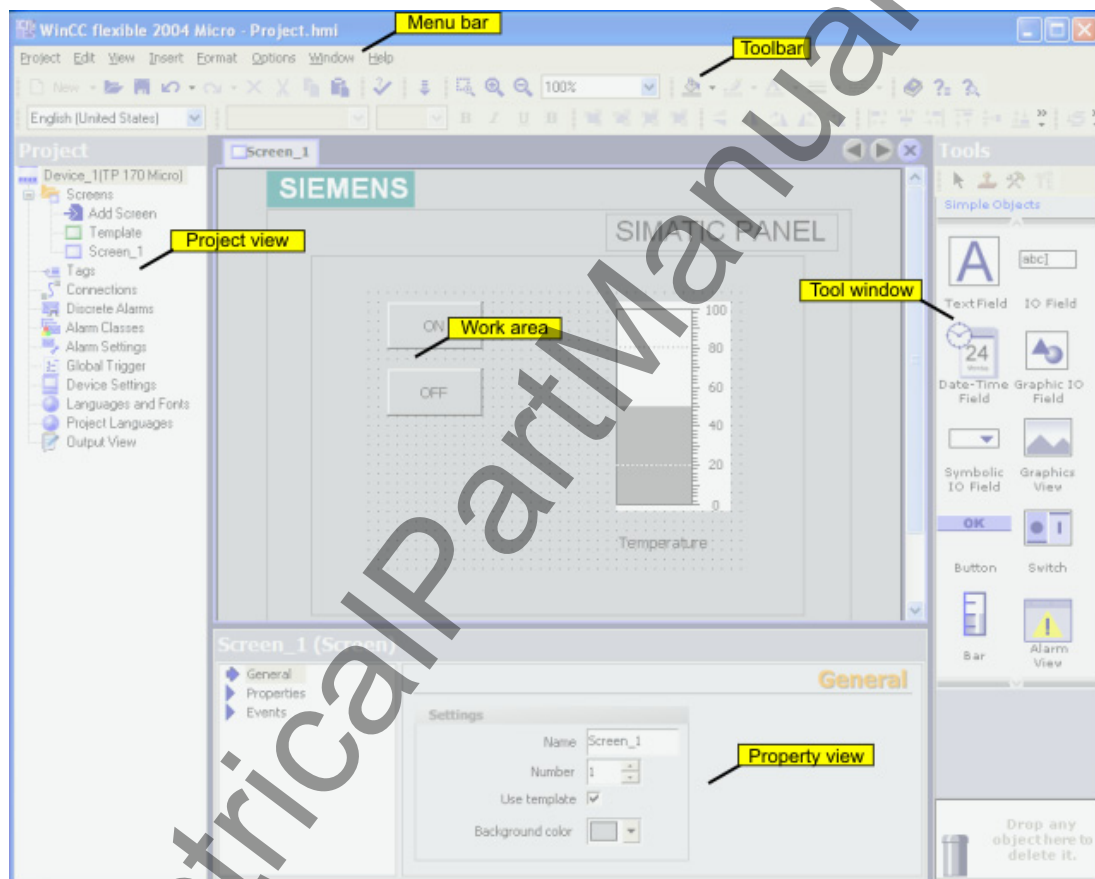


Figure 2-2 WinCC flexible Workbench

The illustrated elements are described in the subsequent chapters.

## 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.

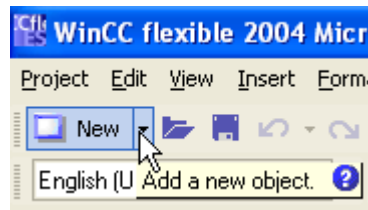


Figure 2-3 Tooltip

### 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:

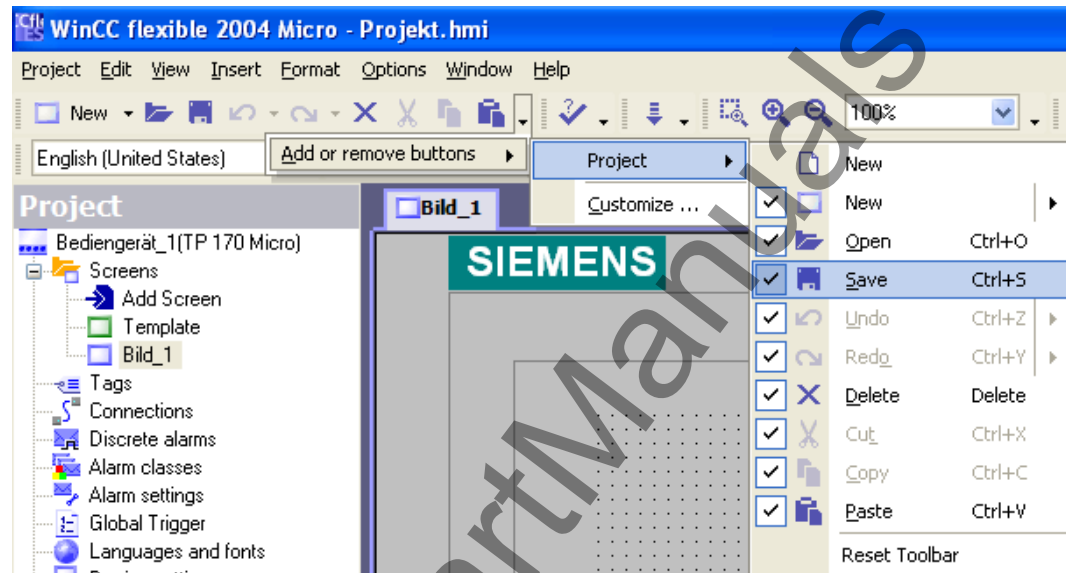
Menu	Brief description
"Project"	Contains commands for project management.
"Edit"	Contains commands for clipboard and search functions.
"View"	Contains commands for opening / closing elements, and for zoom / layer settings. To reopen a closed element, select the "View" menu.
"Paste"	Contains commands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

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



Use the arrow buttons on the toolbar to open the the toolbar menus. Enable or disable the buttons as you wish.

- Changing the position

Position the mouse pointer on the toolbar. Hold down the mouse button and move the toolbar to the desired position.

## 2.2.3 Work area

### Introduction

Project objects are edited in the work area:

- in tabular form, for example tags and alarms.
- graphics, for example objects in a process screen.

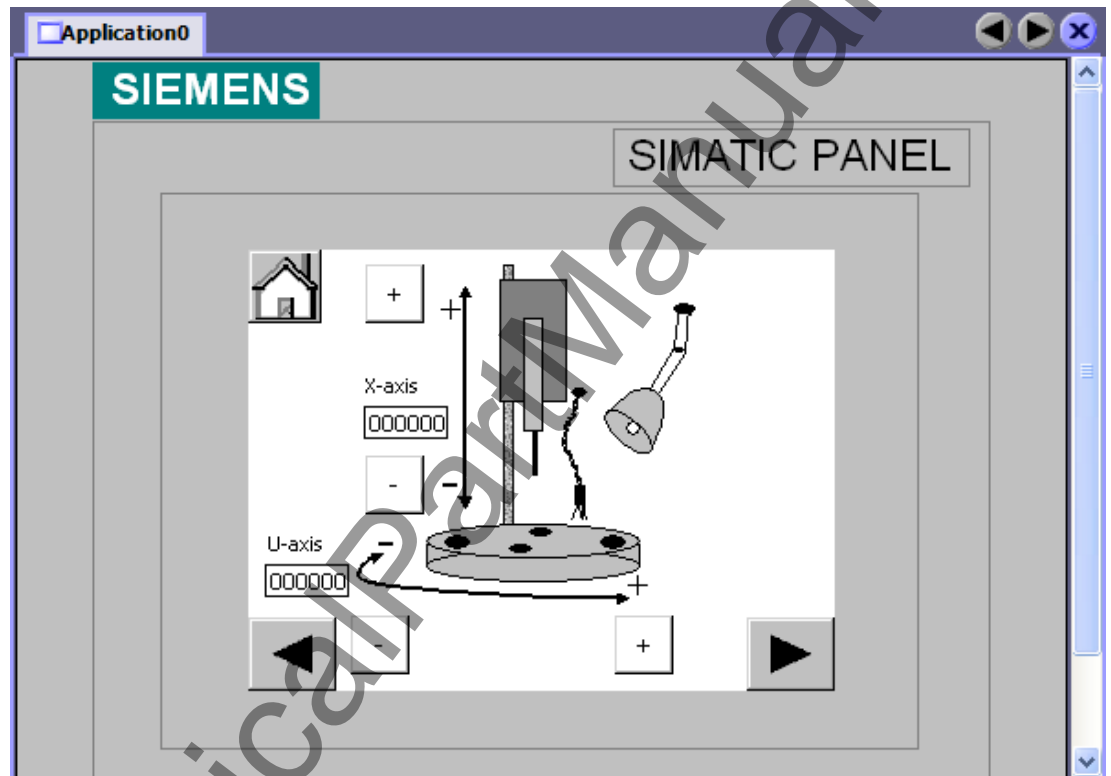


Figure 2-4 Process screen TP 170micro

## Description

Each editor is opened in a separate tab control on the work area. For the "Screens" editor, each element is displayed in a separate tab. Only one tab is active when several editors are open simultaneously. To select a different editor, click on the relevant tab control in the work area. For simple identification, the tab control of a tabular editor displays the name of the editor. In the case of "Screens" editor, the name of the current element is indicated, e.g. "Screen1".

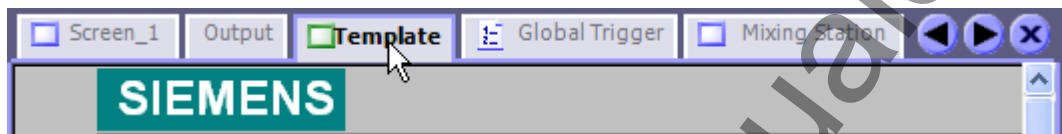


Figure 2-5 Tabs in the work area

## Navigation arrows

If the tab controls cannot be fully displayed in the work area, the navigation arrows become active in the work area.



To access the tab controls that are no longer displayed in the work area, click on the corresponding navigation arrow.

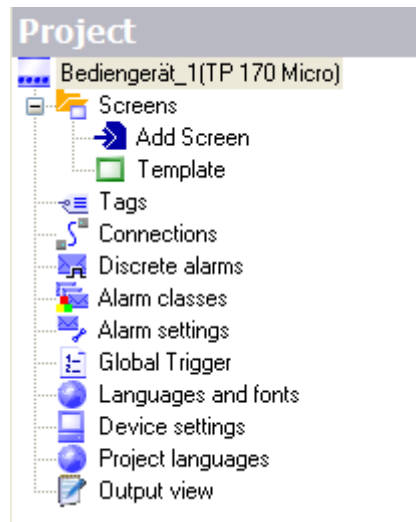
## Closing the Editor

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

## 2.2.4 Project window

### Introduction

The project window is the central control point for project editing. All component parts and available editors in a project are displayed in a tree structure in the project window and can be opened from there. Each editor is assigned a symbol which you can use to identify the corresponding objects. In the project windows you have access to the device settings of the HMI device, the language settings and the output window.



## Description

The project window displays the project structure:

- Process screens in the "Screens" folder
- Editors for editing objects of a project
- Device settings for the HMI device.
- Language support and the output window

The project window is used to create and open objects for editing. Context menus, which consist of the most important commands, are available for all items in the project window. You can also open an editor by double-clicking on the corresponding item in the project window.

## 2.2.5 Properties window

### Introduction

The Property window is used to edit the properties of an object selected from the work area. The content of the "Property window" is based on the selected object. The Property window is only available in specific editors.

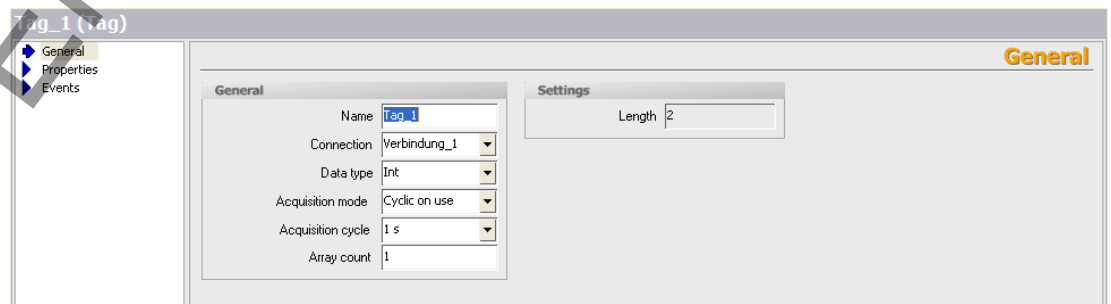


Figure 2-6 Properties window

## Description

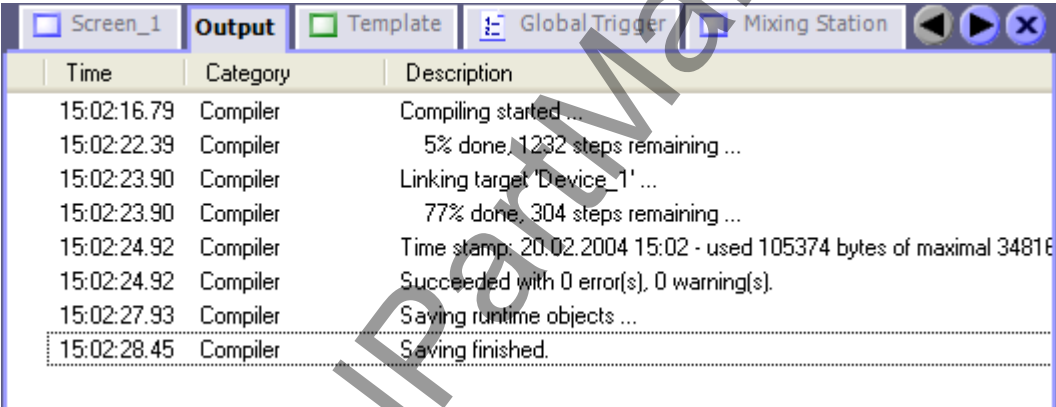
The "Property View" shows the properties of the selected object organized in categories. The changed values take effect directly after exiting from the input field.

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

## 2.2.6 Output window

### Application

The output window displays system alarms generated, for example, in a project test run or during the consistency check of a project.



Time	Category	Description
15:02:16.79	Compiler	Compiling started ...
15:02:22.39	Compiler	5% done, 1232 steps remaining ...
15:02:23.90	Compiler	Linking target 'Device_1' ...
15:02:23.90	Compiler	77% done, 304 steps remaining ...
15:02:24.92	Compiler	Time stamp: 20.02.2004 15:02 - used 105374 bytes of maximal 34816
15:02:24.92	Compiler	Succeeded with 0 error(s), 0 warning(s).
15:02:27.93	Compiler	Saving runtime objects ...
15:02:28.45	Compiler	Saving finished.

Figure 2-7 Output window

## Description

The "Output window" normally displays system alarms in the order of alarms. Different symbols are used to identify system alarms as notification, warning or fault messages. Using the context menu, for example, you can jump to the location of an error or delete the system alarms in a specific category. The categories define the corresponding WinCC flexible module which has generated a system alarm. For example, system alarms for the "Compiler" category are generated during the consistency check.

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

The "Output window" shows all system alarms of the last action. A new action overwrites all previous system alarms.



## 2.3 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

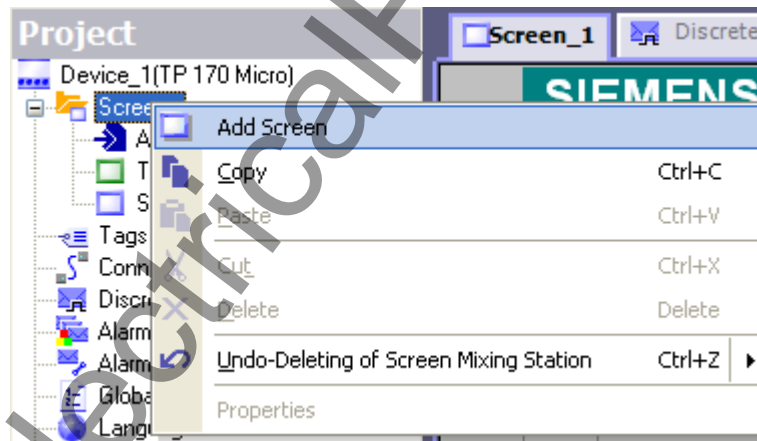
Drag-and-drop makes configuring much easier. To configure a screen change, drag and drop the required process screen onto the process screen displayed in the work area. This generates a button configured to contain a corresponding screen change function.

The drag-and-drop function is available for all objects in the "Project window" and "Object window." 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 displays the commands you can execute in the relevant situation.



