SIEMENS

SIMATIC HMI

WinCC V6.0
Basic Documentation

Manual

Order number
6AV6392-1XA06-0AB0

Release 04/03

A5E00221799

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are marked as follows according to the level of danger:

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.

Attention

indicates that unwanted events or status can occur if the relevant information is not observed.

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

Only **qualified personnel** should be allowed to install and work with this software. Qualified personnel within the meaning of the safety notices of this documentation are persons capable of giving due consideration to safety engineering standards when using this software on systems..

Trademarks

The registered trademarks of the Siemens AG can be found in the preface.

Impressum

Editor and Publisher: A&D PT1

Siemens AG
Bereich Automation & Drives
Geschäftsgebiet SIMATIC HMI
Postfach 4848, D-90327 Nuernberg

Exclusion of liability

We have checked the content of this publication for compliance with the described hard and software. However, discrepancies cannot be excluded, with the result that we assume no guarantee for total compliance. The information in this publication is checked regularly, and any necessary corrections are included in the following editions. We would be grateful for any suggestions for improvement.

Siemens Aktiengesellschaft

Order number 6AV6392-1XA06-0AB0

Foreword

Purpose

The WinCC V6 manual describes the structure and operation of WinCC and its components. The information system integrated into WinCC contains further information: Instructions, examples and reference data are provided in electronic form.

In this manual you will find an overview of

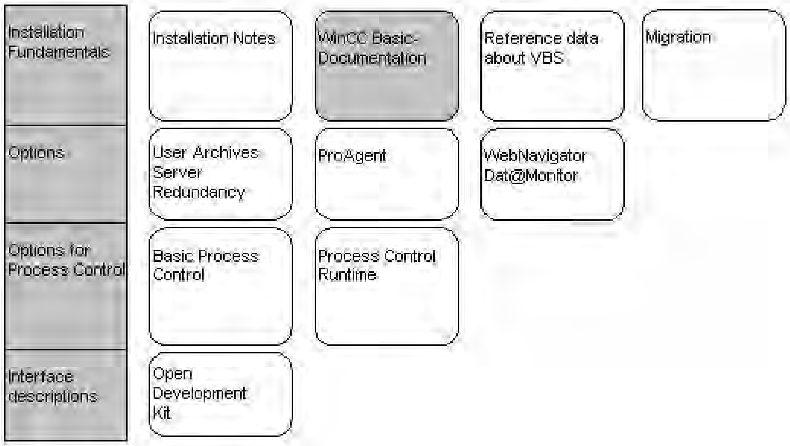
- WinCC
- working with WinCC
- and process communication

Information about the options User Archives, Server, Redundacy and the Configurations can be found in the WinCC V6 Options manual.

History

Issue	Comment
08/1999	WinCC Basic Documentation V5
04/2003	WinCC Basic Documentation V6

Position in the information landscape



Documentation	Content
Installation Notes	Contains important information about the contents of the WinCC package as well as its installation and operation.
WinCC V6 Basic-Documentation	Contains information • about WinCC regarding • working with projects • working with tags • creating process pictures • dynamizing process pictures • creating procedures and actions with VBS • creating C-functions and actions in Global Script • setting up a messaging system • archiving messages and process values • documenting configuration and runtime data • setting up multi-language projects • setting up user administration • the automation of project engineering with VBA • about process communication
Reference data about VBS	contains reference data about VBS
Migration	contains information about the migration of projects, which were created using versions of WinCC prior to V6.
WinCC V6 Options (User Archives, Server, Redundancy)	Contains information about the options • User Archives: creating and using user archives • Server: setting up and operating WinCC in a multi-user system • Redundancy: setting up and operating redundant systems
ProAgent	Contains information about the ProAgent (process diagnosis) option • configuring plant-specific process diagnosis • detecting process errors and their causes

Documentation	Content
WebNavigator Dat@Monitor	Contains information about the WebNavigator and Dat@Monitor options • configuring the Web project • accessing Web project process pictures via the Intra-/Internet • functions for displaying archive data and current process values in Excel • functions for displaying archive data from WinCC Historian in tables and charts • Viewing function for process pictures
Basic Process Control	Contains information about WinCC's process control options • Picture Tree Manager • Horn • Chipcard • Lifebeat Monitoring • Time Synchronization • OS-Projecteditor
Process Control Runtime	Contains information regarding the operation of the process control options of WinCC in runtime
Open Development Kit	Enables the programmer to use the WinCC API functions and access the data

Trademarks

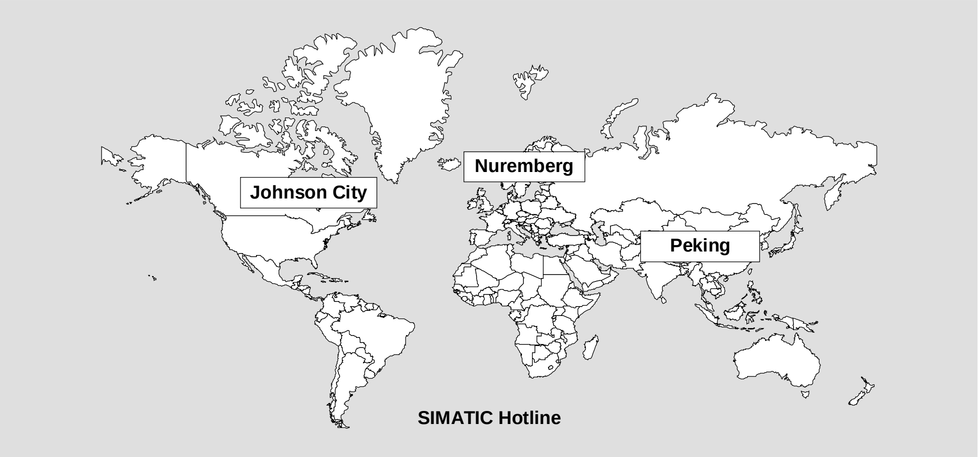
The following names are registered trademarks of Siemens AG:

SIMATIC®, SIMATIC HMI®, SIMATIC Multi Panel®, SIMATIC Panel PC®, SIMATIC Multifunctional Panel®, HMI®, WinCC®, ProTool®, ProTool/Lite®, ProTool/Pro®

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.

Customer and Technical Support

Available round the clock, worldwide:



Worldwide (Nuremberg)		**Worldwide** (Nuremberg)		**United States** (Johnson City)	
Technical Support (FreeContact)		**Technical Support** (fee-based, only with SIMATIC Card)		**Technical Support and Authorization**	
Local time:	Mo.-Fr. 7:00 to 17:00	Local time:	0:00 to 24:00, 365 a day	Local time:	Mo.-Fr. 8:00 to 19:00
Telephone:	+49 (180) 5050-222	Telephone:	+49 (911) 895-7777	Telephone:	+1 423 461-2522
Fax:	+49 (180) 5050-223	Fax:	+49 (911) 895-7001	Fax:	+1 423 461-2289
E-Mail:	techsupport@ ad.siemens.de			E-Mail:	simatic.hotline@ sea.siemens.com
GMT:	+1:00	GMT:	+1:00	GMT:	–5:00
Europa/Afrika (Nuremberg)		**Asien/Australien** (Peking)			
Authorization		**Technical Support and Authorization**			
Local time:	Mo.-Fr. 7:00 to 17:00	Local time:	Mo.-Fr. 8:30 to 17:30		
Telephone:	+49 (911) 895-7200	Telephone:	+86 10 64 75 75 75		
Fax:	+49 (911) 895-7201	Fax:	+86 10 64 74 74 74		
E-Mail:	authorization@ nbgm.siemens.de	E-Mail:	adsupport.asia@ siemens.com		
GMT:	+1:00	GMT:	+7:00		

The languages spoken by the SIMATIC Hotlines are generally German and English.

SIMATIC Customer Support Online Services

The SIMATIC Customer Support team offers you substantial additional information about SIMATIC products via its online services:

- General current information can be obtained

  In the Internet under
  http://www.siemens.com/simatic

- Current Product Information leaflets, FAQs (Frequently Asked Questions), Downloads, Tips and Tricks can be obtained

  In the Internet under
  http://www.siemens.com/automation/service&support

Training Center

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

Telephone:	+49 (911) 895-3200
Internet:	http://www.sitrain.com
E-Mail:	info@sitrain.com

Other Sources of Assistance

In case of technical queries, please contact the Siemens representatives in the subsidiaries and branches responsible for your area.

The addresses can be found:

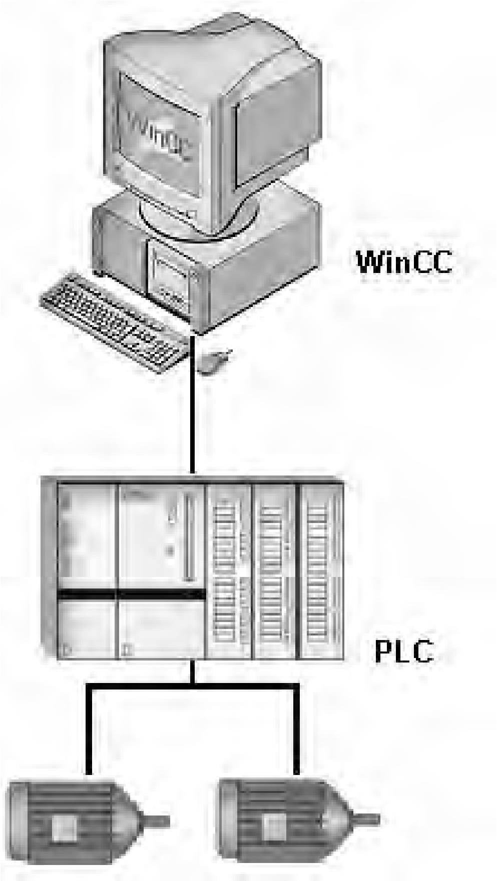
- In the Siemens Catalog ST 80
- In the Internet under
  http://www.siemens.com/automation/partner
- In the Interactive Catalog CA01
  http://www.siemens.com/automation/ca01

Contents

1 SIMATIC WinCC

What is WinCC?

WinCC is a powerful HMI system for use under Microsoft Windows 2000 and Windows XP. HMI stands for "Human Machine Interface", i.e. the interface between the person (the operator) and the machine (the process). The automation process (AS) retains actual control over the process. Communication between the WinCC and the operator on the one hand and WinCC and the automation systems on the other is affected.



WinCC is used to visualize the process and develop the graphic user interface for the operator.

- WinCC allows the operator to observe the process. The process is displayed graphically on the screen. The display is updated each time a status in the process changes.
- WinCC allows the operator to control the process. He can, for example, predefine a setpoint or open a valve from the graphic user interface.
- An alarm will automatically signal in the event of a critical process status. If, for example, a predefined limit value is exceeded, a message will appear on the screen.
- When working with WinCC, process values can either be printed or electronically archived. This facilitates the documentation of the process and allows subsequent access to past production data.

The distinguishing features of WinCC

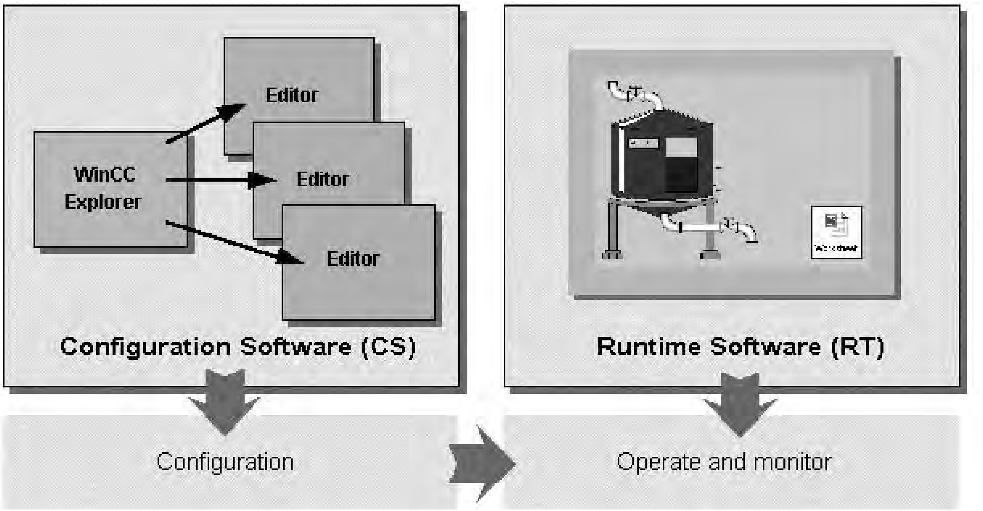
WinCC can be optimally integrated into your automation and IT solutions:

- Being a part of the Siemens TIA concept (Totally Integrated Automation), WinCC works very efficiently with automation systems that belong to the SIMATIC product family. Automation systems from other producers are also supported.
- WinCC data can be exchanged with other IT solutions through standardized interfaces, such as with MES and ERP-level applications (a SAP system for example) or with programs such as Microsoft Excel.
- The open WinCC programming interfaces allow you to connect your own programs and you will be able to control the process and process data.
- WinCC can be optimally customized to meet the requirements of your process. An extensive range of configuration possibilities is supported from single-user systems and client-server systems right up to redundant distributed systems with several servers.
- Your WinCC configuration can be modified at any time even subsequently. This will not interfere with existing projects.
- WinCC is an Internet-compatible HMI system which facilitates the implementation of web-based client solutions as well as Thin-client solutions.

1.1 How WinCC works

Structure of WinCC

WinCC is a modular system. Its basic components are the Configuration Software (CS) and Runtime Software (RT)



Configuration software

WinCC Explorer will open immediately after you start WinCC. WinCC Explorer forms the core of the Configuration software. The entire project structure is displayed in WinCC Explorer. The project is also administered here.

Special editors which can be called from WinCC Explorer have been provided for configuration purposes. Each editor is used to configure a special WinCC subsystem.

The most important WinCC subsystems are:

- The Graphics System the editor that is used to create pictures is known as the Graphics Designer.
- Alarm Logging the process of configuring messages is referred to as Alarm Logging.
- The Archiving System the Tag Logging editor is used to determine which data is archived.
- The Report System the editor that is used to create report layouts is known as the Report Designer.

- User administration the editor that is used to administer users is, as the name implies, known as the User Administrator.
- Communication this is configured directly in WinCC Explorer.

All configuration data is saved in the CS database.

Runtime software

The Runtime software allows the user to operate and monitor the process. It is primarily used to execute the following tasks:

- It reads the data that has been saved in the CS database
- It displays pictures on the screen
- It communicates with the automation systems
- It archives current runtime data, e.g. process values and message events
- It controls the process, e.g. through setpoint input or switching On and OFF

Performance data

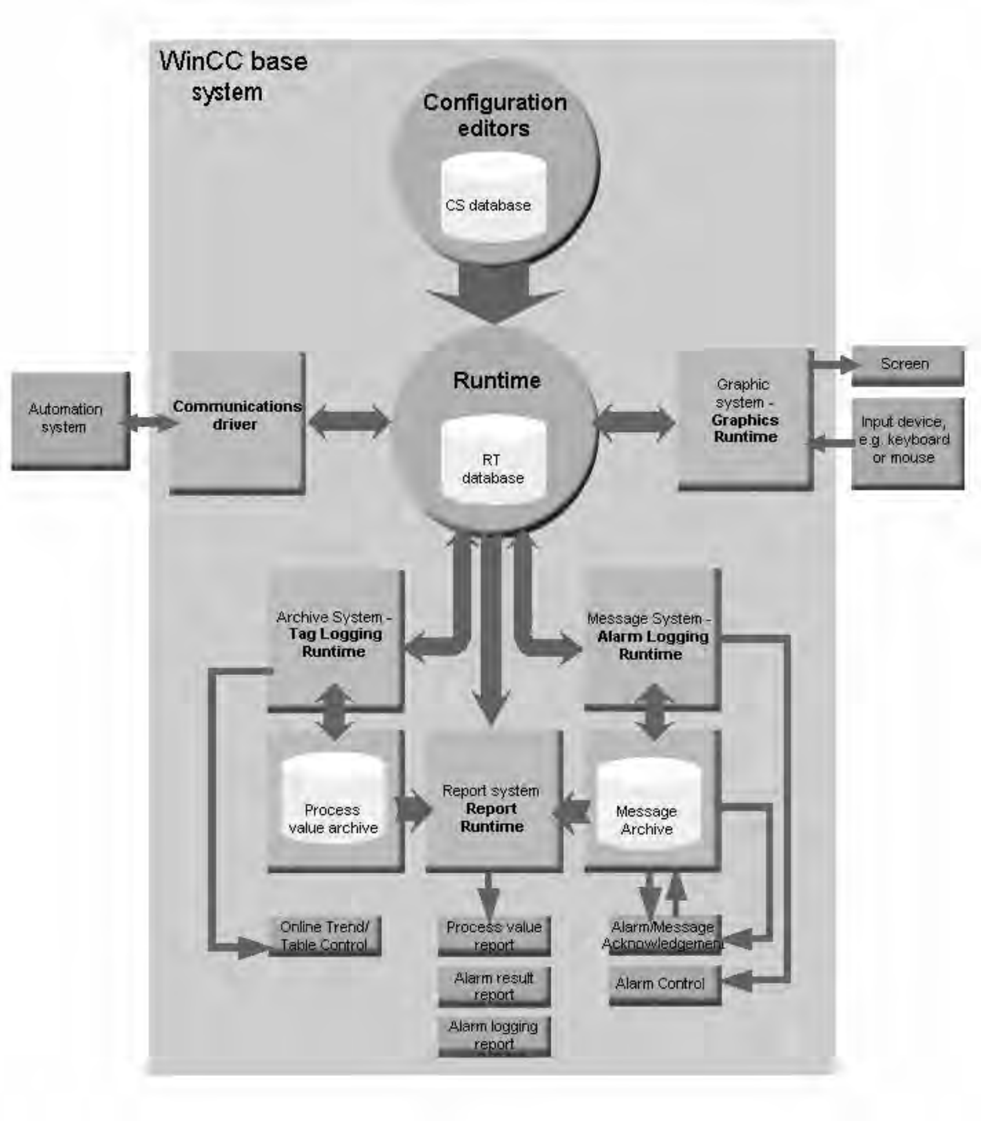
The performance data will be directly determined by the PC hardware that is used and the manner in which the system is configured. You will find examples of different system constellations in the WinCC Information System at "Performance data".

1.2 WinCC Function Chart

Overview

The following graph summarizes the interaction between the WinCC subsystems. This provides important information relating to the sequence that is employed for configuration.

For example, the Report Designer provides the print jobs for the output of reports and logs. Data cannot be printed until you configure the corresponding layout in Report Designer.


WinCC base system
Configuration editors
CS database
Runtime
RT database
Automation system
Communications driver
Graphic system - Graphics Runtime
Screen
Input device, e.g. keyboard or mouse
Archive System - Tag Logging Runtime
Message System - Alarm Logging Runtime
Process value archive
Report system Report Runtime
Message Archive
Online Trend/ Table Control
Process value report
Alarm result report
Alarm logging report
Alarm/Message Acknowledgement
Alarm Control

Workflow

You will use the editors in the Configuration software to create your project. All WinCC editors store their project information in the Configuration database (CS database).

In runtime, the project information is read out of the Configuration database by the Runtime software and the project is executed. Current process data is temporarily stored in the Runtime database (RT database)

- The Graphics System displays pictures on the screen. Conversely, it also accepts operator input, such as when the operator clicks on a button or enters a value.
- Communication between WinCC and the automation systems is effected by means of communication drivers, or "channels". The channels have the task of collecting the process value requirements of all runtime components, reading the values of the process tags out of the automation systems and, if necessary, writing new values into the automation systems.
- The exchange of data between WinCC and other applications might be performed by means of OPC, OLE or ODBC.
- The Archiving System saves the process values in the process value archive. The archived process values are, for example, needed to display the temporal development of these values in Online Trend Control or in Online Table Control.
- The individual process values are monitored by Alarm Logging. If a limit value is exceeded, Alarm Logging will generate a message which will be issued in Alarm Control. The message system also receives the acknowledgements made by the operator and manages the message states. Alarm Logging saves all messages in the message archive.
- The process will be documented by the Report System on request or at predefined times. The Process value archive and the message archive are accessed for this purpose.

You will find more information about WinCC editors and communication in the "WinCC Information System".

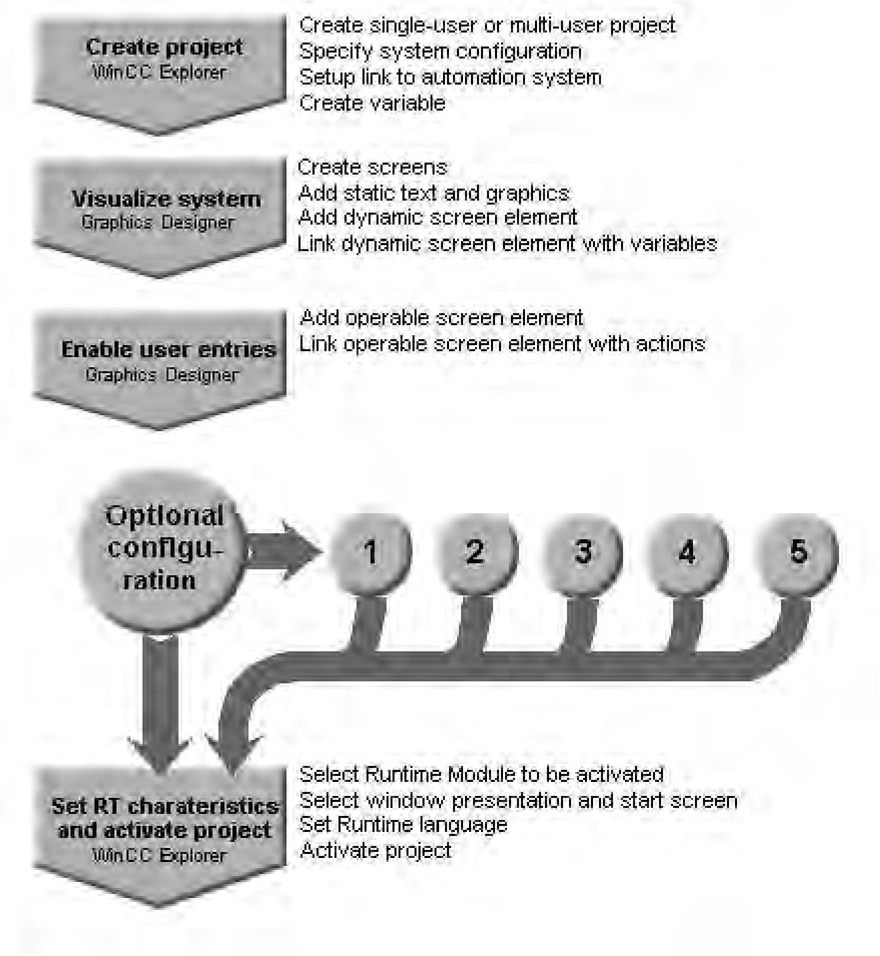
1.3 Guide Through Your Projects

Optimal sequence

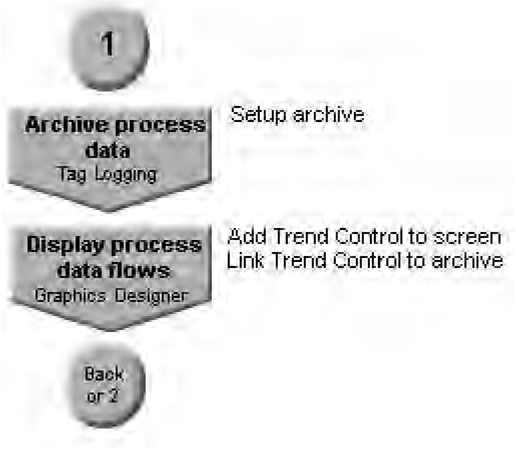
In WinCC, certain configuration steps supplement configuration steps already effected. Therefore, some configuration steps can only be effected after other steps have been carried out.

The following overview serves as a "thread" which will guide you through the configuration phase.

Obligatory tasks

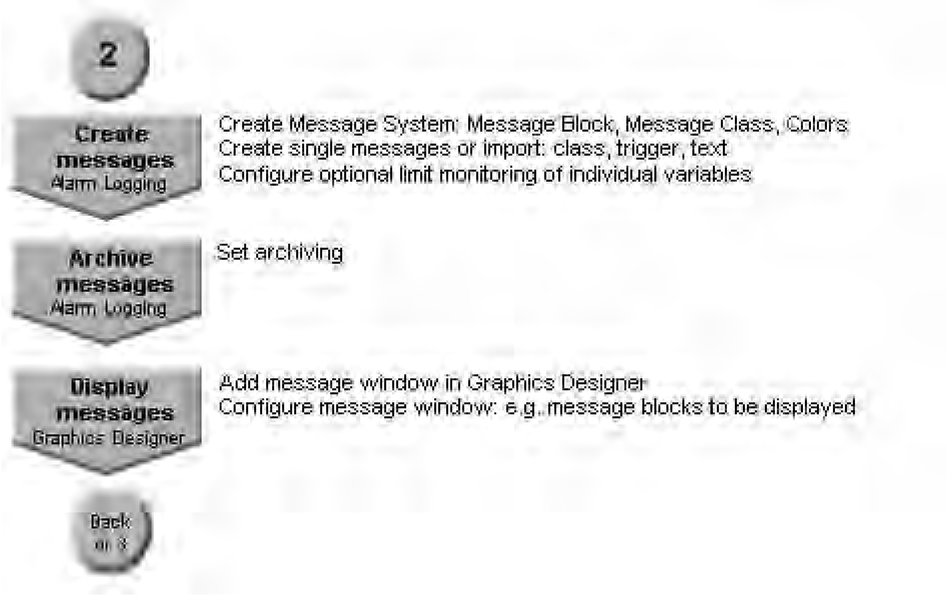

Create project
WinCC Explorer
Create single-user or multi-user project
Specify system configuration
Setup link to automation system
Create variable
Visualize system
Graphics Designer
Create screens
Add static text and graphics
Add dynamic screen element
Link dynamic screen element with variables
Enable user entries
Graphics Designer
Add operable screen element
Link operable screen element with actions
Optional configu-ration
1
2
3
4
5
Set RT charateristics and activate project
WinCC Explorer
Select Runtime Module to be activated
Select window presentation and start screen
Set Runtime language
Activate project

Optional configuration



Setup archive

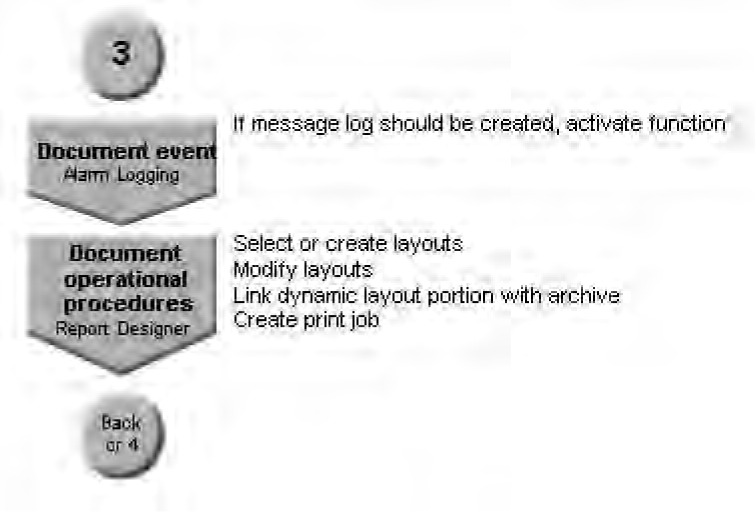
Add Trend Control to screen
Link Trend Control to archive



Create Message System: Message Block, Message Class, Colors
Create single messages or import: class, trigger, text
Configure optional limit monitoring of individual variables

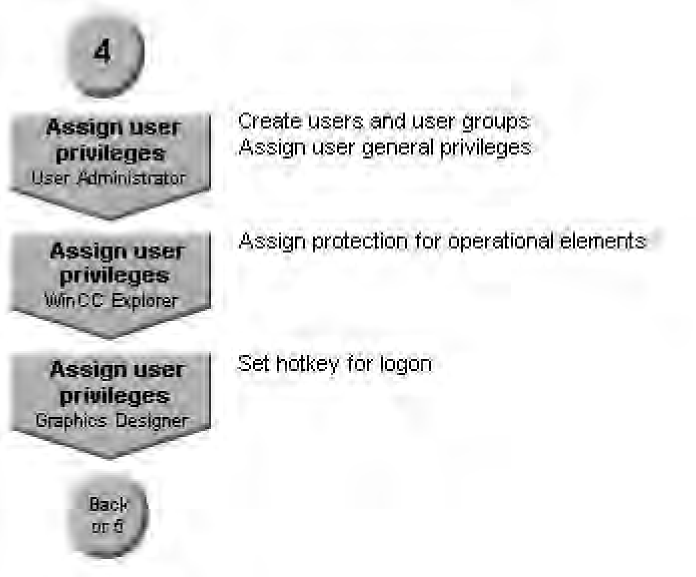
Set archiving

Add message window in Graphics Designer
Configure message window: e.g. message blocks to be displayed



If message log should be created, activate function

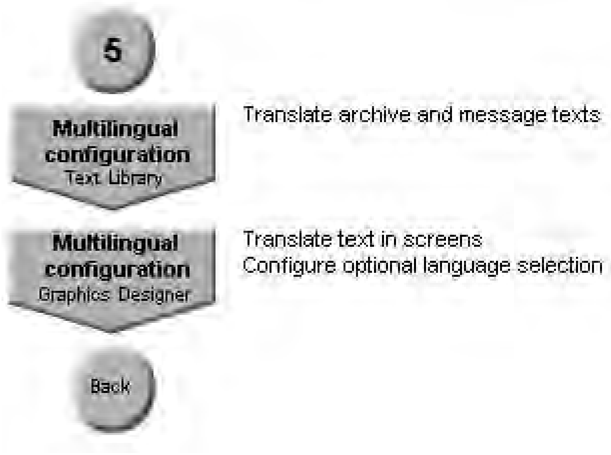
Select or create layouts
Modify layouts
Link dynamic layout portion with archive
Create print job



Create users and user groups
Assign user general privileges

Assign protection for operational elements

Set hotkey for logon



5
Multilingual configuration
Text Library
Translate archive and message texts
Multilingual configuration
Graphics Designer
Translate text in screens
Configure optional language selection
Back

2 Working with Projects

2.1 Editors and Functions in the WinCC Explorer

Introduction

You can see the installed WinCC editors and functions in the navigation window of the WinCC Explorer. The editor belonging to an option is only visible in the navigation window when the option is installed.

Some tools and some WinCC options are not displayed in the WinCC Explorer. You can open the tools in the Windows Start menu in the "Simatic" folder under "WinCC" > "Tools". You can open some of the editors of the WinCC options independently of WinCC. For more detailed information, refer to the help on the specific options.

Opening an Editor

You can open an editor in the WinCC Explorer in the following ways:

- Using the context-sensitive menu of the editor in the data window or in the navigation window
- By double-clicking on the editor in the navigation window
- Using the "Editors" menu in the menu bar

List of Editors and Functions

The following two tables list all the editors and functions that you can open using the WinCC Explorer.

The tables contain the following information:

- Object: Name of the editor or function in the WinCC Explorer.
- Use: This tells you what the object is used for.
- Online Help: This column names the relevant sections in the WinCC Information System that contain more detailed information on the object.
- Import/Export Tools: This column lists tools with which you can import or export data.
- Language Change: This informs you whether or not foreign languages can be configured.

- Online Configuration: This informs you whether or not the object can be used while the project is active in runtime. You will find information on restrictions regarding online configuration in the section "Online Configuration" and in the description of the editors.

Object	**Usage**	**Online Help** [1)]	**Import/Export Tools**	**Language Change**	**Online Configuration**
Computer	Computer name and properties, project properties (client and servers)	Working with Projects	---	Yes	Yes [2)]
Tag Management	Management of tags: creating and editing tags and communication drivers	Working with Tags	WinCC Configuration Tool	---	Yes [2)]
Structure Tag	Creating and editing structure types and structure tags	Making process pictures dynamic	WinCC Configuration Tool	---	Yes [2)]
Graphics Designer	Creating and editing process pictures	Creating process pictures	Export function of the editor	Yes	Yes
Alarm Logging	Configuring messages and archiving events	Structure of a Message System	WinCC Configuration Tool Text Library	Yes	Yes [2)]
Tag Logging	Logging and archiving tags	Working with Process Values	WinCC Configuration Tool	---	Yes
Report Designer	Configuring reports and report layouts	Documentation of Configuration and Runtime Data	---	Yes	Yes [2)]

Object	Usage	Online Help [1]	Import/Export Tools	Language Change	Online Configuration
Global Script	Making a project dynamic with C functions and actions	ANSI-C for creating functions and actions	Export function of the editor	Yes	Yes
Text Library	Creating and editing language-dependent user texts	Structure of Multilanguage Projects	Export function of the editor	Yes	Yes
User Administrator	Managing access permissions for users and user groups	Structure of User Management	Text Library	Yes	Yes
Cross Reference	Localizing, displaying, and rewiring the location at which objects are used	Working with Cross-reference Lists	---	---	Yes
Load Online Changes	Transferring edited data to the operator station	Working with Projects	---	---	Yes [2]
Server Data	Creating and editing packages for multi-user systems	Configurations > Multi-User Systems	---	---	Yes

[1] You will find the chapters listed in the WinCC Information System in the book "Working with WinCC".

[2] with restrictions

Options

Object	Usage	Online Help	Import/ Export Tools	Language Change	Online Configuration
NetCC	WinCC Diagnostics	WinCC Diagnostics > NetCC	Diagnostics file of the editor	---	Yes
Redundancy	Operating two servers at the same time in a redundant system	Configurations > Redundant Systems	---	---	Yes
User Archive	Configurable database system for data from technical processes, for example for recipes and setpoints	Options > User Archives	Text Library	Yes	Yes
OS Project Editor	Initializing and configuring the runtime user interface and alarm systems in PCS 7	Options for Process Control > OS Project Editor	---	---	---
Time Synchronization	Synchronizing the time of day on all clients and servers	Options for Process Control > Timesynchronization	---	---	Yes
Horn	Indicating message-relevant events on signal modules and PC sound cards	Options for Process Control > Horn	---	---	Yes
Picture Tree Manager	Managing picture hierarchies and name hierarchies	Options for Process Control > Picture Tree Manager	Text Library	Yes	Yes

Object	**Usage**	**Online Help**	**Import/ Export Tools**	**Language Change**	**Online Configuration**
Lifebeat Monitoring	Permanent monitoring of the system	Options for Process Control > Lifebeat Monitoring	---	---	---
ProAgent	Configuring process diagnostics to detect and eliminate problems	Options > ProAgent	---	---	Yes
WebNavigator	configuring the Web project	Options > WebNavigator	---	---	---
Dat@Monitor	functions for displaying archive data and current process values	Options > WebNavigator > Dat@Monitor	---	---	---

2.2 Types of Project

Introduction

There are three types of project available in WinCC:

- Single-user project
- Multi-user project
- Client project

2.2.1 Single-User Project

Introduction

If you only want to work with one computer in a WinCC project, create a single-user project.

The WinCC project runs on one computer that functions as the server for processing the data and as an operator input station. Other computers cannot access the project.

Principle

The computer on which you create the single-user project is configured as a server.

The computer is connected to the programmable controller via the process communication.

Redundancy

You can also create a single-user project as a redundant system. In this case, you configure a single-user project with a second redundant server.

Archive Server

You can also create an archive server for a single-user project. In this case, you configure a single-user project and a second server on which the data of the single-user project is archived.

2.2.2 Multi-User Project

Introduction

If you only want to work with several computers in a WinCC project, create a multi-user project.

For a multi-user system, there are two basic options:

- Multi-user system with one or more servers: Several servers with one or more clients. One client accesses several servers. The runtime data is distributed on different servers. The configuration data is on the servers and on the clients.
- Multi-user system with only one server: One server with one or more clients. All the data is on the server.

Principle

You create a multi-user project on the server. The server is connected to the programmable controller via the process communication.

In the multi-user project, you configure the clients that access the server. In a second step, you create the required client projects on the relevant computers.

If you want to work with several servers, duplicate the multi-user project on the second server. Adapt the duplicated project accordingly. You can also create a second multi-user project on the second server that is independent of the project on the first server.

A server can also access another server in the role of client. You can use this option, for example, when you use an archive server or a file server.

2.2.3 Client Project

Introduction

If you create a multi-user project, you must then create the clients that access the server. You create a client program on the computer that will be used as a client.

For a WinCC client, there are two basic options:

- Multi-user system with one or more servers: The client accesses several servers. The runtime data is distributed on different servers. The configuration data on the multi-user projects is on the relevant servers. There may be local configuration data in the client projects on the clients: Pictures, scripts, and tags.
- Multi-user system with only one server: The client accesses a single server. All the data is located on the server and is referenced on the clients.

An archive server or a file server can also access another server in the role of client.

Note:
The WinCC client replaces the clients and multi-clients that were used in WinCC up to version V5.1. Depending on the configuration, a WinCC client takes over the role of a V5.1 client or a V5.1 multi-client.

Principle

You create a multi-user project on the server. The server is connected to the programmable controller via the process communication. In the multi-user project, you create the clients that access the server.

If you configure a multi-user system with only one server, you do not create a separate client project on the WinCC client.

If you configure a multi-user system with several servers, you must create a separate client project on each client. This also applies when you only want to access one server but require additional configuration data on the client.

Multi-User System with One or More Servers

To access more than one server, you create a client project on the client. You specify the project properties on the WinCC client.

On the server, you create packages using the Serverdata component. The packages contain all the important configuration data of the multi-user project. You load the packages on the WinCC client.

You only need to create and compile the packages once manually. If the configuration data on a server is modified, WinCC automatically generates the required packages. The packages can be downloaded to the clients automatically or manually.

Central Server Configuration for a Multi-User System with One Server

If you want to configure a client that accesses only one server, specify all the settings in the multi-user system on the server. When you edit the startup list of the client, you should only start applications that are actually required on the client.

You do not create a separate project on the client. You start the server project using remote access. You will find more detailed information in the WinCC Information System under "Configurations" > "Multi-User Systems".

Web Client

You can configure a client that accesses the server over the intranet or over the Internet. If you require this type of access, you create a Web client with the WinCC Web Navigator option.

2.3 Creating and Editing Projects

2.3.1 Preparing to Create a Project

Introduction

To configure in WinCC, you do not require detailed planning. To create a WinCC project efficiently, you should nevertheless give some initial thought to the structure of the project. Depending on the size of the planned project and number of configuration engineers involved, it may be useful to make certain settings and decide on certain rules.

This section contains information on some of the elements in a project that you can specify before you start the configuration work:

- Project type
- Project path
- Naming conventions
- Tag groups
- Picture hierarchy
- Reusing project sections

Project type

Before you start to plan your project, you should already know whether you require a single-user system or multi-user system. If you are planning a project with WinCC clients or Web clients, make sure you know the factors affecting performance.

Project path

A WinCC project does not need to be created on the same partition in which you installed WinCC. It is sometimes better to create a separate partition for a project. When you create a partition, make sure you have adequate space for the anticipated amounts of data. If you archive a lot of data, the WinCC project can take up several gigabytes of space.

A separate partition also ensures that the WinCC project and all the data contained in it are not lost if there is a system crash.

Naming conventions

Using naming conventions can make it easier to handle large projects. You can increase the clarity in your project particularly if you use conventions for naming tags, pictures, or functions in your project. Note the restriction applying to names in the section "Illegal Characters".

Project name

Changing the name of a project once it has been created involves a number of steps. It is advisable to decide on a suitable name before creating the project.

Tags

You can give tags a prefix that identifies the tag type or the connection assigned to the tag. You could, for example, give all text tags the prefix "txt_" and internal tags the prefix "int_".

If you develop a company standard, the prefixes should be the same for all projects.

Pictures

You can specify prefixes for pictures, for example to identify plant pictures and system pictures.

If you create a large number of pictures, you can include continuous numbers in the picture names.

Functions

With functions, it is useful to introduce a prefix for your company standard. This makes it clear at a glance which functions are required for the standard.

Tag groups

To structure tags, you can create tag groups. In WinCC, you cannot nest groups but can only create one level with tag groups.

Picture Hierarchy

If you want to reduce the configuration time, you should plan the picture hierarchy in your project before you start the project. It is advisable to work out an overview of the pictures you need to create. Using a basic picture and the tag prefix, you can structure navigation within your project.

Reusable Project Sections

You can take various project sections from existing WinCC projects. These include pictures, tags, functions, and actions.

Standard Project

If you do not want to repeatedly take data from an existing WinCC project, you should create a standard project. In this project, you can configure basic project sections to suit your needs. When you create a new WinCC project, you can simply copy the standard project and then work with the copy. This saves you time during configuration.

2.3.2 Setting up and administering projects

Overview

In WinCC Explorer you will be able to set up and administer projects. The Project Assistant will guide you through the setting up phase.

Configuring with the Project Assistant

The Project Assistant will open automatically when you select the menu item "File > New". The Assistant asks you for the project type (single-user or multi-user project), the name of the project and where it is stored.

As soon as the Assistant has set up the project, the basic data of the project that has been created by the Project Assistant will appear in WinCC Explorer. The project name will appear in the title bar of WinCC Explorer.

Configuring with WinCC Explorer

You can also use WinCC Explorer to administer your projects.

- The individual operator consoles are configured using the component "Computer". Here, you also define which runtime components should be started when the project is enabled.
- The connection to the connected automation systems is established under the component "Tag Management". The tags that are required for data exchange with the automation systems are also defined here.

The remaining components have specialized editors for all further configuration tasks. These editors can be selected from the pop-up menu.

2.3.3 Running and Testing Projects

Overview

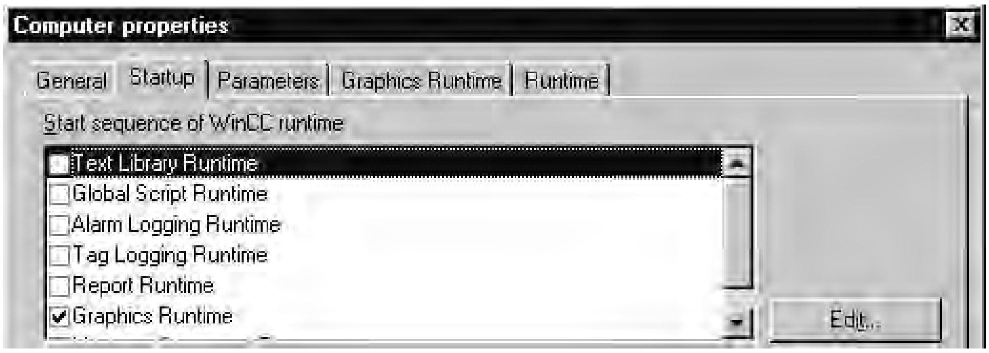
You will require the WinCC Runtime software to run your projects. If the Runtime software was installed together with the Configuration software you will not have to move to another workstation to carry out tests.

Configuring with WinCC Explorer

The runtime properties will have to be specified before you enable your project for the first time. The dialog "Computer properties" has been provided for this purpose. This dialog can be accessed via the pop-up menu of the component "Computer" in WinCC Explorer.

The "Startup" tab is used to specify which runtime components should be activated and which corresponding functions should be available in runtime. If, for example, your project contains cyclic actions, it will be necessary to activate the component "Global Script Runtime".