### 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)	Start an editor in the project window..
<Left mouse button+drag-and-drop>	Generates a copy of the object in the "project window".

## 2.4 Keyboard operation

### 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.

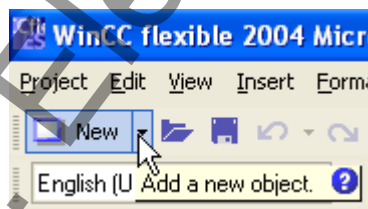
Shortcut keys	Effect
<Ctrl+Tab>/<Ctrl+Shift+Tab>	Activates the next/previous tab control in the work area.
<Ctrl+F4>	Closes the active view in the work area.
<Ctrl+C>	Copies a selected object to the clipboard.
<Ctrl+X>	Cuts an object and copies it to the clipboard.
<Ctrl+V>	Inserts the object stored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

## 2.5 Working with WinCC flexible

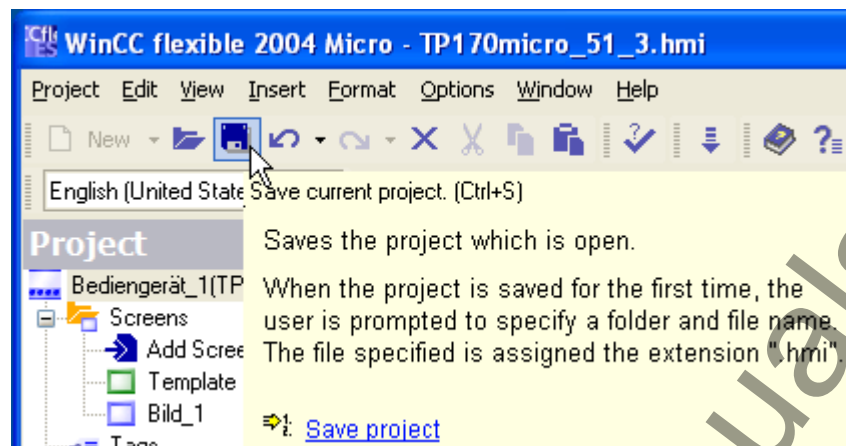
### 2.5.1 Displaying help

#### Context-sensitive help

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



If a question mark is next to the Tooltip, context-sensitive help is available for this user interface element. To call in 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 directly to a detailed description in the online help.

## 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 an opened search tab. Enter the desired search term.

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

The online help is opened in a separate window. The WinCC flexible Portal opens on the home page. The WinCC flexible Portal is organized with overview topics. The overview topics refer to subordinate topics that bring you directly to the information you are looking for.

## 2.5.2 Editor properties

### Introduction

WinCC flexible provides a special editor for each configuring task. WinCC flexible differentiates between two different types of editors: the "Screens" editor as a graphical editor and the tabular editors such as the "Tags" editor.

## Screen editor

The "Screens" editor shows all of the screens contained in the project in the project window of the "Screens" folder. When screens are opened, each screen is opened in a separate window in the work area.

## Tabular Editors

Tabular editors, such as the "Tags" editor, only display the objects contained in the work area. The objects contained there are displayed in a table. You can edit the objects either directly in the table or in the properties window.

## 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 in the "Screens" editor, for example, the change has a direct effect in the "Tags" editor.

- Accepting changes to the project data

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

- Revert working steps or restoring

Every editor has an internal list in which the 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.

## 2.5.3 Opening an editor

### Introduction

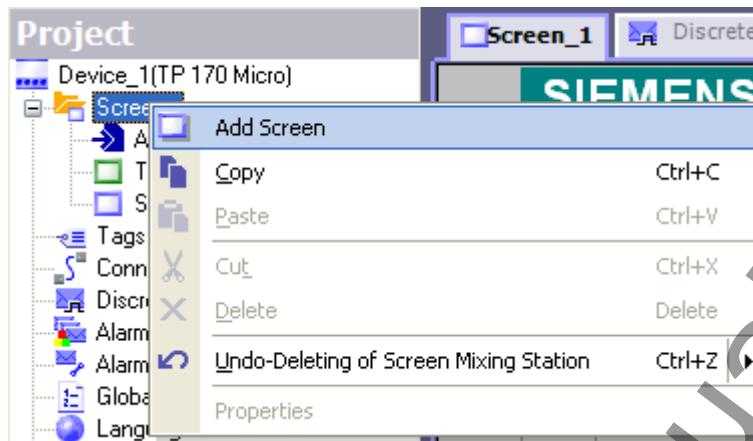
There are several ways to start the editors in WinCC flexible. These options vary depending on the editor involved. A maximum of 20 editors can be opened simultaneously.

### Opening the "Screens" editor

The "Screens" editor is started by either creating a new object or opening an existing object.

To create a new object, proceed as follows:

1. Click on "Screens" in the project window with the right mouse button.
2. Select the "Add operating units" command in the context menu.

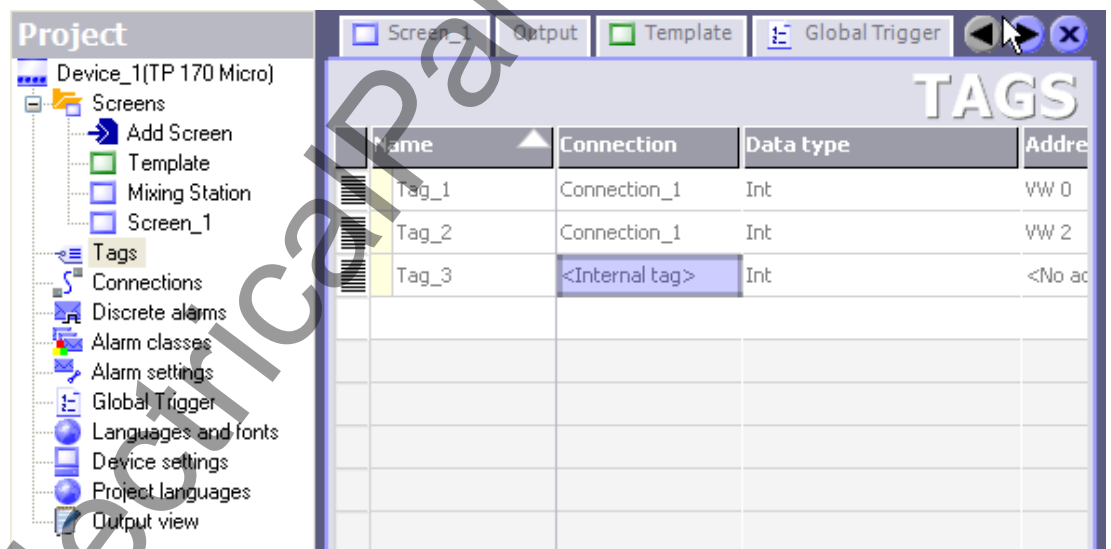


A new screen is created in the project window and displayed in the work area.

To open an existing screen, double-click on the screen in the project window. The screen opens in the work area.

### Opening a Tabular Editor

A tabular editor such as the "Tags" editor is opened by double-clicking on the tabular editor in the project window. The editor then appears in the work area.



A tabular editor can also be activated using the associated context menu.

### Alternative procedure

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

## 2.5.4 Editing several projects with WinCC flexible

### Principle

WinCC flexible only allows one project to be open for editing at any time. If you wish to copy projects globally, restart WinCC flexible and then open the desired project.

Each opened WinCC flexible is shown in the Windows toolbar:



## 2.5.5 Object list

### Introduction

The object list is a helpful feature for configuration tasks in WinCC flexible. You can use the object list to find an existing object of the required object type and configure it directly at the place of use. You can also use the object list to create new objects at the place of use.

### Opening the object list

Objects are usually edited in the properties window but can also be edited directly in the table in the work area when tabular editors are used. The object list opens if WinCC flexible needs a connection to an object. When, for example, you wish to configure a tag for the object, click in the selection field. From the selection field, the object list opens offering you all of the available tags in the project for selection.

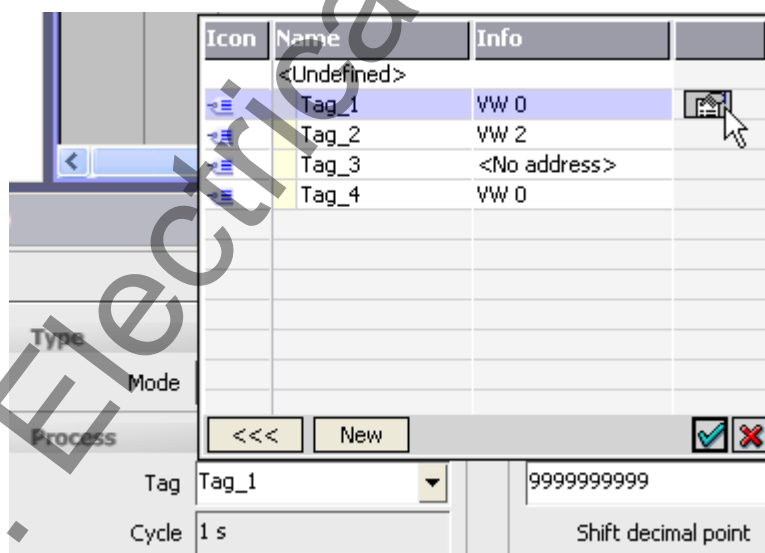



Figure 2-8 Object list

Select the required tag. Confirm your selection by pressing the  button.

## Working with object lists

When a suitable object is not available in the project, create a new one using the object list. To create a new object, click on the "New" button in the object list.

A new object is created and the corresponding dialog for configuring this object opens. Configure the newly created object and close the configuration dialog.

You can also open and configure an existing object from the object list. Select the object from the object list. The icon for editing is displayed in the right column. Click on the icon. The corresponding dialog for editing the object opens. Edit the properties of the object and close the configuration dialog.

## 2.5.6 Function list

### Introduction

You can use function list to link the execution of system functions to an occurring event. 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. Events occur only when the project is in runtime. Events are, for example:

- Value changes of a tag
- Pressing of a button
- Alarm occurrences

You can configure a function list precisely on every event. You can configured several functions in the function list. When the configured event occurs in runtime, the function list is completed from top to bottom. In order to avoid waiting times, system functions with a longer running time, file operations, for instance, are processed simultaneously. A subsequent system function can already be performed even though the previous system function has not yet been completed.

### Configure function list

In WinCC flexible, open the editor containing the object for which you wish to configure the function list. Select the object with the mouse. Click in the property window in the group "Events" on the event on which you want to configure the function list. The function list opens in the properties window.

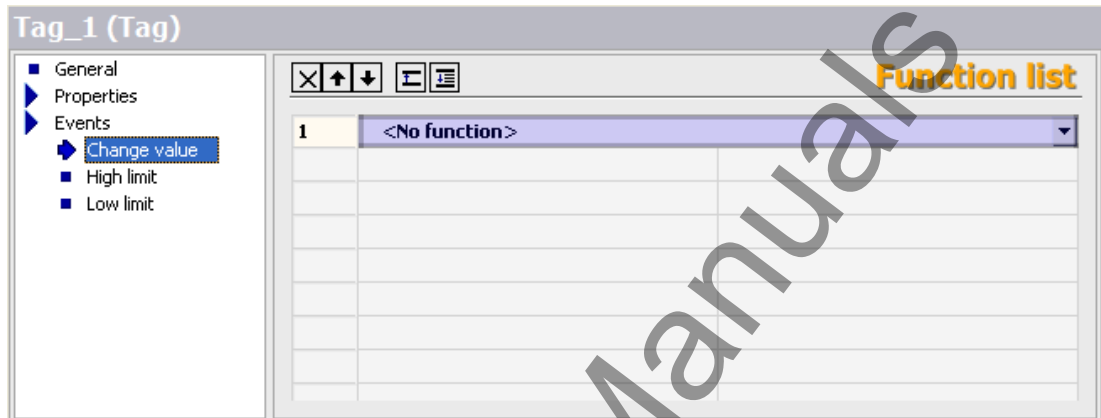


Figure 2-9 Function list

"No function" appears in the first line of the function list when no function has been configured for the object. Click on the "No function" field. A selection button is displayed. Use the selection button to open the list of available system functions. The system functions are arranged in the selection list according to categories.

Select the desired system function.

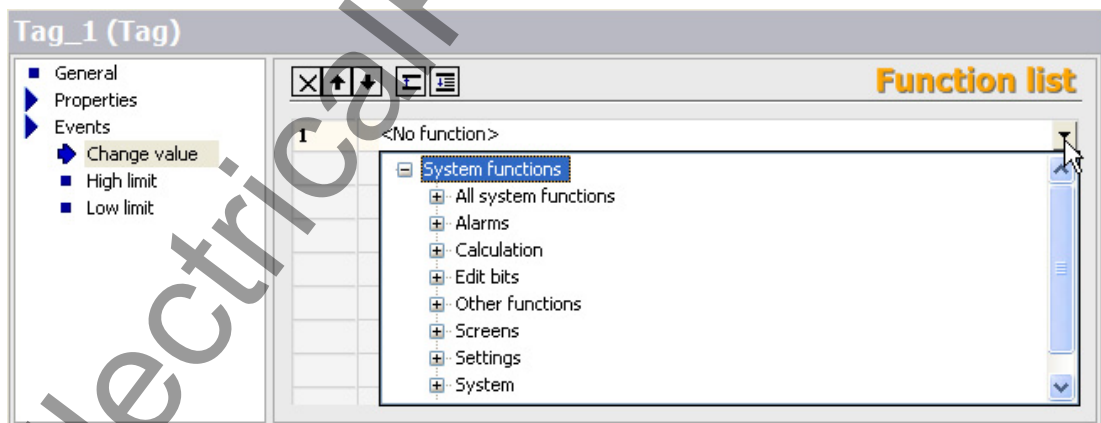


Figure 2-10 System functions

If a parameter is needed for the system function, the entry "No value" appears in the next line. Click on the "No value" field. A selection button is displayed. Use the selection button to open the object list and select the required parameter.



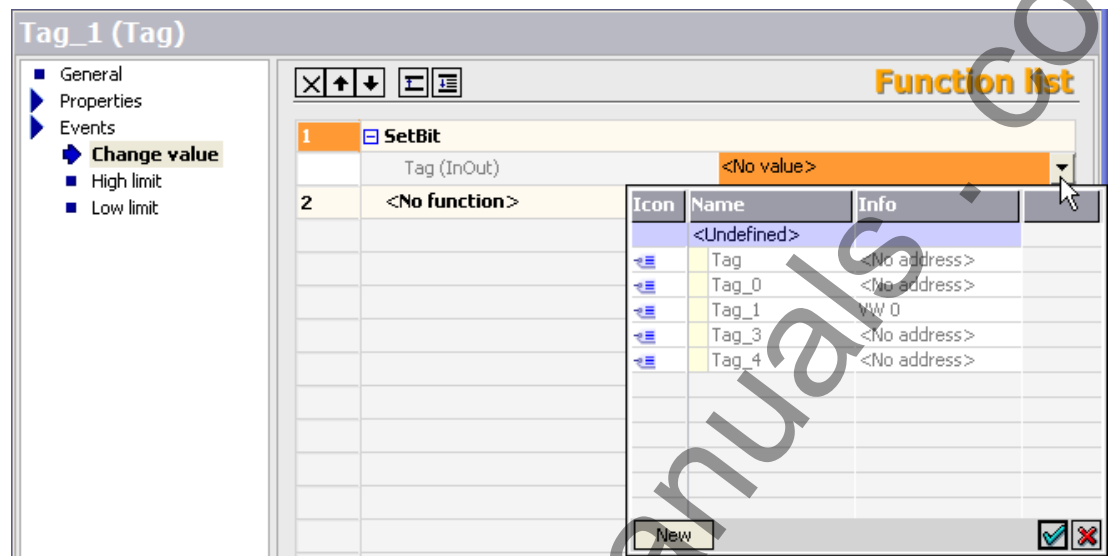


Figure 2-11 Parameter selection

The function is configured in the function list. Configure other functions as required. Use the arrow buttons at the top of the function list to change the order of the configured functions. Select a function and move the function up or down by clicking on the arrow buttons. To delete a function, mark the function with the mouse and press the <Del> key.

www.ElectricalPartManuals.com

## Working with tags

### 3.1 Basic principles

#### 3.1.1 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.

##### Principle

An external tag is the image of a defined memory location in the PLC. You have reading and writing 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.

##### Data types

SIMATIC S7-200

Name	Object	Data type
Tag	V	Char, Byte, Int, Word, DInt, DWord, Real, Bool, StringChar
Input	I	Char, Byte, Int, Word, DInt, DWord, Real, Bool, StringChar
Output	Q	Char, Byte, Int, Word, DInt, DWord, Real, Bool, StringChar
Bit memory	M	Char, Byte, Int, Word, DInt, DWord, Real, Bool, StringChar
Timer	T	Timer
Counter	C	Int

##### Using area pointers

In addition to the external tags, the area pointer "Date/Time PLC" can be used for communication between the HMI device and PLC. The area pointer is used to display the time of the PLC on the HMI device.

You can activate the area pointer in the "Connections" editor. Open the "Area pointer" tab there in the lower section of the work area. Select the connection to the PLC in the

"Connection" column. Select the address for the time of the PLC in the "Address" column. This address has to match the one configured for the time in the PLC program.

To display the time of the PLC on the HMI device, configure a date/time field in a process screen and select the "Display system time" option in the "General" category in the properties window.

### 3.1.2 Internal tags

#### Introduction

Internal tags do not have any connection to the PLC.

#### Principle

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

The following data types are available for internal tags.

Char, Byte, Int, UInt, Long, Ulong, Float, Double, Bool, String and DateTime.

## 3.2 Elements and basic settings

### 3.2.1 "Tags" editor

#### Introduction

Create a tag "Tags" editor. The tag is assigned a basic configuration when it is created. You can use the "Tags" editor to adapt the configuration of the tag to the requirements of your project.

#### Opening

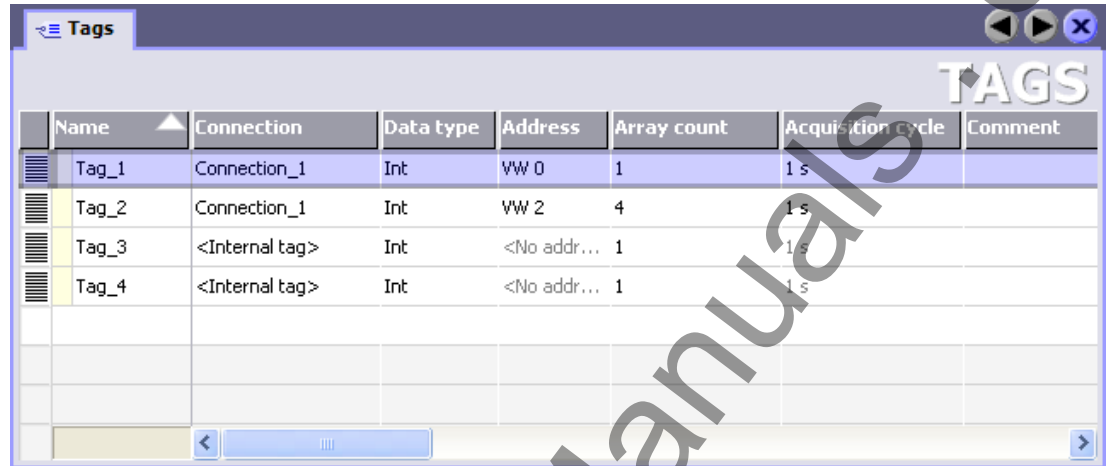
Open the "Tags" editor by selecting the "Tags" item in the project window and then open the context menu with the right mouse button. In the context menu, select the command:

- Open editor
- or
- Add tag

You can also open the "Tags" editor by double-clicking on the "Tags" item in the project window.

## Structure

The "Tags" editor displays all tags of the project.



The screenshot shows the 'Tags' editor window with a table containing the following data:

Name	Connection	Data type	Address	Array count	Acquisition cycle	Comment
Tag_1	Connection_1	Int	VW 0	1	1 s	
Tag_2	Connection_1	Int	VW 2	4	1 s	
Tag_3	<Internal tag>	Int	<No addr...>	1	1 s	
Tag_4	<Internal tag>	Int	<No addr...>	1	1 s	

Figure 3-1 "Tags" editor

## Work area

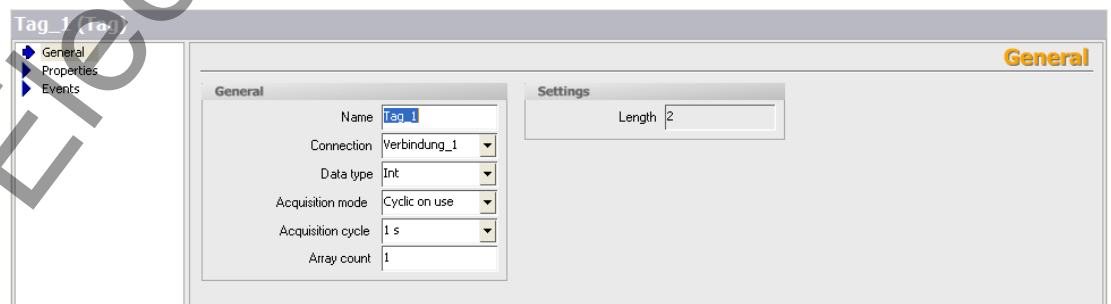
All tags are displayed in a table in the work area. You can sort the table according to the entries in a column by clicking the column header.

You can configure the selection of columns to suit your needs. Move the mouse pointer to a column header and open the context menu by clicking the right mouse button. In the context menu, select which the column to be displayed or hidden. 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.

## Properties window

Here you configure tags.

The left section of the property window shows a variety of categories from which you can select the various subcategories. The fields for configuring the selected properties category are shown in the right section of the properties window.



The screenshot shows the 'Tag\_1 (tag)' property window with the 'General' tab selected. The 'General' section contains the following fields:

- Name: Tag\_1
- Connection: Verbindung\_1
- Data type: Int
- Acquisition mode: Cyclic on use
- Acquisition cycle: 1 s
- Array count: 1

The 'Settings' section contains the following field:

- Length: 2

Figure 3-2 Property window tag

3.2.2 Basic settings for tags and arrays

Introduction

You can configure the properties of tags and array elements using the property window or directly in the respective table cell in the work area.

Structure of the Properties window

The left sections of all property windows show a variety of categories from which you can select the various subcategories. The fields for configuring the properties category are shown in the right section of the properties window.

Property window for tags

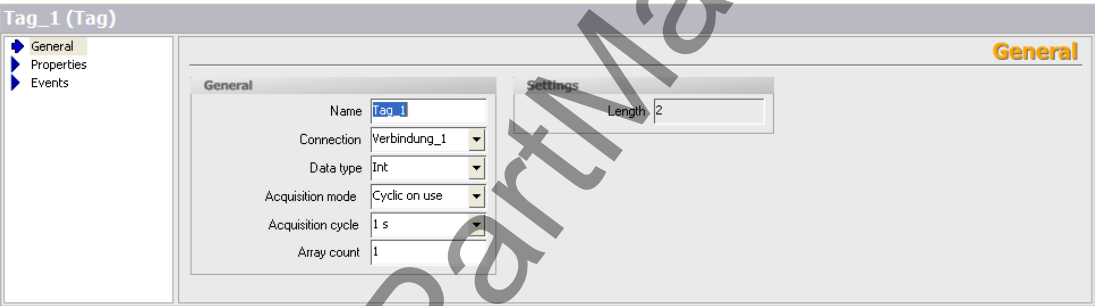


Figure 3-3 Property window tag

You can configure the following properties of the selected tag in the properties window for tags:

Category	Fields
"General"	"Name" "Connection" "Data type" "Acquisition type" "Acquisition cycle" "Array count" "Length"
"Properties"	
"Addressing" (for external tags only)	"Area" "VW" "IW" "QW" "MW" "C"

Category	Fields
"Limits"	"Upper limit - deactivate"
	"Upper limit - constant"
	"Lower limit - deactivate"
	"Lower limit - constant"
	Limit check
"Basic values"	"Update code"
"Comment"	Text field for entering the comment
"Events"	
"Upper limit exceeded"	List of functions that will be processed if the upper limit is exceeded
"Value 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

### Property window for array elements

The properties window of an array element is identical to the properties window of a tag.

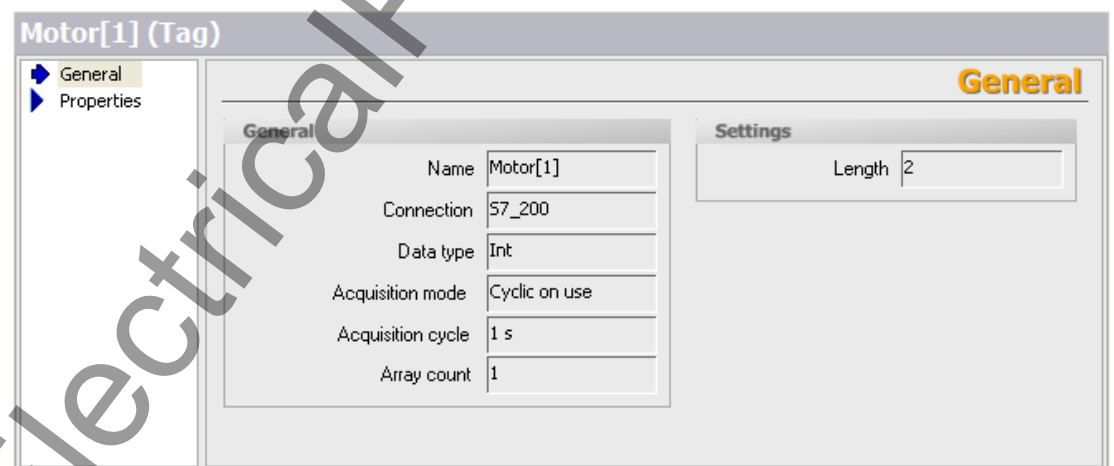


Figure 3-4 Properties window array tag

The properties of an array element are read-only in WinCC flexible Micro. The properties of an array element are taken over by the tags which are composed of the array element.

## 3.3 Working with tags

### 3.3.1 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 configuration.

#### Principle

The following properties can be configured for tags:

- "Name"

Every tag has a name which you can choose. The name must be unique within the project.

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

Create a connection to the PLC for external tags. An external tag is used as an image of the memory area in the PLC and saves the value that is passed from the PLC.

By setting the acquisition cycle, you can determine how often and when a tag is 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 mostly used when working with data of identical form.

- "Comment"

You can enter a comment for every tag.

- "Limits"

You can specify a value range that contains an upper and lower limit for each tag. You can use limits to monitor the value range of a process value configured for the tag.

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.



### 3.3.2 External tags for communication with the PLC

#### Introduction

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

#### Principle

An external tag is used as an image of a defined memory area in the PLC. You have reading and writing 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 following tag properties depend on the configuration options in the properties of the PLC:

- "Addressing"
- "Data type"
- "Acquisition cycle"

#### Addressing

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

Example: You would like to display the status of the output "Q 1.2" of the PLC on the HMI device. To do this, create an external tag and set the output "Q 1.2" as the address.

#### 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. The interval for regular updates is set with the acquisition cycle. For cyclical acquisition, you can choose one of the predefined cycles or define one yourself. The fastest cycle time of the SIMATIC S7-200 controller is 100ms.

However, an external tag can also be updated continuously. Please note that frequent reads cause an increased communication load.

### 3.3.3 Tag limit values

#### Introduction

You can define a value range for numerical tags.

#### Principle

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

If the process value of a tag violates one of the limits, you can have an alarm, e.g. a warning, triggered.

If the process value, which should be stored in the tag, violates a limit, an alarm can be output. If the process value lies outside the value range, a function list for alarm output can be executed. If the operator enters a value for the tag that is outside the configured value range, this value will not be saved.

#### Application example

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

### 3.3.4 Updating the tag value in runtime

#### Introduction

Tags contain data which change in runtime. The value changes are handled in different ways for internal and external tags.

#### Principle

The value of a tag is set to 0 when runtime starts. Tag value changes can occur in runtime.

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

- Executing a system function to change the value, e.g. the system function "SetValue"
- An operator input, e.g. in an IO field
- Changing external tags with the PLC

In the case of external tags, you must ensure that the HMI device is informed of any value change made by the PLC. The tag value must also be updated on the HMI device.

The value of an external tag is updated in the following manner:

- Updating through an acquisition cycle

Normally, tags are updated after an acquisition cycle as long as the tag appears in a screen. The acquisition cycle determines the interval at which the value of the tag will be updated on the HMI device. For cyclical acquisition, you can choose one of the predefined cycles or define one yourself. The lowest value for a cycle depends on the

HMI device. For the TP 170micro the lowest value is 100 ms. The values of all other cycles are always an integer multiple of the smallest value.

- When the setting "Continuous update" is activated

If this setting is activated, the tag will be updated continuously in runtime, even if it is not found in the currently open screen. The setting is activated, for example, for tags which are configured to trigger a function list in the event of a value change. The cycle time configured for the tag is used even with continuous updating.

Only use the "Continuous update" setting for tags that must truly be updated continuously. Frequent reads increase the communication load.

## 3.4 The basics of arrays

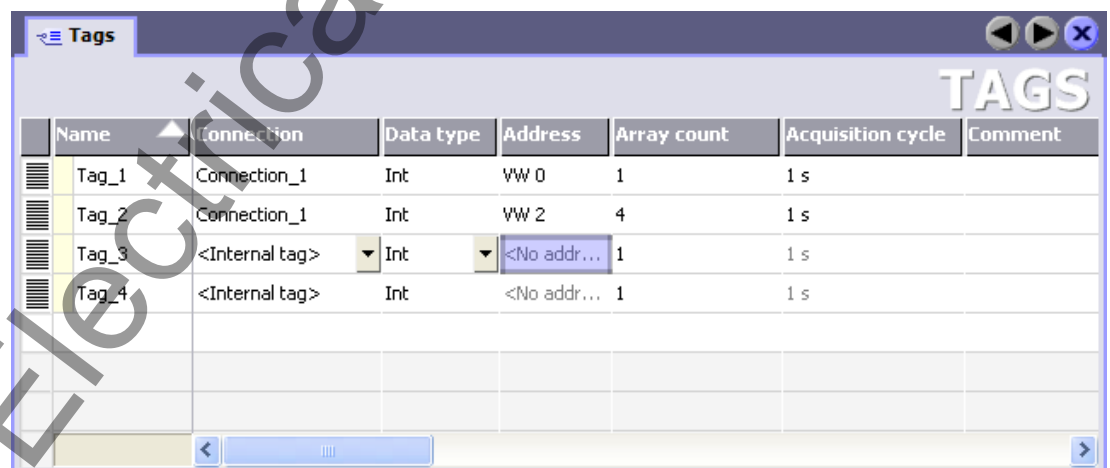
### Introduction

If you want to save a large amount data of the same type, create a tag from multiple elements of the same type (array elements). Array elements occupy a continuous address area.

A tag assembled from array elements is called an array tag. Using an index tag, you can control which array element will be accessed.

### Principle

Array tags consist of a definable number of array elements in which data of the same type can be stored. Each of a tag's array elements occupies the same amount of memory. All of an array tag's array elements are saved consecutively in memory.



Name	Connection	Data type	Address	Array count	Acquisition cycle	Comment
Tag_1	Connection_1	Int	VW 0	1	1 s	
Tag_2	Connection_1	Int	VW 2	4	1 s	
Tag_3	<Internal tag>	Int	<No addr...	1	1 s	
Tag_4	<Internal tag>	Int	<No addr...	1	1 s	

---

#### Note

All read and write accesses will always read or write all of the array elements of a tag. If an array tag is connected with a PLC, all of the contents will be transferred whenever there is a change. This is the reason that the HMI device and PLC cannot write to the same array tag at the same time.

---

#### Array element properties

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

### 3.5 The basics of cycles

#### Introduction

Cycles are used to control project sequences that are run at regular intervals. There is an acquisition cycle and an update cycle. A cycle is configured with a cycle time that determines the time intervals repeated within the cycle.

#### Principle

In runtime, actions that are performed at regular intervals are controlled by cycles. Typical applications for cycles include the acquisition of external tags 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 unnecessarily increase the communication load of the process.

- Update cycle

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

The lowest value for a cycle depends on the HMI device. For the TP 170micro the lowest value is 100 ms. The values of all other cycles are always an integer multiple of the smallest value.

#### Application example

You can use cycles, for example, to regularly update the process value display.

## Creating screens

### 4.1 Basic principles

#### 4.1.1 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, predefined objects are available to support you in visualizing processes and setting process values.

##### Application example

The figure shows a process screen which was created in WinCC flexible. This screen can be used to operate and monitor an oven. The operating mode can be set to "Automatic" or "Manual" using buttons. In the "Manual" mode, the "Temp. +" and "Temp. -" buttons can be used to change the oven temperature. The bar is an analog display of the current oven temperature. The setpoint temperature is displayed in the I/O field "Setpoint temp" and can be changed by input on the HMI device. The "Actual temp" output field is a digital representation of the current temperature. You can use the arrow buttons to navigate to the previous or next process screen in a defined sequence. The button showing the house brings you to the home screen.