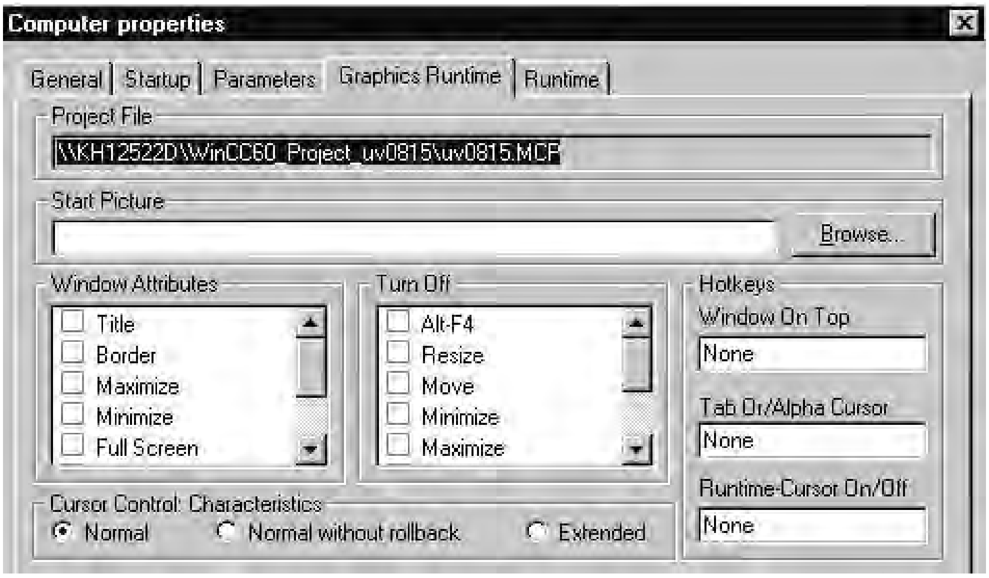
To achieve maximum performance, it is recommended that you only activate the components you really require.



Start sequence of WinCC runtime

- □ Text Library Runtime
- □ Global Script Runtime
- □ Alarm Logging Runtime
- □ Tag Logging Runtime
- □ Report Runtime
- ☑ Graphics Runtime

The "Graphics Runtime" tab can be used to define which screen should be displayed first once your project has been enabled (Start screen). It is also used to define the manner in which the WinCC project should appear on the screen.



Computer properties

General | Startup | Parameters | Graphics Runtime | Runtime

Project File: \\KH12522D\WinCC60_Project_uv0815\uv0815.MCP

Start Picture:

Browse...

Window Attributes:
- □ Title
- □ Border
- □ Maximize
- □ Minimize
- □ Full Screen

Turn Off:
- □ Alt-F4
- □ Resize
- □ Move
- □ Minimize
- □ Maximize

Hotkeys:
- Window On Top: None
- Tab Or/Alpha Cursor: None
- Runtime-Cursor On/Off: None

Cursor Control: Characteristics: ● Normal ○ Normal without rollback ○ Extended

Enabling projects

Once you have defined the runtime properties you will be able to enable the project. The "Enable" command is located in the "File" menu in WinCC Explorer. Alternatively, you can use the button in the toolbar.

When the project has been enabled, the selected components of the Runtime software will be started. You will now be able to control and test the project.

WinCC Simulator

Using WinCC Simulator, you can test your WinCC project during the development phase without connecting to the process peripherals. You can also test your WinCC project while connected process peripherals but without the process running.

- You will be able to define a fixed value for a tag.
- The value of a tag can also be modified automatically over the period of time, e.g. ascending, descending, in the form of a sine curve or on the basis of random variation.

WinCC Simulator can be installed using the WinCC's Setup program.

Testing projects

All projects that are created with WinCC should be subjected to thorough and systematic checks like any other software. The first step involves testing on a module basis with simulated tag values. The second step involves testing the entire functionality of the project with all automation components.

Online configuration

If a fault is ascertained during the testing phase, this can immediately be rectified in WinCC without stopping the process. Switch to the Configuration software using the shortcut <Alt>+<Tab>. Make the alteration, save the data and then return to the Runtime software. The process will run interruption-free with the new data.

Deactivating projects

To deactivate your project, switch to the Configuration software with the shortcut <Alt>+<Tab>. Click on the "Deactivate" button in WinCC Explorer toolbar to stop runtime. Alternatively, you can also assign this function to a button on one of your screens.

2.3.4 Illegal Characters

Introduction

Depending on the language and components, only certain characters are permitted in names.

In WinCC, you can use all the characters of the ASCII character set. We do, however, recommend that you avoid special national characters. Above all, you should avoid special characters in object names in the object name is used in scripts.

The following table lists the characters that must not be used in WinCC components, identifiers, and names.

Illegal Characters in WinCC

Component	Illegal Characters
WinCC Project: Names of WinCC projects	. , ; : ! ? " ' + = / \ @ * [ ] { } < > space case sensitive
Tags: Tag names	. : ? " ' \ * $ % space not case sensitive "@" is used only in system tags. The period is used as a separator in structure tags. Names beginning with "$" are not visible in tag management.
Tags: Names of process tags in Tag Logging	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ * % & § ° [ ] { } < > space
Tags: Names of archive tags in Tag Logging	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ * # % & § ° [ ] { } < > space
Tags: Names of tag groups	? ' \ space not case sensitive

Component	Illegal Characters
Structure types: Names of structure types, structure elements, structure instances	. : ? ' \ @ * % space
Graphics Designer: Names of pictures (PDL files)	: ? " / \ * < >
Graphics Designer: Names of objects in pictures	The name can be a maximum of 180 characters long. If you use special characters, the maximum number of characters is further restricted. Avoid using special characters in the object name is used in scripts. For more detailed information, refer to the documentation on VBS in section "Testing with the Debugger" > "Action and Procedure Names in the Debugger".
Graphics Designer: Text list object type	Restriction for assigned and referenced texts: ;
Graphics Designer: Names in the Dynamic Wizard	%
Alarm Logging: Names of message blocks, message classes, message types, and message texts	' Enter key line feed
Tag Logging: Archive name	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ * # % & § ° ( ) [ ] { } < > space
Tag Logging / Trend Control: Labeling time axis and value axis	Single "&" character is not displayed. Double "&" character is displayed once.
Report Designer: Names in the page layout and line layout	: ? " / \ * < >
User Administrator: User names	' \ not case sensitive
User Administrator: Passwords	' \ case sensitive
User Administrator: User rights	' \

Component	Illegal Characters
User Archives: Names for archives, fields, views, columns	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ @ * # $ % & § ° ( ) [ ] { } < > space The first character must be a letter.
Server Data: Names of packages	, / \ National special characters, for example umlauts (ä, ü etc.) are not permitted.

Illegal Characters for Basic Settings

Component	Illegal Characters
Computer name	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ ¦ @ * # $ % & § ° ( ) [ ] { } < > space only uppercase relevant The first character must be a letter.
DNS host names	, ; : ! ? " ' ^ ´ ` ~ _ + = / \ ¦ @ * # $ % & § ° ( ) [ ] { } < > space
Folder path: Names of folders	: ? " / * < >
WinCC Explorer	Restrictions depending on individual components
Communication: Names of connections under a channel unit	Restrictions as for the SQL database
Communication / OPC: Used names	. : ? " ' \ * % space
Web client: Used names	. , ; : ! ? " ' ^ ´ ` ~ - + = / \ @ * # $ % & § ° ( ) [ ] { } < > space

Illegal Characters when Integrating in the SIMATIC Manager

Component	Illegal Characters
Simatic Manager: Names of WinCC projects	. , ; : ! ? " ' + = / \ @ * % [ ] { } < > space
Compiling OS: PLC/OS connection names	. : ? " ' \ * % space
PCS 7: Hierarchy folder	. " / \ %

2.3.5 Online Configuration

Introduction

You can edit a project in a single-user system or a multi-user system during runtime. Remember, however, that some configuration tasks are not possible.

WinCC also provides the "Load Online Changes" function. With the Load Online Changes function, you can edit a project on one computer while it is running on another computer in runtime. When configuring with Load Online Changes, there are other restrictions than those that apply to online configuration.

Note:
In time-critical projects, remember that online configuration can affect the response of your project over time.

Restrictions

For more detailed information on the possibilities for online data editing, refer to the description of the relevant editor in the WinCC Information System.

You can edit the following elements while the project is activated in runtime:

Element	Create	Delete	Edit	Remarks
Project name / project properties	---	---	No	---
Computer	Yes	No	No	---
Computer properties	---	---	Yes	Edit: Changes are only adopted in runtime, after deactivating the project and restarting runtime.
Timebase (computer properties)	---	---	Yes	Edit: Depending on the configuration, modifications to a picture are only updated in runtime after reselecting the picture.
User cycle (project properties)	---	---	No	---
Tags (tag management)	Yes	Yes	Yes	Delete/Edit: Only possible with external tags if the channel used supports the function. This currently applies only to the S7 channel

Element	Create	Delete	Edit	Remarks
Tag groups (tag management)	Yes	Yes	Yes	Delete: Possible only when the tags contained can be deleted.
Structure types	Yes	Yes	Yes	Delete: Possible only if the structure type does not contain any structure tags. Edit: Possible only when no structure tag has yet been created in the structure type.
Structure tags (structure types)	Yes	Yes	Yes	Delete: Possible only when the structure tag elements contained can be deleted.
Connections (tag management)	Yes	Yes	Yes	Modifications to a connection can lead to loss of data. Delete/Edit: Currently applies only to the S7 channel.
Channels (tag management)	No	No	No	---
Channel units (tag management)	No	No	No	---
Text entries (text library)	Yes	Yes	Yes	---
Pictures (PDL files, Graphics Designer)	Yes	Yes	Yes	A picture is updated in runtime only after reselecting the picture.
Library, color palette (Graphics Designer)	Yes	Yes	Yes	---
Archives, archive tags (Tag Logging, User Archives)	Yes	Yes	Yes	---
Reports, layouts (Report Designer)	Yes	Yes	Yes	Create/delete/edit: Applies only to runtime documentation.
Scripts (Global Script)	Yes	Yes	Yes	---
Access rights (User Administrator)	Yes	Yes	Yes	---

Element	Create	Delete	Edit	Remarks
Other files	Yes	Yes	Yes	---
Packages (Serverdata)	Yes	Yes	Yes	---
Messages (Alarm Logging)	---	---	---	Refer to the following table "Online Configuration with Messages"

Online Configuration with Messages

Element	Attribute	Create	Delete	Edit	Remarks
Single messages		Yes	Yes	Yes	Edit: Possible with most attributes. Exceptions are listed separately:
	-Number -Group	---	---	No	---
	-Class	---	---	Yes	If you modify a message so that acknowledgment is no longer mandatory, the message is deleted. If you change the message class, but not the acknowledgment setting, the message is updated immediately.
	-Type	---	---	Yes	The message is updated immediately.
	-Triggered on falling edge -Message tag -Message bit -Standard DLL	---	---	Yes	If you change the attribute, the message is deleted.
	-Controls the Central Signaling Device -Will be archived -Priority	---	---	Yes	Updated in runtime only when the status of the message changes.

Element	Attribute	Create	Delete	Edit	Remarks
	-Infotext	---	---	Yes	The message is updated immediately.
	- User text block - Process value block	---	---	Yes	Updated in runtime only when the status of the message changes.
Group messages		No	No	No	---
Limit value monitoring		Yes	Yes	Yes	Edit: If you change the "Message" attribute, the message is deleted.
System blocks		Yes	Yes	Yes	Create/Delete/Edit: Updated in runtime only after you reselect the picture. Create/Delete: Reconfiguration in OCX necessary.
User text blocks		Yes	Yes	Yes	Create/Delete/Edit: Updated in runtime only after you reselect the picture. Create/Delete: Reconfiguration in OCX necessary.
Process value blocks		Yes	Yes	Yes	Create/Delete/Edit: Updated in runtime only after you reselect the picture. Create/Delete: Reconfiguration in OCX necessary.
Message classes		Yes	No	Yes	Edit: Possible with most attributes. Exceptions are listed separately:
	-Class name -Status text	---	---	Yes	Updated in runtime only when the status of the message changes or when you reselect the picture.

Element	Attribute	Create	Delete	Edit	Remarks
	- Acknowledgment strategy	---	---	Yes	If you modify a message so that acknowledgment is no longer mandatory, the message is deleted.
Message types		Yes	No	Yes	---
Raw data tags		Yes	Yes	Yes	Change/Delete: The message is deleted.
Tags		---	Yes	Yes	Change/Delete: The message is deleted.

2.3.6 Downloading Changes Online

Introduction

With the Load Online Changes function, you can edit a project on one computer while it is running on another computer in runtime. The computer on which you configure will be called the Engineering Station (ES) in the following description. The computer on which runtime is activated will be called the Operator Station (OS).

If you want the changes to take effect on the operator station, start a download of the modified data. The project is updated in runtime.

You can test the modified project on the engineering station before you user modifications in runtime. You can update the project at any time.

Working with online downloads of changes is possible only when you maintain a consistent project.

Note:
In time-critical projects, remember that online configuration can affect the response of your project over time.

Application Scenarios

During various phases, for example commissioning, operation, or maintenance, you will normally find it necessary to make changes to an existing project. These changes can then be adopted in the activated project; in other words, online.

You can download changes online in the following situations:

- Continuous automation tasks: All modifications are made online from a central configuration station. As a result, you do not need to make a configuration changes directly on site. It is also possible to add, modify, and delete runtime objects, for example tags, alarms, and archives without deactivating WinCC.
- Testing Modifications in a Protected Environment: The intended modifications can be made initially on the configuration station offline before they are downloaded to the activated project over Ethernet. The configuration engineer can therefore test the modifications in a protective environment before adopting them in active operation. This allows configuration errors to be detected before they can cause problems in the process or stop the plant.
- Downloading Mutually Dependent Modifications at the Same Time: Changes in a configuration often involve interdependent settings. Such modifications should be downloaded consistently and take effect at the same time. This situation occurs, for example, when adding a calculated value to an archive display. In may be necessary to create new tags and add them to a new archive. Finally, the archive will be displayed in a picture. In the situation, if the information is not adopted as a unit, this will lead to error messages.

- Editing Projects in a Redundant System: Configuration changes can be transferred to a redundant server using online change downloads without the stored modifications being deleted later. The same changes can be transferred to the second server. This means that the current version is available on both servers in the redundant system practically without any delay.

Principle

A project runs on the operator station during runtime. You edit a copy of the project on a second computer, the engineering station. On the engineering station, you activate the load online changes function in the WinCC Explorer. From this point onwards, WinCC records all the changes made in the project. Online configurations on the Engineering Station are not., however, recorded.

When you have completed your modifications in the project, start the online download of the changes. This exports the modified data to the operator station. The project is updated in runtime.

The modifications are updated in a project in the following order:

- Connections
- Tag groups, tags, structures
- Texts
- Messages
- Archives, archive tags
- User Administrator
- Pictures
- Scripts
- Reports

In a multi-user system, WinCC generates packages following each download, that are then exported to the clients and loaded.

If you use a redundant system, the redundant server data is synchronized following each download.

3 Working with Tags

3.1 Tag Management and tags

Introduction

Communication between WinCC and the automation system takes place by means of communication drivers. Data generated by the automation system or the WinCC project are transferred by means of tags.

This chapter provides infomation on managing tags and on the different types of tags.

3.2 Tag Management

Introduction

Tag Management administers the tags and communication drivers used in the project. You will find Tag Management in the navigation window of WinCC Explorer.

Usage

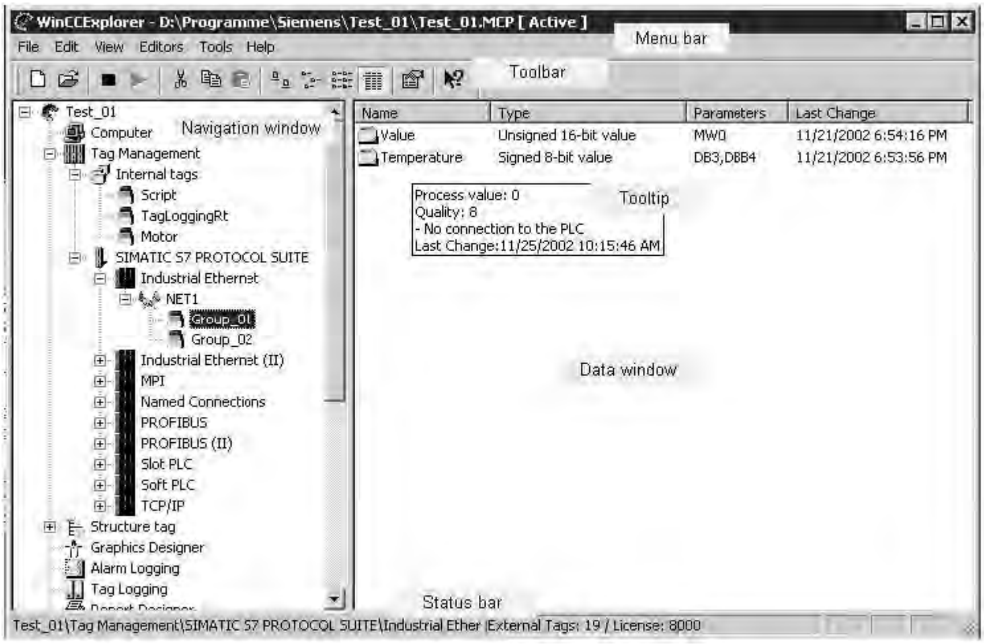
Tags with values supplied by the process are referred to as process or external tags in WinCC. In the case of process tags, Tag Management determines the communication driver by means of which WinCC is connected to the automation system and how the exchange of data is taking place. The associated tags are created in the directory structure of this communication driver.

Tags not supplied with values by the process, known as the "internal tags", are created in the "Internal Tags" directory.

To provide a better overview, you can arrange the tags in Tag Management in groups.

Creating tags in Tag Management leads to a directory structure in which you can navigate in a manner similar to Windows directories.

Configuration of Tag Management


WinCCExplorer - D:\Programme\Siemens\Test_01\Test_01.MCP [Active]
Menu bar
File Edit View Editors Tools Help
Toolbar
Navigation window
Tooltip
Data window
Status bar

- Test_01
  - Computer
  - Tag Management
    - Internal tags
      - Script
      - TagLoggingRt
      - Motor
    - SIMATIC S7 PROTOCOL SUITE
      - Industrial Ethernet
        - NET1
          - Group_01
          - Group_02
      - Industrial Ethernet (II)
      - MPI
      - Named Connections
      - PROFIBUS
      - PROFIBUS (II)
      - Slot PLC
      - Soft PLC
      - TCP/IP
  - Structure tag
  - Graphics Designer
  - Alarm Logging
  - Tag Logging

Name	Type	Parameters	Last Change
Value	Unsigned 16-bit value	MW0	11/21/2002 6:54:16 PM
Temperature	Signed 8-bit value	DB3,DBB4	11/21/2002 6:53:56 PM

Process value: 0
Quality: 8
- No connection to the PLC
Last Change:11/25/2002 10:15:46 AM

Test_01\Tag Management\SIMATIC S7 PROTOCOL SUITE\Industrial Ether External Tags: 19 / License: 8000

Navigation window

You will find Tag Management in the navigation window of WinCC Explorer.

Internal tags and associated tag groups will be found in the "Internal Tags" directory.

WinCC creates a new directory in Tag Management for each communication driver that is set up. Under the communication driver you will find a channel unit, its connections and associated tag groups and process tags.

Tag Management uses the following icons:

Icon	Meaning
▥	Tag Management
🗗	Internal Tags
▯	Communication drivers
▮	Channel Unit
⇄	Connection
▤	Tag Group
▭	Tag
⊢	Structure type

Data window

The data window of WinCC Explorer shows you the contents of the directory you selected in the navigation window.

Tooltip

In Runtime you can view the status information on connections and tags as a tooltip. To view it, move the mouse pointer in the data window to the connection or tag you want.

The tooltip contains the following information:

- with a connection, brief information about the state
- The current process value of a tag
- The quality code of a tag
- The date on which a tag was last changed

Menu bar

At "Edit" you can cut, copy, paste and delete tags and tag groups. At "Edit" > "Properties" you can view the properties of the selected tag, the communication driver, the channel unit or connection.

At "View" you can modify the way in which objects are presented in the data window and update the display.

Toolbar

For tags and tag groups you can use the "Cut", "Copy", "Paste" and "Properties" buttons.

You can modify the display in the data window by means of the "Large Icons", "Small Icons", "List" and "Details" buttons.

Status bar

The status bar can display the following information, regardless of the operated object:

- Current directory path in the project and tips on operation
- External tags: number of configured process tags
- License: number of licensed process tags (power tags)

Find

In Tag Management you can open the search function from the shortcut menu.

You can search for tags, tag groups, connections, channel units and communication drivers.

3.2.1 Tags

Introduction

Data are passed on in a WinCC project by means of tags. A tag has a data address and a symbolic name, which is used in the project. The data address is used in communication with the automation system.

WinCC works with two kinds of tags:

- Process tags
- Internal Tags

WinCC simplifies tag handling by means of two other object types:

- Tag groups
- Structure types

Name conventions

When naming tags, you must respect certain conventions:

- Tag names must be unique throughout the project. When tags are created, WinCC does not distinguish between upper and lower case characters in a name.
- Tag names must not be longer than 128 characters. In the case of structure tags, this limit applies to the whole expression "structure tag name + dot + structure tag element name".
- You must not use certain characters in tag names. You will find the characters you must not include in a name in the WinCC Information System at "Working with projects" > "Authorized characters"

Note:
You must not create tags whose name starts with an '@'. Tags with an @ prefix are created only by WinCC and PCS 7.

Note:
Remember that WinCC distinguishes between upper and lower case characters in a configuration. If you change upper and lower case characters in a tag name, you must adapt your configuration accordingly.

Updating tags

In the case of process tags, the current tag values are transferred over the communication connection between WinCC and the connected automation systems in Runtime. In this instance, WinCC accesses the data area on the automation system that is defined in the tag properties. The current value of the process tag is made available by WinCC in Tag Management after it has been transferred for further processing and evaluation. Conversely, WinCC can also write data back to the automation system.

In WinCC you set the frequency of data transfer and of updating in the display. Remember while you are configuring that periodic updating of tags results in severe loading of the system and can have an adverse effect on performance.

Importing tags

You can also create tags external to WinCC and import them into your project.

For example, you can establish a connection to the WinCC configuration tool for your project and create the tags with the help of the configuration tool in an Excel table. You assign certain properties to these tags in the table. WinCC can use the configuration tool to read in the data and then automatically creates the tags with the specified properties.

3.2.1.1 Process Tags

Definition

You use process tags for communication between WinCC and the automation system.

The properties of process tags depend on the communication driver being used. Therefore, create process tags in Tag Management also under a specific communication driver, its channel unit and its connections.

You can set the following properties for process tags:

- Name
- Data type
- Address on the channel unit
- Adapt format
- Limit values
- Start value
- Replacement value
- Scaling

Note:
Process tags are also referred to "external tags" on the system.

Usable data types

You can use the following data types for process tags:

- Binary tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag, 8-bit
- Text tag, 16-bit character set
- Raw data tag

Authorization for power tags

For WinCC zou purchase authorizations to be able to use a certain number of process tags and archive tags.

Power tags are licensed process tags. With an authorization covering 1024 power tags, you can run a WinCC project on a computer in Runtime that uses up to 1024 process tags. The number of licensed and configured process tags will be seen in the status bar of WinCC Explorer.

Authorizations are subdivided into the following types

- "RC": for configuration and Runtime of a certain number of process tags
- "RC": only for Runtime of a certain number of process tags

3.2.1.2 Internal Tags

Definition

Internal tags are not connected to the process.

You use Internal tags to manage data within a project or to transfer data to an archive.

You can set the following properties for internal tags:

- Name
- Data type
- Updates project-wide/on local computers (only relevant with multi-user projects without additional client projects)
- Limit values
- Start value

Usable data types

You can use the following data types for internal tags:

- Text reference
- Binary tag
- Signed 8-bit value
- Unsigned 8-bit value
- Signed 16-bit value
- Unsigned 16-bit value
- Signed 32-bit value
- Unsigned 32-bit value
- Floating-point number 32-bit IEEE 754
- Floating-point number 64-bit IEEE 754
- Text tag, 8-bit
- Text tag 16-bit character set
- Raw data tag

Authorization for archive tags

For WinCC you purchase authorizations to be able to use a certain number of process tags and archive tags.

Archive tags are the licensed archive tags in Tag Logging. The "WinCC CS" and "WinCC RT" basic authorizations contain up to 512 archive tags. If you wish to configure more than 512 archive tags, you require an additional archive authorization. You can see the number of configured archive tags in the status bar of Tag Logging.

System tags

WinCC applications create tags required for internal administration of the project. The names of these tags begin with an "@" character. You cannot delete or rename these tags. You can evalaute, but not change, the value of the tag.

Exceptions to this rule are tags created with the "Redundancy" option. They can be set by means of scripts, for example:

- @RM_MASTER
- @RM_MASTER_NAME
- @RM_SWITCHER

Note:
You must not create tags whose name starts with an '@'. Tags with an @ prefix are created only by WinCC and PCS 7.

3.2.1.3 Structure types and structure tags

Definition

WinCC Structure types simplify the creation of several tags having identical properties.

Designation	Definition	Displayed in...
Structure type	A structure type denotes a scheme that is used in WinCC to create a certain group of tags. A structure type contains at least one structure element.	Component "Structure Types"
Structure element	A structure element is a component of a structure type. It is used as a template when creating a structure tag with the help of a structure type.	Component "Structure Types" > Structure Type > "Structure Properties" dialog box

Designation	Definition	Displayed in...
Structure tag	A structure tag is a tag that was created with the help of a structure type. The template of a structure tag is a structure element. The name of a structure tag consists of the name of the structure instance and the name of the structure element being used. Both parts of the name are separated by a dot.	Component "Tag Management" > "Internal Tags" or a connection, under which the structure tags were created with the help of structure type.
Structure instance	If the tags defined in a structure type are created with the help of the structure type, a structure instance occurs. The structure instances created with the same structure type are displayed under this structure type in the data window. The name of a structure instance is also referred to as the tag prefix.	Component "Structure Types" > Structure Type

Principle

You define a certain tag structure when you create a structure type.

You create a structure type in WinCC Explorer. You create a structure element for each tag. You assign the properites to a structure element which the tag created by that should possess.

When creating a tag, assign a created structure type as data type. The structure types are displayed in the "Data Type" selection window according to standard data types. WinCC creates a structure instance under the associated structure type and the structure tags in Tag Management. The created structure tags are defined by the structure elements of this type.

In the same structue type, you can define structure elements for internal tags and process tags. When a structure instance is created under the connection of a communication driver, the process tags defined in the structure type are created under the connection. The internal tags, however, are created in Tag Management at "Internal tags".

Modifying structure types

If you have created a structure instance with a structure type, you can no longer modify the properties of this structure type. You must first delete the structure instance. In doing so, you also delete all associated structure tags. After that you can change the properties of the structure type and the structure elemets and regenerate a structue instance.

You can modify the properties of structure instances and structure tags later. If you rename the structure instance, then rename the associated tags at the same time.

Usage

If a certain tag configuration occurs several times in your system, use structure types. In this instance, for example, several temperature controllers can be involved that are always set up according to the same principle.

Consider which tags you require for the temperature controller.

Then create a TEMPERATURE structure type in which each tag is represented by a structure element, for example ELEMENT_1, ELEMENT_2, ELEMENT_3. Each time you create a new structure instance with this structure type, WinCC automatically generates all structure tags for the corresponding temperature regulator.

You use the created structure tags when, for example, you configure pictures which you would like to integrate by means of picture windows.

Structure types in picture windows

You configure a picture, TEMP_REGLER.PDL, for the temperature regulator. You do not connect the objects in the picture with tags but with structure elements.

To do this, you have to enter the name manually, since you cannot directly enter elements by means of a dialog box. Alternatively, you can connect structure tags you have created and delete the first part of the name, which is determined by the structure instance. You link e.g. the structure tags of the structure REGLER_A. In the tag name displayed, e.g. REGLER_A.ELEMENT_1, delete the REGLER_A and the linking point in the input field.

Configure a picture window and integrate the picture TEMP_REGLER.PDL. Open the object properties of the picture window. Accept one of the created structure instances with a dot behind the name, such as "CONTROLLER_A.", as the tag prefix.

Configure a second picture window with the picture TEMP_REGLER.PDL and the next structure instance, e.g. REGLER_B, as tag prefix, e.g. "REGLER_B.".

In Runtime, WinCC composes the names of the required structure tags from the following elements:

- Tag prefixes of the picture window
- Names of the structure elements linked in the pictureTEMP_REGLER.PDL

Then the first picture window displays the state of regulator A and the second picture window the state of regulator B.

3.2.1.4 Tag groups

Definition

You can sort your tags into tag groups in Tag Management.

You create a tag group as a directory at "Internal tags" or at the connection of a channel unit. Subgroups are not possible.

Usage

When creating a large number of tags in your project, you can combine them in groups according to subject.

You can create, for example, a tag group for every picture in your project. You create the tags you use in one of your pictures in the appropriate group.

In this way WinCC makes it easier for you to assign and retrieve tags.

Name conventions

The names of tag groups must be unique in the whole project. You must use a name simultaneously for tags and tag groups. When tags and tag groups are created, WinCC does not distinguish between upper and lower case characters in a name.

You must not use certain characters in the names of tag groups. You will find the characters you must not include in a name in the WinCC Information System at "Working with projects" > "Non-authorized characters"

Note:
Remember that WinCC distinguishes between upper and lower case characters in a configuration. If you change upper and lower case characters in a tag name, you must adapt your configuration accordingly.

3.2.2 Tag selection dialog box

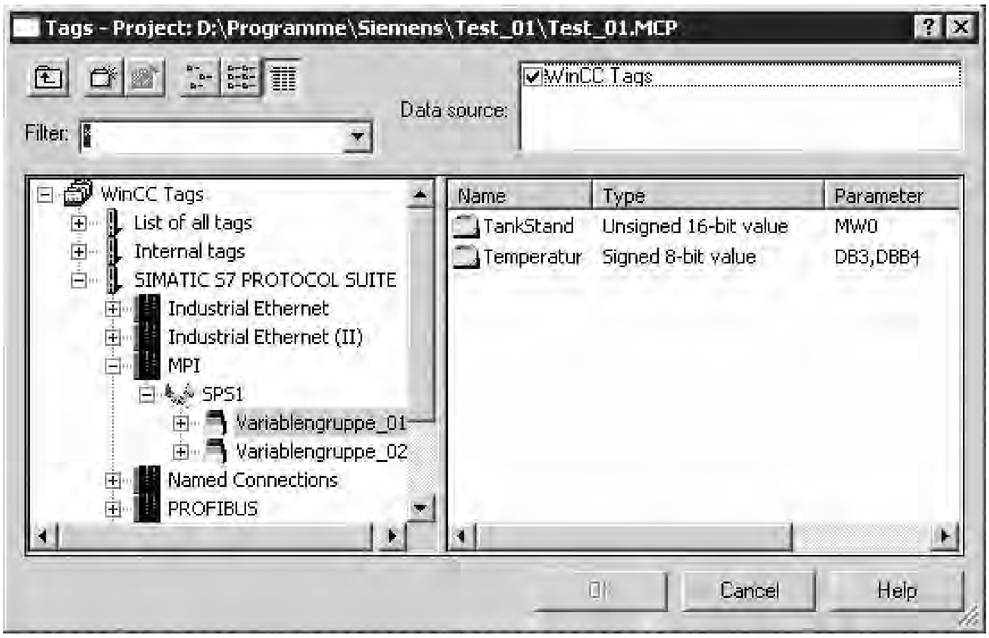
Working with the tag selection dialog box

WinCC opens the tag selection dialog box as soon as you want to attach a tag in an editor.

In Graphics Designer you can view the tag selection dialog box continuously. Call the option "View" > "Toolbars..." on the menu bar. Select the check box next to the "Tags" entry.

In the Global Script editor, open the tag selection dialog box by choosing "Edit" > "Tag Browser".

In the tag selection dialog box, you can view either all tags in the project or navigate in the directory structure. You select a highlighted tag by clicking "OK".



Data source: ☑ WinCC Tags

Name	Type	Parameter
TankStand	Unsigned 16-bit value	MW0
Temperatur	Signed 8-bit value	DB3,DBB4

Drag and drop

You can also connect a tag to an object or object attribute by means of drag and drop. With drag and drop, there are two places above which you can store the tag:

- Above an object in the picture
- Above an attribute in the "Object Properties" dialog box, Properties tab

Store the tag above an attribute in the "Object Properties" dialog box, Properties tab. This ensures that this attribute is dynamized by the tag.

Filter

Using the filter function, you can reduce the selected number of tags or symbols so as to search more efficiently.

When searching, you can use the placeholders "*" and "?". You can only use alphanumeric characters when searching for names. WinCC saves the last 20 search criteria you entered.

Example

The following internal tags have been created in WinCC, for example: "var1"; "var2"; "var3"; "smv2" and "apm1". Enter 'a*' as the search criterion and exit the filter function with 'TAB'. WinCC then shows all the tags whose name starts with "a". In this case, the tag 'apm1' in the 'List of all Tags' folder.

Note

The search operation in the tag selection dialog box can take some time when there is a large number of tags. Searching lasts about one minute with 15,000 tags.

It is sensible to arrive at a preselection in the tag selection dialog box by means of the filter.

Data source

This function is available when you are working with WinCC integrated in STEP 7.

In this box you can select the data source from which you would like to attach a tag.

'WinCC Tags' as data source

WinCC display all configured internal tags and process tags. You can create new tags by means of the "Create New Tag"function. You can edit existing WinCC tags with the "Edit Tag" function.

Note

In certain situations the tag selection dialog box sets a filter to the data type of the tag.

'ES Tags' as data source

WinCC displays all PCS 7 process tags. PCS7 process tags are all operable and visible CFC and SFC modules which were configured in the PCS 7 Engineering System. "CFC" stands for "Continuous Function Chart". "SFC" stands for "Sequential Function Chart". You can update the attributes of the display tag by operating the "Update ES Tags" button. This is necessary only when an attribute is modified in the PCS 7 Engineering System while the tag selection dialog box is open. Attributes are items such as name, type or comment relating to a process tag.

Further information on this is available in the Chapter "Integration in the SIMATIC Manager" and in the "Process Control System PCS7 Operator Station" configuration manual.

"Update ES Tags" button

The button is available to you after selecting the "ES tags" data source.

You can use this button to update the CFC/SFC connections. CFC/SFC stands for "Continuous Function Chart/Sequential Function Chart". When you extend STEP 7 connections, you can update the corresponding engineering station with this function.

'STEP7 Symbols' as data source

WinCC displays all the inputs, outputs, and bit memories in the STEP 7 symbol list and all the global data blocks. The symbol lists and the data blocks are created in SIMATIC Manager. In this case, data blocks themselves cannot be mapped. Only the parameters of the prevailing data block displayed in the data window can be mapped.

"Transfer Data" button

The button is available to you after selecting the " STEP 7 Symbols" data source.

The "Transfer Data" function is used by WinCC to apply the selected symbols or parameters of data blocks to the WinCC database.

3.3 Tag types

Introduction

When you create a tag, you assign one of the possible data types to the tag.

This data type depends on the type of data for which you would like to use the tag.

Note:
If you modify the data type of an existing process tag, the previously defined tag address is deleted. This reason for this is that the PLC address changes when the data type is modified.

Adapt format

The data type of a tag in WinCC can differ from the data type used in the automation system. You can then set that WinCC converts the data type of a tag for transfer to the automation system.

Tag types with type conversion	Tag types without type conversion
Signed 8-bit value	Text reference
Unsigned 8-bit value	Binary tag
Signed 16-bit value	Text tag 8-bit character set
Unsigned 16-bit value	Text tag 16-bit character set
Signed 32-bit value	Raw data type
Unsigned 32-bit value	Structure types
Floating-point number 32-bit IEEE 754	
Floating-point number 64-bit IEEE 754	

You will find detailed information about type conversion in the WinCC Information System at "Communication" > "Process Communication" > "WinCC Process Communication" > "External Tags".

3.4 Editing tags

3.4.1 Copying, moving and deleting tags

Introduction

The shortcut menu and the toolbar can be used to perform the following operations on tags, tag groups and structure types:

- Cut
- Copy
- Paste
- Delete

There is no difference in the ways in which tags, tag groups and structure types behave.

You can select and edit several objects at a time in each case.

Copying, tags, tag groups and structure types

If, for example, you wish to configure several tags having the same properties, you can copy an existing tag and paste it at the desired position. WinCC automatically increments the name or adds a count to the name.

When you copy a tag group, WinCC automatically copies every tag contained in it.

Structure types can be copied separately. Structure tags and structure instances cannot be copied separately but only while the associated structure type is being copied.

Note:
If a tag was created in WinCC using SIMATIC Manager with "Translate OS", you cannot copy or delete the tag in WinCC Explorer.

Moving tags and tag groups

If you want to move tags or tag groups, cut the object and paste it at the position you want.

You can move tags

- from one tag group to another tag group
- from the connection of a communication driver to under the connection of another communication driver
- from "Internal tags" Tag Management to under a communication driver

- from the connection of a communication driver to the "Internal tags" Tag Management

Structure types, structure tags and structure instances cannot be moved.

Note:
If you want to move a tag between the "Internal tags" Tag Management and a Variable or between communication drivers, you must adjust the properties of the tag after doing so.

3.4.2 Editing tags in Runtime

Introduction

If you have activated a project, you can edit tags in parallel. There are a few constraints, however.

Editing in Runtime

You can edit the following elements in Runtime:

	Tags	**Tag groups**	**Structure types**	**Structure instances**
Create	X	X	X	X
Address	X	---	---	X
Cut	X [2]	---	---	---
Copy	X [2]	X	X	X
Paste	X	X	X	X
Delete	X [1] [2]	X [1]	---	X
Modify properties	X	X	X [3]	X
Rename	X [1] [2]	X	---	X

[1] Possible with external tags only if the channel being used supports the function. This currently applies only to the "SIMATIC S7 Protocol Suite" channel.

[2] Does not apply to structure tags.

[3] Not possible if structrue tags are created.

Note:
If a tag was created in WinCC using SIMATIC Manager with "Translate OS", you cannot edit the tag in WinCC Explorer.

3.4.3 Status information of tags in Runtime

Introduction

In Runtime you can view information on connections and tags as a tooltip.

Displaying status information

In Tag Management of the activated project, change to the subdirectory you require.

If you point the mouse pointer to a connection or a tag in the data window, WinCC outputs the status information as a tooltip.

Connections

The tooltip of a connection specifies:

- The status of a connection

Tags

The tooltip of a tag specifies:

- The status of a connection
- The current process value of a tag
- The quality code of a tag
- The date on which a tag was last changed

This status display similarly applies to structure tags.

Quality codes

Quality codes provided encoded information about the status and quality of a tag.

You will find a list of quality codes in the WinCC Information System at "Communication" > "Communication diagnostics" > "Quality codes of tags".

4 Creating Process Screens

Graphics System Tasks

During configuration, the Graphics System is used to create the pictures which display the process in runtime.

The Graphics System handles the following tasks:

- It displays static and operator-controllable picture elements, such as texts, graphics or buttons
- It updates dynamic picture elements, e.g. modifies the length of a bar graph in relation to a process value
- It reacts to operator input, e.g. the clicking of a button, or the entry of a text in an input field

The components of the Graphics System

The Graphics System is made up of Configuration and Runtime components:

- The Graphics designer is the configuration component of the Graphics System. The Graphics Designer is the editor that is used to create pictures.
- Graphics Runtime is the runtime component of the Graphics System. Graphics Runtime displays the pictures on the screen in runtime and administers all inputs and outputs.

4.1 The Graphics Designer in the WinCC Explorer

Introduction

The Graphics Designer is an editor for creating process pictures and making them dynamic. The Graphics Designer can only be started for the project currently opened in the WinCC Explorer. The WinCC Explorer can be used to display an overview of the pictures available in the current project.

For working with the Graphics Designer, the WinCC Explorer offers the following functions and configuration options:

- Starting Graphics Designer
- Creating and renaming pictures
- Configuring object libraries and ActiveX controls
- Converting libraries and pictures from older program versions
- Configuring and starting runtime

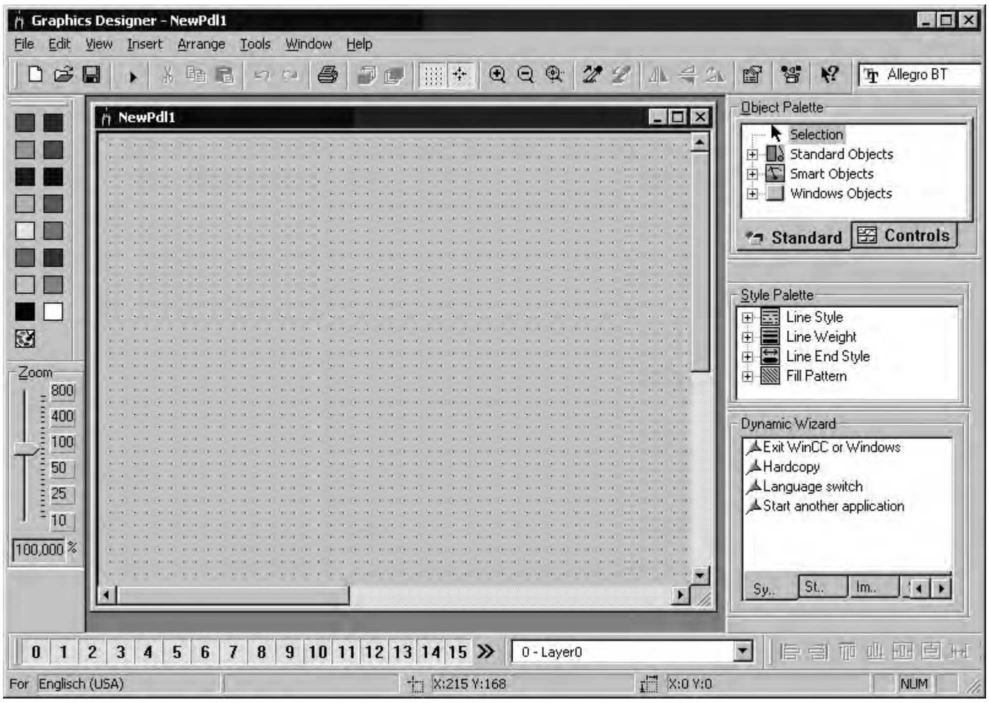
4.1.1 The start screen of the Graphics Designer

Introduction

The Graphics Designer provides objects and tools for creating process pictures. Start the Graphics Designer from the navigation window of the WinCC Explorer.

Layout of the Graphics Designer screen

The Graphics Designer is designed according to the Windows standard. The Graphics Designer has a desktop, toolbar, menu bar, status bar and various palettes. When you open the Graphics Designer, the screen with the default settings appears. As standard, a blank picture is displayed (file format PDL). You can move the palettes and bars on the screen and arrange them as you like.



The toolbars

At "View", "Toolbars...", the following toolbars can be shown or hidden. In the default setting, all toolbars, except for the tags, are shown.

- Objects: for inserting Standard Objects, Smart Objects, Windows Objects and Controls.
- Styles: for changing the line type and thickness, and fill patterns.
- Default: for creating and saving documents, activating the runtime etc.
- Alignment: for aligning a number of objects.
- Colors: for fast changes to the color of an object.
- Font: for changing the font and font size, font color and border color.
- Zoom: for stepless zooming in and out.
- Status: for displaying the language setting and coordinates
- Layers: for showing and hiding individual layers.
- Dynamic Wizard: for making objects dynamic with pre-prepared C Actions.
- Tags: for fast assignment of tags to objects that can be made dynamic.

Note:
To move the visible area in the case of large pictures, you can use the scroll bars in accordance with the Windows standard.
The Graphics Designer provides additional operation options with the mouse wheel: click on the mouse wheel. The mouse pointer changes and you can move the visible picture area simply by moving the mouse.

4.1.2 Customizing the working environment

Introduction

The Graphics Designer provides a great many possibilities to customize the working environment. For example, you can change the toolbar and palettes. You can define your own colors and change the basic settings of the program.

Please bear in mind that the default setting of the Graphics Designer is such that changes to the working environment are saved when you exit from the program. If you do not want to save the settings, open the "Extras" menu, then "Settings..."; select the "Options" tab and disable the "Save settings on exit" check box.

4.2 Working with Pictures

Introduction

In Graphics Designer, a picture is a file in the form of a sheet of drawing paper. The size of the drawing sheet can be adjusted. A drawing sheet has 32 layers, which can be used to improve the organization of the drawing. The files are saved in the project directory, GraCS, in the PDL format. The complete process can be distributed over several individual pictures, which are linked together. Calls to other applications and files can also be included in a process picture.

The more complex the configured process is, the more detailed the planning should be.

Project Planning Notes

- Plan the structure of the representation of the process: determine how many pictures will be needed and their hierarchical order. Example: subprocesses can be shown in individual pictures, which are then merged in a main picture.
- Plan the navigation within a picture and between the individual pictures: the operation sequences, buttons and hotkeys should be consistent in all pictures.
- Create a Master picture in which the picture properties, default settings and object settings are defined. This Master picture should then be used as the template for each individual picture.
- Take advantage of the program functions: Process pictures can be efficiently created by using libraries, copying and transferring properties and by working with layers and multiple pictures.
- To avoid changes in the picture layout, make certain that each picture is displayed - during the design phase - at the same size that it later will be displayed on the target computer.
- To avoid changes in the picture layout, make certain that you only use fonts that are also available on the target computer.

4.2.1 Working with Layers

Introduction

In Graphics Designer, a picture consists of 32 layers in which objects can be placed. The position of an object in a picture is set when it is assigned to a layer. Objects on Layer 0 are located on the lowest layer (background) of the picture; objects on Layer 32 are on the top layer (in the foreground). Objects are always added on the active layer, but can be quickly moved to another layer. An object's assignment to a layer can be changed using the "Layer" attribute in the "Object Properties" window.

In addition, it is also possible to change the positioning of the objects with respect to each other within a layer. Four functions are available for this purpose in the "Arrange / Within the Layer" menu. When creating a process picture, the objects within a layer will be arranged - by default - in the order that they are added: The first object added will placed in the background of the layer, while each additional object is placed one position closer to the front.

Layer Techniques - Principles

When a picture is opened, all 32 layers of the picture will be displayed. This setting cannot be changed. The layer palette can be used to hide all but the active layer. In this manner, it is possible to concentrate on editing the objects of the active layer. Layers are particularly useful when preparing pictures that contain many different types of object types.

For example, you could place all of the "Bar" objects on Layer 1 and all of the "I/O Field" objects on Layer 2. If you later decide to change the font color of all of the I/O fields, you can now display just Layer 2 and then select all of the objects on this layer. The time-consuming selection of individual I/O fields scattered throughout the picture is thus unnecessary.

4.2.2 Working with Multiple Pictures

Introduction

Multiple process pictures are quite useful when working with complex processes. These process pictures are then linked to each other and one picture can be integrated in another. Graphics Designer supports a number of features which ease the process of working with multiple pictures.

- The properties of a picture can transferred to another picture.
- Objects can be transferred from one picture to another.
- Objects can be copied from one picture to another.

4.3 Working with objects

Introduction

This chapter explains

- which basic functions the Graphics Designer offers for working with objects
- which characteristics the individual objects possess
- how to use objects to create process pictures
- how to change object properties as specified

Objects of the Object Palette

In the Graphics Designer the predefined graphic elements which enable efficient creation of process pictures are called "objects" in the Graphics Designer. All objects can be easily inserted into a picture from the object palette. The "Default" register tab of the Object Palette provides the objects in the following object groups:

Standard Objects	Smart Objects	Windows Objects
Line Polygon Polyline Ellipse Circle Ellipse Segment Pie Segment Ellipse Arc Circular Arc Rectangle Rounded Rectangle Static Text Connector	Application Window Picture Window Control OLE Element Both Bar Graphic Object Status Display Text List 3D-Bar Group Display	Button Check Box Option Group Round Button Slider

Combined Objects

The objects of the object palette can be combined by creating a "group" or a "user object" from a multiple selection. Additionally a multiple selection or a combined object can be added to the project library and can be made available as a "library object" for reuse in other process pictures or projects.

4.3.1 The properties of an object

Introduction

Form, appearance, position and process connection of an object are determined by the "Object properties". These properties can be changed as required in the Graphics Designer.

The properties of an object are described by the multitude of "Attributes". An object property can be changed by a new value is assigned to the associated attributes.

The "Object Properties" window on the "Properties" tab contains all the attributes of a selected object or a multiple selection of objects. The attributes are subdivided into property groups such as for example "Geometry" or "Colors". The type and number os property groups depends on the type of object selected. Therefore for example the "Font" property group is only displayed for those object types with which a text can be displayed.

With a multiple selection it is possible that individual property groups for one specific object type are summarized to a shared property group. All attributes of the summarized property groups available in this group are continued under the "Customized" label.

Alternatively to changing the attributes in the "Object Properties" window, objects can also b adapted by operation with the mouse and keyboard or by using toolbars and palettes. However this was it is only possible to change certain object properties such as for example basic geometric sizes, colors and line styles.

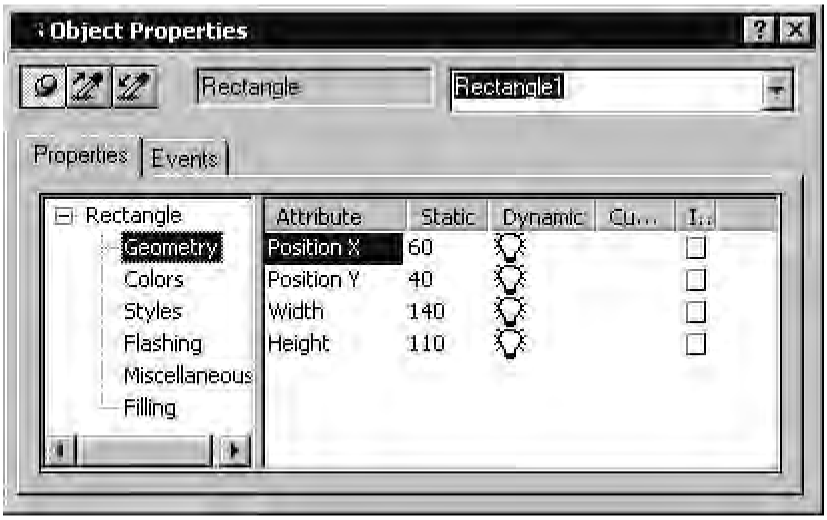
4.3.1.1 The "Object Properties" window

Introduction

In the Graphics Designer the "Object Properties" window represents the central dialog box for changing object properties.

To create process pictures it is firstly important to adapt the static values of attributes to for example define the shape, appearance, position or operability of an object. The "Object Properties" window allows however process pictures to be made dynamic. By linking the attributes with dynamic dialog boxes, C actions or tags, the properties of an object can be dynamically adapted to the requirements of the processes to be displayed. An extensive description of dynamic procedures can be found in the chapter Making "Process Pictures Dynamic".

- The "Object Properties" window can for example have the following appearance for a rectangle with the object name "Rectangle1":



Object Properties

Rectangle

Rectangle1

Properties | Events

- Rectangle
  - Geometry
  - Colors
  - Styles
  - Flashing
  - Miscellaneous
  - Filling

Attribute	Static	Dynamic	Cu...	I...
Position X	60	💡		□
Position Y	40	💡		□
Width	140	💡		□
Height	110	💡		□

The "Object Properties" window can be pinned on top of the Graphics Designer. The position and size of the window can be changed freely.
The "Object Properties" window is subdivided into Elements toolbar, and the "Properties" and "Event" tabs. More information about these elements is provided in their extensive description.

4.3.1.1.1 The "Properties" tab in the "Object Properties" window

Introduction

In the "Object Properties" window, the "Properties" tab represents the central dialog box to change static values of a selected object.

- The "Properties" tab can for example have the following appearance for a rectangle with the object name "Rectangle1":

The width of both window areas and the columns in the attribute display can be changed by moving the vertical separation lines.

Properties folder

The left area shows the selected object with its property groups in the form of a folder tree. The selected object is displayed as a folder.
With a multiple selection or a user object, this folder only receives the shared property groups of the contained individual objects. With a selected group the shared property groups are listed first and then, as a subfolder, the individual objects with their property groups.

It is not possible to change the entries in the property folder. Folders and subfolders can be open or closed by double-clicking them or by clicking the "+" or "-" icons. The attributes available for the selected entry are shown in the attribute display.

Attribute display

The right-hand area contains all the attributes that are available for the entry selected from the property folder. The attribute display is subdivided into 5 columns in which the static values and the dynamics of the selected objects are displayed. The displayed values can be changed by double-clicking or calling the popup menu in the respective column.

Columns of the attribute display

Column	Function	Description
Attribute	Attribute name	All attributes of the selected property group are displayed which are available for the selected object. The name of the attribute cannot be changed. The static value of the attribute can be changed by double-clicking the name of the attribute.
Static	Static value of the attribute	The current value of the attribute is shown for the selected object. Depending on the type of attribute, the value is displayed as a number, text or a graphic representation. The static value of the attribute can be changed by double-clicking the value or name of the attribute. Further information can be found in chapter "Changing Object Properties".
Dynamic	Making the attribute dynamic	The type of dynamics is displayed and marked by one of the following icons: White bulb = no dynamics Green bulb = dynamics with a tag Red lightning = dynamics via a dynamic dialog box Green lightning = dynamics with a C action Yellow lightning = dynamics with a C action not yet translated The dynamics of the attribute can be changed. Right-clicking opens a popup menu on which the required dynamics can be set. Further information can be found in chapter "Making Process Picture Dynamic".

Column	Function	Description
Current	Update cycle of the attribute	The setting for the update cycle is displayed if dynamics of the attribute are set. The update cycle of the attribute can be changed by double-clicking the value. Further information can be found in chapter "Changing Object Properties".
Indir.	Indirect attribute addressing	An attribute can be directly or indirectly made dynamic: Directly = The attribute is directly made dynamic with the contents of the tags. Indirectly = The attribute is linked to a tag of the type "String" which links to another tag. The attribute is made dynamic with the contents of the referenced tag. Indirect addressing of the attribute can be activated by double-clicking the checkbox if the attribute is made dynamic with a tag. Further information can be found in chapter "Making Process Picture Dynamic".

Font styles to display dynamics and events

Dynamics and events are particularly highlighted with the aid of different font styles. The following font styles are used:

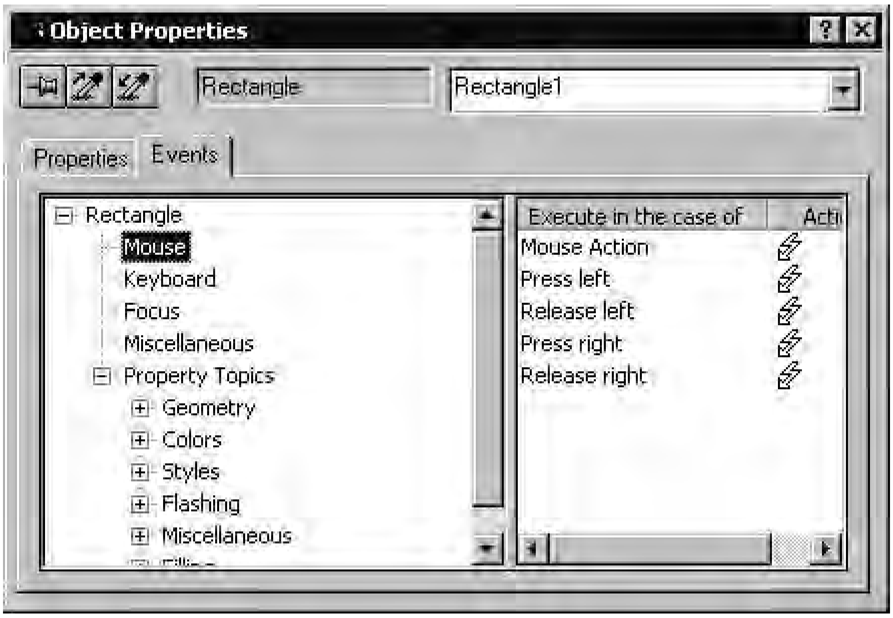
- bold As soon as dynamics or an event are configured to an attribute of the selected object, the attribute is displayed in bold in the attribute display. The associated property group in the property folder and the selected object in object selection of the toolbar are also displayed in bold.
- italics If a direct connection was configured for an event, the target of the direct connection is displayed in italics in the attribute display. The target object is also displayed in italics in the object selection of the toolbar.
- bold and italics If a selected object is both target of a direct connection and is dynamic itself, the attribute in the attribute display and the object in the object selection of the toolbar are displayed in bold and italics.

4.3.1.1.2 The "Events" tab in the "Object Properties" window

Introduction

In the "Object Property" window the "Event" tab represent the central dialog box for configuring events. Configuring events is described extensively in the chapter "Making Process Pictures Dynamic".

- The "Event" tab in the "Object Property" window can for example have the following appearance for a rectangle with the object name "Rectangle1":



The width of both panes and columns in the event display can be changed by moving the vertical separation lines.

Event folder

The left area shows the selected object in the form of a folder tree. This folder shows entries such as for example "Mouse", "Keyboard", "Focus" and "Misc" with which events for the operation of the entire object can be configured. The "Property Theme" displays all property groups of the selected object. An event can also be configured for each individual attribute of a property group. If there are several individual objects in the selected object, the structure is repeated for each individual object.
It is not possible to change the entries in the event folder. Folders and subfolders can be open or closed by double-clicking them or by clicking the "+" or "-" icons. The configurable events for the selected entry are shown in the event display.

Event display

The right-hand area contains all the events that are available for the entry selected from the event folder. The event display is subdivided into 2 columns in which the events and the linked actions are displayed.
The displayed actions can be changed by double-clicking or calling the popup menu in the "Action" column.

Columns of the event display

Column	Function	Description
Run at	Event type	All events are displayed which are available for the selected object and assigned to the entry selected in the event directory. The name of the event cannot be changed. The configured action can be changed by double-clicking the name of the attribute.
Action	Action selection	The action is displayed which is run when event occurs. The selected action is marked with one of the following icons: White lightning = There is no action for the event. Blue lightning = There is an action via a direct connection for the event. Green lightning = There is a C action for the event. Yellow lightning = There is a C action not yet translated for the event. The configured action can be changed by double-clicking in the "Run "at column or by calling the popup menu in the "Action" column. Further information can be found in chapter "Making Process Picture Dynamic".

Triggering events

Events	Run at	Description
Mouse	Mouseclick	Is triggered if the cursor is on the object when the mouse button is pressed an released.
Mouse	Left / right clicking	Is triggered when pressing the mouse button directly on the object.
Mouse	Left/right releasing	Is triggered when releasing the mouse button for the object the cursor was over when the mouse button was pressed.
Keyboard	Pressing	Is triggered when pressing a key on the keyboard. The keys <F10> and <ALT+PRINT> may not be used for process operation.

Events	Run at	Description
Keyboard	Releasing	Is triggered when releasing a key on the keyboard. The keys <F10> and <ALT+PRINT> may not be used for process operation.
Focus	Focus change	Occurs with focus reception or loss, called by a C or VBS action or by selecting the object with <TAB> (TAB order).
Miscellaneous	Object change	Occurs if at least one object attribute changes.
Object attributes	Change	Most object attributes have the Change event. It is used to be able to react specifically to the change of a certain attribute. The event occurs if the attribute value changes. The action linked to this event is logged on individually. With Close Picture, all actions logged on at this point are logged off individually. This leads to an increased system load. To keep the system load low, this event type should be used sparingly and only there where it is absolutely necessary to react to the change. This is the case for example when inputting a valie in the I/O field.

4.3.1.2 Property groups and attributes

Introduction

On the "Properties" tab of the "Object Properties" window all property groups are displayed which are available for the selected object. If one of these properties is selected, the associated attributes for the selected object are listed in the attribute display. The properties of an object can be changed by assigning a new value to the associated attributes.

This chapter introduces all property groups and the associated attributes. The function of an attribute is described briefly and is assigned to the object types for which the attribute is available.

An extensive description of the individual attributes can be found in the "Direct help" which can be called from the popup menu of the required attribute.

4.3.2 Working with standard objects

Introduction

The standard objects include geometric shapes and static text. The geometric shapes draw the individual elements of the process picture. For example, static text can be used for labels.

In the Graphics Designer, the various object types have default properties. When they are inserted the objects import these default properties (except for individual geometric properties). After insertion the properties of an object can be modified. In the same way the default settings for the object types can be modified as required.

Overview

Icon	Object	Description
╱	Line	The line is an open object. The length and angle of a line are determined by the height and width of the rectangle around the object. The line ends can for example be shown as arrows or points.
⌂	Polygon	The polygon is a closed object that can be filled with a color or pattern. A polygon can have any number of corners. The corners are numbered in the sequence of their creation and can be modified individually or deleted.
⌂	Polyline	The polyline is an open object. Even if the start and finish point have the same coordinates, that area cannot be filled. A polyline can have any number of corners. The corners are numbered in the sequence of their creation and can be modified individually or deleted. The line ends of a polyline can for example be shown as arrows or points.
⬬	Ellipse	The ellipse is a closed object that can be filled with a color or pattern. The height and width of an ellipse can be modified as desired to allow it to be aligned horizontally or vertically.
●	Circle	A circle is a closed object that can be filled with a color or pattern. A circle can be resized at will.
◠	Ellipse Segment	The ellipse segment is a closed object that can be filled with a color or pattern. The height and width of an ellipse segment can be modified as desired to allow it to be aligned horizontally or vertically.
◔	Pie Segment	The pie segment is a closed object that can be filled with a color or pattern. A pie segment can be resized at will.

Icon	Object	Description
◝	Ellipse Arc	The ellipse arc is an open object. The height and width of an ellipse arc can be modified as desired to allow it to be aligned horizontally or vertically.
◝	Circular Arc	The circular arc is an open object. A circular arc can be resized at will.
■	Rectangle	The rectangle is a closed object that can be filled with a color or pattern. The height and width of a rectangle can be modified as desired to allow it to be aligned horizontally or vertically.
▢	Rounded Rectangle	The rounded rectangle is a closed object that can be filled with a color or pattern. The height and width of a rounded rectangle can be modified as desired to allow it to be aligned horizontally or vertically. The corners of a rounded rectangle can rounded as much as desired.
A	Static Text	The field for static text is a closed object that can be filled with a color or pattern. The static text is entered into a field of any desired size. One-line or multiline text can be input for all designed languages.
⌐	Connector	The connector is a linear object whose ends can be connected with the connection points of other objects. The number and arrangement of the connection points depend on the specific object type. If connected objects are moved, the connector's length and direction is automatically adapted and the connection is retained.

4.3.3 Working with smart objects

Introduction

Smart objects offer the option of building complex system pictures. The smart objects include items such as various windows, fields and bars, which offer a variety of dynamic options.

In the Graphics Designer, the various object types have default properties. When they are inserted the objects import these default properties (except for individual geometric properties). After insertion the properties of an object can be modified. In the same way the default settings for the object types can be modified as required.

Overview

Icon	Object	Description
🗔	Application Window	The application window is an object that can be supplied from applications of the global script and the logging system. In runtime, these applications open an application window, transfer information and enable operation. The size and properties which an application window accepts in Runtime are defined in the Graphics Designer.
🗔	Picture Window	The picture window offers the option of displaying other pictures that were created with the Graphics Designer in the current picture. For example, the contents of a picture window can be continuously updated with dynamics. The size and properties that a picture window has in runtime are defined in the Graphics Designer.
⚒	Control	The Object Control provides the option to integrate system process control and monitoring elements into a picture. Controls are prefinished objects such as for example alarm windows, measuring windows, selection dialog boxes or buttons. ActiveX Controls, WinCC Controls and controls from other manufacturers are available if they are registered in the operating system. These are changed as required and dynamically integrated into the process. The size and properties which a control accepts in Runtime are defined in the Graphics Designer.
▦	OLE Object	The OLE element enables files created in other programs to be inserted into a picture. Therefore all OLE elements registered in the operating system can be integrated. The size and properties which an OLE element accepts in Runtime are defined in the Graphics Designer. No changes can be made to OLE elements in Runtime.
▣	I/O Field	The I/O Field can be defined as an input field, an output field or a combined input/output field. The following data formats are available: binary, decimal, string and hexadecimal. Limit values such as "Hidden Input" or "Accept on complete input" can also be specified. The size and properties which an I/O Field accepts in Runtime are defined in the Graphics Designer.

Icon	Object	Description
▣	Bar	The bar offers the option of displaying values graphically. The values can also be displayed in a combined view as graphics with freely definable number scale. The size and properties which a bar accepts in Runtime are defined in the Graphics Designer.
🖼	Graphic Object	The graphic object enables graphics created in other programs to be inserted into a picture. Graphics and pictures can be inserted in the following formats: EMF, WMF, DIB and BMP. The size and properties of a graphic object in runtime are defined in the Graphics Designer.
⧓	Status Display	The status display offers the option to display almost any number of different states of an object. The states are implemented via tags whose value corresponds to the respective state. The states are displayed via the assigned pictures. The size and properties which a status display accepts in Runtime are defined in the Graphics Designer.
☰	Text List	The text list offers the option of assigning specific values to a text. The text list can be defined as an input list, an output list or as a combined input/output list. The following data formats are available: decimal, binary or bit. The size and properties which a text list has in Runtime are defined in the Graphics Designer.
■	3D Bar	The 3D bar graph enables values to be displayed graphically three-dimensionally. The size and properties which a 3D bar has in Runtime are defined in the Graphics Designer. The 3D bar is only available if the "Basic Process Control" optional package was installed with WinCC.
▯▯▯▯	Group Display	The group display enables the current states of certain message types to be displayed hierarchically. Four buttons allow messages to be represented and operated. Using a group display, it is possible for example to implement a quick change to represent an error source. The size and properties which a group display accepts in Runtime are defined in the Graphics Designer. The group display is only available if the "Basic Process Control" optional package was installed with WinCC.

4.3.4 Working with Windows objects

Introduction

The Windows objects are elements familiar from Windows applications such as buttons, check boxes, option groups and sliders. A round button can also be designed. The objects can be modified and made dynamic in many ways. They enable you to operate process events and to control processes.

In the Graphics Designer, the various object types have default properties. When they are inserted the objects import these default properties (except for individual geometric properties). After insertion the properties of an object can be modified. In the same way the default settings for the object types can be modified as required.

Overview

Icon	Object	Description
□	Button	The button enables process operation. It is used for example to acknowledge messages. A button can have two different states: "On" and "Off". A button is integrated into the process by providing the corresponding attributes with dynamics.
☑	Checkbox	The checkbox enables the operator to select several options and to check the corresponding checkboxes. It is also possible to check checkboxes by default so that the operator only changes the predefined value as required. A checkbox is integrated into the process by providing the corresponding attributes with dynamics.
◉	Option box	The option box enables the operator to selected one of the presented options and to activate the corresponding options field. It is also possible to activate one of the options fields by default so that the operator only changes the predefined value as required. An option box is integrated into the process by providing the corresponding attributes with dynamics.
○	Round Button	The round button enables process operation. It is used for example to acknowledge messages or to navigate in Runtime. A round button can have three different states: "On", "Off" and "Deactivated". It is also possible to define whether a round button should by default be displayed as pressed and whether it should click when pressed. A round button is embedded into the process by making the corresponding attributes dynamic.

Icon	Object	Description
⬍	Slider	The slider is a slide controller and enables processes to be controlled. For example, it can be used for infinitely variable changes to a value. A slider is embedded into the process by making the corresponding attributes dynamic.

4.3.5 Quick object configuration

Introduction

The Graphics Designer has configuration dialogs for quick configuration of individual objects. the important features of an object can be set in the configuration dialogs.

If the use of configuration dialogs is activated in the "Options" tab of the "Tools / Settings..." menu, the associated dialog is opened when an object is inserted. The configuration dialog can also be opened with the object's pop-up menu.

Configuration dialogs are available for the following objects: I/O field, bar, graphic object, status display, text list, button and slider. The contents of the configuration dialogs are different for different objects.

Short description of the configurable features

Features	Description	Object Type
Update	Specifies the frequency with which the display of the displayed value is updated. Enter the desired interval or select an interval from the dropdown list.	I/O field, bar, status display, text list, slider
Alignment	Specifies whether the slider is moved horizontally or vertically.	Slider
Bar Alignment	Specifies the direction of the coordinate axis to which the maximum value of bar points.	Bar
Operation	An authorization can be assigned for operation of the button. A hot key can also be defined, which allows operation with a key or key combination. You can find the detailed description in the "Button" chapter.	Button
Bit Position	A status can be assigned to a specific bit position. You can find the detailed description in "Status Display".	Status Display
Picture Selection with Preview	The picture selection shows all pictures in the graphics directory "GraCS" of the current WinCC project. The picture selected in the picture selection is displayed as a preview. Select a picture or click on the "Browse ..." button to insert additional pictures from the picture selection. See the "Select Pictures" section for the detailed description.	Graphic object, Status Display

Features	Description	Object Type
Change Picture on Mouse Click	An event can be assigned for the selected button with this function. If the button is clicked with the mouse, the display of the process picture specified here is opened. See the "Select Pictures" section for the detailed description.	Button
Field Type	Specify whether the selected object is used for input, output or input and output of values.	I/O field, Text List
Limits	Enter values as "Max. value" and "Min. value" to define the two ends of the slider. You can also define the number of operation steps for the slider. The number of operation steps specifies the number of increments by which the slider is moved when the mouse is clicked on the slider.	Bar, Slider
Font	Change font options for viewing button labels and display values. Click one of the buttons to open the associated dialog.	I/O field, Text List, button
Text	Enter a text as a button label, which for example describes its function. Only one text line is available in the configuration dialog. Multiline text can be input by changing the "Text" attribute in the "Object Properties" window.	Button
Tag	The selected object can be set dynamically by embedding a tag. With an I/O field and text list the value of the tags can be displayed as output or changed by an input depending on the selected field type. See the "Select Tags" section for the detailed description.	I/O field, bar, status display, text list, slider
Status list	The status list shows the linked pictures and the specified flash frequency for all configured states. States can be inserted and every setting can be changed with a pop-up menu. Pictures of the picture selection can be referenced by dragging them with the mouse to the required position in the state list. You can find the detailed description in "Status Display".	Status Display

4.4 Working with combined objects

Introduction

The objects of the Object Palette can be combined by forming a group or a customized object from a multiple selection. In addition, a group or a combined object can be inserted into the project library. This makes the group or the combined object available as a library object for re-use in other process pictures or projects.

Overview

Object	Description
Group	A multiple selection of objects can be combined into a group. A group of objects can be edited like a single object in the Graphics Designer. It is also possible to edit objects in the group individually. Unlike multiple selection, the selection marks of the individual objects in a group are not shown. The selection marks include the entire group. The selection frame of the multiple selection becomes the rectangle around all the objects of the group.
Customized Object	A customized object allows the properties and events that are displayed and changed in the "Object Properties" window to be configured individually. A customized object can be edited in the Graphics Designer like a single object of the object palette. As in a group the selection marks of the individual objects are no longer shown after creation of a customized object. The selection marks surround the entire customized object. The selection frame of the multiple selection becomes the rectangle around all the objects of the customized object.
Library object	The library of the Graphics Designer is versatile tool for saving and managing graphic objects that can be used for creating process pictures. It is divided into two areas: The "Global Library" offers a variety of prefinished graphic objects that are inserted into a picture as library objects and can be configured as required. The "Project Library" enables a project-specific library to be built.

4.5 Working with customized objects

Introduction

A customized object allows the properties and events that are displayed and changed in the "Object Properties" window to be configured individually.

A customized object can be edited in the Graphics Designer like a single object of the object palette. As in a group the selection marks of the individual objects are no longer shown after creation of a customized object. The selection marks surround the entire customized object. The selection frame of the multiple selection becomes the rectangle around all the objects of the customized object.

Note the following features for working with customized objects:

- The "Undo" function is not available for editing a customized object.
- Configured events of the single objects are deleted when the customized object is created.
- If editing mode is exited without an object having been selected, the customized object is deleted. When editing mode is exited, only the selected object is imported into the customized object.
- Assigning hot keys and specifying a TAB sequence for objects within the customized object is not supported.
- If a group display is embedded in a customized object, the attributes "group value", "group relevant" and "bit pattern group display" are available for the customized object to capture and acknowledge states.

Note:
The text properties of the text object must be dynamic to allow changes from external sources so the language change can function.

4.6 Short Description of the WinCC Controls and Additional Controls

Introduction

ActiveX Controls serve for the monitoring and visualization of measured values and system parameters. By appropriate dynamics they can be employed as control elements of the process control.

The WinCC Controls and Symbol Library are installed during installation of WinCC.

Overview

Detailed descriptions on the following controls can be viewed via the links in the Menu Bar.

Control	Abbreviation	Function
Siemens HMI Symbol Library	Symbol Library	The Symbol Library contains a comprehensive collection of ready symbols for the display of systems and system components in process screens.
WinCC Alarm Control	Alarm Control	Alarm Control can be used to display messages during runtime.
WinCC Digital/Analog Clock Control	Clock Control	The Clock Control can be used to integrate a time display into a process picture.
WinCC Function Trend Control	Function Trend Control	The Trend Control function can be used display the values of tags as functions of other tags and compare the trend with the setpoint trend.
WinCC Gauge Control	Gauge Control	The Gauge Control can be used to display the monitored measurement values in the form of an analog measurement clock.
WinCC Online Table Control	Online Table Control	The online Table Control can be used to display values from archive tags tablular form.
WinCC Online Trend Control	Online Trend Control	The online Trend Control can be used to display values from tags and archive tags as trends.

Control	Abbreviation	Function
WinCC Push Button Control	Push Button	The Push Button can be used to configure a command button, which is connected to the execution of a specific command.
WinCC Slider Control	Slider Control	The Slider Control can be used to display the monitored measurement values in the form of a slider control.
WinCC User Archives Table Element	User Archives Table Element	The User Archives Table element provides options to access user archives and views of the user archives.
IXDiskSpace.Disk Space	Disk Space Control	Disk Space Control enables the monitoring of the capacities available on a storage medium.

Autodesk Volo View Control

The Volo View Control enables the display of the CAD files of the format DXF in a process screen. The DXF files can, for instance, have been created in EPLAN or AutoCAD.

4.7 How to setup a picture for mouseless operation

Introduction

Objects such as buttons or I/O fields must be operated in runtime to influence the process. To ensure convenient operation and that all of the necessary entries are made, you can configure operation via the keyboard. In this case, the process of entering the data is simplified, since the cursor will jump from one object to the next as the operator makes the required entries.

Operable objects

The objects must be enabled for operation and the operator must be granted the required permission. Set the Operator-Control Enable and the Authorization for each object. Click on the object and select the object properties on the "Properties" tabbed card. In the "Miscellaneous" properties group, you can set the Operator-Control Enable and the Authorization.

Mouseless operation

There are two types of mouseless operation.

- "Alpha cursor" In runtime the cursor jumps to only the objects for which values can be entered. These objects are the I/O Field and the TextList.
- "Tab order cursor" Depending on the setting, the cursor jumps to all of the objects for which entries are possible.

The cursor mode can be changed in runtime, but must be specified before starting runtime. The cursor mode is set in the picture's object properties. To accomplish this, click on a free area in the picture and select the "Properties" tabbed card in the object settings. The value of "Cursor Mode" attribute in the "Miscellaneous" properties group can be set to either "Alpha Cursor" or "Tab Order cursor".

Combining the alpha and tab order cursors in a picture

The cursor mode can be switched in runtime, if a hotkey has been assigned for this purpose, see the chapter entitled "Hotkeys for operation and screen navigation". With this hotkey, the operator can then toggle back and forth between using the alpha or tab order cursor in runtime. In this case, the tab sequence must be set for both types of cursor.

Jump from one object to the next in runtime

The tab sequence is used to set the order in which objects are operated in runtime. By default, the cursor is moved in operation with the <TAB> key (or <SHIFT+TAB> for the reverse order). You can also configure the operation to use shortcuts or the mouse. It is also possible to define hotkeys. For more information on this topic, please see the chapter entitled "Setting up Runtime".

5 Dynamizing Process Screens

5.1 Types of Dynamization

Introduction

WinCC offers various ways of dynamizing the objects of a process picture.

Basically, there are two types of dynamization.

- Dynamic objects change their appearance and position depending on, for example, a process value. An example of a dynamic object is a bar whose length is influenced by a current temperature, a pointer instrument with a moving pointer, or an object which changes color depending on the position of a workpiece.
- Operator-controllable objects react to events, e.g. a mouse-click, and enable the operator to actively intervene in the process. Operator-controllable objects can be, for example, buttons, sliders, or I/O fields used for entering certain process parameters.

Note
If you copy a dynamized object in the Graphics Designer, the dynamics are also copied. The copied object will then be affected by the dynamics in just the same way as the original object.

Dynamizing by Means of Direct Tag Connection

When a tag is connected to a property of an object, the value of the tag is transferred directly to the object property. This means, for example, that the value of a tag can be directly influenced by an I/O field.

Dynamization by means of tag connection is indicated in the "Object Properties" dialog by the 💡 symbol and the name of the tag.

Dynamizing by Means of Indirect Tag Connection

When a tag is connected to a property of an object, the value of the tag is interpreted as the tag name. The value of this tag is transferred to the object property. You configure an indirect tag connection by double-clicking the box in the "Indirect" column of the "Object Properties" dialog.

Dynamization by means of tag connection is indicated in the "Object Properties" dialog by the 💡 symbol and the name of the tag. The indirect tag connection is indicated in the "Indirect" column of the "Object Properties" dialog by the ☑ symbol.

Dynamizing by Means of Direct Connection

Direct connection is used for reacting to events. If the event occurs in Runtime, the value of a source element is used for a target element. The values of the source element and target element can be defined by:

- a constant
- a WinCC tag
- the value of an object property

Dynamizing by means of direct connection is indicated in the "Object Properties" dialog by the ⚡ symbol.

Dynamizing Using Dynamic Dialog

The Dynamic dialog is used for dynamizing an object property. You should use the Dynamic dialog if you want to map the value of a tag to a value which can be interpreted by the operator. For example, you can use the Dynamic dialog to map the value range of a tag to color values.

Dynamizing using the Dynamic dialog is indicated in the "Object Properties" dialog by the ⚡ symbol.

Dynamizing Using VBS Action

VBS actions are used to dynamize an object property or to react to events. You should use VBS actions if, for example, you want to process several input parameters in one action or you want to execute conditional instructions (if ... then ...).

Dynamizing using a VBS action is indicated in the "Object Properties" dialog by the ⚡VB symbol.

Dynamizing Using C Action

C actions are used to dynamize an object property or to react to events. You should use C actions if, for example, you want to process several input parameters in one action or you want to execute conditional instructions (if ... then ...).

Dynamizing using a C action is indicated in the "Object Properties" dialog by the ⚡C symbol.

5.2 Types of Trigger

Introduction

Triggers are required so that actions can be executed in Runtime. To allow this, a trigger is connected to an action thus forming the triggering event which calls the action. Actions without a trigger are not executed.

The following triggers are available for dynamizing objects:

- Cyclic triggers
- Cyclic triggers based on the window cycle
- Cyclic triggers based on the picture cycle
- Tag Triggers
- Event-Driven Triggers

Note
It is important to remember that the cycle time has a major effect on the performance of the project. All the actions of a picture must be completed within their cycle time. Apart from the runtimes of the actions, the times required for requesting the tag values and the reaction times of the automation systems must also be taken into consideration. You should only set trigger events with a cycle time under one second if variables which change rapidly have to be queried.

5.2.1 Cyclic Triggers

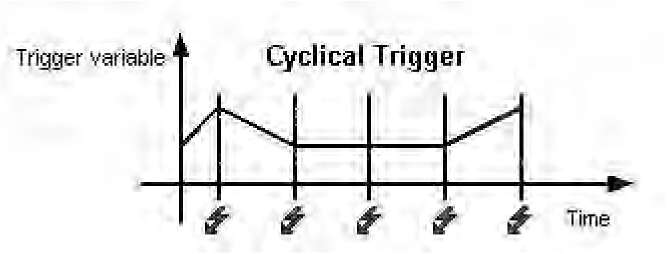
Introduction

Cyclic triggers are a means of processing periodic actions in WinCC. In the case of a cyclic trigger, the action is executed when the trigger event occurs, e.g. every 20 seconds.

Method of functioning

When actions with a cyclical trigger are configured in a picture, each tag is requested individually following selection of the picture.

The beginning of the first interval coincides with the start of Runtime. The length of the interval is determined by the cycle. Cycles between 250 ms and 1 h can be selected. Self-defined user cycles are also available.



The action is always executed when the triggering event occurs.

Note
For system-related reasons, it may not be possible to guarantee that an action with a cyclic trigger is executed at precisely the specified times.

Cyclic Triggers Based on Window Cycle

A cyclic trigger is used as the trigger. The cycle time is defined by the object property "Update Cycle" of the "Picture Window" object. This cycle provides the option of defining the cycles of all the actions used in a picture window centrally.

Cyclic Triggers Based on Picture Cycle

A cyclic trigger is used as the trigger. The cycle time is defined by the object property "Update Cycle" of the picture object. This cycle provides the option of defining the cycles of all the actions used in a picture centrally.

Recommendation

In order that the WinCC project has a high level of performance, it is advantageous to dispense with actions which have a cyclical trigger and to realize periodic actions in the automation system.

5.2.2 Tag Triggers

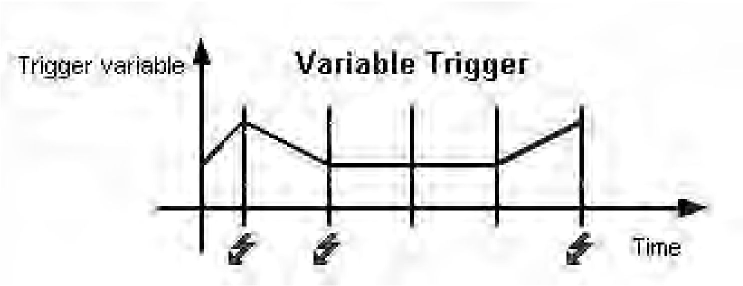
Introduction

Tag triggers consist of one or more specified tags. The action which is connected to such a trigger is executed if a change in the value of one of these tags was detected when the query was launched.

Method of functioning

When actions with tag triggers are configured in a picture, all the tag triggers related to an action are requested block by block when the picture is selected.

The beginning of the first interval coincides with the point at which the picture is first selected. The length of the interval is determined by the cycle. Query cycles between 250 ms and 1 h can be selected. Self-defined user cycles are also available.



The action is only executed when the value of the tag triggers has changed.

Note
If a tag briefly changes within a query cycle and then resumes its original value, the action is not executed.

Upon Change

You can also configure a tag trigger so that the action is executed whenever the tag value is changed.

In the case of process tag, the "Upon change" mode corresponds to a cyclic read job with a cycle tie of 1 second.

Note
If the WinCC diagnostics tags are used as action triggers, this form of the tag trigger should not be used. Diagnostics tags can change very quickly. Since every change could trigger the action, it can lead to high system utilization.

Recommendation

Tag triggers should be used so that the WinCC project has a high performance level:

- With cyclic actions, the action is always executed, e.g. every 2 seconds. The tag trigger only executes the action if a change in the value of the tag has been detected. This reduces the load on the system and increases performance.
- When a picture is selected, all of the tags contained in tag triggers are known and can be requested all at the same time by the automation system. The best possible optimization can thus be achieved from the channel.

5.2.3 Event-Driven Triggers

Introduction

Actions which are connected to an event are executed whenever this event occurs. Events can be, for example, mouse control, keyboard control, or changes in focus.

If the "Mouse Control" event is connected to an action, this action is also triggered by a configured hotkey.

Method of functioning

The action is only executed when the object's triggering event is triggered. If the event occurs, all the tags contained in the action are registered. Process tag are subsequently updated at a cycle time of 1 s.

Recommendation

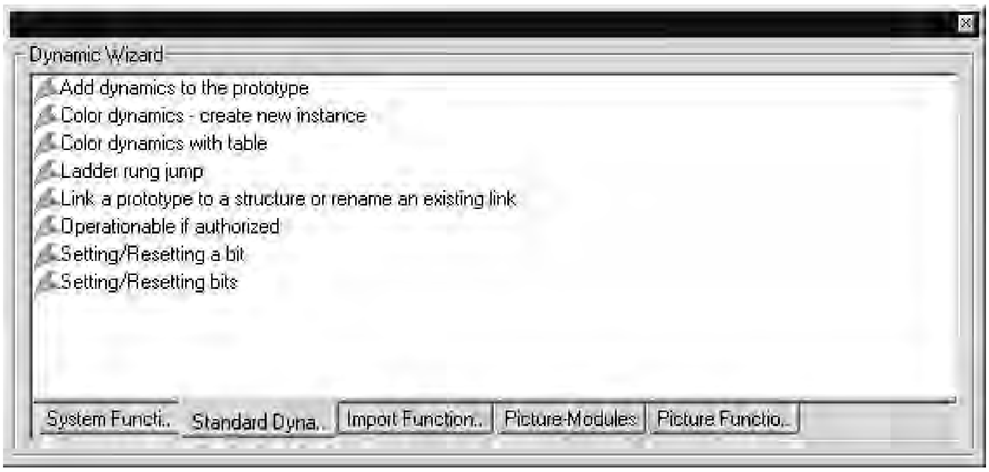
Event-driven triggers are not suitable for bulk configuration in the case of dynamizing with C-actions because each action must be logged on and off individually by the action control.

5.3 Dynamizing Using Dynamic Wizard

Introduction

With the Dynamic Wizard, you can dynamize an object using C actions. When you execute a wizard, pre-configured C actions and trigger events are defined and transferred to the object properties. If necessary, you can change the C actions in the object properties using the Events tab.

Dynamic Wizard



Dynamic Wizard
Add dynamics to the prototype
Color dynamics - create new instance
Color dynamics with table
Ladder rung jump
Link a prototype to a structure or rename an existing link
Operationable if authorized
Setting/Resetting a bit
Setting/Resetting bits
System Functi..
Standard Dyna..
Import Function..
Picture-Modules
Picture Functio..

The pre-configured C actions are divided into the following groups:

- System functions
- Standard Dynamics
- Picture components
- Import functions
- Picture functions
- SFC

The available groups and the wizards contained in the groups depend on the WinCC installation type, the project type, and the selected object.

Note:
You can use the Toolbars... item in the View menu of the Graphics Designer to show or hide the Dynamic Wizard.

5.4 Dynamizing by Means of Tag Connection

Introduction

When a tag is connected to a property of an object, the value of the tag is transferred directly to the object property. This means, for example, that the value of a tag can be directly influenced by an I/O field.

You should always use this type of dynamization if you want to transfer the value of a tag directly to an object property.