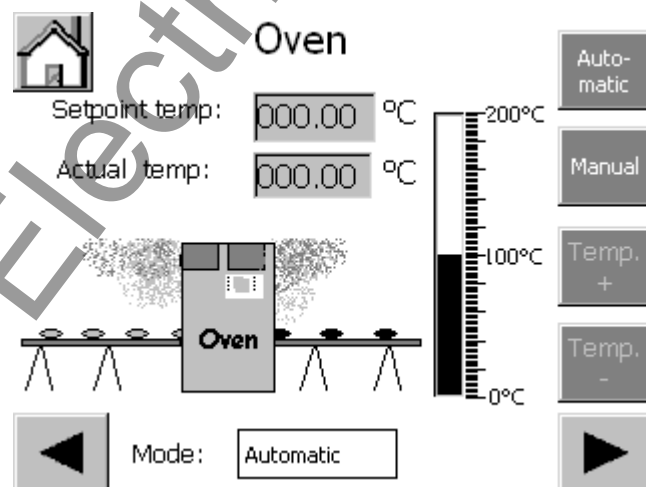


Figure 4-1 Configuration examples

## Screen design

Insert an object you need to represent a process into your screen. Configure the object to correspond to the requirements of your process.

A screen may consist of static and dynamic elements.

- Static elements such as text do not change their status in runtime. In the example of the oven, the labels on the digital temperature displays are static.
- Dynamic elements change their status based on the process. They visualize current process values which are output from the memory of a PLC or HMI device, in the form of a bar in the oven example. Operator input boxes also belong to the category of dynamic elements.

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

## Screen properties

The layout of the screen corresponds to the layout of the user interface of the HMI device. Properties such as the screen resolution and available fonts also depend on the selected HMI.

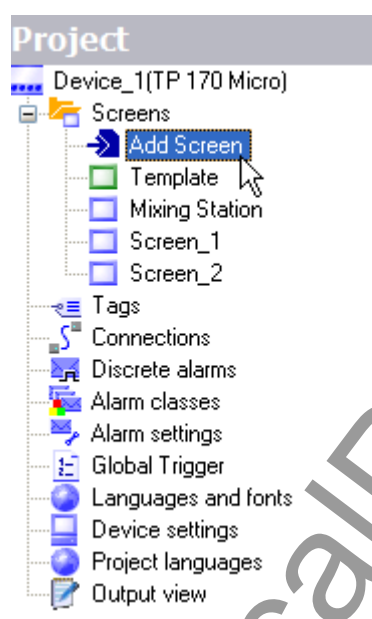
## 4.1.2 "Screens" editor

### Introduction

You configure your screens in the "Faceplate designer". This editor is formed by the combination of a graphic programming software and a process visualization tool. You open the "Screens" editor from the project window.

### Opening

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



The work area opens with a new screen.

## Structure

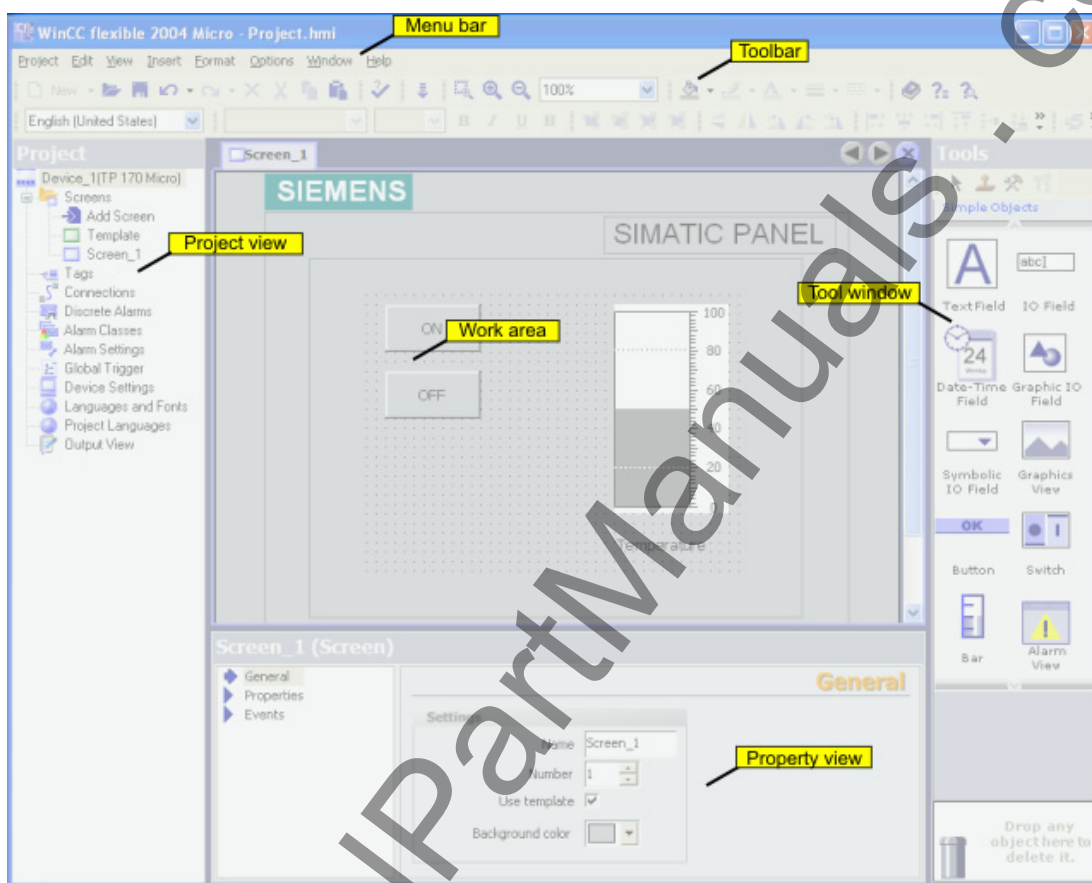


Figure 4-2 "Screens" editor

### 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.

### 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, e.g. graphic objects or operator control elements.



## Properties window

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

- When an object is selected, you can edit the properties of the selected object in the properties window.
- If you have not selected an object on the active screen, the properties of this screen are shown and can be edited in the properties window.

## 4.1.3 Procedures

### Procedures

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

- Planning the structure of the process display: How many screens and what trees are needed?

Example: Partial processes can be visualized in separate screens and merged in a master screen.

- Plan the navigation between the individual screens.
- Adapt the template.

The template which is stored in WinCC flexible for the selected HMI device applies to all your project screens. You can configure objects in the template from a central location. Insert objects needed in every process screen into the template. You only need to configure the properties of the objects once. The template already includes an alarm window that shows occurring system alarms. Delete the alarm window from the template if you do not need it. The objects configured in the template are also displayed in every process screen opened in the work area. You can disable the display of objects from the template with the menu command "Options > Settings > "Screens" editor > Settings for "Screens" Editor.

- Create the screens.

## 4.1.4 Managing screens

### Introduction

In the project window, you can move, copy, rename and delete screens within a project.

### Copy screen

You can copy a screen in WinCC flexible, for example, to use it again as a screen template.

1. Select the screen from the project window.
2. Select "Copy" from the context menu to copy the screen to the clipboard.
3. Select the screen folder in the project window and open the context menu.

4. Select "Paste" from the context menu to paste the screen.

A copy of the screen is inserted with the same name and an incremented sequential number.

As an option, press <CTRL> while you drag the screen to the required position.

### Rename screen

1. Select the screen from the project window.
2. Select the "Rename" command from the context menu.
3. Type in a new name.
4. Press <ENTER>.

As an option, use the <F2> function key to rename the screen.

### Delete screens

1. Select the screen from the "project window."
2. Select the "Delete" command from the context menu.

The screen and all its objects are deleted from the current project.

## 4.2 Objects of the "Screens" editor

### 4.2.1 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.





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 text field and standard operator control elements, such as an I/O field.

## Simple objects

Icon	Object	Note
	"Text field"	You can enter one or several lines of text in a "Text field" and define the font and the font color. You can add a background color or pattern to a text field.
	"IO field"	<p>An IO field may be used to perform the following runtime functions:</p> <ul style="list-style-type: none"> <li>• Output of the values in a tag</li> <li>• Operator input of values; these input values are saved to a tag.</li> <li>• Combined input and output; the operator can here edit the output value of the tag in order to set a new value.</li> </ul> <p>You can define limits for the tag values shown in the IO field.</p> <p>Set "Hide input" if you want to hide operator input in runtime.</p>
	"Date-time field"	<p>A "Date / time field" may be used to perform the following runtime functions:</p> <ul style="list-style-type: none"> <li>• Output of the date and time</li> <li>• Combined input and output; the operator can here edit the output values in order to reset the date and time.</li> </ul> <p>The system time or a corresponding tag may be used as source to define the date and time.</p> <p>The date can be output in extended format, e.g. Tuesday, 31 December 2003, or in short format, e.g. 12.31.2003.</p>
	"Graphic IO field"	<p>A "Graphic IO field" may be used to perform the following runtime functions:</p> <ul style="list-style-type: none"> <li>• Output of images</li> <li>• 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."</li> </ul> <p>Example of its use as output field:</p> <p>To indicate the runtime status of a valve, the "Graphic IO field" outputs the image of a closed or open valve.</p>
	"Symbolic IO field"	<p>The "Symbolic IO field" may be used to perform the following runtime functions:</p> <ul style="list-style-type: none"> <li>• Output of text list entries</li> <li>• Combined input and output; the operator can here select a text from a text list in order to change the content of the "Symbolic IO field."</li> </ul> <p>Example of its use as combination input and output field:</p> <p>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.)</p>
	"Graphic view"	<p>The "Graphic view" shows you on one screen all of the graphics created by means of an external graphic programming tool. Graphics can be shown in the following formats: ".emf", ".wmf", ".dib", ".bmp", ".jpg", ".jpeg", ".gif" and ".tif".</p> <p>In the "Graphic view", you can also integrate graphics of other graphic programming tools as OLE (object linking and embedding) objects, for example. OLE objects can be called and opened for editing directly from the "Properties" window of the graphic programming interface they were created in.</p>
	"Button"	The operator can use a button to control a process. You can configure buttons, for example, with functions for acknowledging alarms. You can integrate a button into your process by assigning it dynamic properties.

Icon	Object	Note
	"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 process variables, for example, the filling levels.
	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 in runtime. The alarm window can be automatically activated when a new alarm occurs. You always edit the template to configure the Alarm window. The template already includes an alarm window for displaying system alarms.

## 4.2.2 Object groups

### Principle

You can organize multiple object in a group. To do this, draw a frame around the desired objects with the mouse. Open the context menu and select the "Group" command. In your screen, you edit an object group in the same way you edit a single object. Only the properties shared by all objects can be edited in the object group. Changing the properties in the group affects every object in the 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. To activate the single-object editing mode, click on an object in the created object group; the object group is selected. Open the context menu and select the "Edit" command. The object group is displayed with a red frame. Click on the object in the object group; the single-object editing mode is activated. The properties of the object are displayed in the property window.

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.

## 4.3 The advantage of layers

### Layers

The layers and the nesting depth of the objects form a feature which allows a 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 rear within the level. Each further object is placed one position further towards the front. The position between objects within the layer can be changed.

### 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 example

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

## 4.4 Security in Runtime

### Principle

Access protection controls access to data and functions in order to protect them against unauthorized operation during runtime. Safety-relevant operations are already limited when a project is being created. A password for access authorization is assigned for this purpose. The authorizations required for operation are configured to the objects.

### Configuring a password

You can configure password assignment in the device settings of the WinCC flexible project. Double click on the "Device settings" node in the project window. The device settings are opened and displayed in the work area. Enter a password in the "Security in Runtime" area. Confirm the password by entering it a second time in the lower field.

### Configuring access protection

Access protection is configured directly on each object you wish to protect. Access protection can be configured for the following objects:

- IO field
- Date / time field
- Graphic IO field
- Symbolic IO field
- Button
- Switch
- Alarm view

- Alarm window

Activate access protection in the properties window under "Properties > Security". Set the "Enabled" check box there. The user must enter the password in Runtime to be able to operate the object. The password cannot be changed during runtime.

# Creating an alarm system

## 5.1 Basic principles

### 5.1.1 Displaying process and system alarms

#### Introduction

- Process 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 the HMI device or the PLC to display particular system states in these devices.

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

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

A state is reported as soon as it occurs.

### 5.1.2 User-defined alarms

#### 5.1.2.1 Available alarm procedures

#### Alarm procedure in WinCC flexible

The alarm procedure identifies the type of information that triggers an alarm, and thus also the alarm properties.

WinCC flexible Micro supports the discrete alarm procedures.

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.

## Alarm statuses

The following alarm statuses exist for discrete alarms:

- When the condition for triggering an alarm has occurred, the alarm status is "Activated." Once the operator has acknowledged the alarm, it has "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 these statuses can be displayed on the HMI device.

### 5.1.2.2 Acknowledging alarms

#### Introduction

For discrete 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.

The following options can be used for acknowledgement by the operator:

- Buttons in screens

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

#### Alarms requiring acknowledgement

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

Alarm classes essentially define how alarms will appear when displayed on the HMI device as well as the acknowledgement behavior.

#### Acknowledging by the PLC

An alarm can be acknowledged by the control program by setting a particular bit within a tag.

Conversely, a bit can be set within a tag when the alarm is acknowledged by the operator.

### 5.1.2.3 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.



### Available alarm class settings

The following settings can be defined for each alarm class:

- Acknowledge: Alarms in this class must be acknowledged.
- Texts, various shades of gray and flash modes to identify each alarm status when alarms are displayed
- A text placed in front of the alarm number to indicate the alarm class when alarms are displayed on the HMI device.

### Predefined alarm classes in WinCC flexible

- "Error" for discrete alarms that indicate critical or hazardous operating and process states. Alarms in this class must always be acknowledged.
- "Event" for discrete 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.

## 5.1.3 Displaying alarms

### 5.1.3.1 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 is not automatically opened when an alarm occurs. The process screen with the alarm view must be opened in order to view the incoming alarms.

The alarm view can be configured in such a way that it includes only one alarm line. The single-line alarm view is called the "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 opening and reclosing of the alarm window. This allows the alarm window to be configured to open when an alarm occurs.

### 5.1.3.2 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.

The table shows the system functions for alarms and manipulating the alarm display.

System function	Effect
ClearAlarmBuffer	Deletes alarms from the alarm buffer on the HMI device.
EditAlarmScreenMessage	Triggers the event "Edit" for alarms selected in the selected alarm view.  The system function only triggers the event. A function must be configured on "Edit" event for the selected alarm.
AcknowledgeAlarmScreenMessage	Acknowledges the alarms selected in the given alarm view.
ShowAlarmWindow	Hides or shows the alarm window on the HMI device.

The "Configuration Examples" section contains further information about the system function "EditAlarmScreenMessage."

#### Events for alarms and for objects used for alarm displays

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
Alarm view	Enable Disable
Alarm window	Enable Disable

## 5.2 Elements and basic settings

### 5.2.1 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 alarm text can contain output fields for the current values of tags. The alarm buffer retains the current 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
- HMI system alarms

- Alarm triggers

- For discrete alarms: A bit within 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.

These components are freely selected or entered for each alarm.

#### Optional alarm properties

The behavior of an alarm can also be defined by the following properties:

- 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.

## 5.2.2 Editors for Configuring Alarms

### 5.2.2.1 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
- "Alarm classes" for changing alarm classes

#### Changing the column display

The column display can be configured as follows:

- You can show or hide individual columns using the context menu in the column header.
- 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 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 and selecting the corresponding command from the context menu.

### 5.2.2.2 "Discrete alarms" editor

#### Introduction

You create discrete alarms and specify their properties in the "Discrete alarms" editor.

#### Opening

Double-click on "Discrete alarms" in the project window.