5.5 Dynamizing by Means of Direct Connection

Introduction

Direct connection can be used as a reaction to events. If the event occurs in Runtime, the 'value' of a source element (Source) is used for a target element (Target).

Constants, tags, or the attributes of the objects in the picture are available as sources.

Tags or the dynamizable attributes of objects and windows or tags can be used as targets.

The advantages of direct connection are the simple configuration and the time response in Runtime. Direction connection has the best performance of all dynamization types.

Copying Objects

If in the Graphics Designer you copy an object whose properties are dynamized with a direct connection, the dynamizations are also copied.

If the direct connection in the original object relates to an object property of this object, the direct connection in the copied object relates to the corresponding property of the copied object.

If the direct connection in the original object relates to an object property of a third object, this third object is influenced by the direct connection of the copied object in exactly the same way as the direct connection of the original object.

5.6 Dynamizing Using Dynamic Dialog

Introduction

The Dynamic dialog can be used for dynamizing object properties. In the Dynamic dialog you formulate an expression by using tags, functions, and arithmetic operators. The value of the expression and the status of the tags used in the expression are used to form the value of the object property in Runtime.

The Dynamic dialog can, for example, be used

- to map value ranges of a tag to colors
- to monitor individual bits of a tag and to map the value of the bit to colors or texts
- to monitor a boolean tag and to map the value of the tag to colors or texts
- to monitor the status of a tag

Note
If several tags or operators are used, the considerable performance advantage of the Dynamic dialog is lost.

Conversion into C Action

An action created using the Dynamic dialog is indicated in the "Object Properties" dialog by the ⚡ symbol.

You can view the code of this action by right-clicking the ⚡ symbol in the "Object Properties" dialog and selecting the "C Action" command in the context menu. The dynamization created using the Dynamic dialog is then converted into a C action.

If the action is stored or the action code is modified, it will no longer be possible to change the action using the Dynamic dialog.

5.7 Dynamizing Using VBS Action

Introduction

Apart from the dynamization options using direct connection, C action, and tags, WinCC also offers VBS actions for dynamizing graphic objects in Runtime.

You should use VBS actions if

- you want to process several input parameters in an action
- you want to execute conditional instructions (if ... then ...)
- you want to change several object properties in an action
- you want to access selection dialogs of the operating system, e.g. file selection dialog or color selection dialog

You create VBS actions in the VBS action editor of the Graphics Designer. The action editor offers a similar range of functions to the VBS editor "Global Script". From the Graphics Designer you can also access procedures which you have created in Global Script.

Actions which you create in the Graphics Designer are always stored with the picture in which they have been configured. In addition to all configured object properties, the configured VBS actions are also documented in the project documentation of the Graphics Designer. All of the VBS actions configured in this picture are displayed if you select a picture in the WinCC Explorer and call up the Properties dialog using the context menu.

Further information on VBS actions can be found in the section "Creating Procedures and Actions Using VBScript".

Application Scenarios

VBS Action for Dynamizing Object Properties

You use VBS actions for dynamizing an object property. You can dynamize the value of the object property in Runtime depending on a trigger, a tag, or the status of other object properties. You should use a VBS action if the options provided by tag connection or the Dynamic dialog are not sufficient to solve the task in question.

VBS Action for Events

You can use a VBS action to react to an event which occurs at a graphic object. You should use a VBS action if the options provided by tag connection or the Dynamic dialog are not sufficient to solve the task in question.

The use of actions to react to changes to object properties influences performance in Runtime.

The event occurs if the value of the object property changes. The action associated with the event is then started. When a picture is closed, all of the started actions are stopped one by one. This can cause high system loads.

5.8 Dynamizing Using C Action

Introduction

C actions can be used for dynamizing object properties and for reacting to events. When object properties are dynamized, the value of the object property is determined by the return value of the C function.

Use C actions if, for example, you want to process several input parameters in one action or you want to execute conditional instructions (if ... then ...). It is advisable to use C actions especially if in the case of recipes, for example, you want to access several tags in the automation system at the same time.

Application Scenarios

C Action for Dynamizing Object Properties

You use C actions for dynamizing an object property. You can dynamize the value of the object property in Runtime depending on a trigger, a tag, or the status of other object properties. You should use a C action if the options provided by tag connection or the Dynamic dialog are not sufficient to solve the task in question.

C Action As Reaction to Events

You can use C actions to react to an event which occurs at a graphic object. You should use a C action if the options provided for direct connection are not sufficient to solve the task in question.

The use of actions to react to changes to object properties influences performance in Runtime.

The event occurs if the value of the object property changes. The action associated with the event is then started. When a picture is closed, all of the started actions are stopped one by one. This can cause high system loads.

6 VBS for Creating Procedures and Actions

6.1 Using Visual Basic Script in WinCC

Introduction

In addition to the C script, WinCC also provides the VBScript program language as a programming interface in order to make the WinCC runtime environment dynamic.

Target Group of the Documentation

This documentation is aimed at project engineers with experience of Visual Basic or WinCC Scriptings (C) used to date.

Application Possibilities

VBScript (VBS) provides access to tags and objects of the graphical runtime system at runtime and can execute picture-independent functions:

- Tags: Tag values can be read and written in order, for example, to specify tag values for the PLC by clicking the mouse when positioned on a button.
- Objects: Object properties can be made dynamic using actions and actions can be triggered by events influencing objects.
- Picture-independent Actions: Picture-independent actions can be triggered cyclically or according to tag values, e.g. for the daily transfer of values into an Excel table.

VBS can be used at the following points in WinCC:

- In the Global Script Editor: This is used to configure picture-independent actions and procedures. The procedures can be used in picture-dependent and picture-independent actions. Several procedures are compiled in a topic-related module.
- In Graphics Designer: Picture-dependent actions can be configured with which the properties of graphic objects can be made dynamic or caused to respond to events in runtime.

Application Scenarios

VBS can be used in runtime, for example:

- to configure setpoint value specification for tags for the operation of a graphic object in order to define a value for the PLC by clicking a mouse, for example.
- to configure switching the runtime language for the operation of a graphic object.
- to configure the change of color, e.g. cyclically (flashing) or to display statuses (motor on).

Apart from the specific WinCC applications, the general functionality of VBS can also be used to customize the Windows environment, e.g.:

- to transfer data to another application (e.g. Excel).
- to start external applications from WinCC.
- to create files and folders.

The automation objects in your environment are available with which to customize the Windows environment.

Note:

All the objects supplied with the Windows Script Host (WSH) from Microsoft can be integrated in the environment using the standard VBS method CreateObject. However, direct access to the WSH object is not possible using VBS from WinCC.

Note:

There is no guarantee for VBS functionally to customize the Windows environment and no WinCC support.

Limits to Other Programming Languages in WinCC

VBS and C

VBScript can be used in WinCC parallel to C-Script, but do not mix the script types:

- VBScripts and C-scripts can be configured within a picture and project.
- C-scripts cannot be invoked in VBScripts and vice versa.
- VBS provides internal interfaces to tags and picture objects while the C environment enables access to other WinCC subsystems (e.g. the report system).

VBS and VBA

VBA is used in WinCC Configuration during the configuration in order to adapt Graphics Designer to your individual requirements and to simplify and automate configuration. VBA programs only run in the WinCC configuration environment.

As opposed to VBA, VB scripts only run in WinCC Runtime and, from there, enable access to graphic objects and tags. Objects and pictures can be neither created nor modified on a permanent basis in VBS, as opposed to VBA.

The main language-related differences between VBA and VBS are e.g.:

- VBS was developed for use in Internet, VBA for the automation of software applications.
- The data type of VBS tags is always VARIANT. VBA, on the other hand, differentiates the individual data types such as INT, DOUBLE, STRING, etc.
- Certain language constructs from VBA have been removed from or added to VBS.
- Faults are handled differently in VBS as compared to VBA.

A complete list of the differences between VBA and VBS is provided in the Appendix in "Basic Principles of VBScript".

Procedures, Modules and Actions

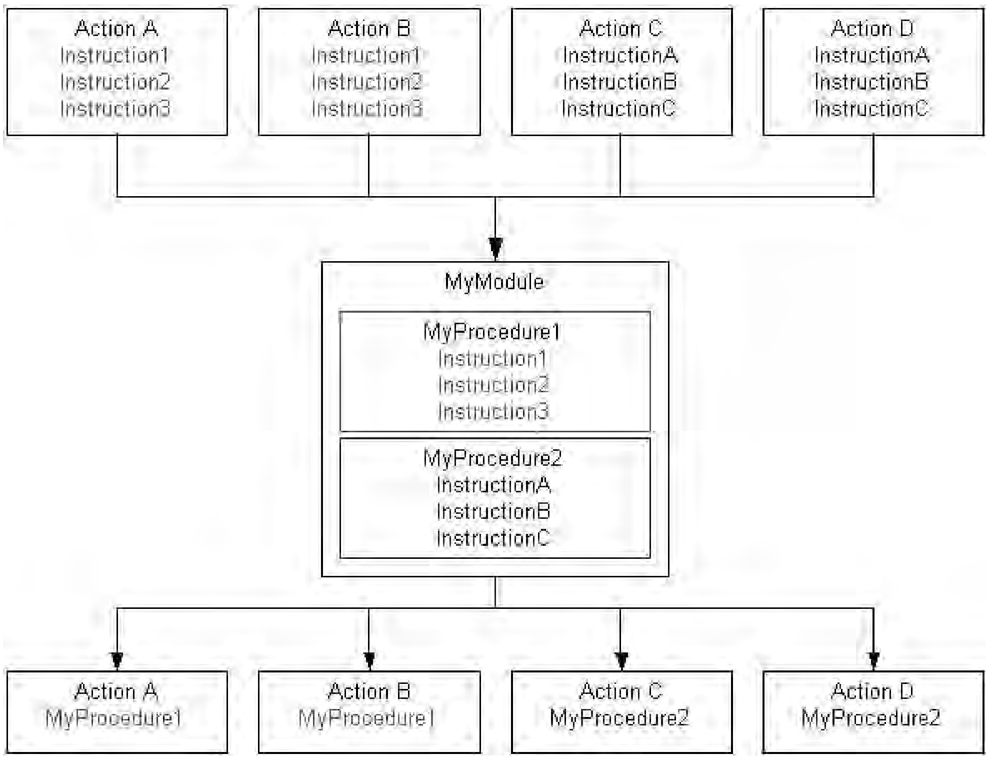
VBS in WinCC enables the use of procedures, modules and actions to make the Runtime environment dynamic:

- Procedures: A procedure corresponds to a function in C. Codes are stored in procedures which are then used at several points in the configuration. Call in the code or another procedure in an action by invoking the procedure name. Procedures can be created in WinCC with or without return values. Procedures do not have their own trigger, they are always called in by an action.
- Modules: It is advantageous to compile related procedures to units in modules. Create modules for procedures, for example, which must be used in a specific picture or belong to a specific topic, such as auxiliary mathematical functions or database access functions.
- Actions: Actions are always activated by a trigger, namely a triggering event. Actions are configured in graphic object properties, in events which occur on a graphic object or globally in a project. Codes used several times can be called, in the form of procedures, in actions.

6.2 Modules and Procedures

Introduction

Procedures are used to make codes, created only once, available at several points in a project. Instead of entering the code several times, simply call in the corresponding procedure. The code is clearer and easier to maintain.

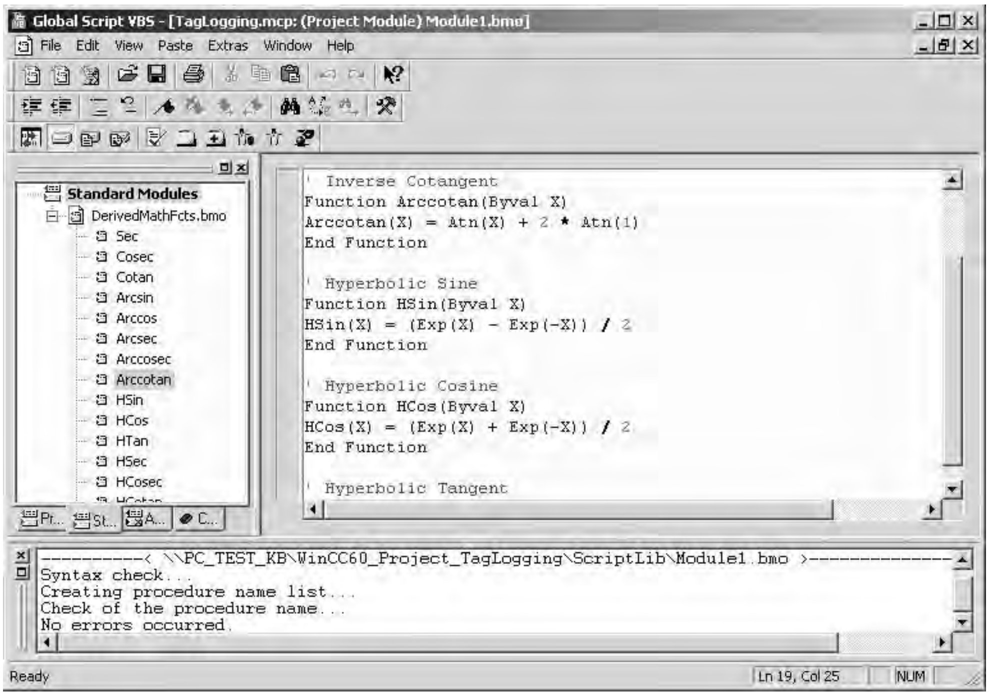


Related procedures should be stored in modules. When a certain procedure is called in via an action during runtime, the module is loaded which contains that procedure. Observe the following when structuring the modules and procedures:

- The more modules which must be loaded when a picture is called in, the worse the performance in runtime.
- The larger a module, the more procedures are contained and, thus, the longer the loading time for the module.

Organize the modules sensibly, e.g. a module with procedures for a specific system part/picture.

Another type of structuring of procedures in modules is the functional structuring, e.g. a module containing mathematical functions. Use this structure, for example, for modules which should be used globally for projects. The following example illustrates a module containing mathematical functions derived from the standard functions:



```
' Inverse Cotangent
Function Arccotan(Byval X)
Arccotan(X) = Atn(X) + 2 * Atn(1)
End Function

' Hyperbolic Sine
Function HSin(Byval X)
HSin(X) = (Exp(X) - Exp(-X)) / 2
End Function

' Hyperbolic Cosine
Function HCos(Byval X)
HCos(X) = (Exp(X) + Exp(-X)) / 2
End Function

' Hyperbolic Tangent
```

```
----------< \\PC_TEST_KB\WinCC60_Project_TagLogging\ScriptLib\Module1.bmo >--------------
Syntax check...
Creating procedure name list...
Check of the procedure name...
No errors occurred.
```

Procedure Features

The procedures in WinCC have the following properties:

- They are created and modified by the user.
- They can be protected against modification and viewing by means of a password.
- They do not have a trigger.
- They are stored in a module.

WinCC does not provide predefined procedures, but does provide code templates and Intellisense, for example, to simplify programming. Procedures differ according to module assignment:

- Standard procedures apply globally to projects located on the computer on which they were created.
- Project procedures can only be used in the project in which they were created.

Module Features

A module is a file in which one or more procedures are stored. The modules in WinCC have the following properties:

- They can be protected against modification and viewing by means of a password.
- The have the file extension *.bmo.

Modules differ according to the validity of their procedures in:

- Standard Modules: Contain procedures which are globally available to the project. Standard modules are stored in the WinCC file system under: <WinCC installation directory>\ApLib\ScriptLibStd\<Module name>.bmo
- Project Modules: Contain project-specific procedures. Project modules are stored in the WinCC file system under: <Project directory>\ScriptLib\<Module name>.bmo. Since the project modules are stored in the project directory, they are copied when the WinCC project is copied.

Note:
If WinCC must be reinstalled and the standard procedures and modules need to be used again, save the module files prior to the reinstallation in a different directory and copy them in the relevant WinCC directory after the reinstallation has been completed. Otherwise, the standard modules in the WinCC installation directory are deleted during the installation routine.

Using Procedures and Modules

Procedures are used in:

- Actions (in Graphics Designer and Global Script)
- Other procedures (in Global Script)

Procedures are structured in modules.

6.3 Actions

Introduction

An action is always started by a trigger. An action, for example, is triggered in runtime when an object is operated by a mouse click, a certain time has occurred or a tag has been modified.

Action Features

Actions are defined once in Global Script and then available independent of the picture. Global Script actions are only valid in the project in which they were defined. Actions linked to a graphic object are only valid in the picture in which they were defined.

Note:
VBS does not currently allow the creation of computer-specific actions.
The following applies for clients in a multi-user system: All global actions configured on a server are also executed on a client when a project is opened.
The following applies for clients in a distributed system: If actions should be used on a client computer, copy all the action files in the corresponding project directory on the client.

Actions have the following properties:

- Actions are created and modified by the user.
- Action in Global Script can be protected against modification and viewing by means of a password.
- Actions have at least one trigger.
- Actions in Global Script have the file extension *.bac.
- Global Script actions are stored in the WinCC file system under: <Project directory>\ScriptAct\Actionname.bac

Action Trigger

Triggers are required to execute actions in Runtime. A trigger is linked to an action thus forming the triggering event which calls the action. Actions without a trigger are not executed.

The following trigger types are available in WinCC:

- Timer: Acyclic or cyclic trigger, e.g. for calling a picture or every hour.
- Tag: Change of value
- Event: Modification of object properties (e.g. change of color) or event on an object (e.g. mouse click).

Processing of Actions in Runtime

In Graphics Designer:

Two actions of the same type can be executed simultaneously in runtime. In order, for example, that cyclic actions are not hindered by an action executed by a mouse click, event triggered actions and cyclic/tag triggered actions in Graphics Designer are executed independently of each other.

Note:
Note that synchronization between the two action types in WinCC can only be achieved by internal WinCC tags. As a result of the separate processing, no common data area exists between event triggered and cyclic/tag triggered actions.

If processing cyclic actions in pictures, for example, is prevented by a high system load or another action, the action is started at the next opportunity. Cycles which are not executed are not retained in a queue but rejected.

After a change of picture, scripts still running are automatically stopped 1 minute after the change of picture.

Scripts which are still running when runtime is terminated are stopped after 5 seconds.

In Global Script

Picture-independent actions from Global Script are executed in runtime in succession after being triggered. If an action is triggered while another action is in progress, the second action is retained in a queue until it can be executed.

Note:
Note that synchronization between actions in Global Script and in Graphics Designer can only be achieved by means of internal WinCC tags. There is no common data area between the actions in Graphics Designer and in Global Script.

Using the Actions

Actions can be used as follows:

- In Global Script: The global actions defined here run picture-independent in runtime.
- In Graphics Designer: The actions defined here only run in the configured picture. An action is configured in Graphics Designer on an object property or an event on a graphic object.

6.4 Use of CrossReference

CrossReference and Tag Trigger

The CrossReference from WinCC can be used to quickly find all the application points of tags, even in VBS actions. Tag triggers in actions in Graphics Designer can be "linked" using CrossReference, i.e. replaced by other tags at all or selected points.

Note:
Tags can also be directly linked in Graphics Designer by marking the graphic object and selecting the "LInking ..." command from the context menu.

Further information on CrossReference is available in the WinCC documentation.

Actions and CrossReference

All the actions used in a picture can be displayed by means of the picture properties. To do this mark the picture in WinCC Explorer and select the "Properties" context menu command. After double clicking on an entry, detailed information on the type of dynamics appears.

It is also possible to display all the tags and pictures used in actions by means of the WinCC CrossReference. CrossReference can also be used for the to link tag connections of Graphics Designer actions easily.

Note:
For pictures and tags addresses in your code, always use the standard formulations
HMIRuntime.BaseScreenName = "Screenname" and HMIRuntime.Tags ("Tagname") to ensure that pictures and tags are detected by CrossReference.

Tags and CrossReference

All tags addressed with the standard formulation

```
' VBS1
HMIRuntime.Tags("Tagname")
```

are automatically acquired by WinCC CrossReference and listed in the picture properties.

If the tags should be addressed with other formulations in the code, this can be notified using the following section of CrossReference:

```
' WINCC:TAGNAME_SECTION_START
Const TagNameInAction = "TagName"
' WINCC:TAGNAME_SECTION_END
```

This section can be inserted in VBS actions as often as required.

Note:
The acquisition of combined tag names fro the CrossReference cannot be guaranteed.

Pictures and CrossReference

All pictures addressed with the standard formulation

```
'VBS2
HMIRuntime.BaseScreenName = "Screenname"
```

are automatically acquired by WinCC CrossReference and listed in the picture properties.

If the pictures should be addressed with other formulations in the code, this can be notified using the following section of CrossReference:

```
' WINCC:SCREENNAME_SECTION_START
Const ScreenNameInAction = "ScreenName"
' WINCC:SCREENNAME_SECTION_END
```

This section can be inserted in VBS actions as often as required.

Note:
For reasons of compatibility to future versions, always write picture names without the file name extension ".PDL".

6.5 Using Global Tags in VBS

Introduction

Global tags can be defined in the Global Script Editor which can then be used in all actions and procedures.

Using Global Tags in Graphics Designer and Global Script

Observe the following conditions when using global tags in Graphics Designer and Global Script:

- In order to use a global tag in an action in Graphics Designer, call in the procedure in which the tag is defined so that the associated module is loaded in runtime.
- In order to use a global tag in an action in Global Script, at least one procedure must be activated from the module in at least one global action in which the tag is defined so that the module is loaded in Global Script Runtime. This need not be the procedure in which the tag was defined.

This process is necessary because actions from Global Script and Graphics Designer are processed independently of each other in runtime. There is no common data area between the two runtime systems.

When actions from Global Script and Graphics Designer must be synchronized, use internal WinCC tags.

Using Global Tags in Graphics Designer

When using global tags in Graphics Designer, observe the following conditions:

- In order to use a global tag in cyclic or a tag triggered action in Graphics Designer, call in the procedure in which the tag is defined. This also applies when the tag has already been called in an event triggered action.
- In order to use a global tag in an event triggered action in Graphics Designer, call in the procedure in which the tag is defined. This also applies when the tag has already been called in a cyclic or tag triggered action.

This process is necessary because the cyclic/tag triggered actions and the event triggered actions in Graphics Designer in Runtime are processed independently of each other. There is no common data area between the two types of action.

When cyclic/tag triggered actions and event triggered actions must be synchronized, use internal WinCC tags.

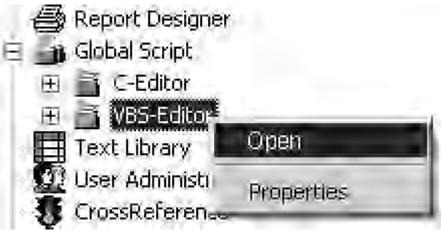
In the case of Graphics Designer, it is also possible to define global tags in a separate declaration section. Since event triggered and cyclic/tag triggered actions are separately processed in runtime, the global tags can only be jointly addressed within the event triggered or cyclic/tag triggered actions.

6.6 VBScript Editors

Introduction

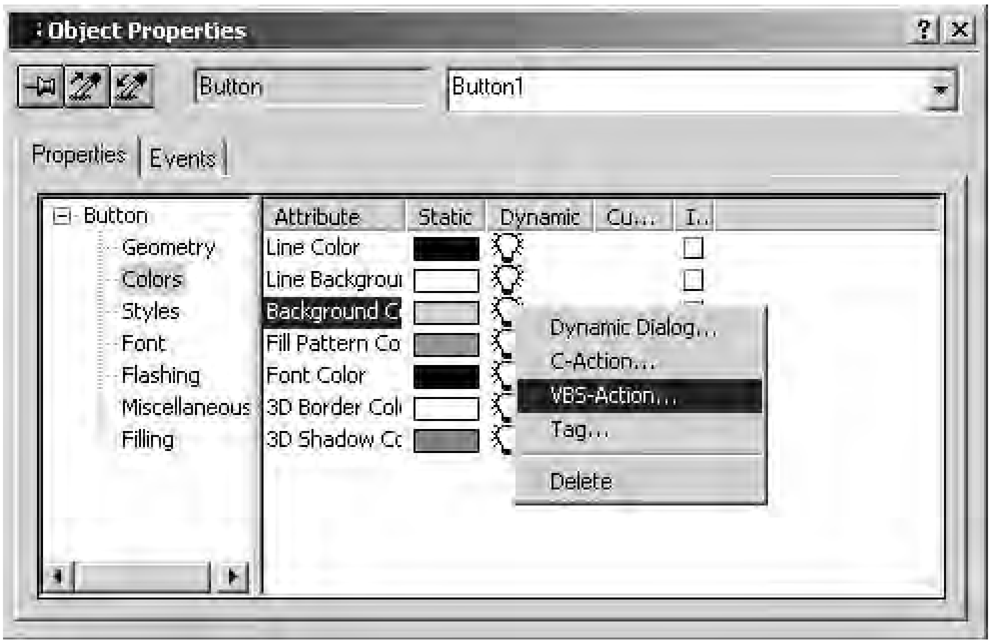
VBScripts can be programmed at two points in WinCC:

- In Global Script: Global Script is the central editor for VBS programming. Call it in via WinCC Explorer.



Global Script can be used to program global actions, which are independent of the picture and are not linked to a graphic object, and procedures, which can be activated in other actions or procedures.

- In Graphics Designer: Graphics Designer can be used to program actions related to object properties or events related to graphic objects. The action editor in Graphics Designer is called in via the context menu in the Properties dialog of a graphic object.



Attribute	Static	Dynamic	Cu...	I...
Line Color	■	💡		□
Line Backgrou	□	💡		□
Background C	▨	💡		
Fill Pattern Co	▨	💡		
Font Color	■	💡		
3D Border Col	□	💡		
3D Shadow Cc	▨	💡		

Graphics Designer can be used to make object properties dynamic and trigger actions through events on objects.

Restrictions, Global Script - Graphics Designer

Graphics Designer can be used to program actions and picture-specific procedures but not global procedures valid for the entire project. However, global procedures which were programmed in Global Script can be called in.

Note:
This documentation is used primarily to describe Global Script and makes reference to deviations in the functionality compared to Graphics Designer. A detailed description of the Graphics Designer action editor is provided in the WinCC help topic "Dynamization".

6.7 Creating and Editing Procedures

Introduction

Projects and standard procedures can be programmed in WinCC using VBS:

- Project procedures can only be called in the current project. Since procedures are stored in the project directory, they are automatically copied when a project is copied.
- Standard procedures can be called in by all computers linked to a project. When a project is copied onto another computer, the standard procedures must be copied into the corresponding directory on the target computer manually.

The copied procedures are directly available for use in runtime. This becomes visible in the editor when the view is updated.

Apart from the procedures programmed by yourself, general VBS functions can also be used (e.g. Abs, Array,... Year). These general VBS functions can be invoked in the code using the "Function List" command from the context menu.

In addition, WinCC provides the most popular instructions as code templates (e.g. If...Then, When...While). The code templates can be moved from the navigation window's Code Templates tab control directly into your code via drag&drop.

If you insert a code template into your code, it is important to note that, for example, conditions in the templates are identified by "_XYZ_". You must replace these placeholders with the appropriate information.

Using Procedures

Procedures are used for the central creation and maintenance of codes which are to be implemented at several point in the configuration. Codes are written and saved in a procedure and the procedure is called in with the current parameters in actions or other procedures instead of repeatedly entering the same code.

Procedures are created for repeated functionalities, e.g.

- Calculations with different starting values (procedure with return value)
- Checking tag values (procedure with return value)
- Executing tasks (procedure with return value)

This is linked to the following advantages:

- The code is only programmed once.
- Modifications are only made at one point, namely in the procedure, not in each action.
- The action code is shorter and, thus, remains clearer.

Related procedures should be stored in modules in WinCC.

Procedures are loaded in runtime when the calling action is executed.

If a procedure (module) used in a picture is modified, the modification is applied the next time the picture is loaded. This means that a picture currently displayed only works with the modified procedure after the picture has been reloaded.

Note:
Procedures can be used in actions in Global Script and Graphics Designer.
In order to use a global tag defined in Global Script in an action in Graphics Designer, observe the following:
In order that access can be made to the tag, it is necessary to call in the procedure in which the tag is defined.
In order to use a global tag in picture-independent actions in Global Script, observe the following:
In order that access can be made to the tag, at least one procedure in the module containing the tag must be called in at least one global action.

Procedure - Action Restrictions

Global procedures valid for the entire project can only be created in Global Script. Graphics Designer can only be used to create picture-specific procedures and call in global procedures in actions. Picture-specific procedures in Graphics Designer are defined in the declaration area of an action.

A procedure is not executed without an action.

File Name and Procedure Name

The procedure name is entered in the first line of the procedure code. The procedure is displayed in the navigation window and called in actions under this name. Procedures do not have a file name of their own but are stored in a module.

Module names are assigned in the editor's navigation window. Use the "Save As" command to save a module under another name in the project directory.

Since procedures in Global Script are valid for the entire project, procedure names must always be unique. Module names can be used more than once within a project, e.g. in different subdirectories or stored separately in the standard and project directories.

Displaying Procedures and Modules

If a module is saved which contains at least one syntactically incorrect procedure, it is displayed in the navigation window with the adjacent symbol.

Caution
If a module contains a syntactically incorrect procedure, the module can no longer be loaded. Procedures can no longer be called from the module.

If a module is saved which only contains syntactically correct procedures, it is displayed in the navigation window with the adjacent symbol.

Procedures and Modules

Procedures are classified as standard or project procedures according to their assignment to standard or project modules. Standard and project modules are located on the corresponding tab controls in the Global Script navigation window.

Use the modules in order to compile procedures to practical function groups. Observe the following when structuring the modules and procedures:

- The more modules which must be loaded when a picture is called in, the worse the performance in runtime.
- The larger a module, the more procedures are contained and, thus, the longer the loading time for the module.

Organize the modules sensibly, e.g. a module with procedures for a specific system part/picture.

6.8 Creating and Editing Actions

Introduction

When using VBS in WinCC, there is no differentiation between local (valid for entire project) and global (valid on all computers) actions. A configured action is always valid globally.

A copied action is available for use in runtime following a restart or opening and saving the action. This becomes visible in the editor wen the view is updated.

VBS actions can be used to make graphic objects and object properties dynamic in runtime or to execute picture-independent actions.

Note:
Please note that the object name length of objects made dynamic in Graphics Designer is limited to approx. 200 characters, and each special character used in an object name is converted to five characters in the script files. The special characters are represented by a four-place hexadecimal code behind the preceding X. If the name of an object made dynamic is too long, a corresponding error message appears. Further information is available in this help under "Structure of VBScript Files".

Caution
If you make an object property dynamic with a VBS action via the return value of a script, the value of the object property is written only if it has changed in relation to the last script run. It is not considered if the value had been changed from another location.
Therefore it is illegal to change properties which have been made dynamic by VBS action via the return value from another location (e.g., other C scripts or VBS scripts).
If you do not observe this, wrong values can be the results.

Using the Actions

Actions can be used as follows:

On graphic objects in Graphics Designer

Making properties dynamic (action with return value), e.g.:

```
Function BackColor_Trigger(ByVal Item)
'VBS143
   BackColor_Trigger = RGB(125,0,0)
End Function
```

Triggered by an event on an object (action without return value), e.g.:

```
Sub OnClick(ByVal Item)
'VBS144
   Item.BackColor = RGB(255,0,0)
End Sub
```

Picture-independent in Global Script

As a cyclic action, e.g. incrementing a tag:

```
Option Explicit
Function action
'VBS145
   Dim objTag1
   Dim lngValue
   Set objTag1 = HMIRuntime.Tags("Tag1")
   lngValue = objTag1.Read
   objTag1.Write lngValue + 1
   action = CLng(objTag1.value)
End Function
```

Executing Actions

An action can be assigned several triggers. The action is always executed when one of the triggering events occurs. Observe the following:

- Actions in Global Script cannot be executed simultaneously. The action triggered last is held in a queue until the action currently being performed is completed.
- When using Graphics Designer, cyclically and tag-driven actions cannot be triggered simultaneously. If the execution of a tag-driven action hinders the execution of a cyclic action, the cyclic action is executed when the tag-driven action has finished. The cyclic action is held in a queue during the non-execution phase. When the current action is completed, the cyclic action is executed with the normal cycle.
- In Graphics Designer, event-driven actions cannot be executed simultaneously.

The action types mentioned do not prevent each other from being executed: The execution of actions in Global Script has no influence on actions in Graphics Designer. In the same way, in Graphics Designer, the execution of cyclically or tag-driven actions has no effect on the execution of event-driven actions.

Note:
Actions in pictures which are still running one minute after the picture has been deselected are terminated by the system. This is recorded in a logfile entry.

Action Details

All the actions used in a picture can be displayed by means of the picture properties. To do this mark the picture in WinCC Explorer and select the "Properties" context menu command. After double clicking on an entry, detailed information on the type of dynamics appears.

It is also possible to display all the tags and pictures used in actions by means of the WinCC CrossReference. CrossReference can also be used for the to link tag connections of Graphics Designer actions easily.

Note:
For pictures and tags addresses in your code, always use the standard formulations
HMIRuntime.BaseScreenName = "Screenname" and HMIRuntime.Tags ("Tagname") to ensure that pictures and tags are detected by CrossReference.

Procedure - Action Restrictions

Actions can be used to program instructions and call procedures. Codes are programmed within procedures for use at several points in a configuration. Actions always have a trigger.

Creating and Editing Actions

Actions can be configured in Global Script and Graphics Designer. Use Global Script to configure global actions which can be executed independently of the picture currently open. Graphics Designer is used to configure actions related to graphic objects which should be executed when the picture is opened in runtime or when the configured trigger occurs.

The script editors in WinCC provide the option of checking that scripts have a correct syntax without executing them. Errors in the script are displayed in the output window under the editor window. Double click on the corresponding error line to access the related point in the code.

Note:
The syntax check can only check objects known to the system at the moment of execution. Therefore, the tags and objects addressed in the scripts must be created in WinCC.
Only syntactically correct actions are executed in runtime.

Note:
The automation objects "PDLRuntime" and "WinCC Runtime Project" cannot be used in VBS actions.

Representation of Actions

If a syntactically incorrect action is stored, it is displayed in the Global Script navigation window with the adjacent symbol.

If a syntactically correct action without a trigger is stored, it is displayed in the Global Script navigation window with the adjacent symbol.

If a syntactically correct action with a trigger is stored, it is displayed in the Global Script navigation window with the adjacent symbol.

Note:
Actions can only be saved in the Graphics Designer if they have the correct syntax. If an action with errors should still be retained and exited, however, enter the comments.

6.9 Diagnostics

Introduction

If the scripts are executed and tested in runtime, the Diagnostics window can be used to display an analysis quickly.

Diagnostics Tools

WinCC provides a range of tools with which to analyze the behavior of actions in runtime:

- The GSC Runtime and GSC Diagnostics application windows
- Use of a debugger

GSC Runtime and GCS Diagnostics

The GSC Runtime and GSC Diagnostics application window are used by inserting them in a process screen. This can be a process screen developed for diagnostic purposes which is called in runtime.

The application windows are used for different strategies:

While runtime is active, GSC Runtime provides information on the dynamic behavior of all (Global Script) actions, enables the individual startup as well as log on and off of each individual action and offers the access point to the Global Script Editor.

GSC Diagnostics issues the chronological sequence of calls of the trace methods contained in the actions. This also applies to trace methods in procedures which were called in actions. The targeted use of trace methods, e.g. to issue tag values, enables the progress of action and the procedures called in them to be traced. The trace methods are specified in the form "HMIRuntime.Trace(<output>)".

The trace output from C and VBS is displayed in the GSC Diagnostics window.

Debugger

In order to test the scripts in runtime, a debugger can be used instead of the Diagnostics window. The scope of material supplied in WinCC contains the Microsoft Script Debugger; its use is described under the help topic "Testing with the Debugger".

6.10 Structure of VBScript Files

Principle

In order not to hinder the simultaneous processing of cyclic and event-driven scripts in the graphical runtime system, the event-driven actions and cyclic/tag-driven actions are strictly separated during processing. In this way, a cyclic action, for example, cannot hinder the execution of an action initiated by clicking a button.

To ensure this, the event-driven actions and the cyclic/tag-driven actions are stored in separate script files when saving a picture. If a global picture section has been defined in actions in Graphics Designer, this is copied into both scripts. In the same way, modules which are used in an action are also copied in both script files.

If a tag from a module should be used, the corresponding module must be called in. Otherwise, the module is not copied in the script file and an error is generated.

Note:
Since the two script files are handled separately, they have no common data area. Therefore, there is no synchronization of global tags between the two script files. If synchronization is necessary, it must be realized via WinCC tags.

Structure of the Script Files

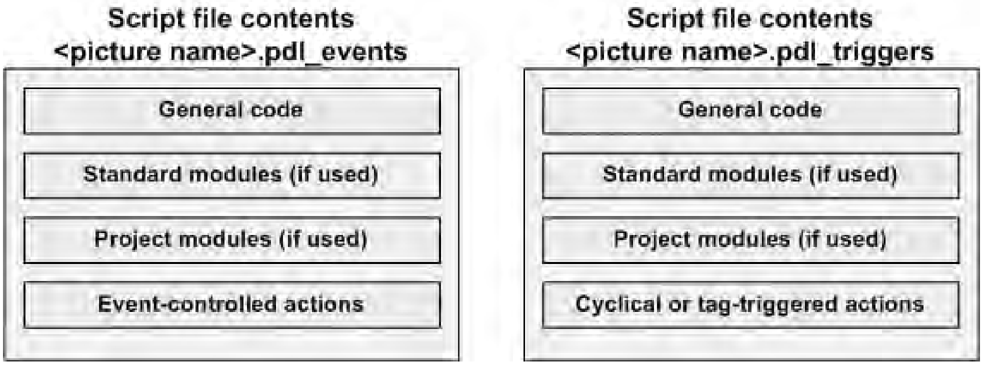
When scripts are debugged with a debugger, the script files always open the different runtime systems.

In the case of the graphical runtime system, this means that you receive two script files per picture:

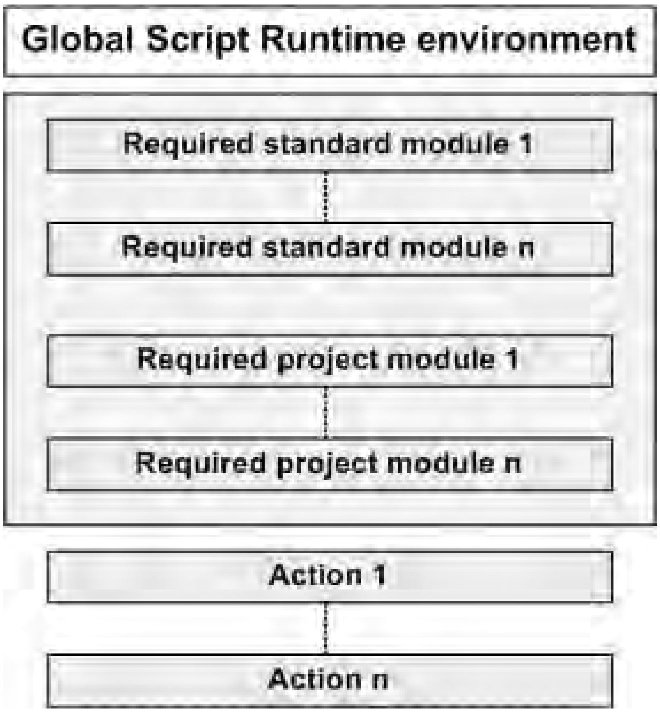
- <Picture name>.pdl_events: Contains the event-driven actions.
- <Picture name>.pdl_triggers: Contains the cyclic and tag-driven actions.

The following section describes how the script files are structured:

Graphical runtime system


Script file contents
<picture name>.pdl_events
General code
Standard modules (if used)
Project modules (if used)
Event-controlled actions
Script file contents
<picture name>.pdl_triggers
General code
Standard modules (if used)
Project modules (if used)
Cyclical or tag-triggered actions

Global Script Runtime


Global Script Runtime environment
Required standard module 1
Required standard module n
Required project module 1
Required project module n
Action 1
Action n

Note:
Please note that the actions and procedures of the graphic runtime system are not displayed with the action name in the script file under which it was saved in WinCC. The name conventions for actions and procedures in the script files are described in "Action and Procedure Names in the Debugger".

7 ANSI-C for Creating Functions and Actions

7.1 Project Function - Features

Project Function - Features

Project functions ...

- you create them yourself
- you can change them
- can be protected against modification and viewing by unauthorized persons through the use of a password
- have no trigger
- are only known within the project
- have file names with the structure "*.fct"

Project functions are saved in the subdirectory "\library" for the WinCC project.

Project Function - Usage

Project functions can be used...

- in other project functions
- in Global Script actions
- in Graphics Designer in C-Actions and within the Dynamic Dialog
- in Alarm Logging within the Loop in Alarm functionality
- in Tag Logging when starting and unlocking archives and when swapping-out cyclic archives

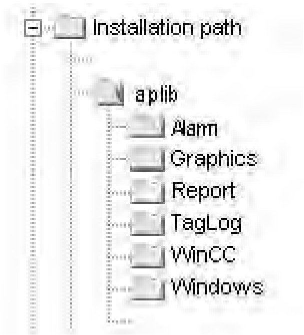
7.2 Standard Functions - Characteristics

Standard functions - characteristics

Standard functions ...

- are included in WinCC
- can also be created by you
- can be modified by you
- can be protected with a password against modification or viewing by unauthorized parties
- have no trigger
- are registered in all projects
- have file names in the form of "*.fct"

Standard functions are saved in WInCC's installation directory in the sub-directory "\aplib".



Standard functions - usage

Standard functions can be used ...

- in project functions
- in other standard functions
- in Global Script actions
- in Graphics Designer in C actions and within the Dynamic Dialog
- in Alarm Logging within the Loop in Alarm functionality
- in Tag Logging when starting and releasing archives and when backing up short-term archives

Note:
The included standard functions can be modified by the user. In the event that WinCC is reinstalled or upgraded, the standard functions, which were modified, will be deleted or replaced by the included standard functions. Therefore, you should save the modified functions before hand.

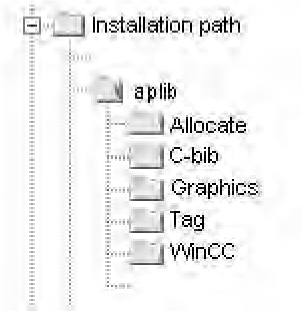
7.3 Internal Functions - Features

Internal Functions - Features

Internal functions ...

- are included in WinCC
- you can**not** create them
- you can**not** change them
- they can**not** be renamed
- have no trigger
- are know project-wide
- have file names with the structure "*.icf"

Internal functions are saved in the subdirectories of "\aplib" in the WinCC installation directory.



Internal Functions - Usage

Internal functions can be used...

- in project functions
- in standard functions
- in actions
- in Graphics Designer in C-Actions and within the Dynamic Dialog

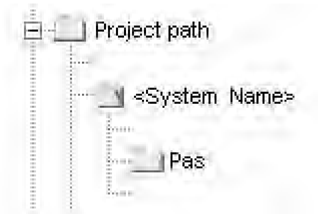
7.4 Local Actions - Features

Local Actions - Features

Local actions ...

- you create them yourself
- you can change them
- can be protected against modification and viewing by unauthorized persons through the use of a password
- have at least one trigger
- are only executed on the assigned computer
- have file names with the structure "*.pas"

Local actions are saved in the subdirectory "\<ComputerName>\Pas" in the project directory.



Local Actions - Usage

Actions are used for picture-independent background tasks, such as e.g. printing a daily report, monitoring tags or performing calculations. An action will be started by the trigger configured for it. In order for an action to be executed, Global Script Runtime must be included in the startup list.

In contrast to global actions, local actions can be assigned to a computer. In this manner, it is possible, for example, to ensure that a report will only be printed on the Server.

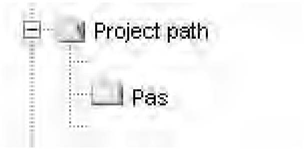
7.5 Global Actions - Features

Global Actions - Features

Global Actions ...

- you create them yourself
- you can change them
- can be protected against modification and viewing by unauthorized persons through the use of a password
- have at least one trigger to start them
- are executed on all of the project computers in a Client-Server project
- have file names with the structure "*.pas"

Global Actions are saved in the WinCC project's subdirectory "\Pas".



Global Actions - Usage

Actions are used for background tasks, such as e.g. printing a daily report, monitoring tags or performing calculations. An action will be started by the trigger configured for it. In order for an action to be executed, Global Script Runtime must be included in the startup list.

In contrast to local actions, global actions are executed on all of the project computers in a Client-Server project. In a single user project there is no difference between global and local actions.

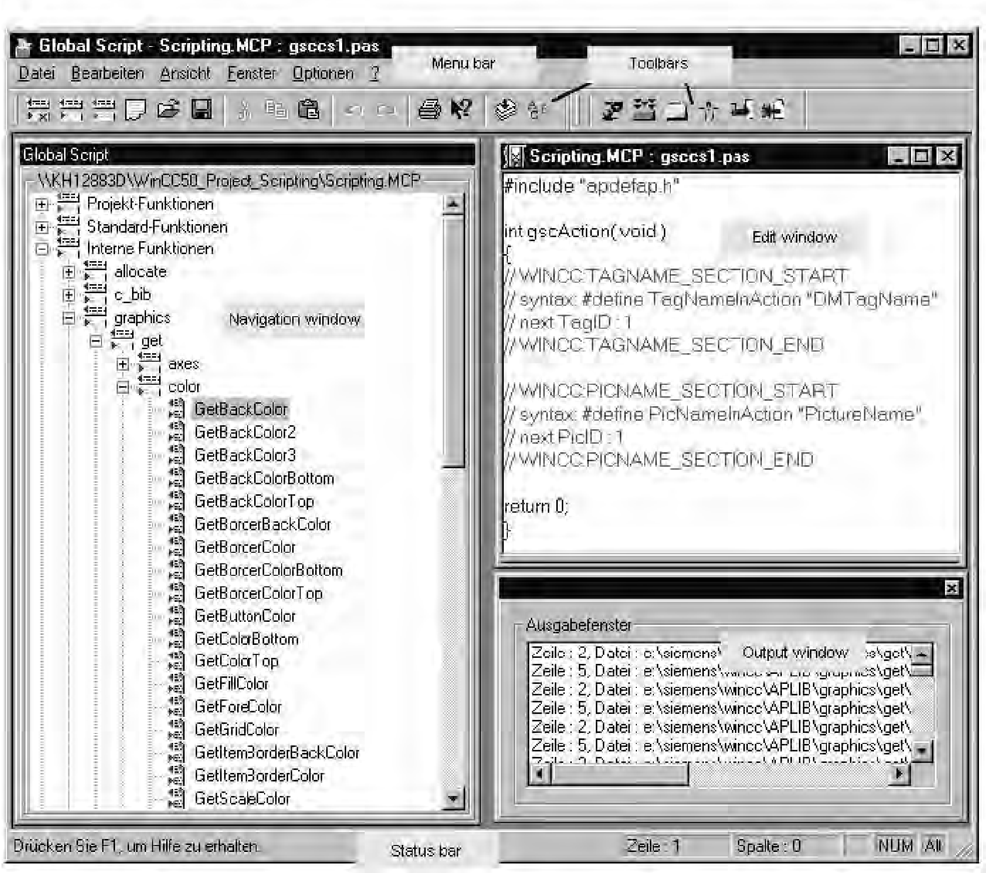
7.6 The Global Script Editor

Introduction

WinCC supports the creation and editing of functions and actions with the Global Script editor. Global Script is started from the project window of WinCC Explorer.

The Global Script Editor's Design

The Global Script editor is designed in accordance with the Windows standards. It has toolbars, a menu bar and a status bar. It has several windows that have pull-down menus.



```
#include "apdefap.h"

int gscAction( void )
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

return 0;
}
```

```
Zeile : 2, Datei : e:\siemens\
Zeile : 5, Datei : e:\siemens\
Zeile : 2, Datei : e:\siemens\wincc\APLIB\graphics\get\
Zeile : 5, Datei : e:\siemens\wincc\APLIB\graphics\get\
Zeile : 2, Datei : e:\siemens\wincc\APLIB\graphics\get\
Zeile : 5, Datei : e:\siemens\wincc\APLIB\graphics\get\
```

The Global Script Editor

The Navigation Window

The navigation window is used to select functions and actions to be edited or to be inserted at the location of the cursor in an editing window.

The functions and actions are organized in a hierarchical system of groups.

Functions will be displayed by function name. In the case of an action, the file name will be displayed.

The Editing Window

Functions and actions are written and edited in an editing window. It will only be visible when a function or action has been opened to be edited. Each function or action will be opened in its own editing window. Several editing windows can be open at the same time.

The Output Window

The results of the functions "Find in files" or "Compile All Functions" will be displayed in the output window. It is normally visible, but can be hidden.

Find in files:

The results of the search will be displayed in the output window as one line per search term found. Each line has a line number and shows the path and file name plus the line number and text of the line in which the search term was found.

By double-clicking on a line displayed in the output window, you can open the associated file directly. The cursor will be placed in the line in which the search term was found.

Compile All Functions:

Any warnings and error messages, which are returned by the compiler, will be output for each function that is compiled. The line following will display the path and file name of the compiled function as well as the compiler's summarizing message.

The Menu Bar

The contents of the menu bar depend on the situation. It is always visible.

The Toolbars

Global Script has two toolbars. They can be made visible as needed and dragged to any place on the screen with the mouse.

The Status Bar

The status bar is located at the lower edge of the Global Script window and can be displayed or hidden. It shows information about the position of the cursor in the editing window and the keyboard setting. In addition, the status bar will show either a quick info for the currently selected Global Script functionality or a tip.

7.7 Creating and Editing Functions

Introduction

A distinction is made between project, standard and internal functions. WinCC is delivered with a with a broad selection of standard and internal functions. Furthermore, you can create your own project and standard functions or modify standard functions. Please remember, however, that the standard functions included with WinCC will be overwritten when WinCC is reinstalled so any modifications will be lost.

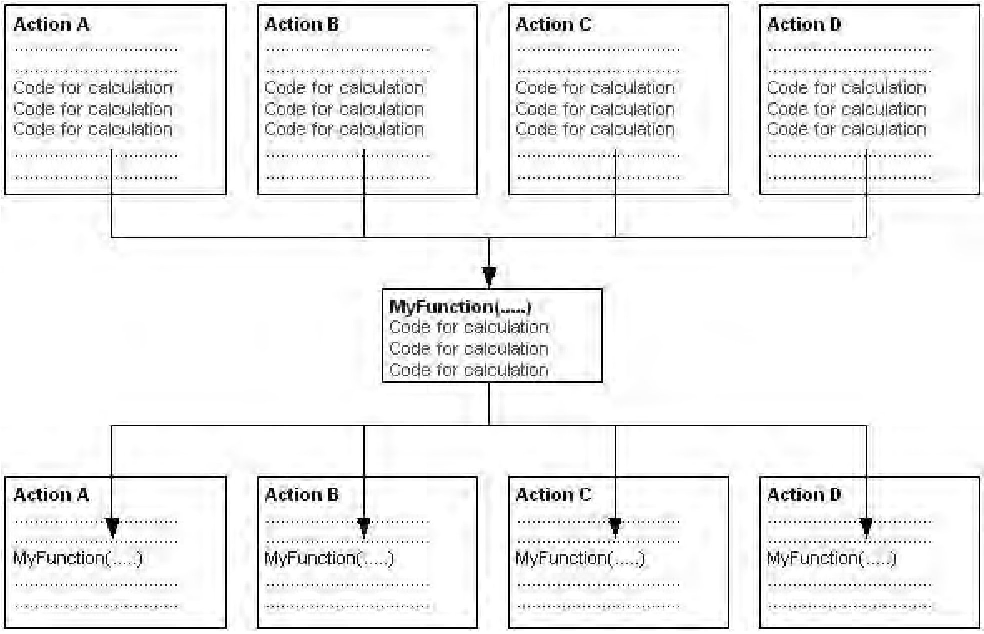
Internal functions cannot be created or modified.

Functions - usage

If the same calculation must be performed - with different starting values - in several actions, it would be to your advantage to program a function to perform this calculation. Afterwards, in the actions, you can simply call this function with the current parameters.

This has several advantages:

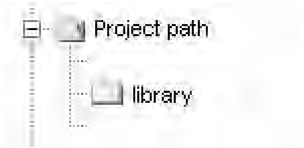
- The calculation is only programmed once.
- Later changes need only be made in one place, namely in the function and not in every action.
- The action code will be shorter and clearer.



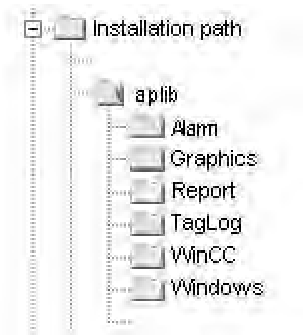
Finding functions

To access existing functions or create new ones, you can either use the "File" menu in Global Script's Navigation window or click on the corresponding button in the toolbar.

Functions are saved in the file system as follows:



Project Functions



- Installation path
  - aplib
    - Alarm
    - Graphics
    - Report
    - TagLog
    - WinCC
    - Windows

Standard Functions

Editing and compiling actions

A function is edited and compiled in its own Editing window. Once the compiler is done, its messages will be displayed in the Editing window. These messages can be warnings and/or error messages. The total number of warnings and error messages will displayed in each case.

What happens, when I rename a function?

In the Navigation window, functions are always displayed with their function names not their Windows file names. If you change the name of a function in the Editing window and then compile it, the name displayed in the Navigation window will no longer match the function name. This will be indicated in the Navigation window by prefixing the name with an asterisk "*". As soon as you save the function, the current function name will be displayed in the Navigation window.

These points must be considered when saving functions.

If you save a function that has not been properly compiled, the symbol will appear in the Navigation window.

If you save a function that has been compiled without errors, the symbol will appear in the Navigation window.

7.8 Creating and Editing Actions

Introduction

A distinction is made between global and local actions. In a client-server project, global actions can be performed on all of the project's systems, while local actions can only be performed on the assigned system.

A global action can be used to e.g. perform a calculation on all of the systems in the project.

A local action can be used to e.g. print a report on a server.

The creation and editing of both types of actions is completely identical.

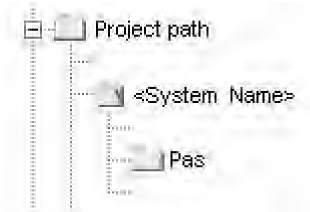
The differences between actions and functions

- In contrast to functions, actions can have a trigger. That means that a function can never be started by itself in Runtime.
- Actions can be exported and imported.
- Actions can be assigned an authorization. The authorization controls the operation of the action in the Diagnostics window in Global Script - Runtime.
- An action has no parameters.

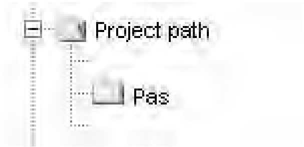
Finding actions

To access existing actions or create new ones, use Global Script's Navigation window.

Actions are saved in the file system as follows:



Local actions



Global actions

Editing and compiling actions

An action is edited and compiled in its own Editing window. Once the compiler is done, its messages will be displayed in the Editing window. These messages can be warnings and/or error messages. The total number of warnings and error messages will displayed in each case.

Representation of Actions

If a syntactically incorrect action is saved, it is displayed by Global Script in the Navigation window with the adjacent symbol.

If a syntactically correct action without a trigger is saved, it is displayed by Global Script in the Navigation window with the adjacent symbol.

If a syntactically correct action with a trigger is saved, it is displayed by Global Script in the Navigation window with the adjacent symbol.

The following must be considered when creating actions

With CrossReference, WinCC offers the option of creating cross-reference lists. To ensure that the software can recognize the variables and pictures used in function calls made within actions, the coding rules given later should be observed.

Renaming actions

In the Navigation window, actions are always displayed with their file names. Therefore, renaming an action implies renaming the file containing the action's code.

System behavior, when actions are modified, deleted or saved in Runtime

If a local action is saved during Runtime, all of the local and global actions on the associated system will be reset.

If a global action is saved during Runtime, all of the local and global actions of the entire project and thus on all of the systems will be reset.

Static variables, which are used in the reset actions, will be reinitialized.

Possible causes, if an action is not performed during Runtime

If an action cannot be performed during Runtime, it may be due to one of the following reasons:

- The action does not have a trigger.
- The action was not compiled.
- Global Script Runtime is not activated in the project's startup list.

Note:
Before creating an action, you should check whether this functionality cannot be implemented in PLC.

7.9 The Runtime Behavior of Actions

WinCC has three tools for analyzing the runtime behavior of actions. These are the application windows GSC Runtime and GSC Diagnose plus the application apdiag.exe.

To use the application windows GSC Runtime and GSC Diagnose, they must be added to a process screen. This can be a process screen made especially for diagnostic purposes. It will be called in Runtime.

With these application windows, the following differing strategies are pursued:

- GSC Runtime supplies information about the dynamic behavior of all of the (Global Script) actions, supports the individual start and the Start and End Action for each individual action and provides an entry point into Global Script Editor, while Runtime is active.
- GSC Diagnose outputs the printf instructions (contained in the actions) in the order in which they are called. That also applies for the printf instructions in functions, which are called in actions. Through a well thought out use of printf instructions, for example to output the values of variables, it is possible to follow the action's flow and that of the called functions. Even error conditions, which result in a call to the OnErrorExecute function, will be displayed in the GSC Diagnose window.

8 Setting Up an Alarm System

Alarm Logging Tasks

Messages provide the operator with information relating to the operating status and the fault status of the process. They inform the operator of critical situations at a very early stage and help to eliminate idle time.

During configuration, you define the events in your process which should trigger messages. An event, for example, may be the setting of a specific bit in an automation system or when a process value exceeds a predefined limit value.

The components of Alarm Logging

Alarm Logging is made up of Configuration and Runtime components:

- Alarm Logging is the configuration component of the message system. Alarm Logging is used to determine when the respective messages should appear and which content they should have. The Graphic Designer also disposes of a special display object, the WinCC Alarm Control, which is used to display messages.
- Alarm Logging Runtime is the runtime component of the message system. When in runtime, Alarm Logging Runtime is responsible for executing the defined monitoring tasks. It also controls the message output operations and administers the acknowledgement of these messages.

The messages are displayed in WinCC Alarm Control in tabular form.

8.1 Functionality

The editor "Alarm Logging" is responsible for the acquisition and archiving of messages, and includes functions for the acquisition of messages from processes, for their preparation, their representation, their acknowledgement, and their archiving.

Alarm Logging

- provides comprehensive information about error and operating states.
- is used for the early detection of critical situations.
- avoids and reduces downtime.
- increases quality.
- provides targeted documentation of error and operating states.

The alarm logging processes results from functions which monitor actions in the process, on the automation level and in the WinCC System. The recorded message events are reported by the system both visually and acoustically. The events are also archived electronically and on paper. The messages can be accessed individually. Supplemental information to individual messages ensures rapid error localization and rectification.

Alarm Logging is divided into two components: the Configuration System and the Runtime System.

Tasks of Alarm Logging Runtime System

Use the Alarm Logging Configuration System (Alarm Logging CS, ALGCS) to configure your messages so that they are displayed in runtime the way you want them to be.

The configuration of the alarm logging can be simplified by the use of the wizards.

- The System Wizard supports you during the setup of the main settings of the alarm logging
- The Linking Wizard facilitates the assignment of process tags to the messages.
- In addition, you can use various projecting dialogues, which enable you to process several messages at the same time.

Tasks of Alarm Logging Runtime System

The task of the Alarm Logging Runtime System (Alarm Logging RT, ALGRT) is to acquire the messages and accept the acknowledgements. It prepares the messages for display and archiving.

8.1.1 Definition of Terms

Archiving

Depending on the message class, changes which occur in message states are written to configurable archives.

- For short-term archives, the size of the archive is defined by the number of messages to be archived. A short-term archive can only be created as a continuous archive. If the maximum number of archived messages has been reached, the oldest messages are overwritten. Up to 10000 message status changes can be stored in a short-term archive.
  Messages which are stored in a short-term archive are displayed in a message window.
- The long-term archiving is don in the message archive. In order to do so, the different parameters such as archive size, the time range, the switchover time etc. are set. If any of the configured criteria is exceeded, the oldest messages in the archive are overwritten. The backup of the archiving data banks can be set with the help of additional settings.
  Messages which are stored in a message archive are displayed in a long-term archive window.

Operator input message

In the case of an operation in the process, for instance, message acknowledgement, an operator input message can be triggered, if necessary.

The configuration of an operator input message depends on the place of the operation:

- An operator input message with the properties of this object can be activated in the Graphics Designer for the operation of an object, , for example, the input of a value into an I/O field. This is done by setting the attribute “Operator Input Message“. The structure of this message cannot be edited.
  By unlocking the attribute „Operator Activities Report“, the user can enter the reason for the operation as a comment on the message after an operation has been done on a dialog. The comment "Operator Activities Report" can be displayed in the Long-Term Archive Window above the command button "Comment".
- An Operator Input Message is configured in the WinCC Alarm Logging for the operation of a message, , for example, acknowledgement locking/releasing.
  In order to do so, the WinCC system messages must be created in the Alarm Logging Editor.
  The trigger events for an Operator Input Message are selected in the WinCC Alarm Control on the General tab on the command button "Operator Input Message" and the Operator Input Message is activated.

An Operator Input Message can be given by stating the registered user and the current computer. In order to do so, the system blocks "User name" and "Computer name" can be entered into the login line of a message in the Alarm Logging Editor and in the Alarm Control.
In order to display/enter a comment about a selected Operator Input Message, a dialog can be opened in the Long-Term Archive Window by using the symbol "Comment". If the system block "Comment" in the Alarm Control is entered into the message line, the message block is displaying the presence of a comment about this message.

Single Message, Group Message

In Alarm Logging there are two different message forms: Single messages and Group messages

- With single messages, every event is assigned a message.
- Group messages are used to summarize multiple single messages. As a result, the events linked to the single messages trigger only the common group message. The single message which triggered the group message cannot be determined.
  One group message can be set up for each message class and message type. Moreover, group messages can also be created using any single messages.

Single Acknowledgement, Group Acknowledgement

A pending message can be acknowledged in one of two ways:

- Messages which do not possess the "group acknowledgement" attribute must be individually acknowledged.
- Group acknowledgement can be used to collectively acknowledge all individual messages possessing the "group acknowledgement" attribute which are visible in a message window.

The printout of the acknowledgement of a message can be done by stating the acknowledged user and the user that was used for it. In order to do so, the system blocks "User name" and "Computer name" can be entered into the login line of a message and in the WinCC Alarm Control. When acknowledging a message, the user and computer names are not displayed in the message line of the incoming messages, but in the Operator Input Message of the acknowledgement in the short-term and the long-term window. The Operator Input Message in the WinCC Alarm Control on the tab General/ command button "Operator Input Message" must be enabled.

Events

With events, a distinction is made between binary events and monitoring events:

- Binary events are changes in the state of internal or external tags.
- Monitoring events are not directly supported by Alarm Logging. Among the monitoring events are: Overflow of archives and hard disks, printer messages, server failure, trouble in process communication.

Initial Value Message, New Value Message

- The term initial value message is used to describe a form of message processing in which the first message in a list of messages to undergo a state change since the last acknowledgement is highlighted.
- The term new value message is used to describe a form of message processing in which the first messages in a list of messages to undergo a state change since the last acknowledgement are highlighted.

Messages

The purpose of a message system is to chronologically signal and archive events at a central point which occur sporadically during the process. A message can be caused by an event or a message frame.

- In general, a distinction is made between event messages, error messages, and system messages. Operating messages are used for displaying a state in the process. Error messages are used for displaying an error in the process. System messages are used for displaying error messages from other applications.
- In Alarm Logging, messages with similar behavior (acknowledgement philosophy, color of message states) can be grouped into message classes and message types.

Message Types, Message Classes

- Message classes differ from one another in acknowledgement philosophy. Messages with the same acknowledgement philosophy can be integrated in a single message class. In Alarm Logging, the message classes "Error", "System, requires acknowledgement", and "System, without acknowledgement" are preconfigured. You can define up to 16 message classes in WinCC.
- Message types are subgroups of the message classes and can differ in the coloring of the message states. Up to 16 message types can be created in WinCC for each message class.

Message Blocks

The change in the state of a message is displayed during runtime in a message line. The information to be displayed in the message line is defined using message blocks. There are three different types:

- System blocks (, for example, date, time, duration, comments, ...) facilitate the specification of information which is predefined and cannot be used freely. With system blocks, the value of the message block (, for example, the time) is displayed in the message line.
- User text blocks make it possible for you to assign a message up to ten, different, freely definable texts. With user text blocks, the content of the message block - the text defined by you - is displayed in the message line. In the message text of a user text block a process value can be displayed the output format of which can be defined.
- Using process value blocks, you can display the value of tags in the message line. The format that is used during this process cannot be defined freely. In the case of process value blocks, the contents of the message block, the value of the tags it has set, will be displayed in the message line.

Message Event, Message State

- Message events are the "Coming", "Going", and "Acknowledging" of messages. All message events are stored in the message archive.
- Message states are the states which are possible for a message: "Came In", "Went Out", "Acknowledged".

Message Window

In runtime, the changes in message states are output in a message window. The appearance and operating options of the message window can be freely defined in Graphics Designer.

A message window contains in tabular form all messages which have yet to be displayed. Each message to be displayed is output in its own line, the message line.

By using definable filters you can affect the contents of the message window or sort it according to message blocks.

Depending on the source of the messages displayed in the message window, a distinction is made between three types of message windows.

- Message lists are used for displaying currently pending messages.
- Short-term archive windows are used for displaying messages which are stored in a short-term archive.
- Long-term archive windows are used for displaying messages which are stored in a message archive.

Message Frames

Message frames originate from either the process or from the process-control monitoring system. They are exchanged by the control system (or the monitoring application) and Alarm Logging.

Message Tag

In the bit message procedure, the control system signals the occurence of an event in the process via the message tag. Several messages can be masked via one message tag.

Message Procedure

The alarm logging supports several message procedures: The bit message procedure, the procedure of the chronological reporting as well as the procedure of the analog alarms about the limit value control.

- In the bit message procedure, the control system signals the occurence of an event via the message tag. The time stamp (date and time) of the message is made by Alarm Logging.
- With chronologically correct messages, the control system transmits a telegram with the message data when the event occurs. This is analyzed by Alarm Logging. The time stamp (date and time) of the message is made by the control.
  If the procedure "Acknowledgement triggered messaging" (QTM) is enabled on an AS S7-400, not signal change is reported to the OS. According to the QTM procedure, in the case of messages that occur very frequently and are caused by wavering signalers, a new message about a signal change is sent from the AS only after the positive signal change (incoming message) that was reported last was acknowledged in the OS.
- With analog alarms, the course of an analog tag can be monitored for violation of the upper and lower limit values.

Message Line

In a message window, each message is displayed in its own message line. The contents of the message line are dependent on the message blocks which are to be displayed. With system message blocks (, for example, date, time), the value of the message block is displayed; with process and user text blocks, the contents (, for example, the text defined by you) are displayed.

Reports

There are three types of message reports available in Alarm Logging.

- With message sequence reports, all state changes (came in, went out, acknowledged) from all currently pending messages are output to a printer.
- The online message report is used to print the messages from the alarm control, which have been sorted chronologically, on a printer with an opened process message window by using the button "Print".
- With archive reports, all changes in message states stored in an archive are output to a printer.

Acknowledgement Philosophy

Acknowledgement philosophy is the way in which a message is to be displayed and processed from the time it "came in" to the time it "went out". In Alarm Logging, the following acknowledgement philosophies can be realized:

- Single message without acknowledgement
- Single message with "coming" acknowledgement
- Single message with "going" acknowledgement
- Initial value message with single acknowledgement
- New value message with single acknowledgement
- New value message with double acknowledgement
- Message without "Went Out" state without acknowledgement
- Message without "Went Out" state with acknowledgement

Acknowledgement Tag

In the acknowledgement tags, the "acknowledgement state" of a message is stored. A central signaling device can also be controlled using the acknowledgement tag.

Unlocking, locking of messages

In order to reduce the number of message events, known and repeating messages can be locked and enabled again. A distinction is made between the active and passive locking/unlocking of messages. For an active locking, the message source must support the locking/unlocking of messages with confirmation and a valid date/time log. In addition, a general query of the source must deliver the currently locked messages. If the message source fulfills these requirements, the messages are locked/unlocked active, otherwise the messages of WinCC are locked/unlocked passive.

- In the case of active locking, a disabling request is sent to the message source (, for example, AS). The message is disabled in WinCC only after the source replies the message as locked. The locking of the messages is done analogue. Only the messages that have been projected chronologically on the AS level are locked/unlocked active.
- In the case of passive locking, the message is locked/unlocked in the alarm server of WinCC. The message source is not included.

Lock List

In the lock list, all locked messages that are delivered in the system are shown. The structure corresponds to that of the message window. Locked messages can be unlocked by a button in the toolbar.

Status Tag

The "Came In / Went Out" status of the message type and an identifier for messages to be acknowledged are stored in the status tag.

8.1.2 Structure of a Message

Messages in runtime are displayed in the form of rows of a table. Here, an individual message is composed of information which can be displayed in the fields of the table. These individual information are called Message Blocks.

The message blocks can be divided into three areas.

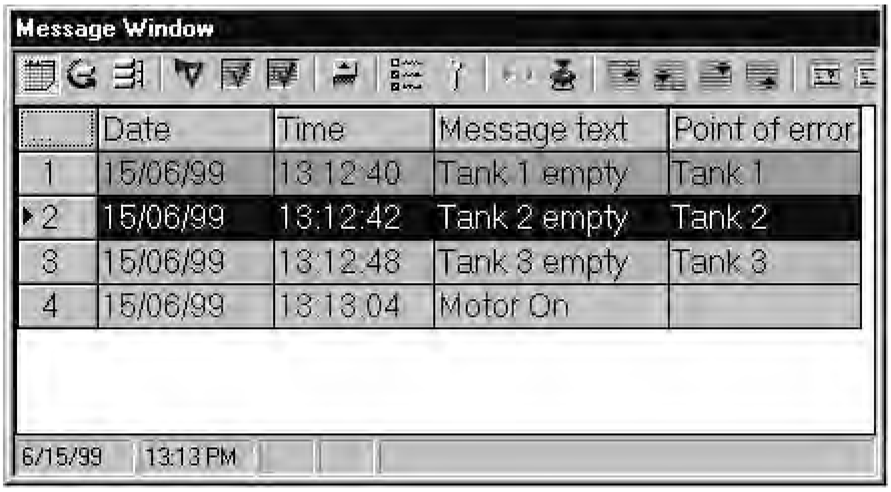
- System blocks
  It includes system data, which are provided by the alarm logging. These data include date, time, report ID, etc.
- Process value blocks
  includes values that have been provided by the process, , for example, critical fill level, temperatures etc.
- User Text Blocks
  Text that contribute to the general information and comprehensibility, for example, explanations about messages, place of error, message cause etc.

Whereas the contents of the system blocks are fixed, the contents of the process value blocks and user text blocks can be modified by the user. Selections in system blocks affect only their display; their reporting is not affected.

8.1.3 Displaying Messages in Runtime

In runtime, the events coming from the controls or process control system are detected by Alarm Logging, analyzed and displayed in a message window. The appearance and operating options of the message window can be freely defined in Graphics Designer.

Message Window as an ActiveX Control

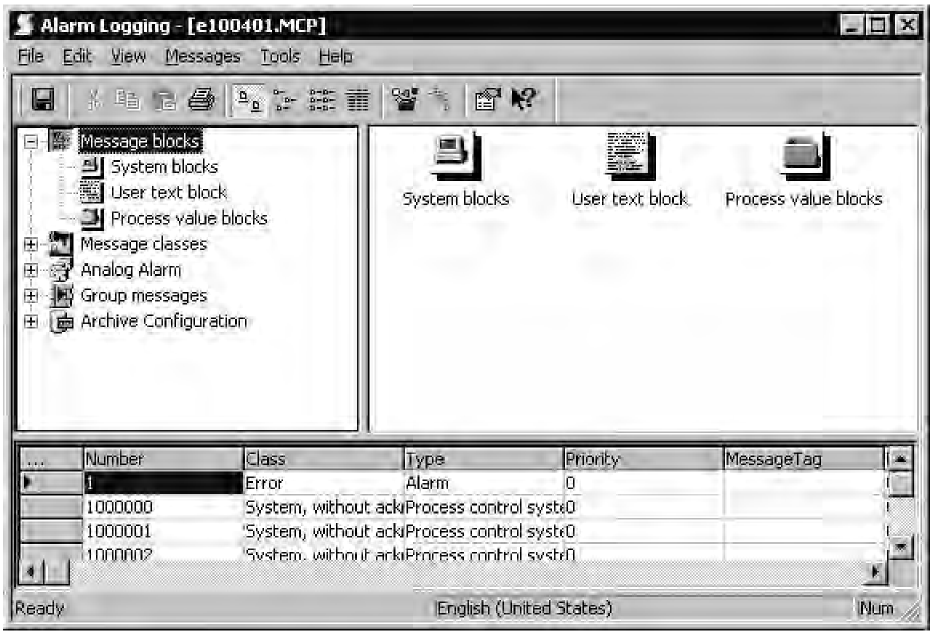


	Date	Time	Message text	Point of error
1	15/06/99	13:12:40	Tank 1 empty	Tank 1
2	15/06/99	13:12:42	Tank 2 empty	Tank 2
3	15/06/99	13:12:48	Tank 3 empty	Tank 3
4	15/06/99	13:13:04	Motor On	

6/15/99 13:13 PM

The message window that is used as an ActiveX Control can be distinguished by the simple configuration of the messages. All information to be displayed can be set at the object, i.e. in Graphics Designer.

8.2 Structure of the Configuration System



...	Number	Class	Type	Priority	MessageTag
▶	1	Error	Alarm	0	
	1000000	System, without ack	Process control syst	0	
	1000001	System, without ack	Process control syst	0	
	1000002	System, without ack	Process control syst	0	

After you open the Alarm Logging configuration system, a window that is divided into four sections appears.

The top border of the window contains the menu bar. It contains the drop-down menus.

The toolbar is directly under the menu bar. Functions which are used often are loaded as icons on this bar.

The Project window is divided into the Navigation, Data and Table window.

You can change the size of the individual window sections. The window is limited at the bottom by the Status bar, where information about the current Operation are displayed.

8.3 Configuring a Message System

This chapter explains how to:

- configure messages and message classes
- import and export single messages
- connect messages to the process
- configure the WinCC Alarm Control system for displaying messages in runtime.

How to Construct a Message System:

1. Use the System Wizard to create the basic structure of the message system.

   The System Wizard

2. Configure the message blocks in accordance with your requirements.

   Configure Message Blocks

3. Configure the message system.

   Configure message classes

   Configure Message Types

   Configure Single messages

   Configure Group messages

4. Set the type of data archiving.

   Configure Archives

5. Configure the WinCC Alarm Control System in "Graphics Designer" for displaying the messages in runtime.

   The WinCC Alarm Control.

Note:

Use the available Wizards and configuration dialogs to completely create the messages.

8.3.1 Wizards

System Wizard

The Wizard is used for preconfiguring the message system. Frequently used message blocks, message types and message archives are created during this preconfiguration process. You can modify the objects created by the Wizard at a later time to meet your requirements.

The System Wizard always generates the messages in German, English, and French. This is independent of whether the languages are installed or whether the languages were set up in the text library. After the Message Wizard has run, you can delete the languages you do not desire from the text library.

How to Configure a Message System with the System Wizard:

1. Select the "Select Wizard" command from the File menu, then select the System Wizard from dialog which opens. The Wizard is started after you click on "OK" to exit the dialog.
2. Following the introductory picture, you can use the "Select Message Blocks" dialog to specify which message blocks are to be created by the System Wizard.
3. Use the "Preset Classes" dialog to create two message classes with different acknowledgement philosophies with the corresponding message types.
4. In the "Final!" dialog, you obtain a summary of the message blocks and message classes to be created by the Wizard.

Note:

All settings changed after the System Wizard has run will be lost if the Wizard is run again.

Linking Wizard

dThe Linking Wizard provides support for connecting the message and acknowledgement tags of messages with the tags of the WinCC Explorer. The tag connections generated by the Wizard can be changed later in the table window.

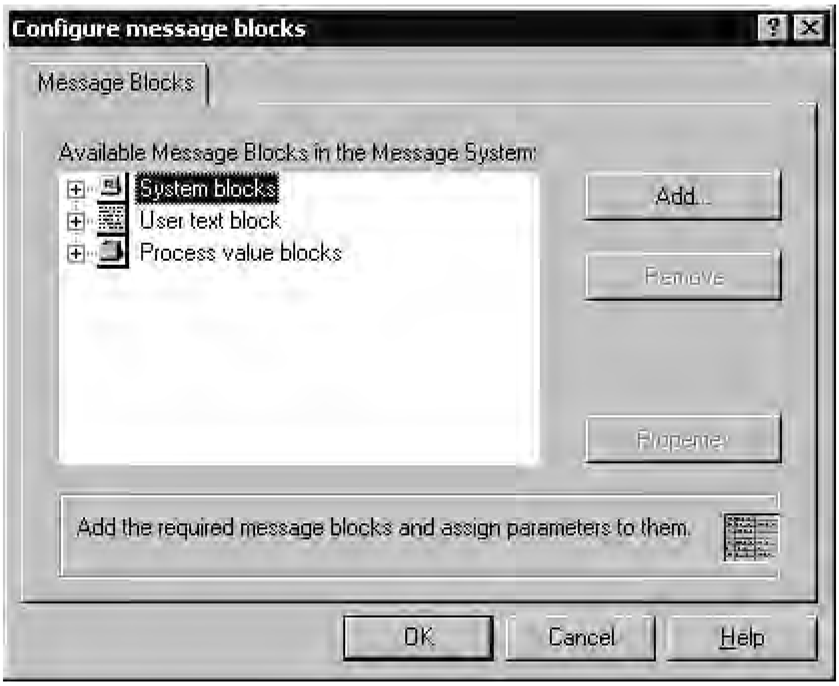
How to Create Links with the Linking Wizard:

1. Open the "Messages" menu.
2. Select "Start Interconnection Wizard...."
3. In the dialog box "Link tags: Message tag" use the button ... to select a tag, which you want to use as a message tag.
4. Enter the numbers of the messages to which you would like to link the specified message tag.
5. In the dialog box "Link tags: Acknowledgment Tag" the button ... to select a tag, which you want to use as a message tag.
6. Enter the numbers of the messages to which you would like to link the specified acknowledgement tag.
7. When the Wizard is ended, the selected tags are interconnected to the messages.

8.3.2 Message Blocks

Message blocks, which are displayed as columns of a table, are used for constructing messages. Specific data are entered in each block. Message blocks can be created or edited for the three different block types:

- System Blocks
- User Text Blocks
- Process Value Blocks



In the "Available Message Blocks" area, the structure of the message blocks is displayed. You can work in this window as you would in the project navigation window. There are, however, no pop-up menus available.

Button	Description
Add	A dialog box is opened in which message blocks can be added. Depending on whether system message blocks, user text blocks or process value blocks were selected, a corresponding selection of message blocks is presented.
Remove	The selected message block is removed.
Characteristics	The properties dialog for the selected message block is opened. The parameters of the message block can be changed.

General properties of Message Blocks

A message that requires acknowledgement or individual Message Blocks of it can be displayed flashing in the message window in runtime, if the following conditions are provided:

- The property "Flashing On" must be activated in the message class which the message is assigned to. (Dialog "Configure message class"; Tab "Acknowledgment")
- "Flashing On" must be enabled under Properties in the message block which will be flashing as well (Dialog message blocks). If several or all Message Blocks one message should have this property, it must be enabled at each message block of this message.

8.3.2.1 System Blocks

The message data that are internal to the system are entered in system blocks.

A list of system blocks is provided in the Appendix.

The System block list provides you with information about the individual system blocks.

The alarm logging can be modified with the help of a Selection dialog.

You can remove the system blocks or add them from a provided selection list.. When configuring, you change the properties of a system message block

Note:

You can activate the properties dialogs by double clicking the left mouse button on the appropriate object in the data window.

8.3.2.2 User text blocks

The alarm logging can be modified with the help of a Selection dialog.

You can remove the user blocks or add them from a provided selection list. When configuring, you change the properties of a system user block

Note:

The maximum length of text blocks is 255 characters.

8.3.2.3 Process value blocks

Process value blocks are used for connecting the messages to the process. Each message system has up to 10 individually configurable message blocks available for each single message.

The alarm logging can be modified with the help of a Selection dialog verändert werden.

You can remove the process value blocks or add them from a provided selection list. When configuring, you change the properties of a process value block.

Note:

The display of the process value blocks for each text in runtime is limited to 32 characters during the chronological reporting, and to 255 characters during the bit message procedure.

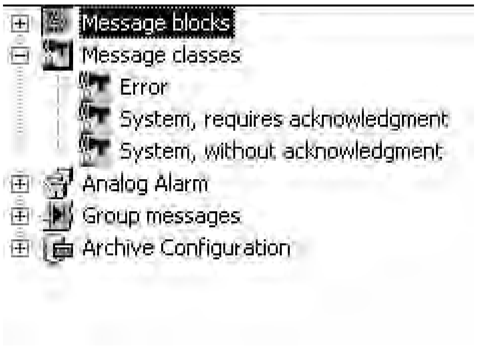
8.3.3 Message classes

For configuring messages, WinCC provides 16 message classes with 16 message types each. A single message is assigned a message type. A message type belongs to a message class. The assignment is based on system properties and attributes that are assigned to the classes and types. Single messages can be combined to form group messages.

The following message classes are offered standard in WinCC:

- Fault
- System, requires acknowledgement
- System, without acknowledgement
- sing message classes,
- the acknowledgement philosophy,
- the related status text and
- the output of acoustic/optic signals is defined for all message types of this message class.

You can edit message classes using the "Message Classes" object in the project navigation window.



Message blocks
Message classes
Error
System, requires acknowledgment
System, without acknowledgment
Analog Alarm
Group messages
Archive Configuration

You can perform the following functions with message classes:

- add
- delete
- configure

8.3.3.1 Message Types

Use the "Message Types" item to set uniform text and background colors for all messages that belong to a message type within a message class. These settings identify the different message types. This way, you can differentiate among messages in runtime even when they are in different states.

Note:

You can activate the properties dialogs by double clicking the left mouse button on the appropriate object in the data window.

8.3.3.2 System Message Classes

System messages means messages that are created in the system, e.g. messages from controls, system failures.

WinCC provides special system classes for processing system messages. You cannot expand these system classes. The acknowledgement philosophy is preset.

System Message Class That Requires Acknowledgement

As the name of the class implies, these are incoming messages that need to be acknowledged before they leave. The message disappears immediately following an acknowledgement.

You can set the parameters in the properties dialog for message classes.

The following message types are available:

- Process controlling messages
- System messages

Note:

The "Went Out" status is not registered or stored in the archive.

System Message That Does Not Require Acknowledgement

These are messages that do not require any acknowledgement. You can set the parameters in the properties dialog for message classes.

The following message types are available:

- Process controlling messages
- Operator Messages

System Message Types

WinCC Alarm Logging knows three system message types. These are

- **Process controlling messages**
  Messages created by the process control system, for example, system startup.
- **System Messages**
  Messages created by the system, for example, failure of system components.
- **Operator messages**
  Messages created during user operation, for example, operation of a component.

Note:

Please remember that a failed Operation is recorded in an Operator Input Message as well.

8.3.4 Single and Group Messages

8.3.4.1 Single Message

Single messages should be edited in the table window. A single message consists of the defined message blocks. The editable message blocks can be edited by selecting or a pop-up menu.

Multiple Single messages can be combined into a Group message.

In the table window you can

- Create Single Messages
- Copy Single Messages
- Delete Single Messages
- or change the properties of a single message. The parameter of a single message can be edited either directly in the boxes of the table window or by means of the "Properties" item in the pop-up menu.

Message Tag of a Single Message

The message is triggered by means of the message tag. The structure of the message tags is not set, meaning you have the option of using one message tag for each message or of triggering multiple messages with one message tag. The differentiation then takes place via the message bit.

Status Tag of a Single Message

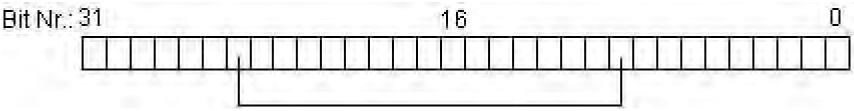
A single message always contains two deciding message states:

1. The "Came In / Went Out" status shows if the message came in or went out.
2. The Acknowledgment Status shows if the message requires acknowledgement and was not acknowledged yet.

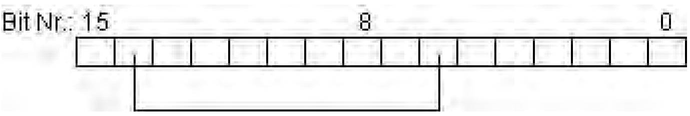
Both states of the single message are stored in a status tag. At the same time - depending on the data type of the tag, up to 16 single messages can be recorded in a status tag. Every single message occupies 2 bits in the status tag.

The position of the bit with the "Came In / Went Out" status in the status tag is marked by the *status bit*. The position of the second bit is dependent on the data type of the status tag. For "8 Bit unsigned" tags the distance to the bit with the "Came in / Went out" status is 4 bit, for "16 Bit unsigned" tags the distance is 8 bit and for "32 Bit unsigned" tags the distance is 16 Bit.

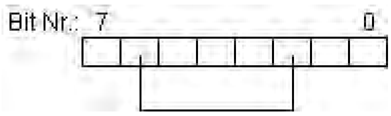
If the status tag has the data type "32-bit unsigned" and the status bit = 9, then bit 9 of the status tag marks the "Came In / Went Out" status of the single message and bit 25 indicates whether this single message has to be acknowledged. The bits 0-16; 15-31, etc. correspond accordingly in a 32-bit status tag.



If the status tag has the data type "16-bit unsigned" and the status bit = 3, then bit 3 of the status tag marks the "Came In / Went Out" status of the single message and bit 11 indicates whether this single message has to be acknowledged. The bits 0-8; 5-13, etc. correspond accordingly in a 16-bit status tag.



If the status tag has the data type "8-bit unsigned" and the status bit = 3, then bit 3 of the status tag marks the "Came In / Went Out" status of the single message and bit 7 indicates whether this single message has to be acknowledged. The bits 0-4; 2-6, etc. correspond accordingly in an "8-bit" status tag.



Acknowledgement Tag of a Single Message

In Single messages a bit of the acknowledgement tag is used to trigger the acknowledgement and display the status. If an acknowledgement bit has the value 1, the associated single message has been acknowledged. Otherwise, it has not been acknowledged.

The structure of the acknowledgement tags is not set, meaning you have the option to use one acknowledgement tag for each single message or summarize multiple single messages into one acknowledgement tag. The differentiation of the single messages then takes place via the acknowledgement bit.