## Structure

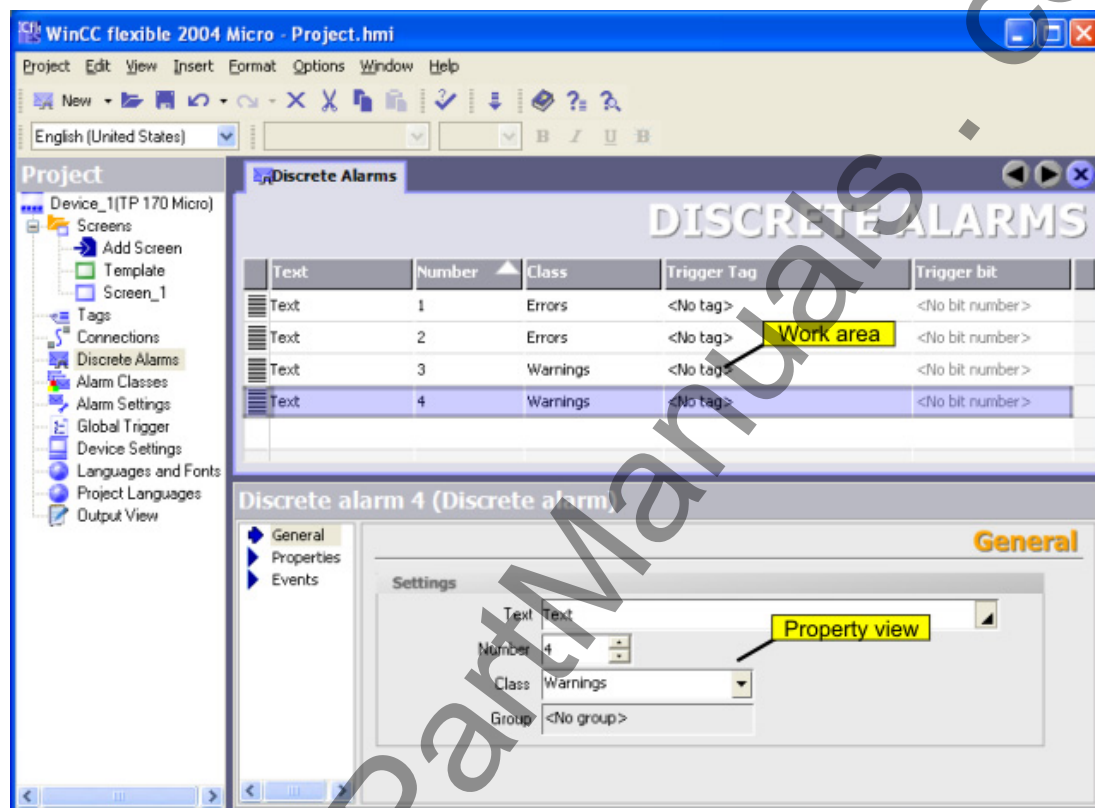


Figure 5-1 "Discrete alarms" editor

## Work area

All the discrete alarms are displayed in a table in the work area. Edit the properties of the discrete alarm directly in the table cell or in the properties window.

## Properties window

Configure the discrete alarm in the property window. Select a discrete alarm in the table in the work area. Select the corresponding category in the left section of the properties window. Make the desired changes in the right section of the properties window.

### 5.2.2.3 "Alarm classes" editor

#### Introduction

Define the properties of the alarm classes in the "Discrete alarms" editor.

#### Opening the "Alarm classes" editor

Double-click in the project window on "AlarmClasses."

## Structure

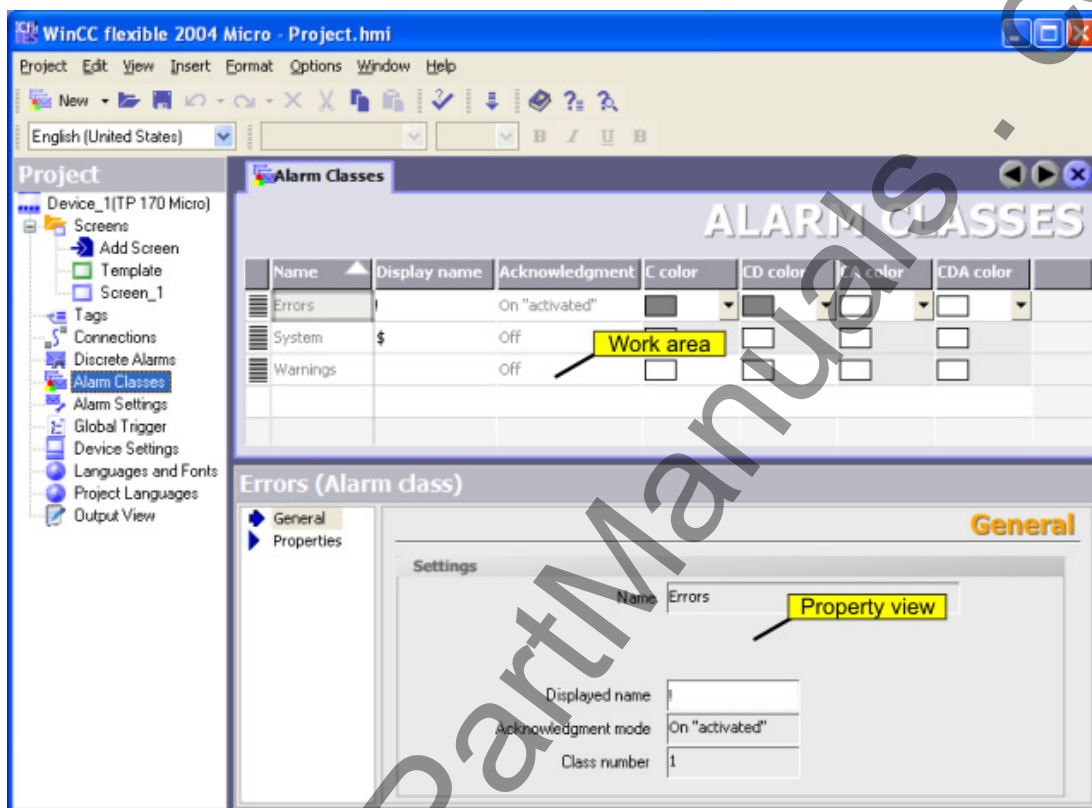


Figure 5-2 "Alarm classes" editor

## Work area

All the alarm classes are displayed in a table in the work area. Edit the properties of the alarm classes directly in the table cell or in the properties window. When you select an alarm class or a field of an alarm class, its properties are displayed in the properties window.

## Properties window

Configure the alarm class in the property window. Select an alarm class in the table in the work area. Select the corresponding category in the left section of the properties window. Make the desired changes in the right section of the properties window.

### 5.2.3 Basic settings for the alarm system

#### Introduction

WinCC flexible is delivered with basic settings for the alarm system. The alarm system can be used in the project with the basic settings. Change the basic settings if your project requires custom settings.

#### Opening the basic settings

Double-click on "Alarm settings" in the project window.

#### Structure

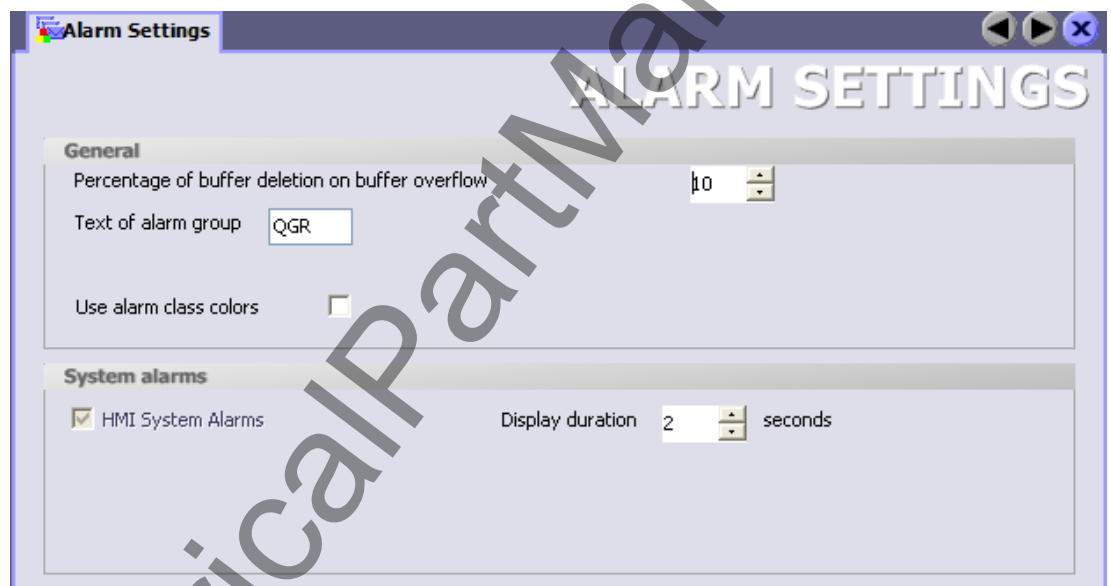


Figure 5-3 Alarm settings

#### Work area

You define the settings for the alarm system in the work area. In the "System alarms" area you select, for example, the amount of time that system alarms are to be displayed on the HMI device.

www.ElectricalPartManuals.com



# 6

## Configuring a connection

### 6.1 "Connections" editor

#### Introduction

The "Connections" editor is used to create a connection to the PLC. The connection is assigned a basic configuration when it is created. You can use the "Connections" editor to adapt the configuration of the connection to the requirements of your project.

#### Opening

You can open the "Connections" editor by selecting the item "Connections" in the project window and then opening the context menu with the right mouse button. In the context menu, select the command:

- Open editor
- or
- Add connection

As an alternative, you can open the "Connections" editor by double-clicking on the "Connections" item in the project window.

Structure

The "Connections" editor displays the configured connection to the PLC.

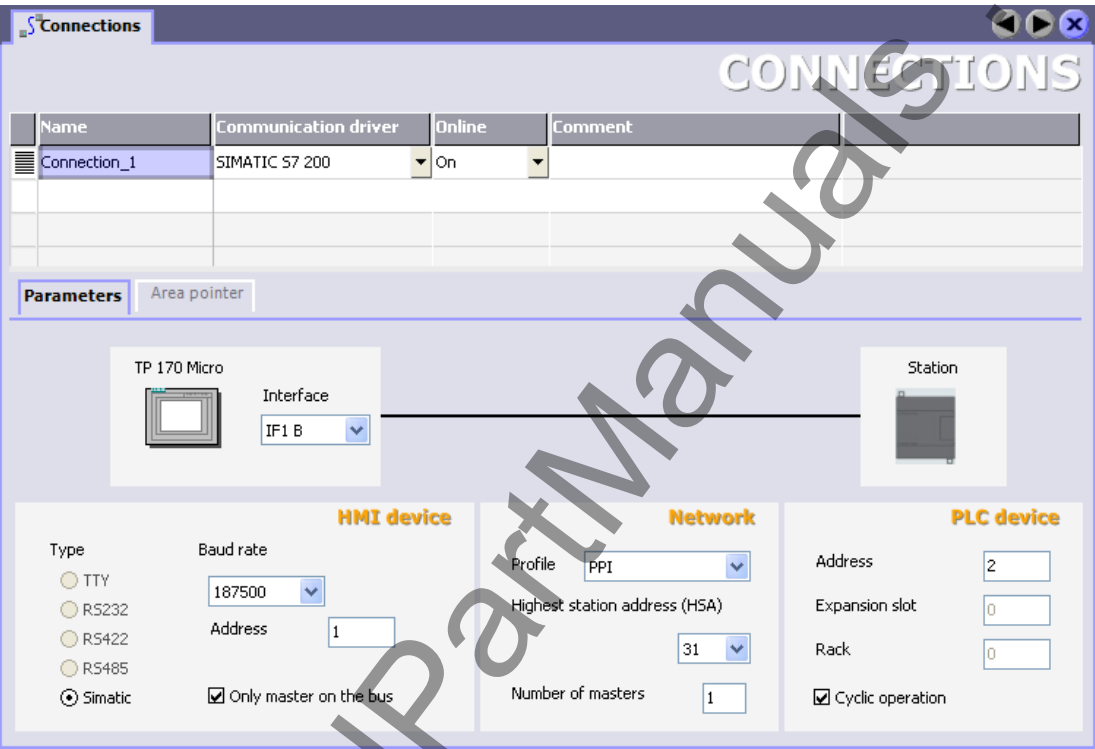


Figure 6-1 "Connections" editor

Work area

The connection to the PLC is displayed in the table in the work area.

You can rename the connection in the "Name" field. The communication driver used is shown in the "Communication driver" field. The communication driver cannot be changed here because you can only have a connection to a SIMATIC S7 200 type controller when the TP 170micro HMI device is used. The connection can be activated or deactivated in the "Online" column.

## "Parameters" tab

You can configure the connection to the PLC in the "Parameters" tab in the work area.

Connection parameters:

Area	Field	Value
HMI device	Baud rate	Sets the transmission rate of the connection.
	Address	Sets the station address for the HMI device.
Network	Profile	Sets the network protocol for the connection. The following protocols are available: <ul style="list-style-type: none"> <li>• PPI</li> <li>• MPI</li> <li>• DP</li> </ul>
	Highest station address	Sets the highest station address allowed.
Controller	Address	Sets the station address for the PLC.
	Cyclic operation	If cyclic operation is activated, the controller optimizes data transfer between the HMI device and controller. This achieves better performance.

[www.ElectricalPartManuals.com](http://www.ElectricalPartManuals.com)

# Using global events

## 7.1 Use cases for global triggers

### Definition

The global trigger is used to execute event-controlled jobs. A job consists of a triggering event and a "function list". You can link one or more system functions to an event in the global trigger.

For example, you can link the system function "ClearAlarmBuffer" to the "AlarmBufferOverflow" event to clear the alarm buffer when an overflow occurs. A job then exists. When the event occurs, the linked function list is called and executed.

## 7.2 Working with events

### Introduction

The global trigger responds to specific events. "ScreenChange" or "AlarmBufferOverflow" are examples of events.

The TP 170micro HMI device supports the following events:

- Runtime shutdown  
Task is performed at runtime shutdown.
- Screen change  
Task is performed when a screen change occurs.
- Alarm buffer overflow  
Task performed when the alarm buffer overflows.
- Change user  
Task is performed when the user logs on or off.

### Sequence of a global trigger

When an event occurs, the global triggers starts the execution of the function list configured for the event. Global triggers are executed consecutively. A global trigger is executed by executing the function list line-by-line.

Only one global trigger can be configured and executed per event.

## 7.3 Elements of global triggers

### 7.3.1 Work area of the global triggers

#### Introduction

The work area shows the planned triggers which consist of the triggering event and the function list.

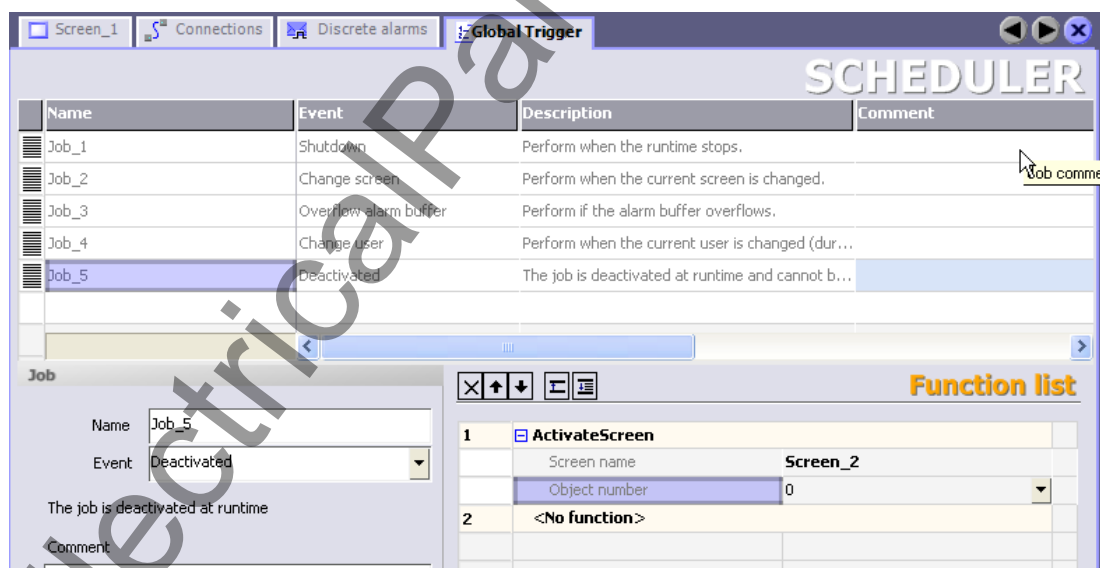
In the global trigger, configure a function from the function list for the event from the selection list of events.

#### Opening

You can open the global trigger in the project window by double-clicking on "Global trigger".

#### Layout

The work area consists of the table of triggers, the properties, and the function list.



You assign the label and a comment and select the event. The global trigger summarizes a description of the task.

The properties also show the trigger along with the triggering event. You can specify the event in the properties.

In the function list you configure the functions to be executed in the job.

---

**Note**

The description provides a written summary of the trigger including the planned event. 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



# Structure of multi-lingual projects

## 8.1 Language terminology 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 when you install WinCC flexible.

- 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.

### User interface languages

English is generally installed as the user interface language. The following languages can also be installed:

- Western European languages
  - German
  - Spanish
  - Italian
  - French

### 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
- Hungarian
- Japanese

You can generally also configure in any language available in Windows. However, restrictions may apply when some languages are used for configuring, such as:

- Right-to-left languages (such as Hebrew or Arabic) cannot be entered on the HMI device.
- 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.

- Editing language

You create the translations of the texts in the editing language. You first create the text of an object in the reference language, switch to the editing language and then enter the foreign language text in the object.

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. The runtime languages must be supported by the HMI device.

You must configure appropriate operator control elements so that the operator can switch between languages during runtime. 5 runtime languages are allowed for the TP 170micro HMI device.

## 8.2 Multilingual configuration

### Multilanguage configuration

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

WinCC flexible can be used to configure in all languages installed in the operating system.

Use the following editors to translate texts for a multilingual configuration 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

You can configure texts for the foreign languages directly at the respective object.

### WinCC flexible multilanguage user interface

During configuration in WinCC flexible, it is possible to switch the language of the configuration user interface, e.g. when several configuration planners of different nationality work with WinCC flexible.

Use the menu command "Options > Settings" to switch the user interface language. The "Settings" dialog is opened. Double-click on the "Workbench" entry in the "Settings" dialog and select the sub-entry "User Interface Language". In the "Language" section, select the user interface language you require. Close the dialog with the "OK" button. The user interface language is changed.

## 8.3 Language settings

### 8.3.1 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
- Language-dependent format for dates, times, currency, and numbers

#### 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.

#### Language-dependent format for 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 the 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 before translation.

## 8.3.2 "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.

### Opening

To open the "Project languages" editor, double-click "Project languages" in the "Localize" group in the "Project window."

### Structure

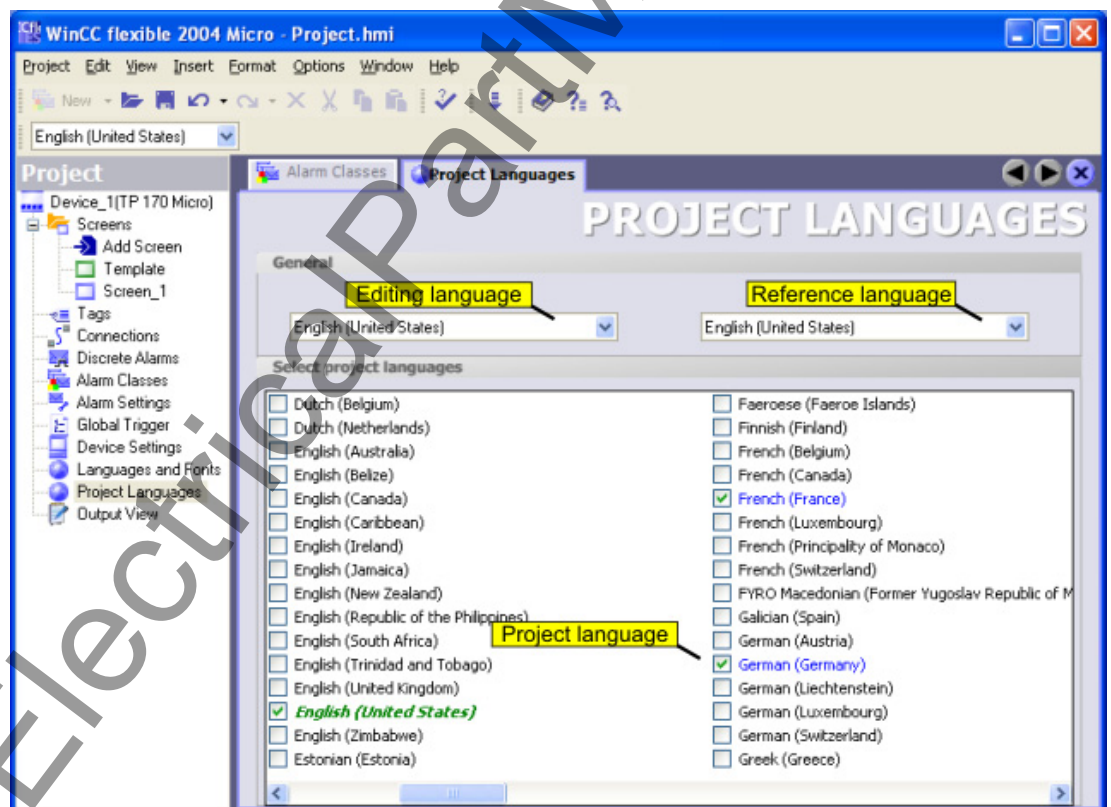


Figure 8-1 "Project languages" editor

### 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.

## 8.4 Languages during runtime

### Using multiple runtime languages

You can decide which project languages are to be used as runtime languages on a particular HMI device. To enable the operator to switch between languages during runtime, you must configure corresponding operator control elements.

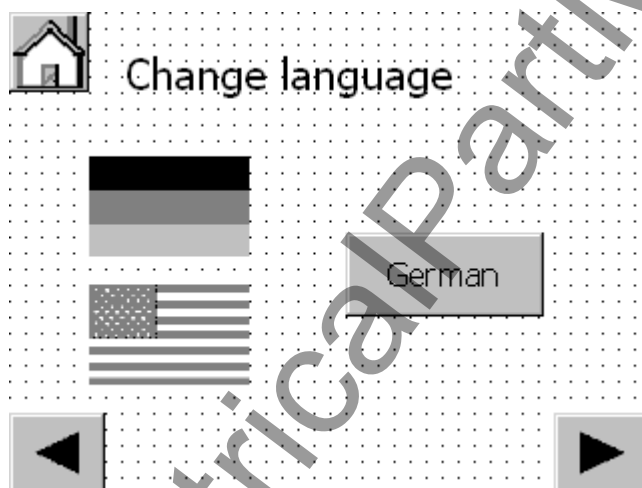


Figure 8-2 Process screen for language switching

When runtime starts the first time, the language with the lowest number in the "Order for language setting" is displayed. When runtime starts thereafter, the project is displayed according to the most recent language setting.



Figure 8-3 Order of runtime languages

### Setting runtime languages during configuration

Activate the check box for selecting runtime languages in the "Runtime languages" column. To change the order of runtime languages, select a language. Use the arrow keys to move the selection up or down.

www.ElectricalPartManuals.com



# Project documentation

## 9.1 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:

- All objects of an editor
- Single or multiple selected objects

The selection of the data for the output depends on the objects or the selected editor. The system assembles the data when the project report is created. The data are output in as a five-column table. The five most important attributes of an object are output. The five attributes which are to be output are preset in the system. The selection of these attributes cannot be modified.

The project report is opened in a preview before the output starts. In this preview you can verify the project report before you output it.

### Output medium

Project reports are sent to the printer from the preview.

## 9.2 Selecting the objects for the project documentation

### Introduction


WinCC offers several options for outputting the configuration data from one or more object of a Editor. Start the output via:

- The main menu
- The toolbar
- The context menu of selected objects

## Outputting data in the project report


Selecting objects:

- Select the screen in the project window to print the configuration data of an individual process screen.
- Select the "Screen" folder in the project window to print the configuration data of all process screens.
- Select the desired editor in the project window to print the data of a WinCC flexible editor.
- Open the editor in the work area to print a specific object of the editor. Select the desired object in the work area.

Select the command "Print Selection" from the context menu of the selection to begin printing. WinCC flexible determines the data for the printout and opens the project report in the preview. Check the data in the preview for the printout. Click on the  button in the preview to start printing.

## Alternative procedure

WinCC flexible provides several options for outputting the data. After selecting the object, you can also start the printout using the:

-  button.
- "Project > Print selection" command in the menu bar

The configuration data are inserted into the "Default layout" and opened in the preview window. Start the printing from the preview window.

## Transfer

### 10.1 Basic principles of the transfer operation

#### Transfer

A transfer operation refers to the transfer of a compiled 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 a successful 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 device. The transfer can also be started without first performing a consistency check. The system will then perform the consistency check and compiling in the background.

The HMI device must be connected to the configuration computer to transfer the project data.

#### Basic procedure

1. Enter the transfer settings for the HMI device in your WinCC flexible project.
2. Set the transfer mode on the HMI device.
3. Transfer the compiled project file from the configuration computer to the HMI device.

#### Transfer mode

The HMI device must be in "transfer mode" for the transfer operation.

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

If the initial settings of the commissioning have been changed, you have to restart the HMI device and call the transfer applet from the Start menu or configure the "SetDeviceMode" system function in your project.

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

## 10.2 Transfer settings

### Introduction

The compiled project file is sent to the configured HMI device during the transfer. The dialog "Select HMI device for the transfer" opens prior to the transfer of the project to the HMI device. Make the settings for the transfer in this dialog.

### Specifying transfer settings

To specify the transfer settings, select the menu command "Project > Transfer > Transfer Settings." The "Select HMI device for transfer..." dialog is displayed. The data transfer to the TP 170micro HMI device is a serial process.

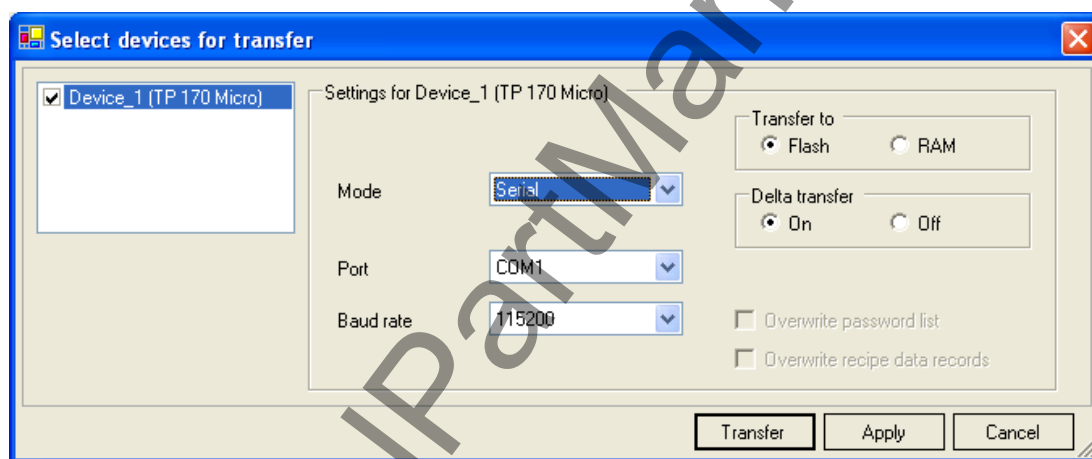


Figure 10-1 Transfer settings

Select the connected port from the "Port" selection field. Select the highest transmission rate from the "Baud rate" selection field.

The check box for the HMI device must be activated in the left section of the transfer settings.

### Transfer mode

You can use the following transfer modes for the TP 170micro HMI device:

- Direct connection via serial cable

Transfer is performed via a "PC/PPI" adapter cable connecting the configuration computer and the HMI device.

The transfer mode setting is also used if the HMI device is selected in the project view and one of the commands in the "Project > Transfer" menu is selected (for example, for a delta transfer operation or an operating system update on the HMI device).

### Transfer destination

On the TP 170micro HMI device, you can store the compiled project file in the flash memory or RAM of the HMI device.

### Delta transfer

Delta transfers can be performed on the TP 170micro HMI device to save transmission time. 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 existing 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 the TP 170micro HMI device. 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

Only one connection can be made to the TP 170micro HMI device, either from the configuration computer or from the PLC.

---

## 10.3 Managing files on the HMI device

### 10.3.1 ProSave

#### Introduction

The ProSave service tool is supplied with WinCC flexible. The functionality of ProSave is integrated in the WinCC flexible user interface on the configuration computer. ProSave can also be installed as a stand-alone program on a computer where WinCC flexible is not installed ("stand-alone operation").

#### Functional scope

ProSave provides all of the functions needed to transfer files to the HMI device.

- Data backup and restoration of backed-up data
- Updating the operating system

### Integrated operation on the configuration computer

ProSave is installed on the configuration computer during the 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 having to install WinCC flexible (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. WinCC flexible is not required for this procedure.

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.

## 10.3.2 Data backup for the HMI device

### Introduction

The data on an HMI device should be backed up at regular intervals.

A data backup gives you the option to recommission your project quickly after a system failure or a device is replaced. The backed-up data is simply transferred to the new HMI device, thus reproducing the original state.

### Data backup using WinCC flexible or ProSave

If an HMI device is connected to a configuration computer, you can use WinCC flexible to back up and restore the data on the HMI device from the configuration computer.

If the computer does not have WinCC flexible installed, you have the option of using ProSave to perform a backup.

### Scope of data backup

The backup and restoration for the TP 170micro HMI device includes the following project data:

- Full backup (runtime, firmware, image of operating system, configuration, password, settings)

A data backup from an HMI device generates a compressed file with the extension \*.psb.

---

**Note**

Note the following when performing a full data backup and restore operation:

When a full data restoration is performed, all of the data that were previously on the device, including 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 by means of the serial interface using the "bootstrap" mechanism before resuming the data restoration.

---

### 10.3.3 Updating the operating system

#### Introduction

If the operating system version on the HMI device is not compatible with the configuration, the transfer of the project is aborted. A message appears indicating the operating system must be updated.

#### Updating the operating system

If an HMI device is connected to a configuration computer, you can use WinCC flexible to update the operating system from the configuration computer.

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.

When an operating system is updated, all of the data on the destination device are deleted. If you want to continue to use any user data stored in the internal flash memory following the update of the operating system, you must export the data to an external data memory beforehand and reload them onto the HMI device following the update. The configuration computer or a PC with a ProSave installation can be used for external data storage

#### "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, the HMI device must be switched off and on (booted), so that the HMI device can make contact by means of the boot loader.

www.ElectricalPartManuals.com



# Configuration examples

11

## 11.1 Create screen

### Introduction

In this example you will create a screen template which can be used as the basis for different screens in a project. In the screen you will configure the navigation buttons for selecting various screens in the project. You will also create a button for selecting your startup screen.

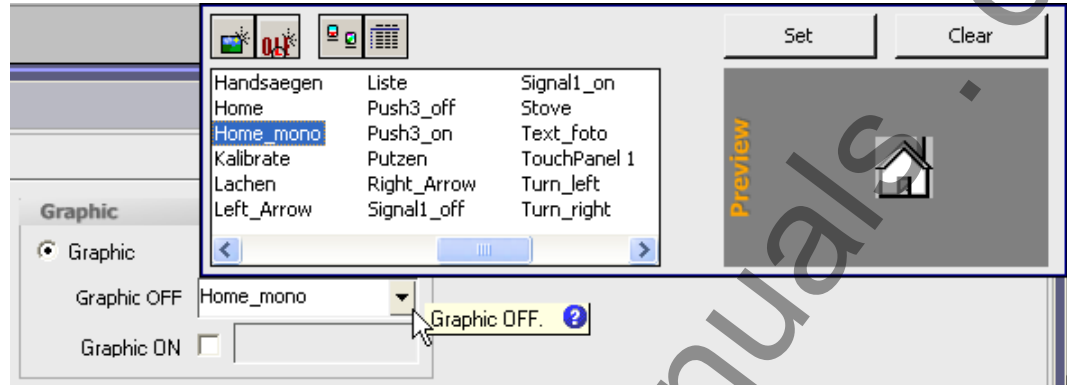
### Requirements


- WinCC flexible is open.
- A project is open.
- An approximately 30x30 pixel graphic is available for the "Home" button.
- Two approximately 30x30 pixel graphics are available for the "Forward" and "Back" buttons.

### Create the screen and buttons

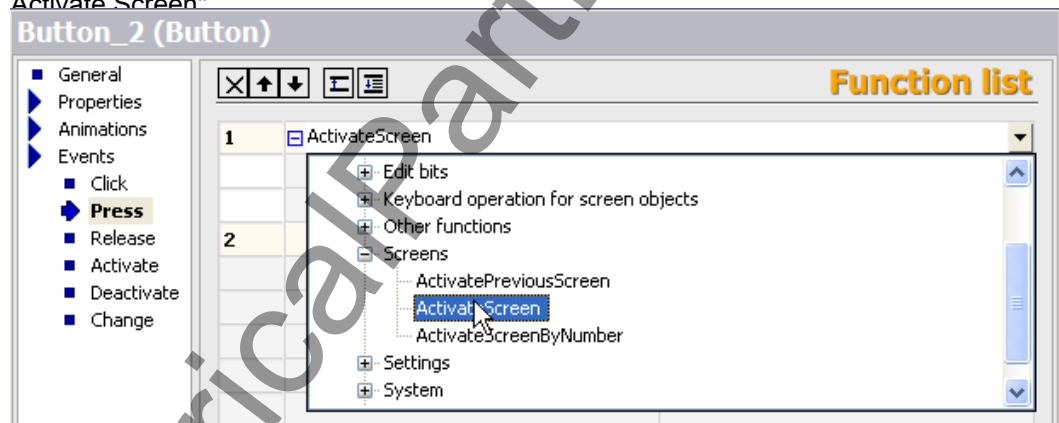
1. Double-click on "Add screen" in the "Screens" folder of the project window, A new WinCC flexible screen is created and opened in the work area.
2. Select a button from the toolbar, drag it onto the work area holding down the left mouse button drawing it to the desired size.
3. In the properties window, select the "General" category and select the "Graphic" option in the "Type" area.