At runtime, this bit is set when a single message is acknowledged.

Note:

For the acknowledgement of the single message through the WinCC Alarm Control, the acknowledgement bit that is assigned to the single message will be set.

8.3.4.2 Group message

A group message is the combination of multiple single messages to form a comprehensive message.

There are two types of group messages:

1. The summary of **all** messages of a message class: Group message from a Message Class. If a new message class is created, the corresponding group message will also be created.
2. The summary of **any** messages: Customized Group message User-defined group messages can contain single messages as well as other group messages and can be edited using the pop-up menu. A single message, though, can only be contained in a group message.

Note:

If a group message occurs in runtime, the single message which triggered the group message cannot be determined.

A group message can be assigned one of the tags which returns the message state. Use this Status tag to query the message status of the Group messages from the other WinCC components.

An acknowledgement tag can be assigned to a group message, which is used to trigger an acknowledgement of the group message.

Status Tag of a Group Message

A group message always contains two deciding message states:

1. The "Came In / Went Out" status shows whether one of the subordinate messages (it can also be another group message) came in or went out.
2. The Acknowledgment Status shows that the single message that triggers the Group message requires acknowledgement and was not acknowledged yet.

Both states of the Group message are stored in a status tag. At the same time - depending on the data type of the tag, up to 16 group messages can be recorded in a status tag. Every group message occupies 2 bits in the status tag.

Bit "Acknowledgment Status"

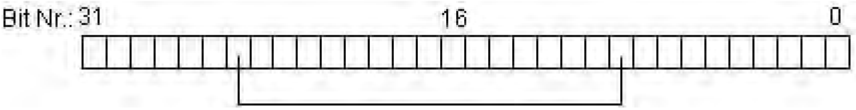
The Acknowledgment Status bit in the status tags changes to the state "1", as soon as at least one single message that belongs to the Group message and which requires acknowledgement has not arrived and not been acknowledged.

The Acknowledgment Status bit changes to the state "0", as soon as all the single messages that belong to the Group message and which require acknowledgement have been acknowledged.

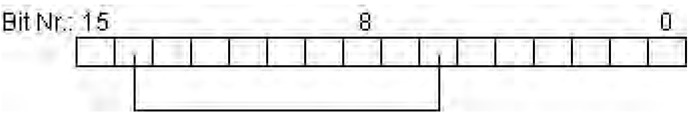
Position of the Bits "Came in / Went out Status" and "Acknowledgment Status"

The position of the bit with the "Came In / Went Out" status in the status tag is marked by the *status bit*. The position of the second bit "Acknowledgment Status" is dependent on the data type of the status tag. For "8 Bit unsigned" tags the distance to the bit with the "Came in / Went out" status is 4 bit, for "16 Bit unsigned" tags the distance is eight bit and for "32 Bit unsigned" tags the distance is 16 Bit.

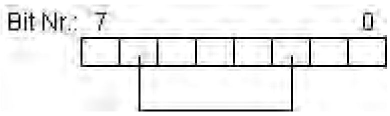
If the status tag has the data type "32-bit unsigned" and the status bit = 9, then bit 9 of the status tag marks the "Came In / Went Out" status of the group message and bit 25 indicates whether this group message contains a message that has to be acknowledged, which is not acknowledged yet. The bits 0-16; 15-31, etc. correspond accordingly in a 32-bit status tag.



If the status tag has the data type "16-bit unsigned" and the status bit = 3, then bit 3 of the status tag marks the "Came In / Went Out" status of the group message and bit 11 indicates whether this group message contains a message that has to be acknowledged, which is not acknowledged yet. The bits 0-8; 5-13, etc. correspond accordingly in a 16-bit status tag.



If the status tag has the data type "8-bit unsigned" and the status bit = 3, then bit 3 of the status tag marks the "Came In / Went Out" status of the group message and bit 7 indicates whether this group message contains a message that has to be acknowledged, which is not acknowledged yet. The bits 0-4; 2-6, etc. correspond accordingly in an "8-bit" status tag.



Acknowledgement Tag of a Group Message

The acknowledgment tag of a group message is used for the acknowledgement of all acknowledgements that belong to this group message. In the acknowledgement tag an acknowledgement tag will be set, which will serve the acknowledgement. The structure of the acknowledgement tags is not set, meaning you have the option to use one acknowledgement tag for each group message to be acknowledged or summarize multiple group messages into one acknowledgement tag. The differentiation of the group messages then takes place via the acknowledgement bit.

During runtime, the related acknowledgment bit is set for the acknowledgment of the group message. This is done by an application program such as a button in the WinCC Process image or the program in the AS. The resetting of the acknowledgment bit requires the application program as well.

Note:

When acknowledging a single message through the WinCC Alarm Control, the acknowledgement bit of the Group message (message class/ message type) that is assigned to the single message or the customized Group message is not set.

Group messages from Message Classes

If you create a new message class, a Group message is created automatically for this Message class as well. All message types within the message class are taken over into the group message.

Use the pop-up menu to change the properties of the Message classes and the Message Types in the group message independent of each other and so to assign different status, lock and acknowledgement tags.

User-Defined Group Messages

User-defined group messages allow you to build a message hierarchy fitting to your needs. You can summarize single messages, as well as other group messages, into one comprehensive message. A single message, though, can only be contained in a user-defined group message.

A maximum of six nestings of customized group messages is possible.

Use the pop-up menu of the type "Customized group message" to create customized group messages, or to display and change the properties of a new customized group message.

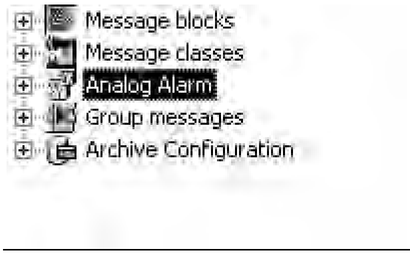
Use the pop-up menu of a user-defined group message to

- Add more group messages to an existing group message,
- Add more single messages to an existing group message, or
- to change the properties of a Group message or of a Single message in it.

8.3.5 Analog Alarm

The analog alarm allows you to set any number of limit values for a tag. If one of these limit values is violated, a message will be generated in runtime.

The analog alarm is an add-in to WinCC. Integrate "Analog Alarm" into the message system via the "Add Ins..." entry of the "Tools" menu. The "Analog Alarm" entry will then be listed in the navigation window of the message system.



Note:

Changes in the dialogs of the analog alarm will be accepted into the database tables immediately after exiting the dialog.

If the configuration takes place online, the message system must be saved before the changes will take effect in runtime.

Analog Alarm Messages

In the "Properties - Tag” and "Properties - Limit Value” dialogs, the number of the message which is to be generated if a limit value violation occurs can be set.

- If you enter a number of a message that does not exist yet, a message will be created according to the requirements of the analog alarm.
- If you enter a number of an already existing message, several properties of the message will be changed. The entries pertaining to the event and status tags, as well as the process value block connections, will be deleted.

Note:

The values of the limit value, hysteresis, and trigger value are stored in process value blocks 1 to 3.

8.3.6 Data Archiving

In the Alarm Logging Editor you can configure the short-term and long-term archiving of messages.

The short-term archive is used to reload a configured number of messages into the message window after power failure.

The long-term archiving of messages is done with the message archive. For further information refer to "Message archiving in WinCC".

Note:

A message requires approximately 120 bytes of memory space.

If changes are made to configured messages in the configuration system, changes are also made to the corresponding messages stored in the archives.

8.4 WinCC Alarm Control

The "WinCC Alarm Control" object is used as a message window to display message events. By using Alarm Control, you achieve a high degree of flexibility while you are configuring, since the message view, message line and the message blocks you want to have displayed are all configured in Graphics Designer.

The messages to be displayed in runtime are accessed from the Alarm Logging configuration data.

Configuring WinCC Alarm Control

To display messages in runtime using the WinCC Alarm Control, the following steps must be performed:

In Alarm Logging

1. Specify the properties of the required message blocks in the configuration system, such as the length of the message text in the table window.
2. Modify the properties of message classes and message types such as the coloration of the Alarm message class, according to your requirements.
3. Configure the required single and group messages.

In Graphics Designer

4. Link the WinCC Alarm Control to a Graphics Designer picture and edit the properties of the ActiveX Control.
5. To be able to check the message window settings, buttons must be configured in the Graphics Designer to set and reset message tags.

Note

The message window can be influenced by means of standard functions. If you do not wish to use the toolbar functions to operate the message window, any Graphics Designer objects can be made dynamic using the appropriate functions.

9 Message Archiving

9.1 Message Archiving in WinCC

Introduction

The Archive Management feature in WinCC gives you the opportunity to archive process values and messages in order to create documentation for specific operational and fault statuses. Microsoft SQL Server is used for archiving.

Messages configured in Alarm Logging are output in runtime when the associated event occurs, for example a fault or an above limit value. Messages are archived if events known as message events occur, for example:

- when the message occurs
- when the status of the message changes (e.g. from "Message Came In" to "Confirm Message")

You can save the message events in an archive database and archive them as a hard copy message report. Messages archived in the database can be output in runtime, for instance in a message window.

Archived Message Data

All the data associated with a message, including the configuration data, is saved in the message archive. You can therefore read out all the properties of a message from the archives, including such matters as its message type, time stamp and texts. A subsequent change to the configuration data of a message causes a new archive to be created with the new configuration data. This ensures that the change has no effect on messages that were archived before the change was made.

Note:
The time stamp on archived messages is always in standard UTC format (Coordinated Universal Time).

Since messages are configured on a language-dependent basis, archives contain a table with the configuration data for each designed language.

9.2 Configuring Message Archiving

Introduction

For archiving messages, WinCC uses cyclic archives of a configurable size, and you can configure them with or without backup.

Storing Archive Files

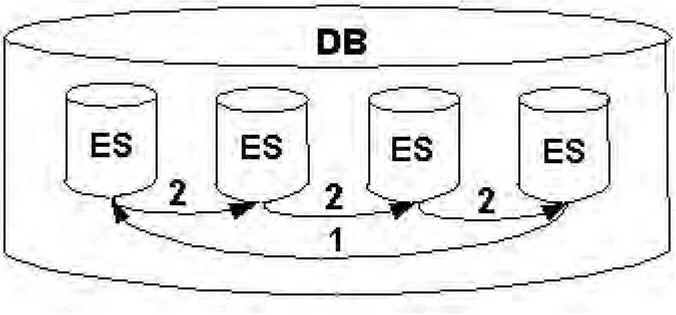
Archive files are always stored so that they are local to the computer in the associated project.

Note
It is not currently possible to use a central archive server in a PCS7 environment because messages and their tag connections cannot be disconnected on different computers.

Properties of Message Archives

A WinCC message archive consists of multiple individual segments. You can configure both the size of the message archive and the size of the individual segments in WinCC:

- The size of the message archive or individual segments, e.g.: The message archive is 100 MB at 5 MB per individual segment.
- The archiving period of the message archive or the individual segments, e.g.: The message archive archives all messages which occur within one week, each individual segment archives the messages which occur within one day. The configuration of the archiving period is called archive partitioning.



You can always configure both conditions at the same time. If either of the two criteria is exceeded, the following happens:

- Criteria for message archive (DB) is exceeded: The oldest messages (i.e. the oldest individual segment) are deleted.
- Criteria for individual segment is exceeded: A new individual segment (ES) is created.

A new individual segment is also created when you configure message data online.

Backing Up Archived Messages

The data from the message archives can be stored in a backup.

9.3 Outputting Message Archive Data in Runtime

Principle

You have the following options for outputting the messages stored in the message archive in runtime:

- Display the archived messages in a message window, with the ability to load into the message system from the archive, following a power failure, the messages that were queued at that time, with the correct time stamp (Reload after power failure).
- Output an archive report in hard copy form.
- Access the message archive database via OLE-DB in order to output archived messages.
- Access via ODK.
- Access via appropriate client applications.

9.4 Direct Access to the Archive Database

Introduction

Various providers provide interfaces which can be used to access data on the database. These interfaces also enable direct access to archive databases from WinCC. In direct access you can for instance read off process values which you can then process further in spreadsheet programs.

Access to the archive databases using ADO/OLE DB

The process values are partly stored in the archive databases in compressed form. Use the WinCC OLE DB Provider in order to access the compressed process values. In the case of data which is not compressed and data in the message archiving, the ADO/OLE DB interfaces form Microsoft can also be used, for example. You can use programming languages such as Visual Basic or Visual C++.

In addition to the ADO/OLE DB interface, access to non-compressed data can also be made using the ODBC interface to access data in the archive databases.

Note:
When making direct access to the archive database using ADO/OLE DB or ODBC, please note that the table structure in a new WinCC version may be different.

Further information on this topic is available in Internet under:

- "http://www.microsoft.com/data/oledb/default.htm" and
- "http://www.microsoft.com/data/odbc/default.htm".

Access to the archive database using OPC

OPC (OLE for Process Control) provides interface ports for communication with process automation systems. Using the OPC interface, devices and applications from various manufacturers can be connected to each other in a uniform way. WinCC can be implemented as an OPC client in order to access process data or messages on an OPC server. If WinCC is implemented as an OPC server, it can be used to manage archive data.

Further information on this topic is available:

- In "WinCC Information System" under "Communication" > "OPC",
- In the "WinCC Information System" under "System Overview" > "Openness"
- In Internet under "http://www.opcfoundation.org"

Access to the archive database using C-API/ODK

Using the option "WinCC Open Development Kit", you can access WinCC data and functions via open programming interfaces.

For further information in this topic:

Documentation on WinCC Open Development Kit

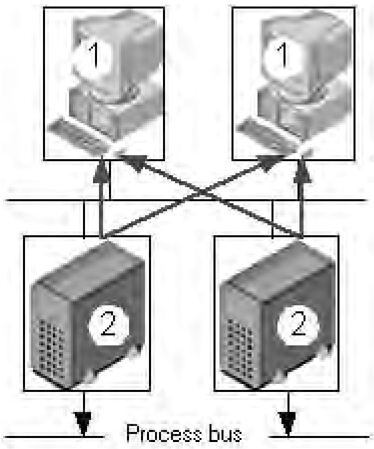
9.5 Message Servers

Introduction

When using WinCC to configure a client-server system, you can configure different scenarios for message archiving.

Archiving on WinCC servers

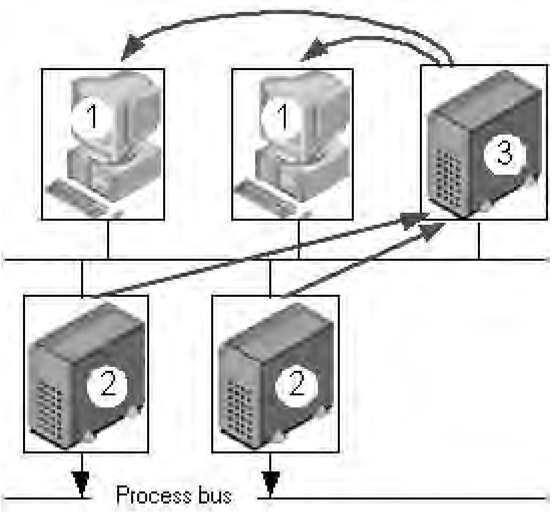
The messages are archived on each server (2) in a client-server system. All runtime applications including archiving run on each server. The clients (1) have the access to the message archives on the server.



Use this configuration if the servers are distributed from the technological point of view. Each server monitors a special part of the plant as well as managing its own message system and archive system.

Archiving on archive servers without a process driver connection (server-server communication)

Archiving on a central archive server (3) on which the message archives (and process value archives, if necessary) are stored. Only the archiving applications run on the archive server. The archive server receives the data to be archived from the servers (2). The clients (1) access the archive server (3) so that archived data can be displayed. All other runtime data is received from the servers (2):



Note
This configuration is not possible for message archives in a PCS7 environment, since messages are generated on each server individually and cannot be brought together on a single server.

10 Archiving Process Values

10.1 Basics of Archiving Process Values

Introduction

The purpose of process value archiving is to acquire, process and archive process data from an industrial plant. The process data obtained in this way can be used to obtain key managerial and technical criteria relating to the operating state of a plant.

Mode of operation

In runtime the process values that are to be archived are acquired and processed and then stored in the archive database. In runtime you can output current process values or values that have already been archived, as either a table or a trend. It is also possible to print out archived process values as a log.

Configuring

You configure process value archiving in the Tag Logging editor. This is where you configure process value archives and compressed archives, define the acquisition and archiving cycles and select the process values that you want to be archived.

In the Graphics Designer you configure ActiveX controls for displaying process data in runtime. You can output the process data in the form of trends or tables.

In the Report Designer you configure the way that archived process data is output in a log. The process values in the log can be output in table form or as a trend diagram.

Usage

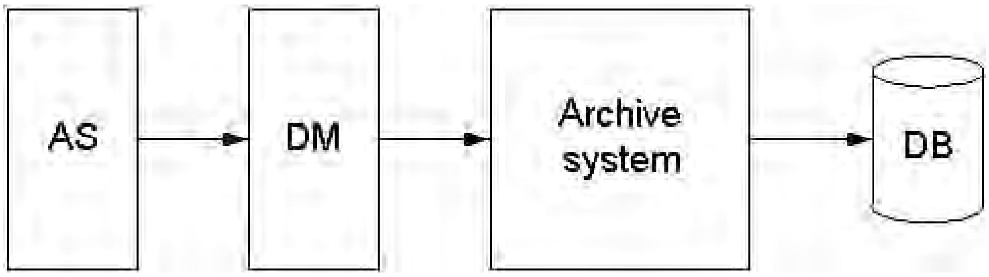
You can use process value archiving for the following tasks:

- Early detection of dangerous states and fault conditions
- Raising productivity
- Increasing product quality
- Optimizing maintenance cycles
- Documenting the history of process values

10.1.1 Process Value Archiving in WinCC

Introduction

The archive system is responsible for archiving process values in runtime. The archive system processes the process values temporarily stored in the runtime database and writes them to the archive database.



The following WinCC subsystems are involved in the archiving of process values:

- Automation system (AS): saves the process values, which are sent to WinCC via communication drivers.
- Data manager (DM): processes the process values and returns them to the archive system via process tags.
- Archive system: processes the acquired process values (e.g. forming the average value). The method of processing depends on the way the archive is configured.
- Runtime database (DB): saves the process values that are to be archived.

Definition of terms

The question of whether and when process values are acquired and archived is dependent on a variety of parameters. Which of the following parameters you configure depends on the archiving method used:

- Acquisition cycle: determines when the value of a process tag is read out in the automation system. You configure an acquisition cycle, for example, for the cyclic archiving of process values.
- Archiving cycle: determines when the processed process value is saved in the archive database. You configure an archiving cycle, for example, for cyclic and cyclic-selective process value archiving.
- Start event: starts process value archiving when a specified event occurs, for example when the plant is turned on. You configure a start event, for example, for the acyclic archiving of process values.
- Stop event: terminates process value archiving when a specified event occurs, for example when the plant is turned off. You configure a stop event, for example, for the cyclic-selective archiving of process values.

10.1.2 Process Value Archiving in Multi-user Projects

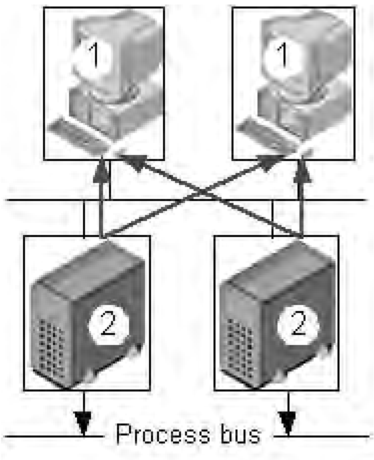
Introduction

In multi-user projects, process values can be archived by various servers. The following client-server scenarios illustrate possible configurations.

Client-server scenarios

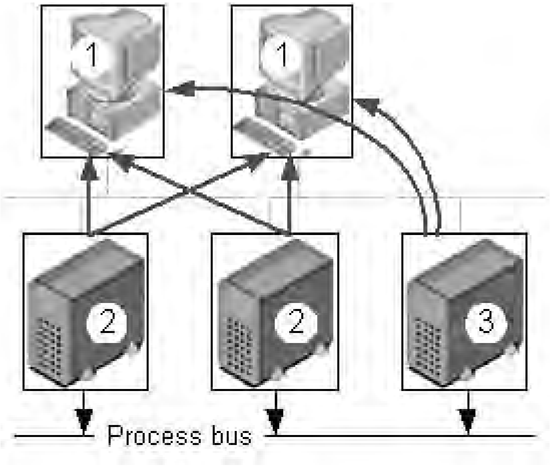
Scenario 1

Process value archiving runs on several servers (2). Clients (1) can access the process value archives on the available servers:



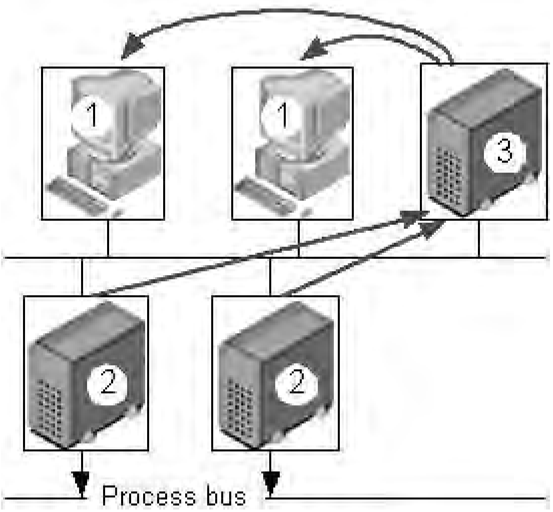
Scenario 2:

Message and process value archiving runs on one server (3), together with process communication. All other servers (2) deal with process-oriented tasks (e.g. displaying pictures). Clients (1) can access all available servers:



Scenario 3:

A server (3) without a process driver connection deals solely with message and process value archiving. The server (3) collects the data from all other servers (2) through server-server communication and distributes it to the clients (1):

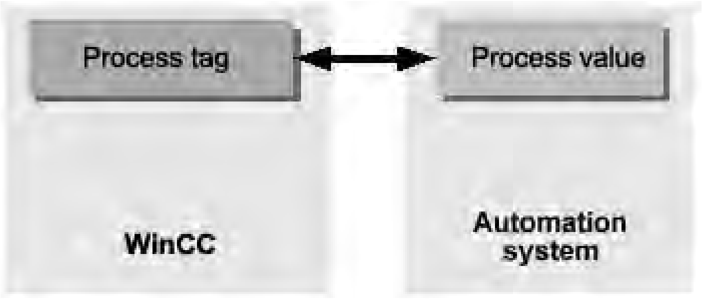


10.1.3 Process Values and Tags

Principle

Process values are data stored in the memory of one of the connected automation systems. They represent the status of a plant, for example, temperatures, fill levels or states (e.g. motor off). To work with the process values, you define tags in WinCC.

The link between WinCC and the automation systems is provided by process tags, which correspond to a certain process value in the memory of an automation system. The value of a process tag is obtained by reading out the process value from the memory of the automation system. Conversely, a process value can also be written back to the memory of the automation system.



10.1.3.1 External and Internal Tags

Introduction

External tags in WinCC are used for acquiring process values and access a memory address in the connected automation system. External tags are therefore referred to as process tags. Internal tags do not have a process link and only carry values within WinCC.

External and internal tags can be stored in archive tags in the process value archive.

Note:
Because the primary task of process value archiving is the archiving of external tag values, this documentation shall refer only to process tags.

Archive tags

Process values are stored in archive tags in the process value archive. A distinction is drawn between the following types of archive tags:

- An analog archive tag stores numerical process values, e.g. the fill level of a tank.
- A binary archive tag stores binary process values, e.g. whether a motor has been switched on or off.
- A process-controlled tag stores process values that have been sent to the archive system as a message frame, e.g. process values in a series of measurements.

As well as this, archived process values can be compressed. This compression is achieved through the application of mathematical functions (e.g. averaging). Compressed process values of this type are stored in compressed tags in a compressed archive.

10.1.3.2 Message Frame Tags

Introduction

Message frame tags are needed for the acquisition of quickly changing process values, or if you want to combine several measuring points from one plant.

Note:
Message frame tags are of the type "raw data type" in WinCC and are therefore also referred to a "raw data tags".

Principle

In the automation system the process values are written to a binary file, and are sent as a message frame to WinCC where they are stored in a raw data tag.

Archiving message frames

If you want the acquired process values belonging to a raw data tag to be archived, you need to configure a process-controlled tag in the process value archive. To enable the archive system to process the message frame in the process-controlled tag, select a format DLL. The format DLL is supplied with the automation system that you are using and dismantles the message frame (e.g. to ascertain the process values). The process values are then written to the archive database.

A format DLL for the SIMATIC S7 is included as standard in the scope of delivery of WinCC.

10.1.4 Archiving Methods

Introduction

You can use various archiving methods for archiving process values. For example, you can monitor a single process value at certain times and make this monitoring dependent on certain events. You can archive rapidly changing process values, without increasing system load as a result. You can compress previously archived process values in order to reduce the volume of data.

Archiving methods

The following archiving methods are available in runtime:

- Cyclic process value archiving: continuous process value archiving (e.g. monitoring a process value).
- Cyclic-selective process value archiving: event-controlled, continuous process value archiving, for example for monitoring a process value within a specific period of time.
- Acyclic process value archiving: event-controlled process value archiving (e.g. archiving a current process value when a critical limit value is exceeded).
- Process-controlled process value archiving: archiving of several process tags or of rapidly changing process values.
- Compressed archive: compression of individual archive tags or of entire process value archives (e.g. the hourly averaging of process values archived every minute).

10.1.4.1 Cycles and Events

Introduction

Process value archiving is controlled using cycles and events. The acquisition and archiving cycles ensure the continuous acquisition and storage of process values. As well as this, process value archiving can also be triggered and ended by events. You can combine cycles and events (e.g. a process value can be regularly acquired; archiving, however, is not triggered until a binary event occurs).

Acquisition cycle

The acquisition cycle determines when the process value of a process tag is read out. The smallest value that can be set is 500 ms. All other values are always an integral multiple of this value.

Note:
A short acquisition cycle can lead to a high system load. If there are frequent or rapid process value changes, use message frame tags.

Archiving cycle

The archiving cycle determines when a process value is saved in the archive database. The archiving cycle is always an integral multiple of the acquisition cycle. All process values that have been read out from the process tags in the period between acquisition and archiving are processed by the archiving function.

You can use one of the following archiving functions in a process value archive:

- Maximum: saves the highest of all acquired process values.
- Minimum: saves the lowest of all acquired process values.
- Actual value: saves the most recently acquired process value.
- Average value: saves the average value of all acquired process values.
- Sum: saves the sum of all acquired process values.
- Action: the last acquired process value is calculated by a function created in Global Script.

Start/stop events

Events start and end process value archiving. The conditions that trigger an event can be linked to tags and scripts (C, VBS). In WinCC a distinction is drawn between the following events:

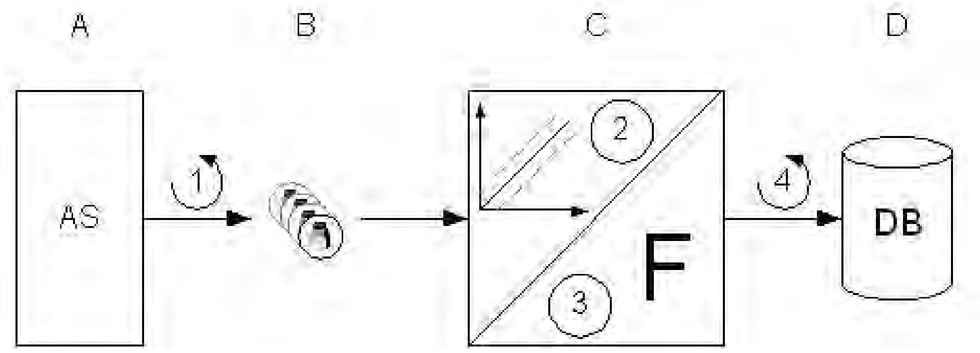
- Binary event: responds to a change to a Boolean process tag. For example, the switching on of a motor can start process value archiving.
- Limit value event: responds to a value falling below or rising above a limit or a limit value being reached. The limit value change can be absolute or relative. For example, archiving can be triggered in the event of temperature fluctuations of more than 2%.
- Time-controlled event: responds to a fixed point in time or an elapsed period of time after the start of process value archiving. For example, a log is output at the time of each shift change.

10.1.4.2 Cyclic Process Value Archiving

Introduction

Cyclic process value archiving begins when runtime is started. The process values are acquired in constant time cycles and are stored in the archive database. Cyclic process value archiving ends when runtime is ended.

Mode of operation



The process tags in WinCC (B) correspond to a particular process value in the memory of one of the connected automation systems (A). The acquisition cycle (1) governs when the process value is read out from the memory of the connected automation system.

The runtime component of the archive system (C) processes the process value:

- Whether the process value is archived at all depends on the way you have configured the system. For example, it may be the case that the process value has to change by a certain amount or percentage (2).
- The archiving function (3) determines how the acquired process values are to be processed (e.g. averaging).

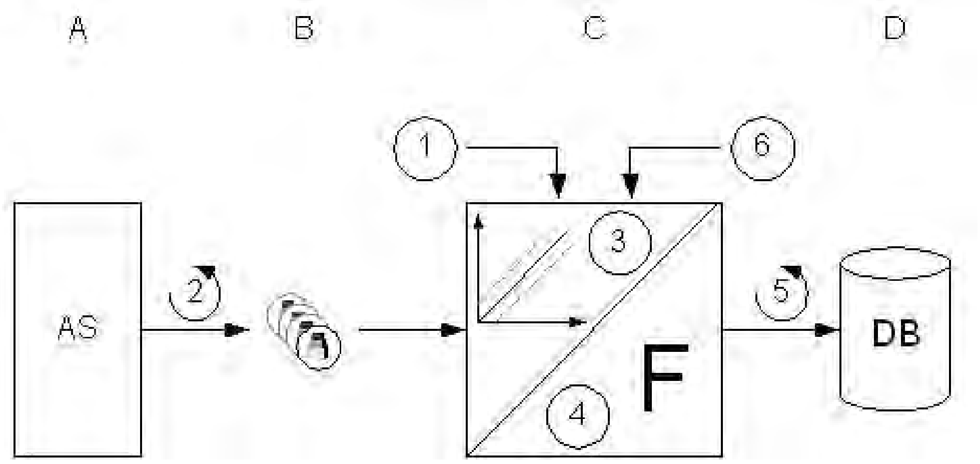
The archiving cycle (4) determines when the processed process value is written to the archive database (D).

10.1.4.3 Cyclic-Selective Process Value Archiving

Introduction

Cyclic-selective process value archiving begins in runtime with the occurrence of a Start event. The process values are acquired in constant time cycles after the start and are stored in the archive database. Cyclic process value archiving ends either with the occurrence of a Stop event or when runtime is ended. When a Stop event occurs, the most recently acquired process value is also archived.

Mode of operation



The process tags in WinCC (B) correspond to a particular process value in the memory of one of the connected automation systems (A). Process value archiving begins on the occurrence of the Start event (1). The acquisition cycle (2) governs when the process value is read out from the memory of the connected automation system.

The runtime component of the archive system (C) processes the process value:

- Whether the process value is archived at all depends on the way you have configured the system. For example, it may be the case that the process value has to change by a certain amount or percentage (3).
- The archiving function (4) determines how the acquired process values are to be processed (e.g. averaging).

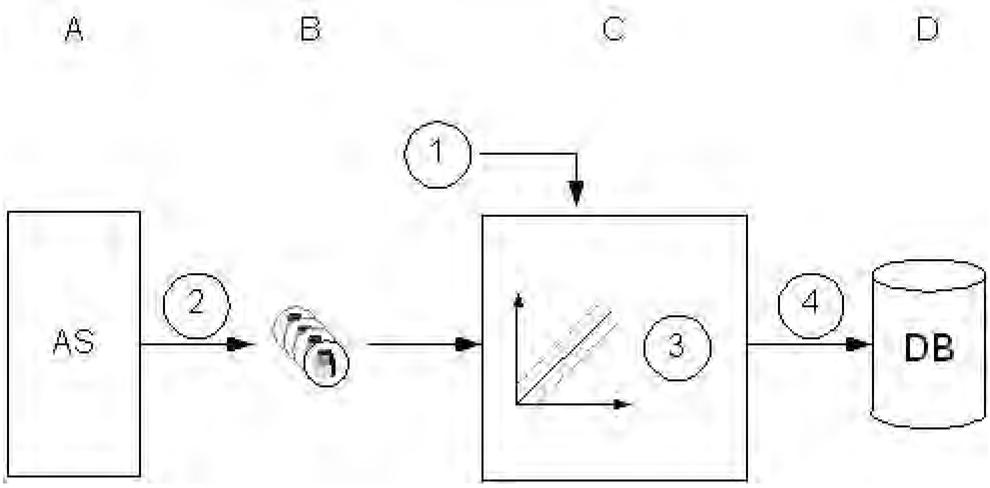
Until the occurrence of the Stop event (6), the archiving cycle (5) determines when the processed process value is written to the archive database (D).

10.1.4.4 Acyclic Process Value Archiving

Introduction

Acyclic process value archiving saves the current process value in the archive database in runtime on the occurrence of a Start event, or when there is a change in the value of the process tag. Acyclic process value archiving ends when runtime is ended.

Mode of operation



The process tags in WinCC (B) correspond to a particular process value in the memory of one of the connected automation systems (A). When the Start event occurs (1) or when there is a change in the value of the process tag, the process value is read out from the memory of the connected automation system (2).

The runtime component of the archive system (C) processes the process value:

- Whether the process value is archived at all depends on the way you have configured the system. For example, it may be the case that the process value has to change by a certain amount or percentage (3).

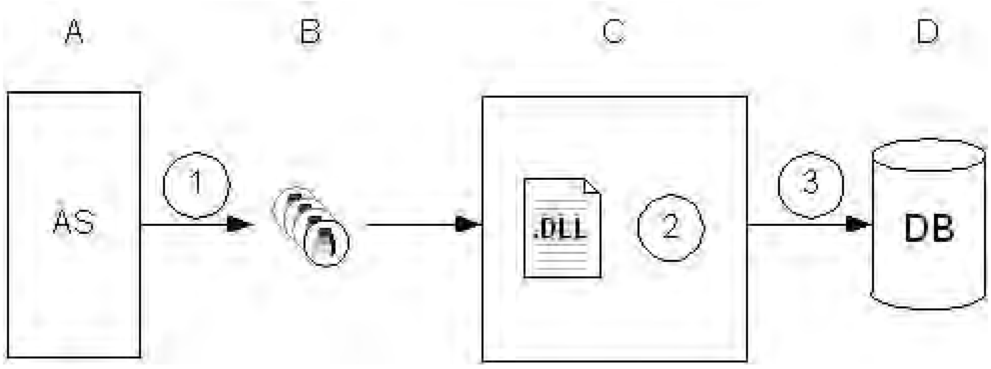
The actual value of the process value is then written to the archive database (D) (4).

10.1.4.5 Process-Controlled Process Value Archiving

Introduction

Process-controlled process value archiving is used for archiving several process tags or rapidly changing process values. The process values are written to a message frame tag, which is decoded by the archive system. The process values obtained in this way are then stored in the archive database.

Mode of operation



The process tags in WinCC (B) correspond to a particular process value in the memory of one of the connected automation systems (A). At the start of runtime, the process values of the selected process tags are read out (1) and written to the configured message frame tag as binary data.

The runtime component of the archive system (C) processes the message frame tag:

- The format DLL (2) is part of the archive system and decodes the binary data of the message frame tag.

The decoded process values are then written to the archive database (D) (3).

10.1.4.6 Compressed Archive

Introduction

In order to reduce the volume of data in the archive database, the archive tags for a specified period can be compressed. To do this a compressed archive is created, which stores each archive tag in a compressed tag. The archive tags are retained, but they can also be copied, moved or deleted. The compressed archive is stored in the archive database in the same way as the process value archive.

Mode of operation

The compression is achieved through the application of mathematical functions. To achieve this, one of the following functions is applied to the archived process values in a specified period of time:

- Max. value: stores the highest process value in the compressed tag.
- Min. value: stores the lowest process value in the compressed tag.
- Average value: stores the average value of the process values in the compressed tag.
- Sum: stores the sum of the process values in the compressed tag.

What happens to the old archived process values after compression depends on which method of compression is used:

- Calculate: the process values of the archive tags from the specified period are read out and compressed. The process values of the archive tags are retained.
- Calculate and copy: the process values of the archive tags from the specified period are read out, compressed and additionally copied to the compressed archive.
- Calculate and delete: the process values of the archive tags from the specified period are read out, compressed and subsequently deleted.
- Calculate, copy and delete: the process values of the archive tags from the specified period are read out, compressed and additionally moved to the compressed archive.

Example

The following example illustrates the way that the compressed archive works:

A process value is archived once every minute and returns 60 values in one hour. Compression (e.g. averaging) is to be carried out over the period of one hour. Accordingly, every hour, the average value is formed from the 60 values and is stored in the compressed tag. What happens to the 60 values depends on the compression method described above.

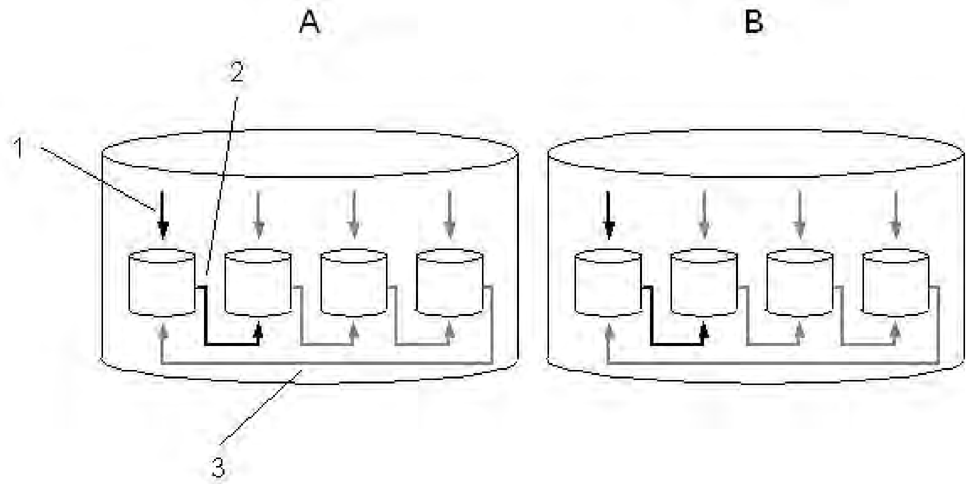
10.1.5 Storing Process Values

Introduction

Process values can be stored either on the hard disk in the archive database or in the main memory of tag logging runtime.

Storing in the archive database

The process values to be archived are stored in two separate short-term archives (A, B) in the archive database. Each short-term archive consists of a configurable number of data buffers. A size in MB and a period of time (e.g. one day) are defined for the data buffer.



The process values are written continuously to the data buffer (1). When the configured size of the data buffer is reached or the time span is exceeded, the system switches to the next one (2). When all data buffers are full, the process data in the first data buffer is overwritten (3). To prevent process data from being lost by being overwritten, it can be swapped out.

Short-term archive A stores process values whose acquisition cycle is shorter than or equal to one minute. These process values are first stored in a binary file and compressed. The binary file is not stored in the short-term archive until it has reached a certain size.

Short-term archive B stores process values whose acquisition cycle is longer than one minute and compressed archives. This data is immediately written to the short-term archive and is not compressed.

Storing in the main memory

In contrast with storage in the archive database, process values archived in the main memory are only available for as long as runtime is active. Storing in the main memory has the advantage, however, that the values can be written and read out very quickly. The process values stored in the main memory cannot be swapped out.

Note:
Compressed archives cannot be stored in the main memory.

10.1.6 Swapping Out Process Values

Introduction

You can swap out process values from the archive database as a backup. All process values contained in a data buffer are swapped out. When the swapping-out actually takes place depends on how you configure the system.

Principle

You configure the swapping-out of process values in Tag Logging in the "TagLogging" dialog. This is where you configure the data buffer on the hard disk and the swap-out settings:

On the Archive Configuration tab enter the settings for the individual data buffers and specify what period of time is to be contained in the archive.

On the Backup Configuration tab specify whether you want to create a backup of the archived process values and where the backup is to be stored.

Long-term archive server

The long-term archive server is used for backing up process value archives. There are three ways in which to access the swap files:

- Copy the swap files onto the configuration computer on which runtime is also running. Link the swap files to the project in AlarmLogging or Tag Logging. The archived valued are displayed in runtime.
- Access using OLE DB
- Access using Dat@Monitor Web Edition

To set up a long-term archive server, run the WinCC "Fileserver Setup" on the computer. A minimum installation of WinCC is then installed on the computer. In order to be able to access the stored process value archive with ADO/OLE DB, the MS SQL server is also installed.

10.2 Configuring Process Value Archiving

Introduction

You configure process value archiving in the Tag Logging editor. This is where and when you define which process values are to be archived. WinCC provides you with the process value archive and the compressed archive for archiving the process values.

General procedure

Configuring of process value archiving is broken down into the following steps:

1. Configuring the process value archive: Use the "Archive Wizard" to create the process value archive in Tag Logging and select the process tags.
2. Creating the archive tags: Define whether and when a process value is to be archived for each archive tag.

10.2.1 Tag Logging

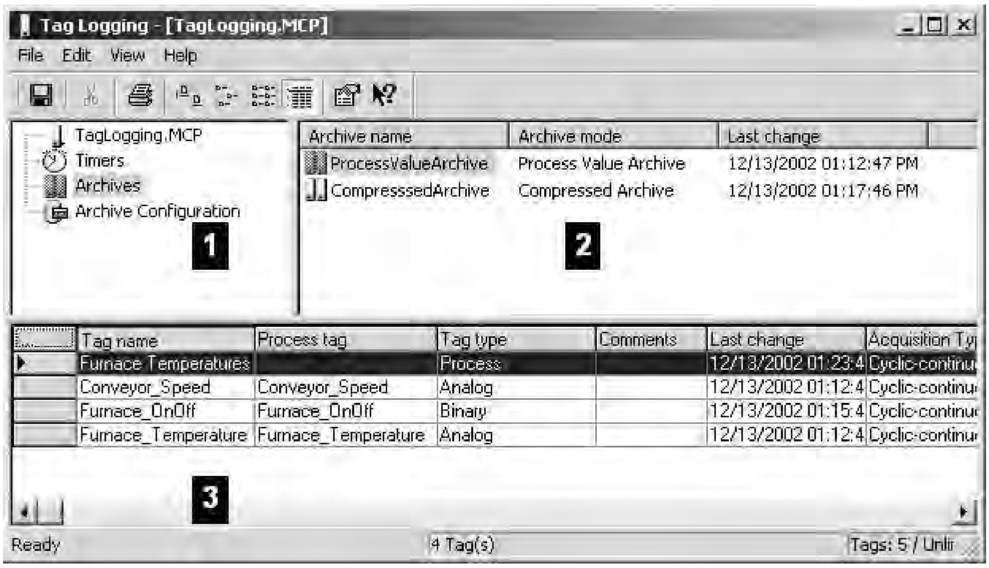
Introduction

Tag Logging is where the archives, the process values that are to be archived, and the times for the acquisition and archiving cycles are configured. In addition, you also specify the configuration of the data buffer on the hard disk and the swapping-out of process values in Tag Logging.

As with all editors in WinCC, to start Tag Logging double-click on it in WinCCExplorer.