- In the "Graphic" area, open the selection button under the entry "Graphic OFF." A dialog for selecting the graphic opens.



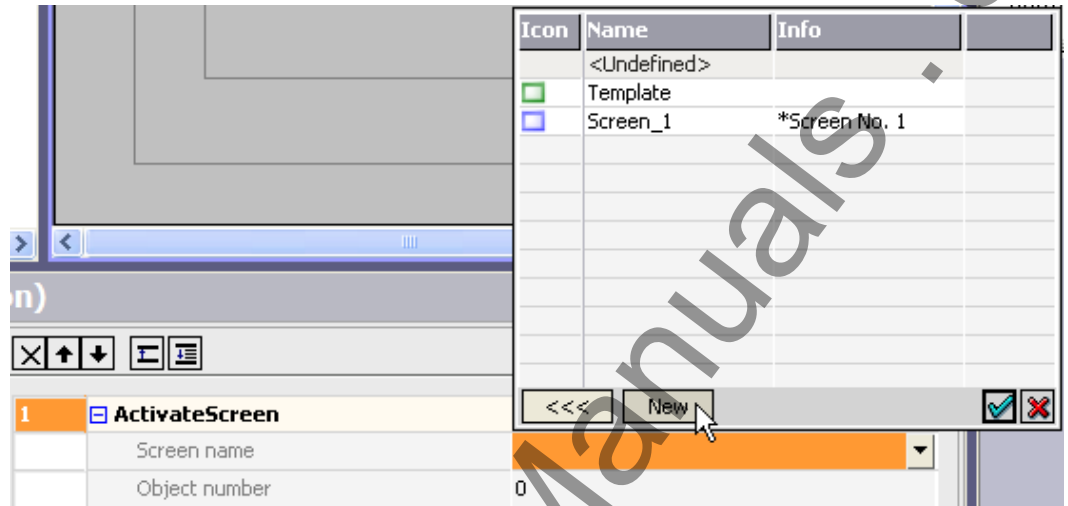
In the dialog for selecting the graphic, press the  button and navigate to the screen you want for the "Home" button. Select the screen and press the "Open" button. The screen is inserted into the button.

Switch to the category "Events > Press" in the properties window and open the function list using the selection button. Select the system function "System Function > Screens > Activate Screen"



The "Activate Screen" system function is configured for the button.

5. Click in the "Screen name" field. A selection button is displayed in the value field. Open the object list using the selection button. Click on the "New" button here.

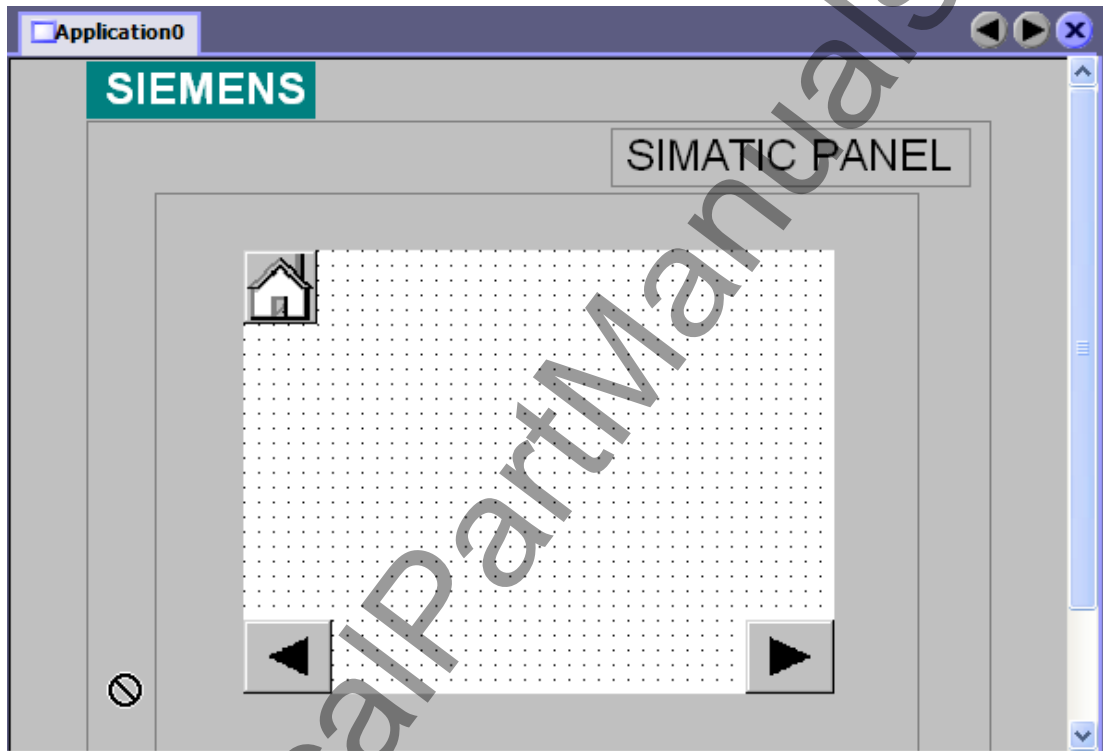


A dialog for creating a new WinCC flexible screen opens. Enter the "Home" in the "Name" field and confirm your entry by pressing "OK". The WinCC flexible "Home" screen is displayed and is linked to the "Home" button with the system function "ActivateScreen".

6. Repeat steps 2 to 6 for the "Forward" and "Back" buttons.  
The screen template is now completed and can be used to create more WinCC flexible screens.
7. Configure the desired sequence of WinCC flexible screens for the the "Forward" and "Back" buttons. To do this, select the "Forward" or "Back" button. In the function list, click on the "Screen name" field. A selection button is displayed in the value field. Open the object list using the selection button and select the destination screen.

## Result

The screen template is now complete and features 3 buttons. The "Home" button is linked to the "Home" screen. Configure the system function "ActivateScreen" for the "Forward" and "Back" buttons. Configure the screen names for the respective destinations as described in Step 7.



To use the screen template again, select the screen in the project window and open the context menu. Select the "Copy" command from the context menu. Select the "Screens" folder and open its context menu. Select the "Paste" command from the context menu. The screen will be inserted along with its name and a sequential number.

## 11.2 Creating external tags

### Introduction

For an external tag, you must set at least the following properties:

- "Name"
- "Connection" to PLC
- "Data type"
- "Address"

For documentation purposes, it is worthwhile to enter a comment for every tag.

## Requirements

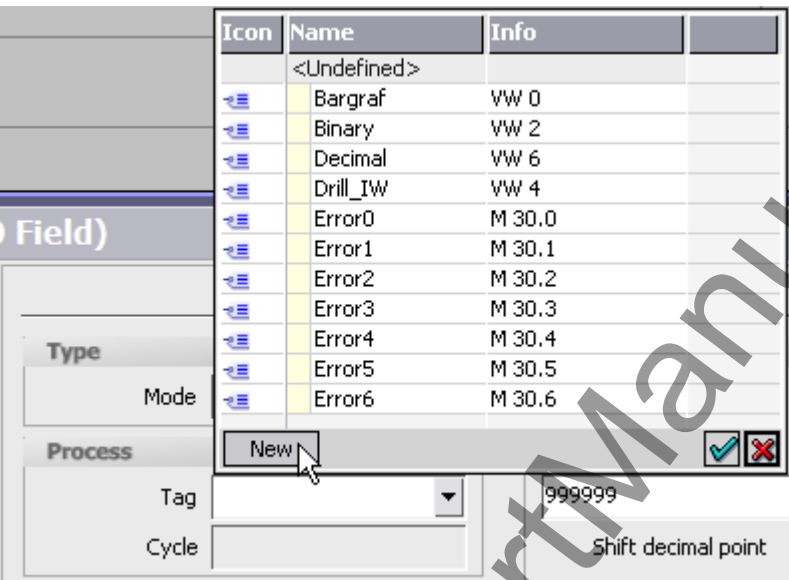
You have opened the project.

## Procedure

1. Select the "Add tag" command in the context menu of the "Tags" work area.  
The "Tag" editor opens.
2. In the "General" group in the property window enter an unambiguous tag name in the "Name" field.
3. In the "Connection" field, select the connection to your SIMATIC S7 200 PLC. If the desired PLC is not shown, you must first connect to the PLC using the object list or the "Connections" editor.
4. Select the desired data type.
5. For tags to contain text, e.g. of the "String" data type, define the length by entering the maximum number of characters that the tag should contain. The length is automatically defined for all other data types.  
If you want to assemble the tag from array elements, enter the desired number of array counts. If the number of array elements is greater than 1, the length will now refer to the length of the individual array elements.
6. If you want to change the acquisition cycle, select another acquisition cycle or use the object list to define your own cycle.
7. Click on "Address" in the "Properties" group. Enter the address of the PLC where you wish to access the external tags.
8. As an option, you can enter a comment regarding the use of the tag. To do this, click on "Comment" in the "Properties" group and enter the text of your comment.

Alternative procedure

You can also create new tags where they are required in the project. Open the object list using the selection button for the tag. Click on "New" in the "Object list".



The dialog for creating a new tag opens. Configure the new tag and confirm your settings by pressing "OK".

Result

You have now created an external tag which you can use in your project.  
You could now configure a tag, for example, by setting the limits.

11.3 Configuring an alarm view

Introduction

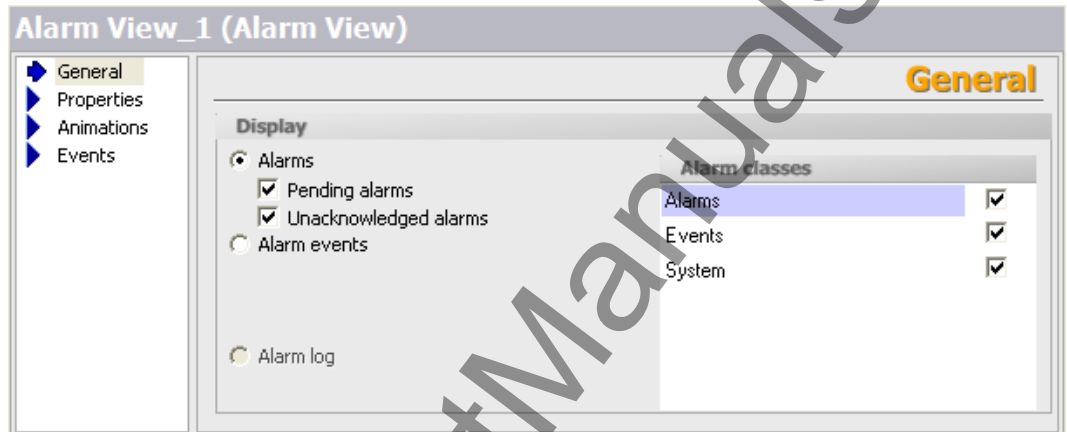
The alarm view shows selected alarms or events from the alarm buffer.  
The display of alarms and events can be combined with all available alarm classes.

Requirements

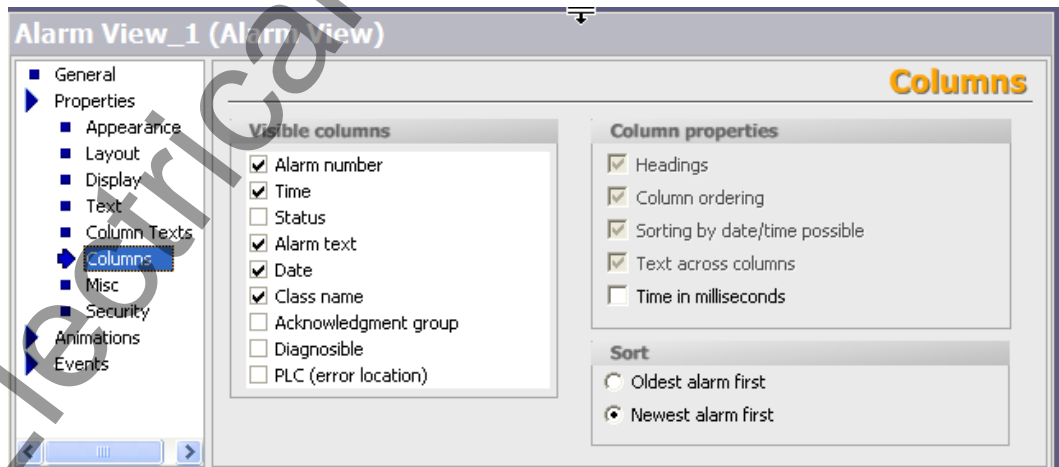
A screen or the template is open in the "Screens" editor.  
The toolbox window is displayed.

## Procedure

1. Insert an alarm view into the screen from the toolbar and draw it to the desired size using the mouse.
2. In the property window, select the "General" group.  
In the "Use" area, select the content of the alarm view:  
Alarms or events from different alarm classes.



3. Click "Layout" in the "Properties" group in the property window. Set the number of lines per alarm and the number of visible alarms in the "Display" area.
4. Click "Display" in the "Properties" group in the property window. Activate the operator control elements of the alarm view that you wish to have available on the HMI device. Click "Columns" in the "Properties" group in the property window.



- In the "Set of visible columns" area, select the columns to be displayed in the alarm view.
5. In the "Sort" area, select the sort order of the alarms.

## Result

Alarms of selected alarm classes are output in the alarm view during runtime.

## 11.4 Configuring an alarm window

### Introduction

The alarm window shows selected alarms or events from the alarm buffer or alarm log. In contrast to the alarm view, you can configure the alarm window to automatically open as soon as an alarm occurs. You can use an alarm window, for example, to immediately display critical error alarms. The template already includes an alarm window that is displayed when system alarms occur during runtime. In the screen template, you can edit or expand the preconfigured alarm window used to display system alarms.

The alarm window can only be inserted into the screen template.

In this example, you will configure an alarm window to be activate when an error alarm occurs.

### Requirements

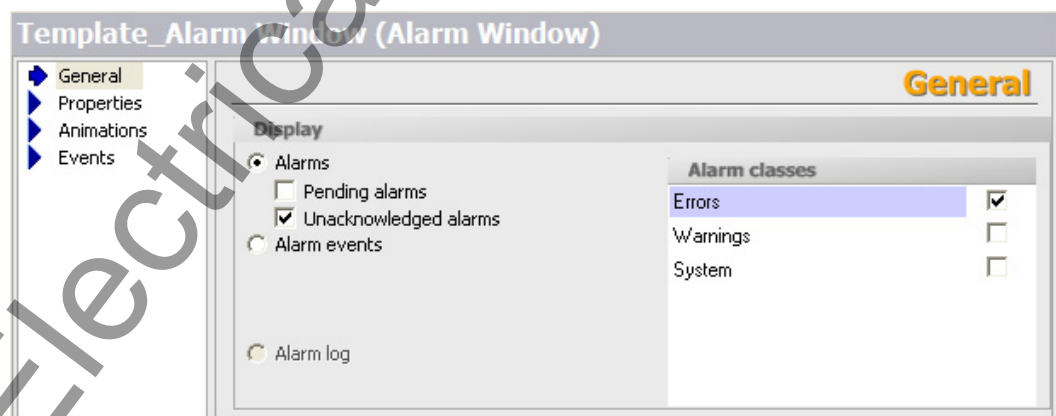
The template from the screens folder is open in the "Screens" editor.

The toolbox window is displayed.

### Procedure

1. Insert an alarm window into the screen from the toolbar and draw it to the desired size using the mouse.
2. In the property window, select the "General" group.