Structure of Tag Logging

Tag Logging is divided into the navigation window, data window and table window:



Archive name	Archive mode	Last change
ProcessValueArchive	Process Value Archive	12/13/2002 01:12:47 PM
CompresssedArchive	Compressed Archive	12/13/2002 01:17:46 PM

	Tag name	Process tag	Tag type	Comments	Last change	Acquisition Ty
▶	Furnace Temperatures		Process		12/13/2002 01:23:4	Cyclic-continu
	Conveyor_Speed	Conveyor_Speed	Analog		12/13/2002 01:12:4	Cyclic-continu
	Furnace_OnOff	Furnace_OnOff	Binary		12/13/2002 01:15:4	Cyclic-continu
	Furnace_Temperature	Furnace_Temperature	Analog		12/13/2002 01:12:4	Cyclic-continu

(1) Navigation window

Here you select whether you want to edit times or archives.

(2) Data window

Depending on the selection you made in the navigation window, here you can edit existing archives or times or create new onew.

(3) Table window

The table window is where the archive tags or compressed tags are displayed that are stored in the archive that is selected in the data window. You can change the properties of the displayed tags here or add a new archive tag or compressed tag.

10.2.2 Configuring Archives

Principle

In the configuring of archives a distinction is drawn between the following archive types:

- The process value archive stores process values in archive tags. When configuring the process value archive you select the process tags that are to be archived and the storage location.
- The compressed archive compresses archive tags from process value archives. When configuring the compressed archive you select a calculation method and the compression time period.

10.2.3 Creating Archive Tags

Principle

Archive tags store the process values that are to be archived. The following archive tags can be used in a process value archive:

- A binary archive tag is used for storing binary process values.
- An analog archive tag is used for storing numeric process values.
- A process-controlled tag is used for storing process values which have been sent to the archive system as a message frame.

In a compressed archive each compressed process value is stored in a separate compressed tag.

General procedure

In the case of binary and analog archive tags you specify the acquisition type (e.g. cyclic) and acquisition and archiving cycles. Depending on the acquisition type, select events which trigger or end archiving. Depending on the type of archive tag, specify the display limits and parameters for processing the process values.

When configuring a compressed tag, select the compression function that for example forms the average value from the process values that are to be compressed.

10.3 Output of Process Values

Introduction

You can output process values in process displays and as a log. You can also directly access the archive database via various interfaces.

Process value output in process displays

In runtime you can output process values in tabular form or as trends. To do this you can load process values from the archive database or monitor the running process directly.

Process value output in the log

You can print out process values from the archive database as a log. In this output form, you can choose between tabular form and trends. Both output forms are available to you as ready-prepared layouts in the Report Designer.

Direct access to the archive database

--

10.3.1 Display of Process Values in Tables

WinCC permits you to display tag values in a tabular format.

The display of the tag values takes place in Runtime in an ActiveX Control. This ActiveX Control has to be created in "Graphics Designer" and be linked to the process value archives configured in "Tag Logging".

10.3.1.1 Table Display Types

Two basic display types are available for displaying tag values in tabular form.

Shared time column

In this display type, the table contains only one time column. This column displays the archived timers of the tag output in the first column. A table in which the values of three tags are to be displayed consists of four columns in runtime.

Note

The display of tags with different acquisition timers in a table with a shared time column is problematic, since the tag values might not correspond to the indicated archived timers of the time column. Values such as these are shown as being struck through.

Separate time columns

In this display type, the archived timers of each tag are displayed in separate columns. A table in which the values of three tags are to be displayed consists of six columns in runtime.

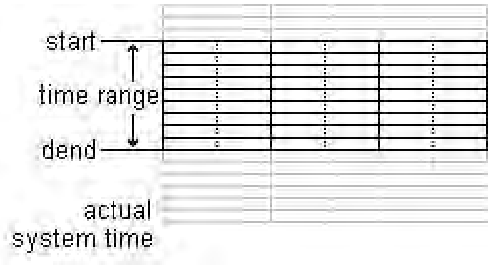
The display type to be used is specified on the "General" tab of the "Properties of the WinCC Online Table Control" dialog.

10.3.1.2 Time Range of a Trend Display

With regard to their time ranges, tags can be displayed in tables in several ways:

Static display

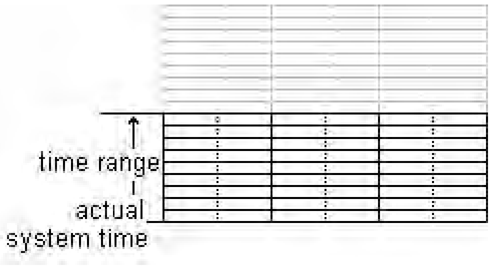
This type of display can present the archived values of a tag within a defined time interval.



The static display type is configured on the "Column" tab of the "Properties" dialog of the "WinCC Online Table Control" by deactivating the "Update" option. The time range to be displayed is set either by specifying a start and end time or by specifying a start time and a time range.

Dynamic display

With this type of display, the last point of the table always corresponds to the current system time. New measurement values coming in will be included in the table.



The dynamic display type is configured on the "Column" tab of the "Properties" dialog of the "WinCC Online Table Control" by activating the "Update" option. The time range to be displayed is set either

- by specifying a time range, or
- by the time difference between the specified start and end times.

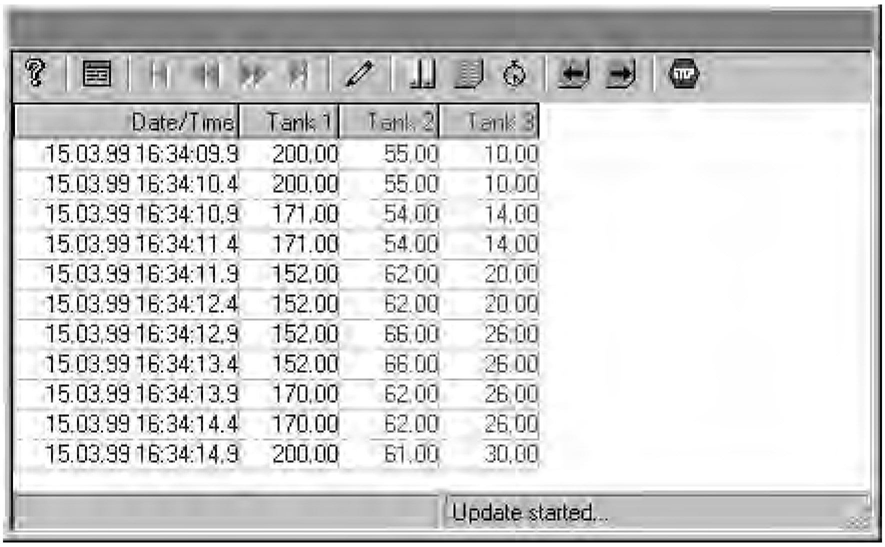
Note

Depending on the setting, the tag values to be displayed are either read from the archive or set to zero when runtime is started.

10.3.2 WinCC Online Table Control

The "WinCC Online Table Control" object is used for displaying process data as a table. Use of the Table Control gives you a high degree of flexibility during the configuration, since the table window configuration is performed exclusively within the Graphics Designer.

To make possible access to the tags which are to be displayed, process value archives and the tags contained therein must be configured in Tag Logging.



Date/Time	Tank 1	Tank 2	Tank 3
15.03.99 16:34:09.9	200.00	55.00	10.00
15.03.99 16:34:10.4	200.00	55.00	10.00
15.03.99 16:34:10.9	171.00	54.00	14.00
15.03.99 16:34:11.4	171.00	54.00	14.00
15.03.99 16:34:11.9	152.00	62.00	20.00
15.03.99 16:34:12.4	152.00	62.00	20.00
15.03.99 16:34:12.9	152.00	66.00	26.00
15.03.99 16:34:13.4	152.00	66.00	26.00
15.03.99 16:34:13.9	170.00	62.00	26.00
15.03.99 16:34:14.4	170.00	62.00	26.00
15.03.99 16:34:14.9	200.00	61.00	30.00

Update started...

Configuring the WinCC Online Table Control

To display process values in runtime using the WinCC Online Table Control, the following steps must be performed:

In tag logging

1. Create a process value archive with the aid of the Archive Wizard.
2. Configure the process value tags contained in it.

In Graphics Designer

3. Link the WinCC Table Control to a Graphics Designer picture.
4. Link the columns to be displayed in Table Control to the tags of the process value archive.

Note

The table window can be influenced by means of standard functions. If you do not wish to use the toolbar functions to operate the table window, any Graphics Designer objects can be made dynamic using the appropriate functions.

10.3.3 Display of Process Values in Graphs

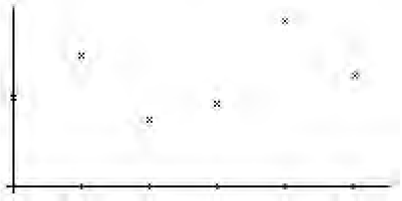
For the graphical display of tags, WinCC provides the option of displaying tag values as trends. The trend type to be used can be selected. The values to be displayed may be either current or archived.

A tag's course is displayed in runtime by an ActiveX Control, which is set up in the "Graphics Designer". Online tags or archive tags can be displayed.

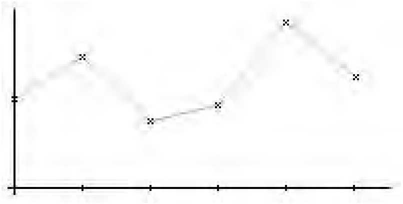
10.3.3.1 Trend Display Types

To graphically display tag values, three basic trend types are available.

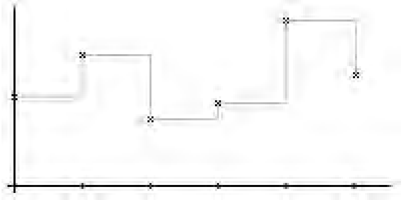
Individual values



Linear interpolation



Stepped trend



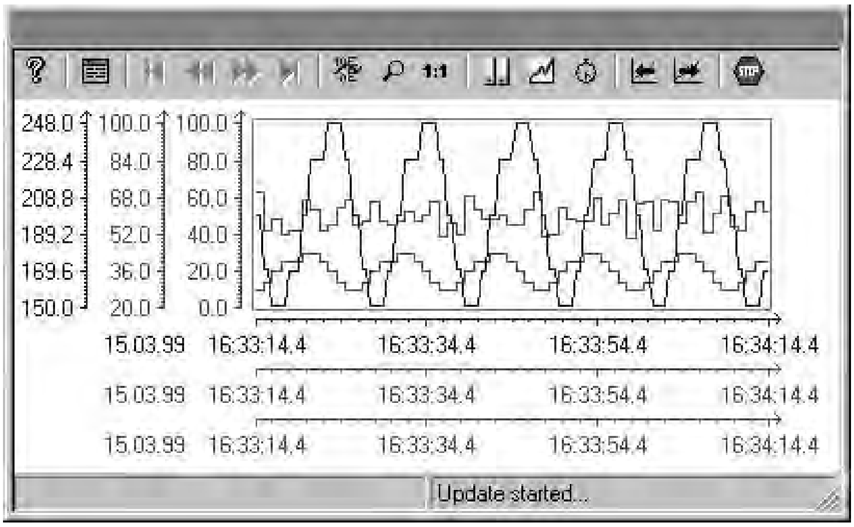
With the exception of when displaying individual values, the area under the trend can also be displayed filled.

The trend type to be used is specified in the "Value Axis" tab of the "Properties" dialog of the WinCC Online Trend Control.

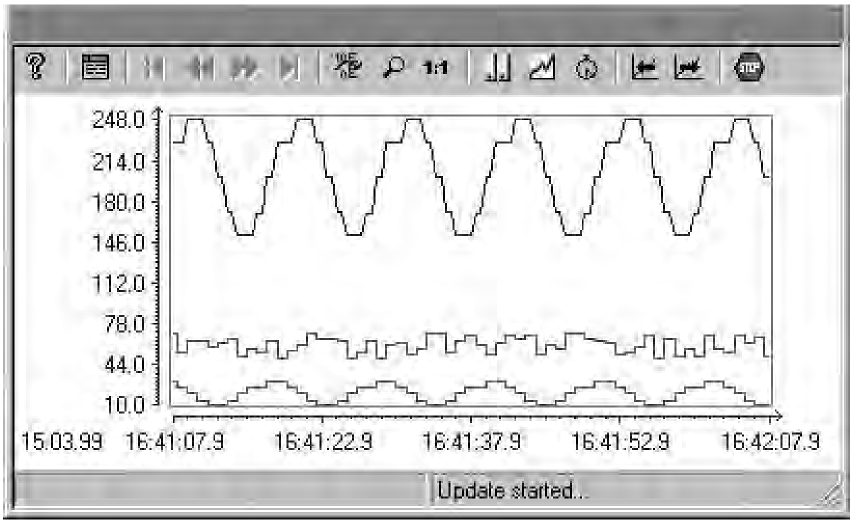
Using Shared Axes

If multiple trends are to be displayed in a trend window, you have the choice of using individual axes for each trend or using shared X/Y axes for all trends.

If the tag values to be displayed in a tag window differ greatly, the use of shared axes for the trend display is not recommended. The tag values can be read more easily if different axial scales are used.



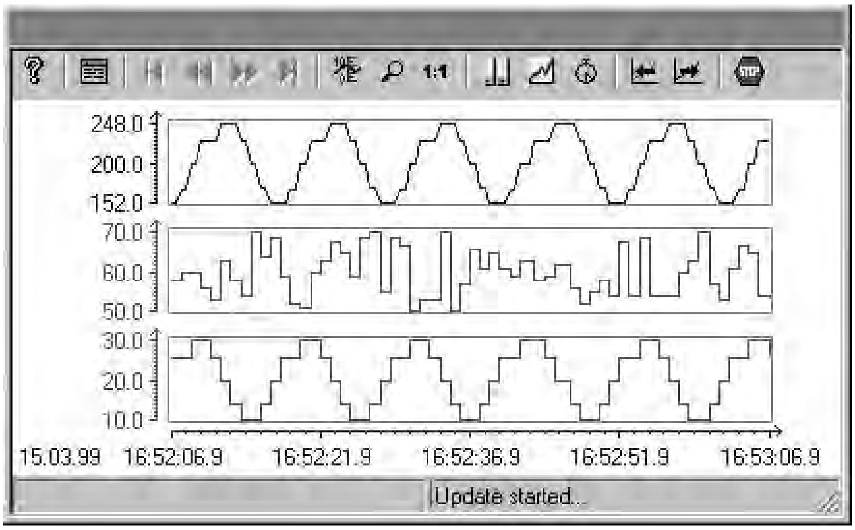
If you want to display multiple trends in a trend window and are particularly interested in the comparability of the trends, the trends should be displayed with shared axes. In runtime, the exact tag values can also be determined by enlarging a segment or by inquiring about the coordinates.



On the "General" tab in the "Properties of WinCC Online Trend Control" dialog, it is possible to configure independent of each other the use of a shared X axis and a shared Y axis.

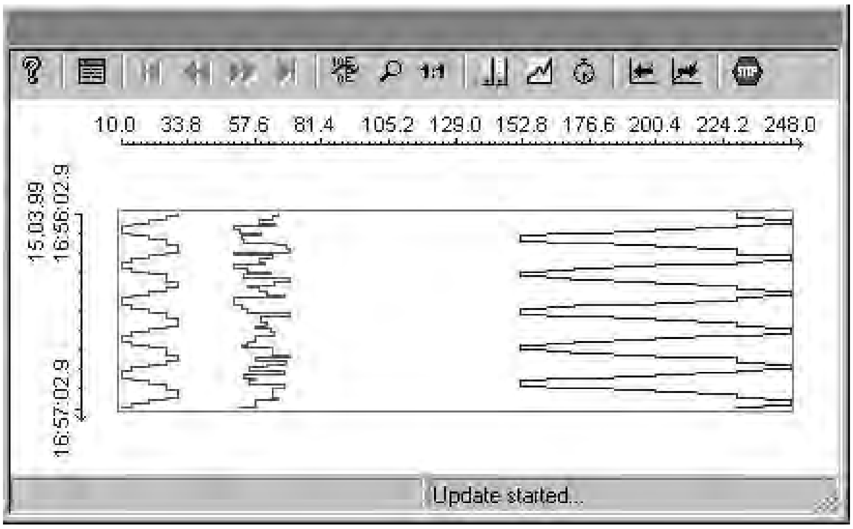
Staggering Trends

In a staggered display, the trends are displayed offset over each other in the trend window. For each trend, the value range to be displayed can be set for the Y axis.



Writer function

With the "Write Direction" option, you specify where the current measurement values are to be displayed. The "from the bottom" setting specifies that the current measurement values are to be displayed at the lower border of the trend window.



Note

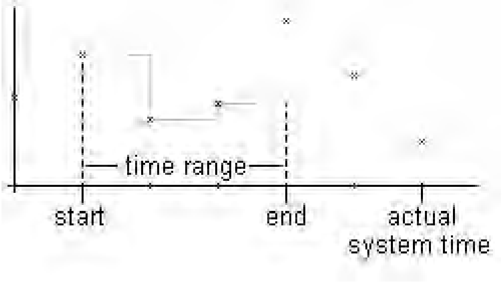
If you select as write direction either "from the top" or "from the bottom", you should use True-Type fonts within the trend window to ensure a clean display of the time axis labeling.

10.3.3.2 Time Range of a Trend Display

With regard to the time ranges to be displayed, there are a number of ways to present a trend display of tags:

Static display of a trend

Based on the archived values, this form of display can present the course of a tag within a defined time interval.



The static display type is configured in the "Time Axis" tab of the "Properties" dialog by deactivating the "Update" option. The time range to be displayed is set either

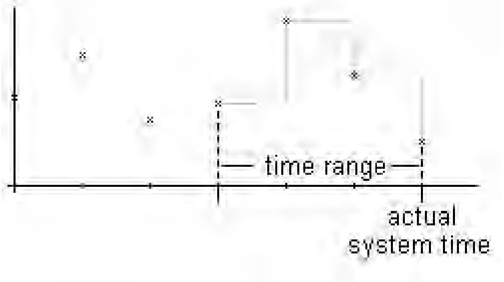
- by entering a start time and a time range.
- by entering a start time and an end time.
- by entering a start time and the number of measurement points which are to be displayed.

Note

This type of display can only be used for archive tags.

Dynamic display of a trend

With this type of display the last point of the trend always corresponds to the current system time. New measurement values coming in will be included in the display.



The dynamic display type is configured in the "Time Axis" tab of the "Properties" dialog by activating the "Update" option. The time range to be displayed is set either

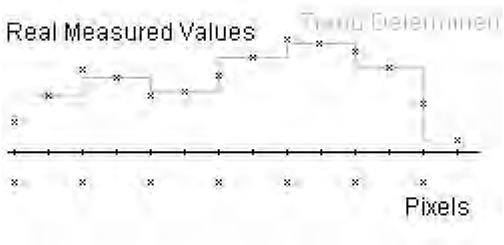
- by entering a time range.
- by entering the number of measurement points to be displayed.
- by the time difference between the specified start and end times

Note

If a trend window picture is opened in runtime, the tag values to be displayed are either read from the archive or set to zero. This function is set in the "General" tab of the "Properties" dialog.

Limiting the resolution of a trend display

The number of trend values that can be displayed on the screen is limited by the screen resolution and selected size of the trend window. Therefore, when displaying trends, it is possible that there are fewer values displayed in the trend window than are actually archived.

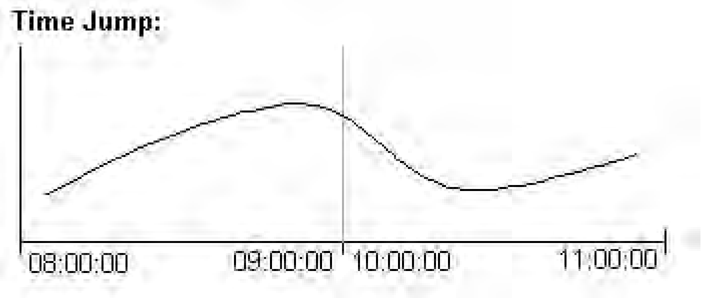


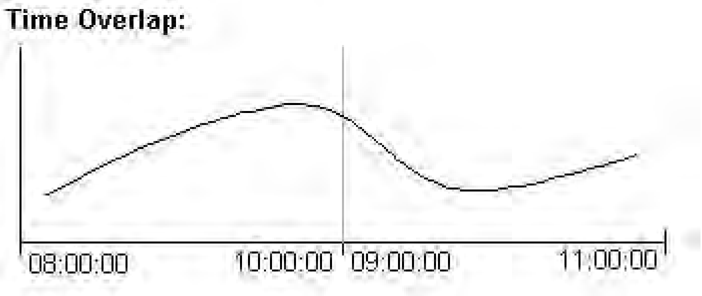
For example, when 200 measured values are archived in a range with 100 pixels, each pixel on the screen will represent two measured values. The value shown on the screen will be that of the most recent data (most recent time stamp).

Time Jumps, Time Overlaps

By changing the time, e.g. by switching from daylight saving to standard time or vice versa, time jumps or time overlaps may occur in the archive.

When displaying a trend for archived values, the labeling of the time axis is modified and marked with a vertical line should a time jump or time overlap be present. On the "Limit Values" tab of the "Properties of WinCC Online Trend Control" dialog, the marker line can be assigned a color.





10.3.4 WinCC Online Trend Control

The "WinCC Online Trend Control" is used to display process data as a trend. Using the Online Trend Control gives you a high degree of flexibility during the configuration, since the table window configuration is performed exclusively within the Graphics Designer.

To allow access to the tags that have to be displayed, they have to be configured in WinCC Explorer first. Online tags or archive tags can be selected and displayed.

Configuring the WinCC Online Trend Control

To display process values in runtime using the WinCC Online Trend Control, the following steps must be performed:

In tag logging

1. Create a process value archive with the aid of the Archive Wizard.
2. Configure the process value tags contained in it.

In Graphics Designer

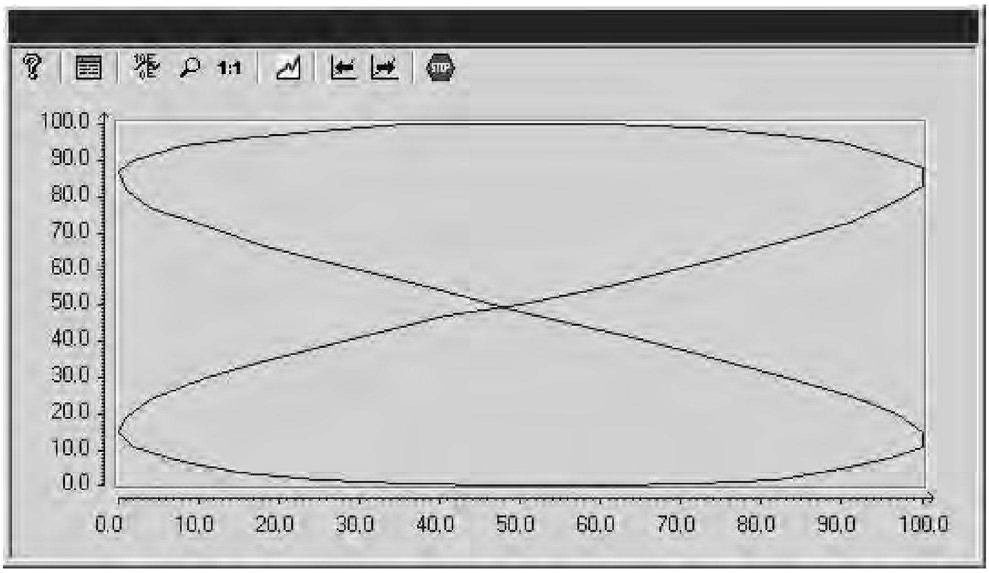
3. Link the WinCC Online Trend Control to a Graphics Designer picture and edit the properties of the ActiveX Control.

10.3.5 WinCC Function Trend Control

Introduction

For the graphical presentation of tags, the Function Trend Control feature of WinCC provides the option to display tag values as a function of another tag. For instance temperature trends can be displayed as a function of pressure. At the same time trends can be compared with an ideal trend.

Trends are displayed in runtime in an ActiveX Control, which is inserted in a picture and configured with the aid of the Graphics Designer.



Requirements

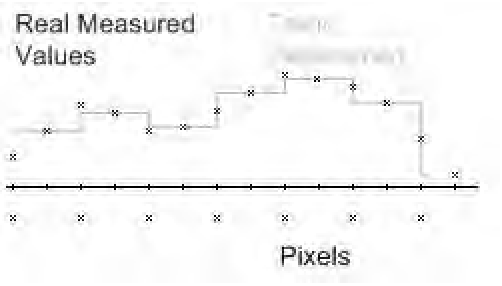
The following prerequisites apply to displaying trends in WinCC Function Trend Control:

- Any number of trends can be displayed in a WinCC Function Trend Control. However, we recommend configuring a maximum of 8 trends.
- A maximum of 10000 value pairs can be displayed in each trend.
- Trends can be based on online tags, archive tags or data from user archives.
- The online tags in a trend must have the same updating cycle.
- The archive tags in a trend must originate from a process value archive on a server, have the same updating cycle and be acquired in a continuous cycle.
- Ideal trends can be based on data from user archives.
- Tags can only be displayed as a function of time if the data source for the trends is connected over the application programming interface (API). You should use WinCC Online Trend Control to display tags as a function of time.

Resolution of a Trend Display

The number of trend values that can be displayed on the screen is limited by the screen resolution and selected size of the trend window. Therefore, when displaying trends, it is possible that there are fewer values displayed in the trend window than are actually archived.

For example, when 200 measured values are archived in a range with 100 pixels, each pixel on the screen will represent two measured values. The value shown on the screen will be that of the most recent data (most recent time stamp).



10.3.5.1 Representing trend lines

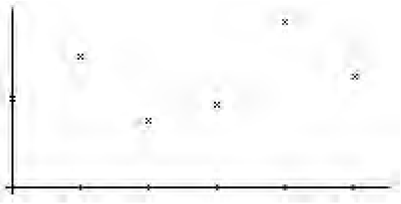
Introduction

WinCC Function Trend Control provides a number of options for representing a trend line.

To graphically display tag values, three basic representation types are available.

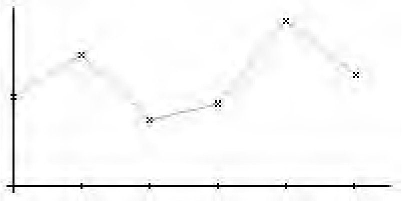
Individual Values

Value of measurement points are shown as dots.



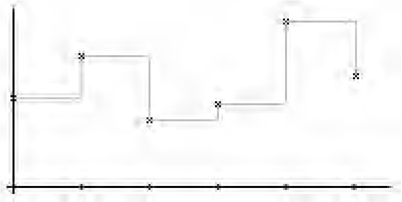
Linear Interpolation

The trend line is interpolated on a linear basis from value of measurement points. The trend is represented as either a solid line or a dashed line. The area under the curve can also be displayed in a solid color.



Stepped Trend

The trend line is determined from the value of measurement points and represented as a stepped trend. The trend is represented as either a solid line or a dashed line. The area under the curve can also be displayed in a solid color.

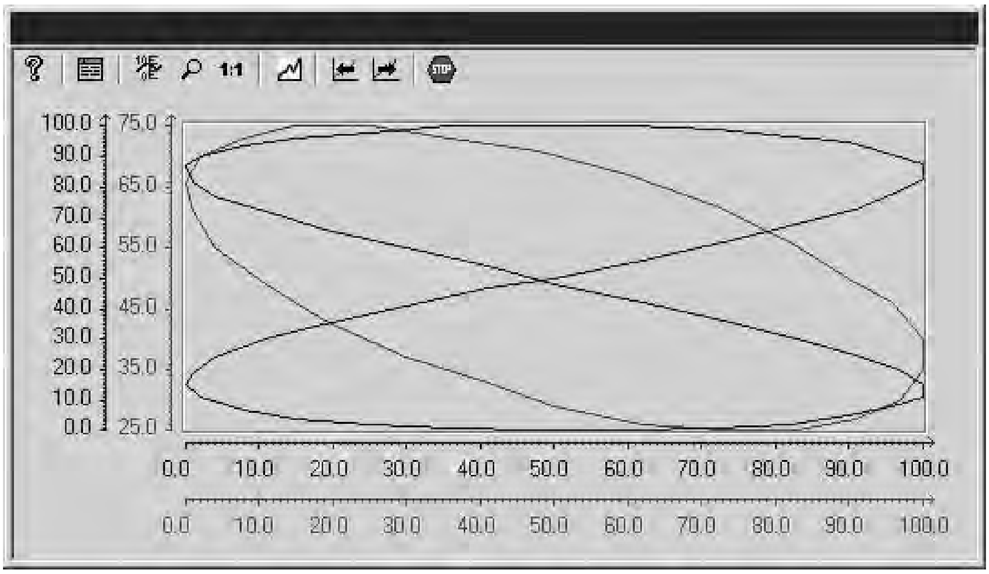


Configuration Stage

The type of representation to be used for the trend is configured in the Trends tab of the "Properties of WinCC Function Trend Control" dialog box.

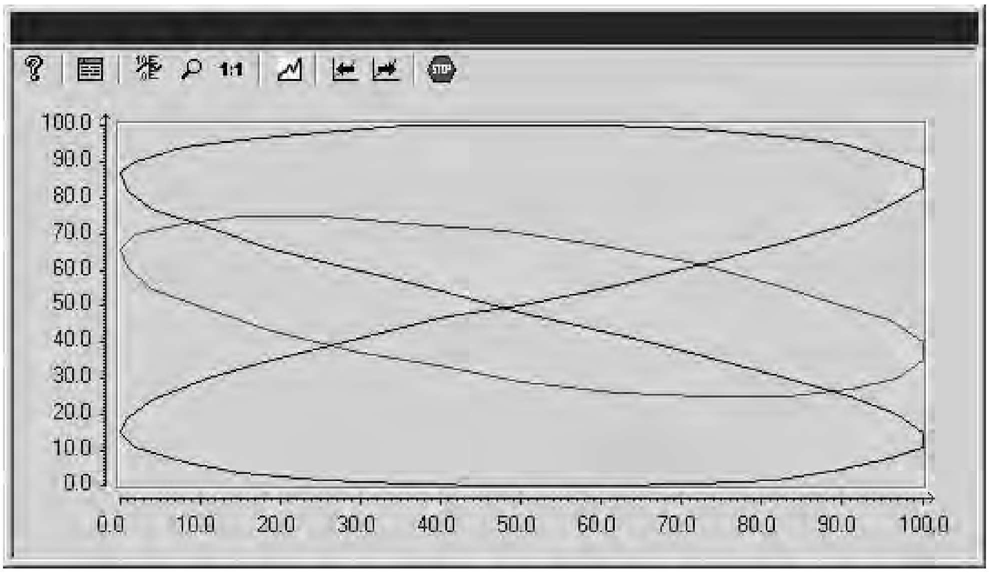
Representation using different axes

If the tag values to be displayed in a tag window differ greatly, the use of shared axes for the trend display is not recommended. The tag values can be read more easily if different axial scales are used.



Representation using common axes

If it is important to be able to compare the trend lines, the use of shared axes for the trend display is recommended. In runtime, you can also determine the exact tag values by zooming in on a segment or by inquiring about the coordinates.

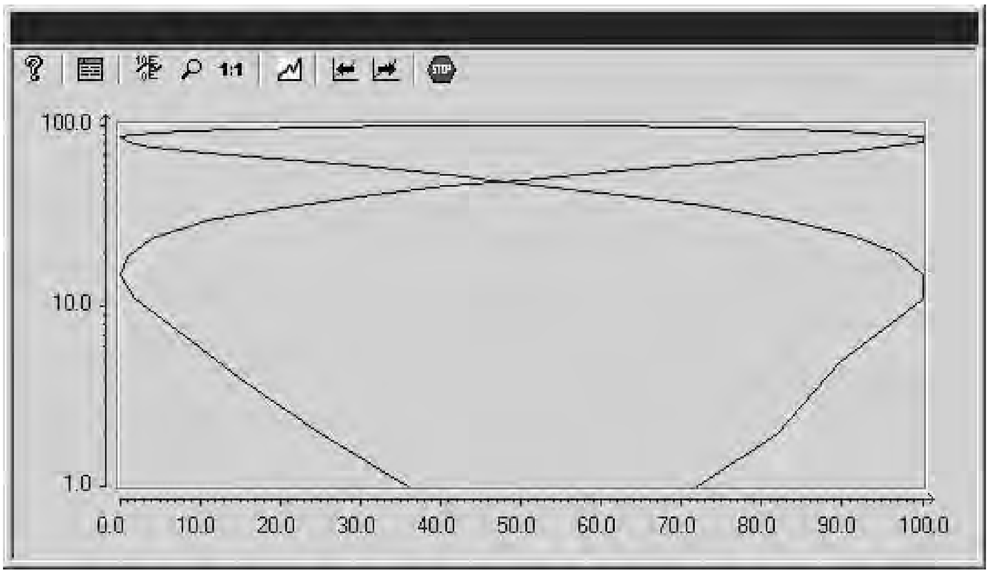


Configuration Stage

Representation using common axes is configured in the "General" tab of the "Properties of WinCC Function Trend Control" dialog box.

Representation using logarithmic axes

The axes in the trend window can be scaled on either a logarithmic or a linear basis. Representation using logarithmic axes means that no negative values can be displayed, and representation using negative logarithmic axes means that no positive values can be displayed.

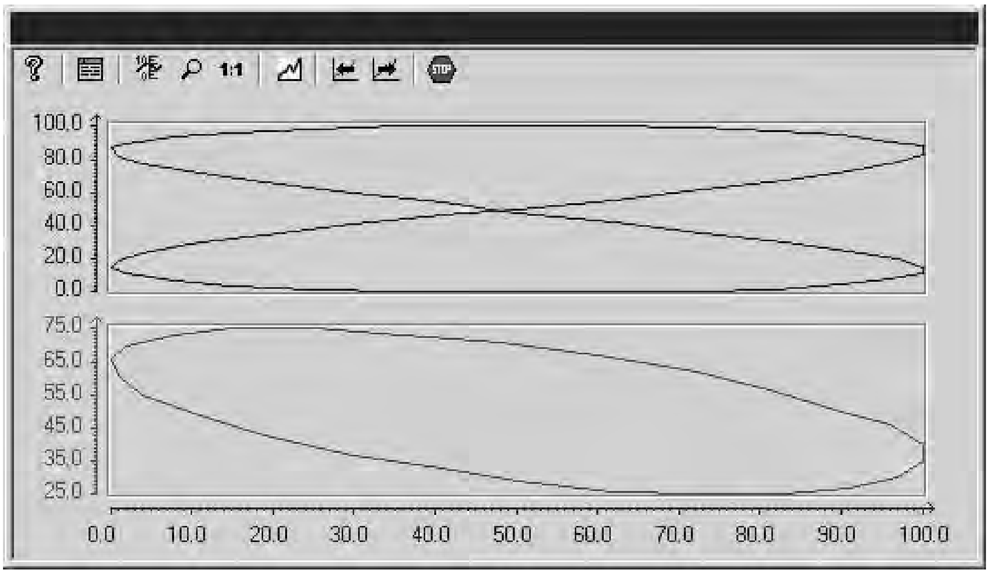


Configuration Stage

Representation using logarithmic axes is configured in the X axis or Y axis tab of the "Properties of WinCC Function Trend Control" dialog box.

Representing staggered trends

In a staggered display, the trends are displayed offset over each other in the trend window. For each trend, the value range to be displayed can be set for the Y axis.



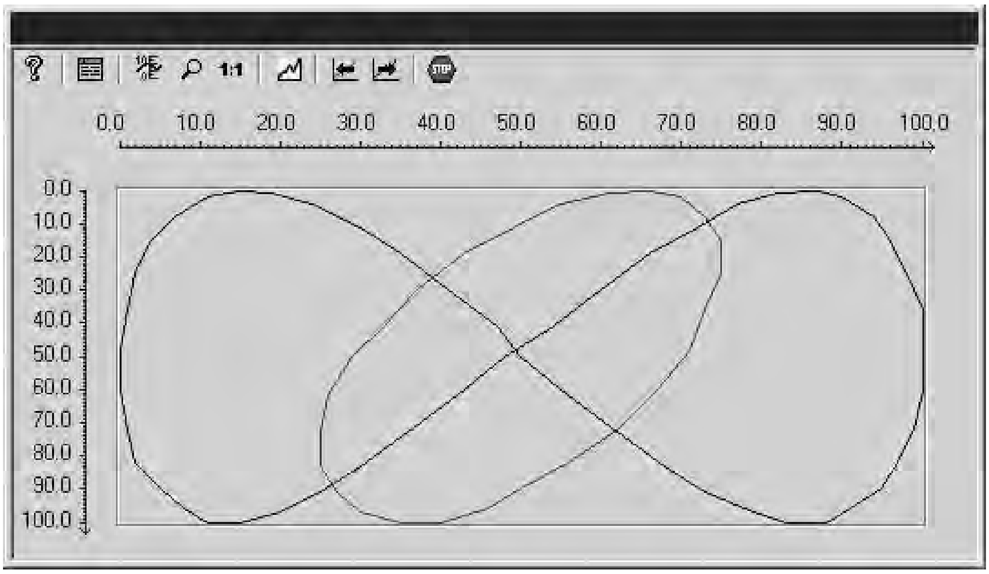
Configuration Stage

Representation of staggered trends is configured in the "General" tab of the "Properties of WinCC Function Trend Control" dialog box.

Write Direction

The "write direction" defines the direction in which the positive values on the axes are depicted. Thus the setting "from the bottom" means that positive values on the Y axis will be shown in the downward direction.

If you select either "from the top" or "from the bottom" as the direction to be written, you should use True-Type fonts within the trend window to ensure a clean display of the vertical axis labeling.



Configuration Stage

The write direction is configured in the "General" tab of the "Properties of WinCC Function Trend Control" dialog box.

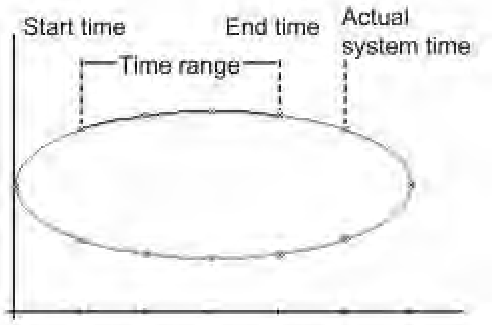
10.3.5.2 Time Range of a Trend Display

Introduction

With regard to the time ranges to be displayed, there are a number of ways to present a trend display of tags:

Static representation of a tag

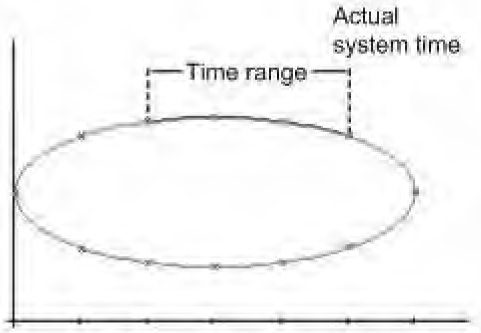
Static representation presents the course of a tag within a defined time interval, based on the archived values.



If display updating is stopped in runtime, the Data Linking tab of the "Properties of WinCC Function Trend Control" dialog box can be used to configure the time range that needs to be represented. This time range is derived either by defining a starting time and finishing time or by defining a starting time and the number of value pairs to be represented.

Dynamic representation of tags

When displaying one tag as a function of another tag in dynamic mode, the value of measurement point associated with the current system time changes along the function graph. New measurement values coming in will be included in the display.

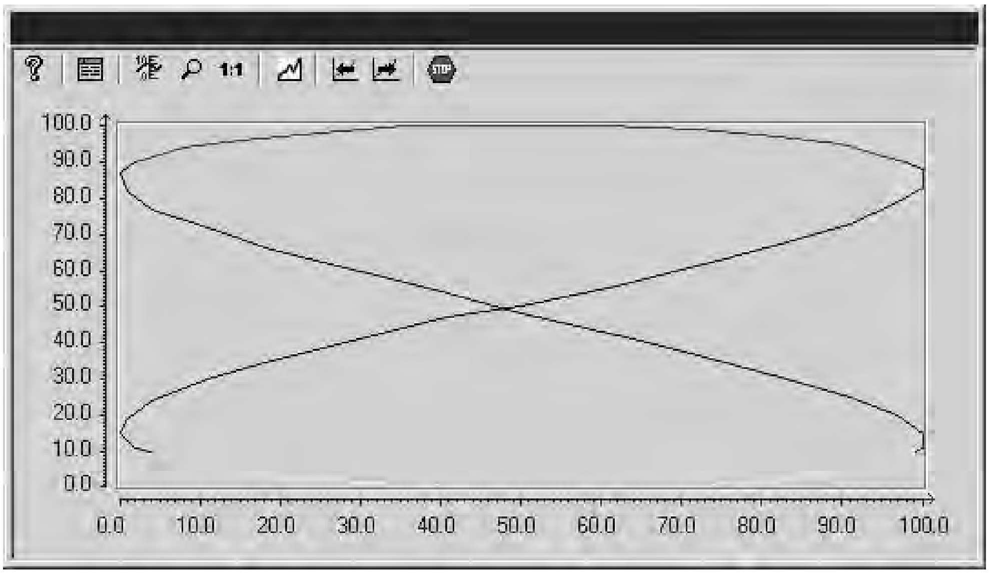


The time range to be represented is configured in the Data Linking tab of the "Properties of WinCC Function Trend Control" dialog box. This time range is derived either by defining a starting time and finishing time or by defining the number of value pairs to be represented.

10.3.5.3 Identification of special values

Introduction

In any trend, values may occur which are above or below a limiting value, or which have an uncertain status. Such special values can be highlighted in color. Values with an uncertain status are those whose initial value is not known after runtime has been activated, or for which a substitute value is used.

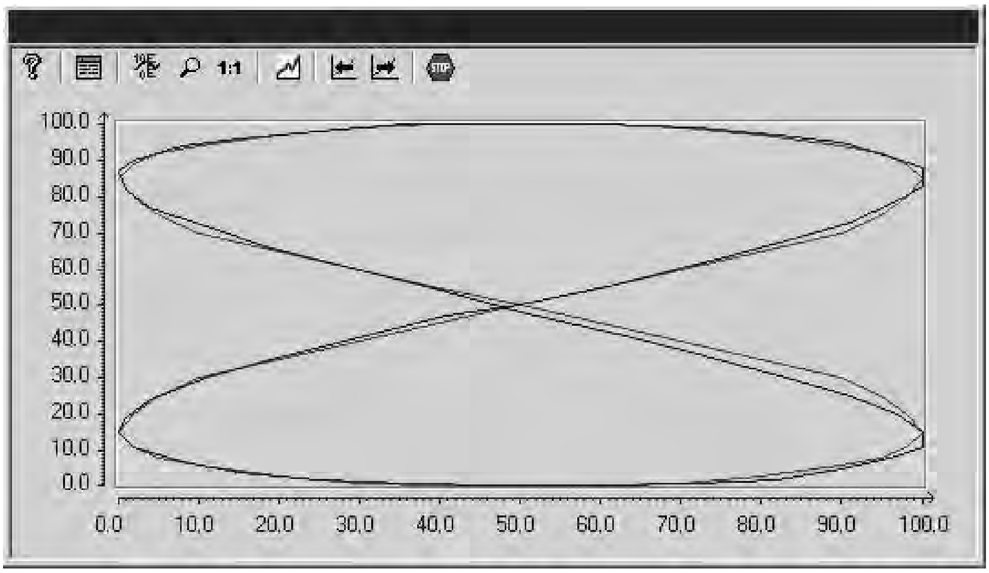


Configuration Stage

Color-coded identification of special values is configured in the Limiting Values tab of the "Properties of WinCC Function Trend Control" dialog box.