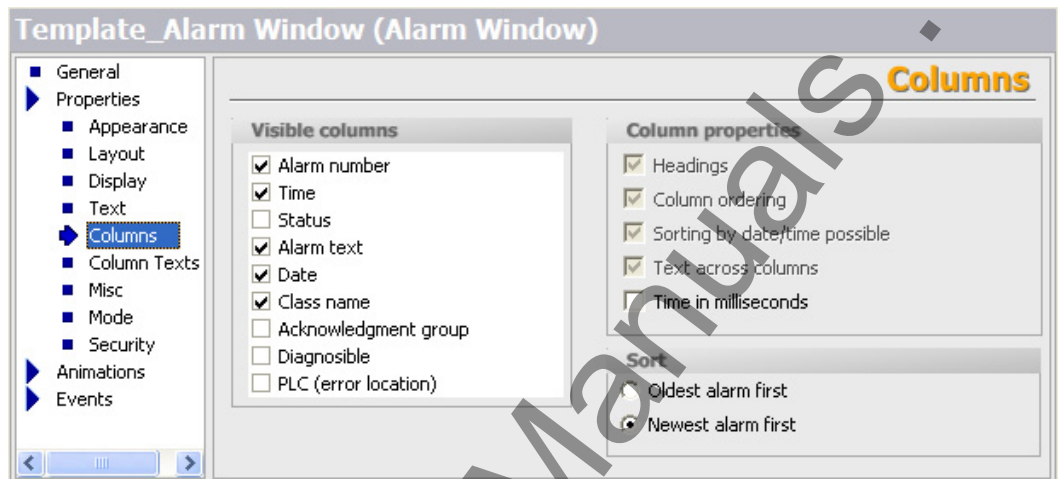
In the "Use" area, select the content of the alarm view:  
"Queued alarms" from the "Error" alarm class.



3. Click "Layout" in the "Properties" group in the property window. Set the number of lines per alarm and the number of visible alarms in the "Display" area.

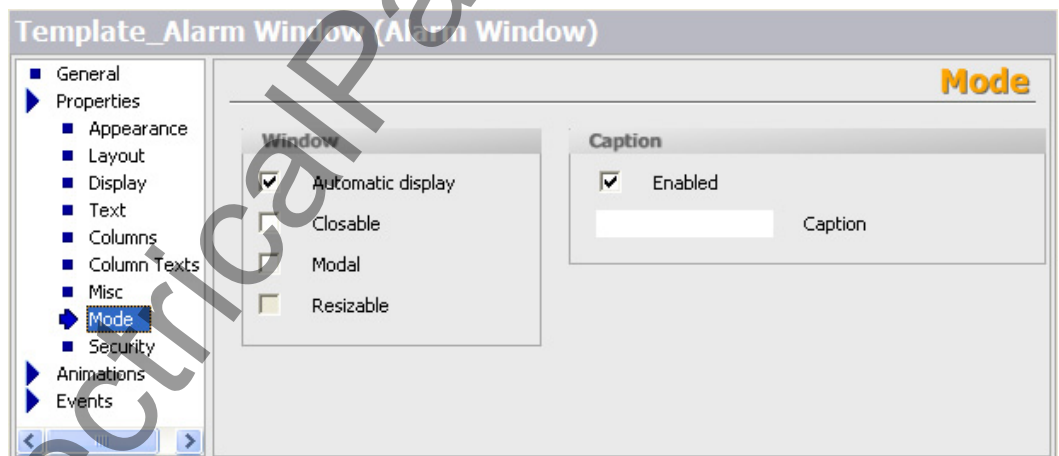


- Click "Display" in the "Properties" group in the property window. Activate the operator control elements of the alarm view that you wish to have available on the HMI device.  
Click "Columns" in the "Properties" group in the property window.



In the "Set of visible columns" area, select the columns to be displayed in the alarm view.

- Click "Mode" in the "Properties" group in the property window.  
Select the options "Automatic display" and "Closable" in the "Window" section.



## Result

When an error alarm occurs in runtime, the alarm window will be automatically activated and display the occurring alarm.

## 11.5 Configuring discrete alarms

### Introduction

For a new discrete alarm, you must configure at least the following properties:

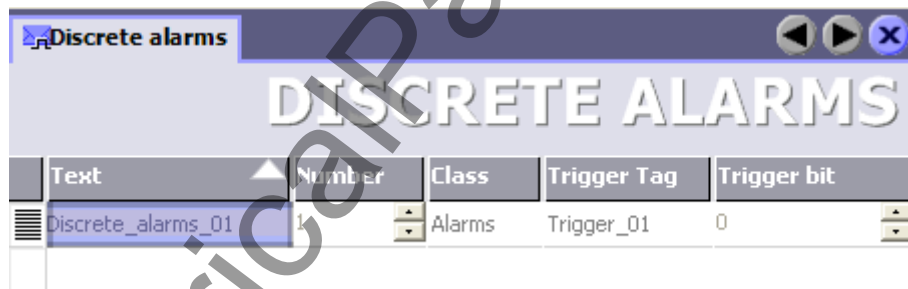
- Alarm text
- Alarm class
- Trigger tag and bit number

### Requirements

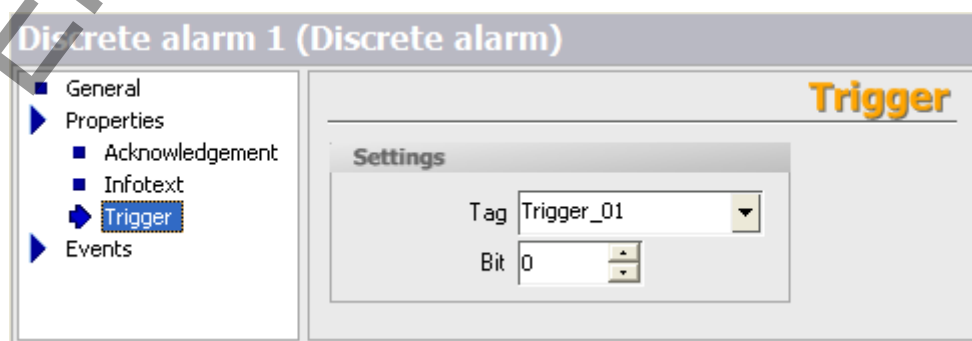
A project is open.

### Procedure

1. Select the "Discrete alarms" entry in the project view and open its context menu. Select the "Add discrete alarm" command in the context menu. The "Discrete alarms" editor is displayed with a new discrete alarm.



2. In the property window, select the "General" group.
3. Enter the alarm text.
4. Select the alarm class.
5. In the property window, select the "Properties > Trigger" group. Select the tag and the bit that triggers the alarm.



**Result**

The required settings are defined for the new discrete alarm.

**Optional settings for discrete alarms****Alarm acknowledgement by the control program**

1. In the property window, select the "Properties > Acknowledgement" group.
2. In the "Acknowledgement write tag" category, select the tag and the bit used to acknowledge the alarm.

**Sending alarm acknowledgement to the PLC**

1. In the property window, select the "Properties > Acknowledgement" group.
2. In the "Acknowledgement read tag" category, select the tag and the bit set by the alarm acknowledgment.

**Additional settings for discrete alarms**

1. To assign the alarm to an alarm group, select the alarm group in the "General" group in the property window.
2. To carry out event-triggered tasks, select the "Events" group in the property window and configure a function list for the required event.

## 11.6 Using the system function "AlarmViewEditAlarm"

**Introduction**

You get an error alarm in runtime and want to open the specific process screen where you can fix the cause of the error. This is done by using the system function "AlarmViewEditAlarm."

**Requirements**

You have configured an error alarm that is triggered by an error event, for example, when a limit is exceeded.

## Procedure

1. Select the alarm in the "Discrete alarms" editor and select the command "Events > Edit" in the properties window.
2. Select the system function "Activate screen" under "Screens" in the function list. A screen for the system function is expected in the next line of the function list.
3. Click in the "No value" field and in the object list that opens select the process screen where you can influence the cause of the alarm.
4. In a process screen, configure an alarm view to display alarms of the "Error" alarm class.
5. Create a button in the same process screen. Select the button and select "Events > Press" in the properties window.
6. Select the system function "AlarmViewEditAlarm" from the function list. A screen object for the system function is expected in the next line of the function list. Click in the "No value" field and in the object list that opens select the alarm view you have just created.

## Result

When the alarm is triggered in runtime, select this alarm from the alarm view. Now when you press the button you have created, the process screen where you can influence the cause of the alarm opens.

## Features

### 12.1 General technical data

#### 12.1.1 Released operating systems

##### Released operating systems

Operating systems released for WinCC flexible:

- Windows 2000 Professional SP4<sup>1)</sup>
- Windows XP Professional SP1<sup>1)</sup>

<sup>1)</sup> 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 under "Start > Settings > Control Panel > System."

---

#### 12.1.2 Legal characters

##### Introduction

WinCC flexible supports the full ASCII character set. We advise user, 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

Illegal characters are: '

## 12.2 System limits

### Introduction

The following list of system limitations provides 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. 500 alarms can be configured if no further objects are used. Simultaneous output of 500 alarms and 250 screens, each with 20 screen objects, is not possible.

In addition to the limitations specified, the restrictions of configuration memory resources must also be considered.

### Overview

#### Micro Panels

	TP 170micro
Number of tags in the project	250
Number of PowerTags	100
Number of elements per array	100
Number of local tags	--
Number of alarm classes	32
Number of discrete alarms	500
Number of analog alarms	--
Length of alarm in characters	80
Number of process values per alarm	8
Size of the alarm buffer	128
Number of queued alarm events	16
Number of screens	250
Number of fields per screen	20
Number of tags per screen	20
Number of complex objects per screen (e.g. scroll bars)	5
Number of recipes	--
Number of elements per recipe	--
User data length in Bs 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	--
Number of logs	--
Number of entries per log (incl. all log segments)	--
Number of log segments	--
Cyclical trigger for logging tags	--

	TP 170micro
Number of trends	--
Number of graphic lists	--
Number of text lists	--
Number of all lists	--
Number of entries per text or graphic list	--
Number of graphic objects	500
Number of text elements	500
Number of scripts	--
Number of connections	1
Number of connections based on "SIMATIC HMI http Protocol"	--
Maximum number of connected Sm@rtClients	--
Length of a operator note in characters	--
Number of runtime languages	5
Tasks	--
User groups	2
Authorizations	32
Passwords	1
Size of the user memory (*.fwx)	256 KB

www.ElectricalPartManuals.com



# Index

## A

- Acknowledgement tag
  - reading, 5-5
  - Writing, 5-5
- Acknowledging alarms, 5-2
- Acquisition cycle
  - Tag, 3-7
  - Tags, 3-9, 3-10
- Alarm
  - Acknowledging by the PLC, 5-5
  - Base values, 5-9
  - Component, 5-5
  - Configure, 11-10
  - Editor, 5-6
  - Property, 5-5
- Alarm class, 5-2
  - Alarm classes editor, 5-7
- Alarm classes editor, 5-7
- Alarm line, 5-3
- Alarm number, 5-5
- Alarm procedure, 5-1
- Alarm status, 5-2
  - Layout, 5-3
- Alarm text, 5-5
- Alarm view, 4-6, 5-3
  - Configure, 11-6
- Alarm window, 4-6, 5-3
  - Configure, 11-8
- Alarms, 5-1
  - Acknowledging, 5-2
  - Basic principles, 5-1
  - Display on the HMI device, 5-3
  - Events, 5-4
  - System functions, 5-4
- Area pointer
  - Date/time PLC, 3-1
  - using, 3-1
- Array, 3-9
  - Properties window, 3-5
- Authorization, v

## B

- Bar, 4-6
- Base values
  - Alarm, 5-9
- Bootstrapping
  - Operating system, 10-5
- Button, 4-6

## C

- Communication
  - between PLC and tag, 3-7
- Configure
  - Alarm view, 11-6
  - Alarm window, 11-8
  - Discrete alarms, 11-10
- Configuring
  - Toolbar, 2-4
- Connections editor, 6-1
  - Work area, 6-2
- Context menu, 2-9
  - calling, 2-9
- Controller
  - Linking tags, 3-7
- Copy
  - Screen, 4-5
- Creating
  - External tag, 11-4

## D

- Data backup
  - HMI device, 10-4
- Data type
  - Internal tag, 3-2
- Date / time field, 4-6
- deleting
  - Screen, 4-5
- Delta transfer, 10-3
- Discrete alarm procedure, 5-1

Discrete alarms  
    Configure, 11-10  
    Discrete alarms editor, 5-6  
Discrete alarms editor, 5-6  
Displaying  
    Project report, 9-2  
Drag-and-drop, 2-9

## E

Editing language, 8-2  
Editor, 2-11  
    Closing, 2-6  
    Configuring alarms, 5-6  
    open, 2-12  
    Project languages, 8-5  
    Properties, 2-11  
    Screens, 2-12  
    Tabular, 2-12

## F

Field of application  
    Global trigger, 7-1  
Function list, 2-15  
    Configure, 2-16  
Functional scope  
    ProSave, 10-3

## G

Global trigger, 7-1  
    open, 7-2  
    Sequence, 7-1  
Global triggers  
    Work area, 7-2  
Graphic IO field, 4-6  
Graphic view, 4-6  
Grouping  
    Objects, 4-8

## H

HMI device  
    Data backup, 10-4  
    Restoring data, 10-4

## I

Introduction  
    Project documentation, 9-1  
IO field, 4-6

## L

Language switching  
    ProSave, 10-4  
Language-dependent format, 8-4  
Layout  
    Language-dependent format for date, time,  
    currency, and numbers, 8-4  
Limit range  
    Tags, 3-8

## M

Menu, 2-3  
    Command, 2-3  
Mouse functions, 2-9  
Move  
    Screen, 4-5  
Multiple selection  
    Creating object groups, 4-8

## N

Navigation arrow, 2-6

## O

Object  
    Selecting for project report, 9-2  
Object group, 4-8  
Object list  
    open, 2-14  
    Working with, 2-15  
Offices, iv  
Online help, 2-11  
    Displaying, 2-11  
open  
    Editor, 2-12  
Opening  
    Global trigger, 7-2  
Operating system  
    Updating on HMI device, 10-5

- Operation, 2-9
  - Hotkeys, 2-10
  - Mouse functions, 2-9
- Output medium
  - Project report, 9-1
- Output window, 2-8

## P

- Print
  - Project documentation, 9-1
- Procedures
  - to create screens, 4-5
- Project
  - Multilanguage projects, 8-3
  - Several projects, 2-14
- Project data, 2-12
  - Updating, 2-12
- Project documentation
  - Introduction, 9-1
- Project language, 8-1
- Project languages
  - Editor, 8-5
- Project report, 9-1
  - Displaying, 9-2
  - Output medium, 9-1
  - Selecting objects, 9-2
- Project window
  - Working with, 2-7
- Project window, 2-6
- Properties
  - Tag, 3-6
- Properties window, 2-7
  - Array, 3-5
  - Properties, 2-8
  - Tag, 3-4
- Property
  - Alarm, 5-5
- ProSave, 10-3
  - Language switching, 10-4

## R

- Read continuously
  - Tags, 3-9
- Reference language., 8-2
- Rename
  - Screen, 4-5
- Representatives, iv

- Restoring
  - Data on HMI device, 10-4
- Restoring data
  - HMI device, 10-4
- Runtime language, 8-3
- Runtime languages
  - Order, 8-7
  - using, 8-6

## S

- Save
  - Data on HMI device, 10-4
- Screen
  - Copy, 4-5
  - deleting, 4-5
  - Inserting, 4-5
  - Move, 4-5
  - Rename, 4-5
  - Steps in creating a, 4-5
- Screen editor
  - Layout, 4-3
- Screens
  - Editor, 2-12
- Selecting
  - Objects for project report, 9-2
- Sequence
  - Global trigger, 7-1
- Service
  - In the Internet, iv
- Setting
  - Alarm, 5-9
  - For transferring, 10-2
  - Languages in the operating system, 8-4
- Simple objects, 4-6
- Start
  - Editor, 2-12
- Start value
  - Tags, 3-8
- Support
  - In the Internet, iv
- Switch, 4-6
- Symbolic I/O field, 4-6
- System event, 7-1

## T

Tab, 2-6

### Tag

Acquisition cycle, 3-7, 3-9, 3-10

Addressing, 11-4

Array, 3-9

Comment, 11-4

Communication with a PLC, 3-7

Connection to PLC, 11-4

Creating external tags, 11-4

Data type, 11-4

External tag, 3-1

Internal tag, 3-2

Length, 11-4

Limit range, 3-8

Name, 11-4

Properties, 3-6

Properties window, 3-4

Read continuously, 3-9

Start value, 3-8

Tag editor, 3-2

Work area, 3-3

Technical Support, v

Text box, 4-6

### Time

Display PLC, 3-1

Toolbar, 2-4

Configuring, 2-4

Positioning, 2-3

Toolbox, 4-3

Training Center, iv

### Transfer

Basic principles, 10-1

Delta transfer, 10-3

Methods, 10-2

Transfer settings, 10-2

### Transfer mode

On the HMI device, 10-1

## U

Update cycle, 3-10

### Updating

Operating system on the HMI device, 10-5

User interface language, 8-1

## W

WinCC flexible

Multilanguage user interfaces, 8-3

WinCC flexible, 2-1

Work area, 2-5

Connections editor, 6-2

Of the screen editor, 4-3

Planning global triggers, 7-2

Tag editor, 3-3

Working step

Restoring, 2-12

Reverting, 2-12

[www.ElectricalPartManuals.com](http://www.ElectricalPartManuals.com)