Comparison with an ideal trend

Any trend can be configured with an ideal trend. It is then possible to make an effective comparison between the actual trend line and the specified ideal trend. The values for the ideal trend must be held in a user archive. You can create user archives with the aid of WinCC option User Archives.



Configuration Stage

The values for the ideal trend are configured in a user archive.

The display for an ideal trend is configured in the Trends tab of the "Properties of WinCC Function Trend Control" dialog box.

10.3.5.4 Configuring WinCC Function Trend Control

Introduction

The steps you need to take in order to configure WinCC Function Trend Control are dependent on the tags you wish to display.

Displaying online tags

- The tags you wish to display need to be configured in WinCC Explorer.
- Link the WinCC Function Trend Control to a Graphics Designer picture and edit the properties of the ActiveX Control.

Displaying archive tags

1. The tags you wish to display need to be configured in WinCC Explorer.
2. Use the Archive Wizard to create a process value archive.
3. Configure the process value tags it will contain.
4. Link the WinCC Function Trend Control to a Graphics Designer picture and edit the properties of the ActiveX Control.

Displaying values from user archives

1. Create a user archive.
2. Configure the fields it will contain.
3. Link the WinCC user archive table element to a Graphics Designer picture and edit the properties of the ActiveX Control.
4. Link the WinCC Function Trend Control to a Graphics Designer picture and edit the properties of the ActiveX Control.
5. Enable runtime and input the display values into the user archive or import them from a csv file.

Comparison with an ideal trend

1. Create a user archive.
2. Configure the fields it will contain.
3. Link the WinCC user archive table element to a Graphics Designer picture and edit the properties of the ActiveX Control.
4. Link the WinCC Function Trend Control to a Graphics Designer picture and edit the properties of the ActiveX Control.
5. Enable runtime and input the ideal trend values into the user archive or import them from a csv file.

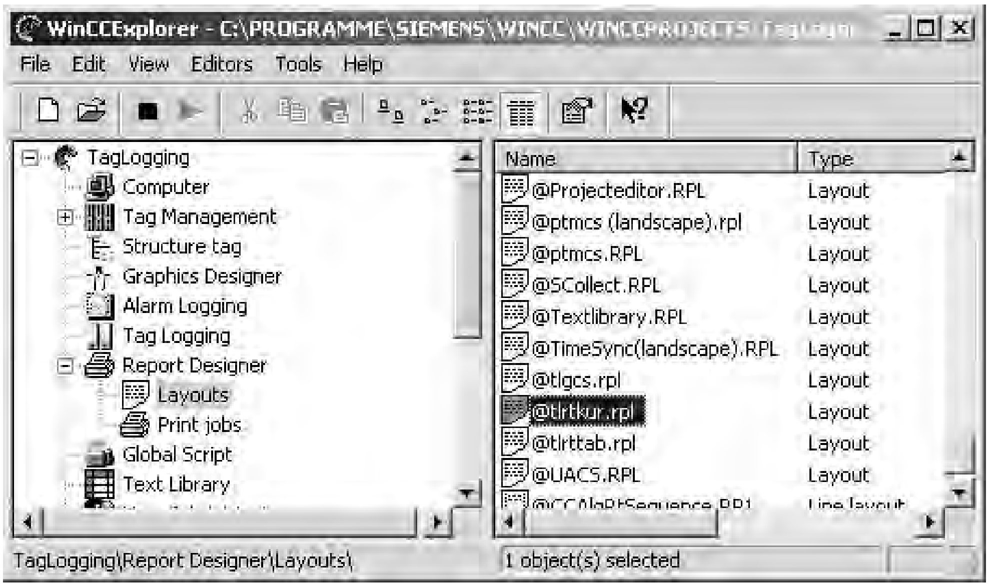
10.3.6 Process Value Output in the Log

Introduction

You can output process values in a log. For example, after every shift change you can load and print out the production figures from the process value archive. You create the logs in a separate editor, the Report Designer.

Report Designer

The Report Designer contains the prepared standard layouts for logs, which you can adapt to suit your requirements. You also use the Report Designer to create the print jobs for starting log output:



Name	Type
@Projecteditor.RPL	Layout
@ptmcs (landscape).rpl	Layout
@ptmcs.RPL	Layout
@SCollect.RPL	Layout
@Textlibrary.RPL	Layout
@TimeSync(landscape).RPL	Layout
@tlgcs.rpl	Layout
@tlrtkur.rpl	Layout
@tlrttab.rpl	Layout
@UACS.RPL	Layout
@CCAlgRtSequence.RPL	Line layout

The following prepared layouts are available for process value output in the log:

- @tlrtkur.RPL: process value output as a trend
- @tlrttab.RPL: process value output as a table

General procedure

Configuring of process value output in a log is broken down into the following steps:

1. Select process data and adapt layout: select the process data that you want to display in the log and if necessary adapt the layout of the log.
2. Configure print job: configure the print output of the log by defining the period, number of pages and the output medium.

11 Working with Cross-Reference Lists

Use the WinCC Component "Cross Reference" (cross-reference list) it is possible to do the following:

- to find all application places of specific object such as , for example, Tags, Screens
- to have the application places of an object be displayed directly, in order to change or delete them (Function "Application place jump")
- to change the names of one or several tags by the function "Linking" for tags, without any inconsistencies in the configuration. This function can also be used to search and replace Character strings in the Tag names.

The implementation of Cross Reference is an advantage, for example, in the following cases:

- Searching for the application places of deleted Process tags to change or delete the application place ("Application place jump") , for example, for output fields, in Trend Displays.
- Finding relationships between the existing process screens of a project, in order to structure the process visualisation for a project update exactly as the existing parts.

Terminology explanation

For the description of the component "Cross Reference" several terms need to be explained:

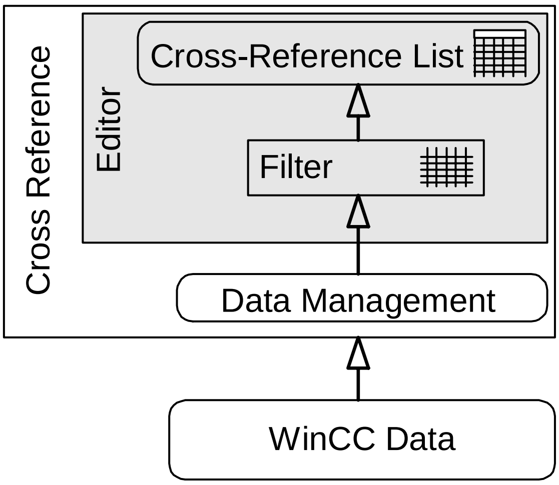
- Cross-reference
  Indicates the WinCC component for itself and is displayed in the WinCC Explorer in the editors.
- Editor
  The editor Cross Reference is started in WinCC Explorer and provides the surface for the working with cross-reference lists.
- Internal data storage
  This includes all the cross-references which need to be managed and updated. The editor uses a filter on this "complete cross-reference list" for the query specific cross-references and receives a filtered result, the cross-reference list.
- cross-reference list
  This is the result of a filter which is displayed on the editor and which has been applied on the data of the internal data storage. The cross-reference list can be saved with the filter with the help of the used editor.

- Filter
  A set of criteria that is applied to the internal data storage and result in a "filtered" cross-reference list. The filter criteria are specify with the help of the editor. .
- Result file
  This is the saved form of a cross-reference list and in addition to the data it contains also the basis filters.
- Application place jump
  This function can be used to jump from a displayed cross-reference list to the application place of a found object. Depending on the selected object, for example, the respective editor is opened.
- Relinking
  This signifies the function, which can be used to change one or also several Tag names in a limited area of the project, without any inconsistencies.

Properties

The Component Cross Reference consists of two parts:

- The editor is responsible for the manipulation of the filter criteria and the Display of the Results of the cross-reference lists. It offers also the Functions of the direct jump to the application place of a found object and that of the "Linking", that is, the Changing of Tag names.
- The second component is the internal data storage. It is responsible for the management and the updating of the relevant configuration data for the function "cross-reference list".



Use the adjustable filters to search for the following Object Types and to create the cross-reference list:

- Process tags and Internal tags
- Pictures and Picture Windows
- Project and Standard functions
- OLE Elements
- ActiveX Controls

The following table shows how the Search for specific object is supported.

	Possible usage in			
Searched Objects	Archive	Messages	Function	Picture
Tag: Process & internal	X	X	X	X
Function Project & Standard	--	X	--	--
Picture (PDL File)	--	X	X	X
OLE Elements	--	--	--	X
ActiveX Controls	--	--	--	X

The search results can be saved, printed or exported for further processing with other Windows applications (CSV or Excel Format).

A found object in the displayed cross-reference list can then be used either for the triggering of the application place jump or for the function "Linking".

The updating of the data that are relevant for the function "cross-reference list" can be done automatically or manually. Inconsistencies between the configuration management and the displayed cross-reference list are shown by the respective progress indicators or a respective warning during the creation of a cross-reference list.

Note:

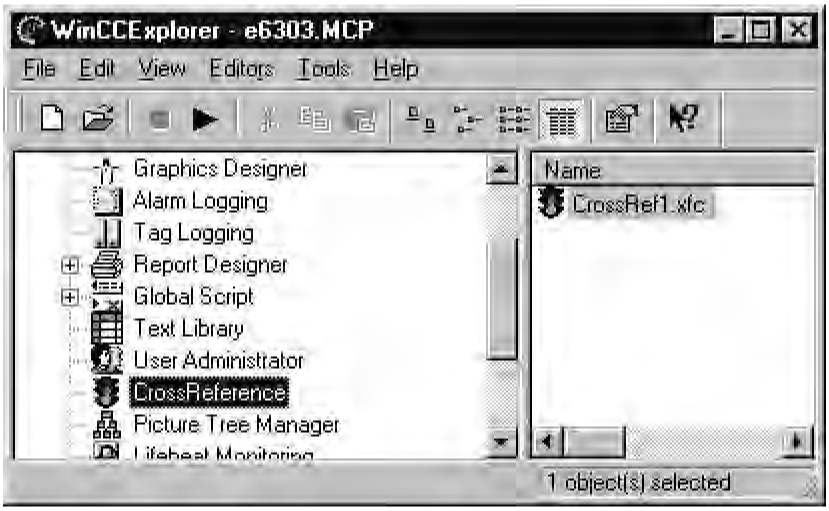
- The **Search for messages** is **not** supported by the Cross Reference, since the messages are displayed on the message windows of a screen in runtime or are used in a function as the parameter of an API activation.
  The **Search in configured messages** is supported. Here an analysis of Single messages is done according to the used Tags.
- **Tags in Functions** can be found and replaced by the Cross Reference only if the configuration regulation that has been explained in the appendix is kept for tag and picture names. This is also true for the configuration of **C-Actions in Screens**.

- For **Tags which are monitored** in Alarm Logging **for limit violation,** the assignment for a message is not included. That is, this assignment does not apply for the Cross Reference as a so-called application place. So, a tag which is, for example, monitored **only** for limit violation, is shown in the cross-reference list as "Not used".
- Cross Reference treats the application places of objects exclusively in the Context of WinCC. Therefore, no **Operands** are adopted from the STEP 5 / STEP 7 into the cross-reference list, which are based on WinCC tags, and neither is an entry possible at the application place in a STEP 5 / STEP 7 project.

The Search for **Standard and Windows objects** in Screens such as, for example, lines, circles is not supported.

11.1 Cross Reference in WinCC Explorer

The editor is visible in the navigation window of the WinCC Explorer and can be started here via the pop-up menu or via the Menu "Editors".



Used File Name

The following file names and Add Ins are used in relation with the function "cross-reference list":

- XFC (**X**re**F**-**C**lient file) for the cross-reference list
  If the entry "Cross Reference" is selected in the Navigation Window, all the files with the extention "XFC", which have been saved in the CrossRef folder in the project directory structure appear in the data window on the right.
  Each of the displayed files represents a cross-reference list with the Filter Conditions and can be opened with a double click.
- XFS (**X**re**F** **S**erver file) Contains all relevant project data in an internal data storage.
  This "complete cross-reference list" has been saved in the project folder and is normally not displayed for the user.

11.2 Data Window

In the editor the result of a filter is shown on the internal data management as cross-reference list in table format.

In a line of the table the following are shown:

- Marks on the reference type, (Used, Not used, Not existing)
- the searched object (e.g. Tags, Pictures)
- where it is referred to (Type, Containing element, Object),
- and on which properties it has an effect (properties/action).

The content of the Columns depend on the searched Object type.

Sorting the Table can be done by a mouse click on the respective column heading increasing or decreasing and it is shown by a plus or minus sign in the Column Heading.

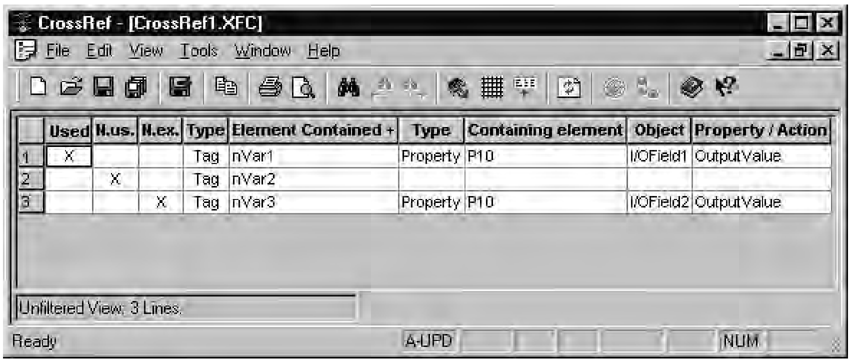
If the display of the Table is filtered through the function "AutoFilter", the respective column heading is shown colored and the criterion is shown in the title .

The Display of the Columns can be changed by the function "Hide/unhide columns" in the menu "View" or also by the pop-up menu of the column heading.

The Column Width itself can be changed by selecting and moving the limits of the column header with the mouse or by the menu item "Tools / Settings". The Column Widths can for instance also be set individually or collectively by the pop-up menu of the column heading to an optimum value, to display all the Data in the Table.

The contents of the Columns "Used", "Not used" and "Used, Not existing" are specified also by the selection in the sub menu of "View / Displays ": If one of these sub items is elected, only one display of the respective Line is done in the Table. For the other, not selected Options, the Column headings are displayed in grey and the Columns stay empty.

In the status bat at the bottom border of the data window you will see whether the view of the Table was done without filters or whether it was filtered, e.g. by the function "Autofilter" or the menu item "View / Displays".



	Used	N.us.	N.ex.	Type	Element Contained +	Type	Containing element	Object	Property / Action
1	X			Tag	nVar1	Property	P10	I/OField1	OutputValue
2		X		Tag	nVar2				
3			X	Tag	nVar3	Property	P10	I/OField2	OutputValue

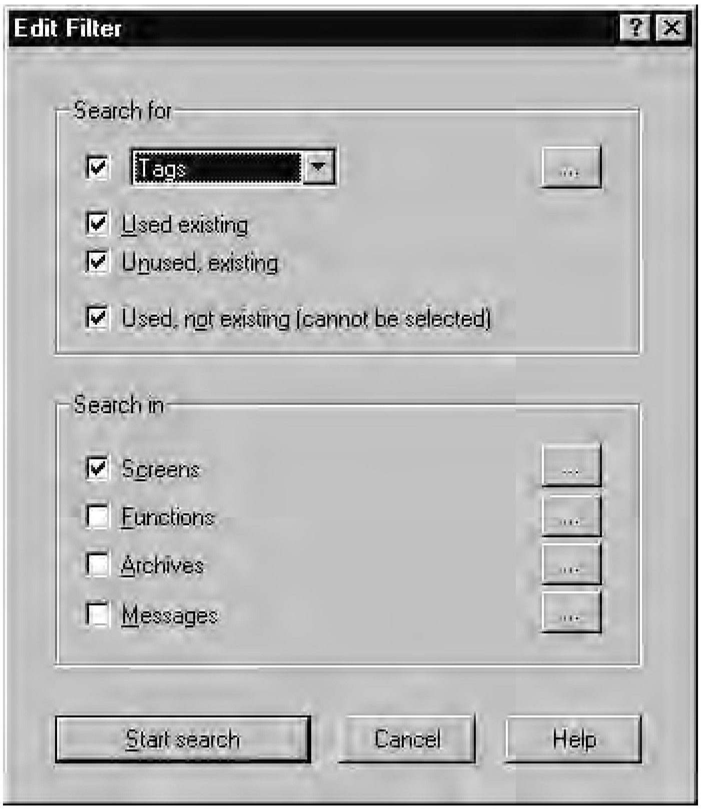
Unfiltered View: 3 Lines.

Example for the structure of the table for the search for tags in screens

11.3 Create a Cross Reference List (Filter Selection Dialog)

The search criteria for a cross-reference list are set in the filter selection dialog.

The dialog appears during the selection of the menu items "File / New" or "Edit / Filter".



Edit Filter

Search for

☑ Tags

☑ Used existing

☑ Unused, existing

☑ Used, not existing (cannot be selected)

Search in

☑ Screens

☐ Functions

☐ Archives

☐ Messages

Check box next to the selection window

If this check box is selected, all objects of the selected type will be searched for. If this is not desired, you can select a part by the button "...". In this case the check box appears "grey highlighted".

Selection Window

Here you can select the object types you want to search for. You can select between the object types tags, pictures, functions, OLE objects and ActiveX controls.

Check box "Used existing" , "Unused, existing" , "Used, not existing"

By enabling these check boxes you can modify the search for the selected object type according to the specified selected terms.

Term	Description
Used existing	All objects of the current cross-reference list which exist and are used (referred to) in another object are displayed.
Unused, existing	All objects which exist and are not referred to in another object are displayed.
Used, not existing	All objects which contain a reference to a not existing object, e.g. a picture where a deleted tag is referred to, are displayed. (so-called open reference)

Check box for the object types

For each object type that is supported for the search for objects, a check box is provided here. After this has been set, the references are searched in all or the selected objects of this type.

If no part has been selected in the selection dialog, the check box appears "grey highlighted".

Buttons "..." to open a selection dialog

The user can use the button "..." to limit the search for or in objects accordingly.

This is done by the corresponding object type dependent selection dialogs.

- If "Tag" was selected as searched object type, the tag selection dialog will open.
- or a search in the object types functions, archives and messages, a selection dialog is opened by the respective button "...", which e.g. shows all existing PDL files in the WinCC project.

In the types OLE objects and ActiveX controls the search can be made only for "All", no dialog 's opened here.

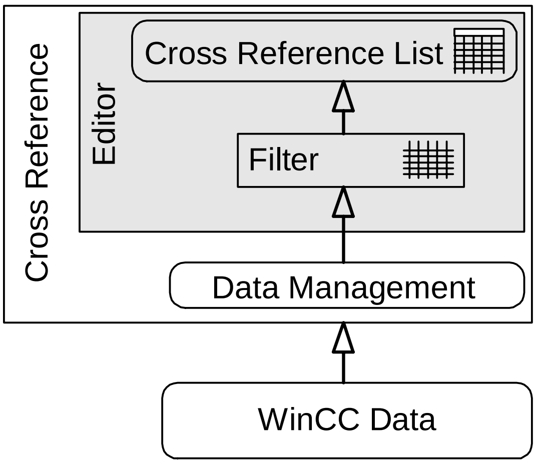
Note:

Combinations of illogical filter criteria are suppressed in the dialog by the default settings. So, either the button "Start search" stays inactive or e.g. during the search for pictures the check box for "Archives" is not provides in the area "Search in".

11.4 Update a Cross Reference List

The cross-reference list that was displayed in the data window is based on the data of the internal data management and a filter that was used on it.

However, this display does not have to correspond to the current configuration of WinCC, since an updating of the internal data management can be done updating or manually, while the updating of the displayed cross-reference list is done always manually.



The following functions can be used for the updating:

Function	Description
Automatic update (Menu "Tools")	switches the "Automatic Update" of the internal data management on or off permanently. The current setting can be recognised at the mark next to the menu item and at the display "A-UPD" in the status bar. All changes of the configuration are stored starting with the time of the enabling of the function.
Update Data Management (Menu "Tools")	triggers only the single updating of the internal data management, the display of the cross-reference list is not updated automatically at the same time.
Update ("View" Menu)	single updating of the displayed cross-reference list, that is, the current filter criteria are applied to the internal data management again and the result is displayed. Corresponds to the menu item "Edit / Filter", however without the previous filter selection dialog.

Update data management and view (Menu "Tools")	triggers the updating of the internal data management and the view of the cross-reference list.

If the Automatic Update is disabled, Inconsistencies can occur between configuration, the internal data management and displayed cross-reference list.

Possible inconsistencies are reminded as follows:

- Inconsistency between the WinCC project and the internal data management by an appropriate warning during the creation / change of a cross-reference list and progress indicator "UPD!"
- Inconsistency between the internal data management and the displayed cross-reference list
  by the progress indicator "OLD"

In order to keep a display of the cross-reference list updated to the latest configuration status, you can select to do one of the following items:

- Restart the function "Automatic Update" (permanent) and triggering the function "Update data management and view"
  The manual function is necessary because of two reasons. On the one hand, because after the starting moment of the Automatic Update all changes of the configuration are managed in the internal data management, for the period of before the reactivation up back to the last updating of the internal data management however inconsistencies can occur. In addition, the activation does not lead to any updating of the cross-reference list .
- Manual, one time triggering of the function "Update data management and view".

Note:

The functions "Update data management and view" and "Update Data Management" can take some time (possibly hours), depending on the memory equipment and the project size.

If the Windows-Explorer is used to copy a New picture into the project folder, the Cross Reference does not recognise any inconsistency of the cross-reference list and in this case the user must start the function "Update Data Management" manually.

After updating you must also remember that the Display could possible be affected by the set filter criteria through the "AutoFilter" .

However, you could be wanting to turn off the Automatic Update:

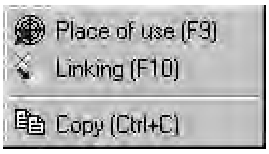
- In order to "freeze" a specific configuration status in the internal data management and create a specific cross-reference lists of it.
- In order to save system resources, since the disabled "Automatic Update" does not require resources for the updating of the data management always.

For a saved cross-reference list there can be an inconsistency of the data, since these saved lists are based on the internal data at the time of the saving and/or the internal data management could have been not current.

11.5 Jump to the Editors ("Application Place Jump")

When application places are displayed in a cross-reference list of the editor it is possible to jump to this application place in the respective WinCC editor.

In order to do so you must select the respective object in the data window of the editor. Use the menu item "Edit / Go to the application place" or the pop-up menu to trigger the so-called application place jump.



The following table shows, which editor is started during the entry into the application place:

Application place	Editor	Action during the application place jump
Archive	Tag Logging	Start of the editor, no further action
Message	Alarm logging	Start of the editor, no further action
Function (Project and Default)	Global Script	Start of the editor and display of the function For internal functions there is no action.
Picture object	Graphics Designer	Start of the editor and focussing of the object
Tag	WinCC Explorer	Explorer is set to the foreground, no further action

Example:
The editor shows a dynamics of a property in a graphic object of a screen. The dynamics was created by a process tag and therefore its usage is shown in the editor accordingly. The Graphics Designer is opened by the application place jump from the editor and the respective object is moved to the view.

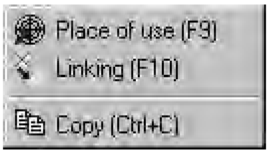
11.6 Linking of a tag

Another usage of the cross-reference lists is the function "Linking". This can be used to change the name of one or several tags in the application places of the project, without inconsistencies occuring in the naming of the used tags.

The linking can also be done on a non existing tag. Cross Reference recognizes this and an a dialog is opened automatically to create this tag.

Using the function "Linking" you can also find and replace single or multiple characters in the tag names.

In order to link you must select the tag, the application place or the respective object in the data window of the editor. Then use the menu item "Edit / Linking " or the pop-up menu to start the linking function.



Note:

If the selected tag is not used, the item "Linking" appears disabled. In this case, a renaming can be done by the WinCC explorer.
In order for the Cross Reference to recognize and rename the tag names, which are used in functions, the configuration regulation in the appendix must be kept. Existing functions must be modified accordingly.

12 Documentation of Configuration and Runtime Data

12.1 Project Documentation

12.1.1 Documentation of Configuration and Runtime Data

Introduction

To document configuration and runtime data, reports and logs are created in WinCC, for which predefined layouts are supplied. These predefined layouts will cover most of the cases in which you need to document data. You can use Report Designer to edit predefined layouts or create new layouts to suit your requirements and wishes.

Applications

The reporting system allows you to output:

- Configuration data in a report
- Runtime data in a log

Usage

The documentation of configuration data is referred to below as project documentation. It allows the configuration data of a WinCC project to be output in a report.

The documentation of runtime data is referred to below as runtime documentation. It allows the process data to be output in a log during runtime. In order to output the runtime data, the corresponding application must be in runtime.

Report Designer makes print jobs available for outputting the reports and logs. The scheduling, output medium and extent of output definitions can be found in the print jobs.

The dynamic objects of Report Designer are used for data output. These dynamic objects are associated with the corresponding applications.

Selection of data for output is application-dependent and is carried out when the layout is created, when the print job is created, or when printing is initiated.

Not all applications offer all three data selection options.

The dynamic objects are supplied with the current values during output of the reports and logs.

The structure and configuration of the reports for project documentation and logs for runtime documentation are largely identical. The essential differences lie in the connection of the data source with the dynamic objects and in the initiation of printing.

Output Media

The reports and logs can be output:

- To a printer
- To a file
- To the screen

Output Formats

The reports and logs can be output with the following layouts:

- A page layout
- A line layout (message sequence report only)

Editors for the Layouts

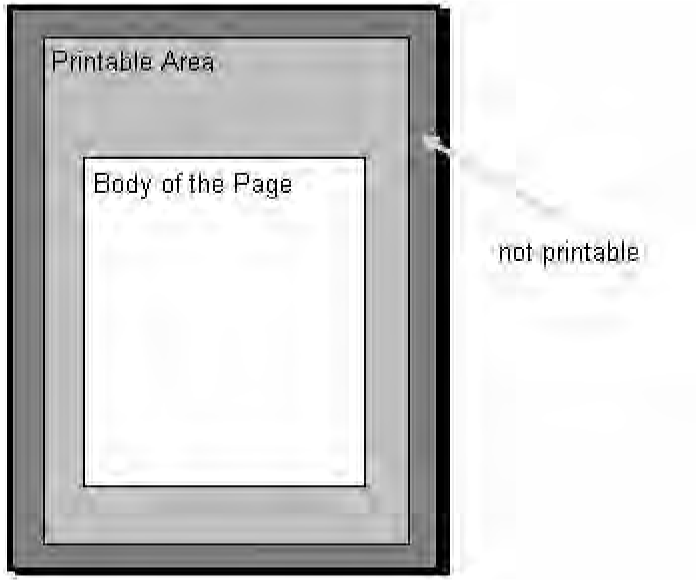
Report Designer provides the page layout editor for editing page layouts. Both the page layouts for the project documentation reports and the logs for runtime documentation are configured in the page layout editor.

Report Designer provides the line layout editor for editing line layouts. Only line layouts for the output of the message sequence report are configured in the line layout editor.

12.1.2 Structure of Reports and Logs in the Page Layout

How the Areas of a Page Layout Are Divided

Page layouts are divided geometrically into a number of different areas. The page range corresponds to the total area of the layout. The print margins can be specified for this area. It makes sense to begin by configuring the margins of the printable area for the header and footer or for a company logo and only then to configure the rest of the printable area for the output of report data. This area defined within the printable area is referred to as the "body of the page".



The report and log layout contain a static layer and a dynamic layer. The static layer contains the header and footer of a layout for outputting the company name, company logo, project name, layout name, page number, time, etc. The dynamic layer contains the dynamic objects for outputting the configuration and runtime data.

Only static and system objects can be inserted in the static layer. Both static and dynamic objects can be inserted in the dynamic layer.

Objects that are inserted in the dynamic part of a page layout are extended dynamically, if required. If an object in the dynamic table is supplied with data, for example, this table is extended to allow all the data in the table to be output. If there are other objects in the dynamic part of the layout, these are moved accordingly. Objects that are to have a fixed position must therefore be inserted in the static part of the layout.

Pages in the Page Layout

Each page layout consists of three pages:

- The cover sheet
- The report contents
- The final page

Cover sheet
The cover sheet is a fixed component of a page layout. It is therefore possible to design a separate cover sheet for each report.

Report contents
The structure and contents of the report for output are defined in this part of the page layout. There are static, dynamic, and system objects available for defining the report contents.
The report contents have a static and a dynamic component (configuration layer).
If necessary, the dynamic part of the contents of the report is spread across the various subsequent pages at output, since it is not known until the time of output how much data there is.

Final page
The final page is a fixed part of a page layout. It is therefore possible to design a separate final page for each report.

The creation and output of the cover sheet and final page are optional. By default, a cover sheet is output, but a final page is not output. You can edit a page layout in the page layout editor.

The cover sheet and final page also have a static and a dynamic component (configuration layer).

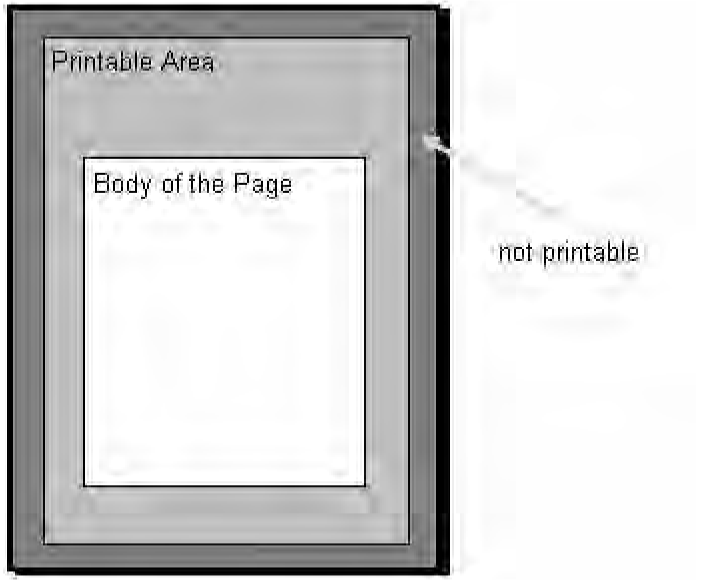
Important:

If you use dynamic objects on the cover sheet or final page, only some of this data may be included at output. This happens when the data of dynamic objects do not fit on a single page. The cover sheet and final page consist of a single page and never contain a page break.

12.1.3 Structure of the Logs in the Line Layout

How the Areas of a Line Layout Are Divided

Line layouts are divided geometrically into a number of different areas. The page range corresponds to the total layout area. The print margins can be defined for this area. It makes sense to begin by configuring the margins of the printable area for the header and footer or for a company logo and only then to configure the rest of the printable area for the output of the log data. This area defined within the printable area is referred to as the "body of the page".



The line layouts contain a static layer and a dynamic layer. The static layer contains the header and footer for outputting the company name, project name, layout name, etc. as pure text. The dynamic layer contains a dynamic table for outputting Alarm Logging messages.

How a Page Is Divided

Each line layout consists of three areas:

- The header
- Log contents (table)
- Footer

Header — The header is a fixed component of a line layout that is output together with every page. A header in the line layout can consist of up to 10 lines. Graphics cannot be inserted.

Log contents (table) — The structure and contents of the log for output are defined in this part of the line layout. The Alarm Logging selection options and filter criteria for defining the contents of the log are available to you for alarm output.
The design depends on the width of the individual columns and of the font size set.

Footer — The footer is a fixed component of a line layout that is output together with every page. A footer in the line layout can consist of up to 10 lines. Graphics cannot be inserted.

The creation and output of a header and footer are optional. By default, 3 lines each are output for the header and footer.

Use of the Line Layout

The layout in line format is used exclusively for the message sequence report. You edit it in the line layout editor. For a line layout there is only one valid print job, which is integrated in the WinCC on a fixed basis. In order to be output, the message sequence report must be activated in the startup list of the computer carrying out the logging.

12.1.4 Print Jobs in WinCC

Introduction

Print jobs in WinCC are of central importance to the output of project and runtime documentation. In the layouts you configure the external appearance and data supply for output. In the print jobs you configure the output medium, how much is to be printed, the time at which printing is to start, and other output parameters.

Each layout must be associated with a print job in order to be output. Various print jobs are provided in WinCC for the purpose of project documentation. These system print jobs are already associated with the corresponding WinCC applications. They can be neither deleted nor renamed.

You can create new print jobs in WinCC Explorer in order to output new page layouts. A special print job is provided in WinCC for outputting line layouts. Line layouts can only be output using this print job. It is not possible to create a new print job for a line layout.

12.2 Introduction to Project Documentation

Introduction

To create project documentation in WinCC, you can start the output of the reports from within the various WinCC editors. For this purpose, the editors are associated with print jobs, which call a predefined layout for output. Default parameters for output are already set in the predefined layouts. More information can be found on the output parameters in the sections on project documentation in the various WinCC components.

Components from Which Project Documentation Can Be Called

You can start project documentation directly from the following components of WinCC:

- WinCC Explorer
- Graphics Designer
- Alarm Logging
- Tag Logging
- Global Script
- Text Library
- User Administrator
- User Archives
- Time Synchronization
- Horn Editor
- Picture Tree Manager
- Lifebeat Monitoring
- OS Project Editor

The reports for project documentation are output by the user. On the File menu of the various WinCC components you will find the following commands:

Menu Command	Function	Special Characteristics
Print Project Documentation	Prints a report immediately. The layout that is set in the print job is used. The report is output to the printer/file that is set in the Print Job Properties dialog.	A printer selection dialog is not displayed; printing starts immediately. Depending on the size of the project, this process can take some time. The process cannot be canceled while the data is being read from the system.
View Project Documentation	Opens a preview of the report to be output.	Depending on the size of the project, this process can take some time. The process cannot be canceled while the data is being read from the system.
Project Documentation Setup...	Opens the Print Job Properties dialog. You can specify the layout to be used, the printer or file to which the report is to be output, and which pages are to be output.	More information can be found in the sections "Print Jobs in WinCC" and "Changing an Existing Print Job".

12.2.1 Project Documentation in WinCC Explorer

Introduction

WinCC Explorer is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

Data for Output

The following data are available for project documentation in WinCC Explorer:

Object	Output Options	Function
Computer	-	Serves to output the name of the computer whose data is to be documented.
Tags	Selection of tags Selection of tag groups Selection of tag parameters Format	Serves to output tags, tag groups, and tag parameters from the tag management system. Formats the tag list.
Connections	Selection of connections in a selection dialog	Serves to output the configured connections to subordinate controllers.

12.2.2 Project Documentation in Graphics Designer

Introduction

Graphics Designer is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

The print jobs for project documentation can be started from the menu of the WinCC component or directly in WinCC Explorer. However, there is an important difference to note in the case of Graphics Designer project documentation. When a print job is started in Graphics Designer, the data of the open picture are output. If the print job is started in WinCC Explorer, the data of all the Graphics Designer pictures in the project are output. This applies to all the documentable parameters of Graphics Designer pictures.

Data for Output

The following data are available for project documentation in Graphics Designer:

Actions in Graphics Designer

Object	Output Options	Function
Info	The object is available for: - Actions on the property - Actions on the object	Serves to output general information such as the picture name, object name, and property name.
Trigger description	The object is available for: - Actions on the property	Serves to output the trigger type.
Source text	The object is available for: - Actions on the property - Actions on the object	Serves to output the source text of an action.

Graphics Designer Picture Data

Object	Output Options	Function
General display	Element selection: All available elements The selected element	Serves to output the view of a Graphics Designer picture in a report.
Picture statistics	Selection of statistics data for the picture	Serves to output statistics data on a Graphics Designer picture.
Attributes	Selection of attributes Selection of table data	Serves to output the attributes of a Graphics Designer picture
Embedded objects	Object selection	Serves to output the embedded objects in a Graphics Designer picture.
Actions	Selection of action types	Serves to output the actions in a Graphics Designer picture.
Trigger	Object selection Selection of actions	Serves to output the triggers for the actions in a Graphics Designer picture.
Source text	Selection of actions	Serves to output the source text of actions in a Graphics Designer picture.
Direct connections	Element selection: All available elements The selected element	Serves to output the direct connections in a Graphics Designer picture.

Graphics Designer Object Data

Object	Output Options	Function
Object statistics	Object selection Selection of statistics on the object	Serves to output statistics data on objects in a Graphics Designer picture.
Attributes	Object selection Selection of attributes Selection of table data	Serves to output the attributes of objects in a Graphics Designer picture.
Actions	Object selection Selection of actions	Serves to output the actions on objects in a Graphics Designer picture.
Trigger	Object selection Selection of actions	Serves to output the triggers for the actions on objects in a Graphics Designer picture.
Source text	Object selection Selection of actions	Serves to output the source text of actions for the selected objects.
Direct connection	Object selection	Serves to output the direct connections of objects

Points to Note About Object Data Output

- Control-Object slider: The following applies to the "PictureBack" and "PictureThumb" attributes in project documentation: If there is no picture entered, a hyphen (-) is output in the project documentation. If a picture is entered, "none" is output.
- Control-Object Push Button: The following applies to the "PictureSelected" and "PictureUnselected" attributes in project documentation: If there is no picture entered, a hyphen (-) is output in the project documentation. If a picture is entered, "none" is output.

12.2.3 Project Documentation in Alarm Logging

Introduction

The Alarm Logging editor is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

Data for Output

The following data are available for project documentation in the Alarm Logging editor:

Object	Output Options	Function
Message classes	-	Serves to output the existing message classes with their properties.
Single messages	Selection of single messages	Serves to output the selected single messages with their properties.
Archives	-	Serves to output the existing message archives with their properties.

Data of the Message Blocks

Object	Output Options	Function
System blocks	-	Serves to output the existing system blocks with their properties.
User text blocks	-	Serves to output the existing user text blocks with their properties.
Process value blocks	-	Serves to output the existing process value blocks with their properties.

Data of the Group Messages

Object	Output Options	Function
Message class	-	Serves to output the group messages of message classes.
User defined	-	Serves to output user-defined group messages.

12.2.4 Project Documentation in Tag Logging

Introduction

The Tag Logging editor is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

Data for Output

The following data are available for project documentation in the Tag Logging editor:

Object	Output Options	Function
Timers	-	Serves to output the existing timers with their properties.

Data of the Archives

Object	Output Options	Function
Process value archive	Archive names Process value archive data	Serves to output the configuration data of process value archives
Compressed archive	Archive names Process value archive data	Serves to output the configuration data of compressed archives

Object	Output Options	Function
Process value archive tag	Tags Tag data	Serves to output configuration data of the selected process value archive tags
Compressed archive tag	Tags Tag data	Serves to output configuration data of the selected compressed archive tags

12.2.5 Project Documentation in Global Script

Introduction

There are two editors available in Global Script: the C editor and the VBS editor. Both of these editors are prepared for project documentation. If you open an action, a function, or a module in one of these editors, the following menu commands are available to you on the editor's "File" menu:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

The output applies to the the open function, action, or module.

Documentation of all the actions, project functions, and standard functions can be output by means of three of the print jobs listed below. The output is started in WinCC Explorer.

The same predefined layouts are used to output the project documentation in the C editor and the VBS editor. The identifier "Language: C Script" or "Language: VB Script" is output in the information box at output. This indicates which of the two editors the documentation comes from.

Note:
The documentation of all actions, functions, and modules can be very extensive. To check the number of pages, open the print preview of the print job, and check the page number there (format: page 1 of x).

Data for Output

The following data are available for project documentation in the Global Script editor:

Object	Output Options	Function
Info	The object is available for: - Actions - Standard functions - Project functions - Standard modules - Project modules	Serves to output general information such as who it was created/changed by, the creation date, data of change, version, and a comment.
Trigger description	The object is available for: - Actions	Serves to output the trigger type.
Source text	The object is available for: - Actions - Standard functions - Project functions - Standard modules - Project modules	Serves to output the source text of an action/function or a module.

12.2.6 Project Documentation in the Text Library

Introduction

The Text Library is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

Data for Output

The following data are available for project documentation in the Text Library:

Object	Output Options	Function
General data	-	Serves to output the configuration data of the Text Library.
Text table	Language selection for selecting the languages for output	Serves to output the texts from the Text Library.

12.2.7 Project Documentation in the User Administrator

Introduction

The User Administrator editor is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation
- View Project Documentation
- Project Documentation Setup

Data for Output

The following data are available for project documentation in the User Administrator editor:

Object	Output Options	Function
Groups	-	Serves to output the user groups from the User Administrator.
Functions	-	Serves to output the texts from the configured functions from the User Administrator.

12.2.8 Project Documentation in the Time Synchronization Editor

Introduction

The Time Synchronization editor is prepared for project documentation. The following buttons are available to you in the editor's dialog:

- Print
- Preview
- Setup

Data for Output

The following data are available for project documentation in the Time Synchronization editor:

Object	Output Options	Function
Devices	-	Serves to output the devices and services used.
General	-	Serves to output the general settings in the Time Synchronization editor.

12.2.9 Project Documentation in the Horn Editor

Introduction

The Horn Editor is prepared for project documentation. The following buttons are available to you in the editor's dialog:

- Project Documentation
- Print
- Preview
- Setup...

Data for Output

The following data are available for project documentation in the Horn Editor:

Object	Output Options	Function
Horn tags	-	Serves to output the assignment of tag and message class.
General settings	-	Serves to output the general settings in the Horn Editor.
Horn signals	-	Serves to output the assignment of signals or signal modules to tags.
Additional reset tags	-	Serves to output multiple acknowledgment via additional acknowledgment tags.

12.2.10 Project Documentation in the Picture Tree Manager

Introduction

The Picture Tree Manager is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print Project Documentation Hierarchy
- View Project Documentation Hierarchy
- Setup Project Documentation Hierarchy
- Print Project Documentation Group Display
- View Project Documentation Group Display
- Setup Project Documentation Group Display

Data for Output

The following data are available for project documentation in Picture Tree Manager:

Object	Output Options	Function
Picture Tree Manager table	-	Serves to output the configured container hierarchy and the pictures it contains.
Group displays table	-	Serves to output the configuration data of the Group Display.

12.2.11 Project Documentation in Lifebeat Monitoring

Introduction

The Lifebeat Monitoring editor is prepared for project documentation. The following commands are available on the "File" menu for this purpose:

- Print
- View
- Printer Setup

Data for Output

The following data are available for project documentation in the Lifebeat Monitoring editor:

Object	Output Options	Function
Lifebeat Monitoring table	-	Serves to output the configuration data from Lifebeat Monitoring.

12.2.12 Project Documentation in the OS Project Editor

Introduction

The OS Project Editor is prepared for project documentation. The following buttons are available for it on the General tab:

- Print
- View
- Printer Setup

Data for Output

The following data are available for project documentation in the OS Project Editor:

Object	Output Options	Function
ProjectEditor table	-	Serves to output the coordinate settings of the monitor, overview area, working area, and key area.
TopField table	-	Serves to output the settings for the display of circular pictures in the runtime window.
TopfieldConfig2 table	-	Serves to output the settings for the display of group pictures and general pictures in the runtime window.
TopfieldConfig3 table	-	Serves to output the settings for the display of trend groups in the runtime window.
AlarmConfig table	-	Serves to output the settings for the display of the small alarm window.
AreaButton table	-	Serves to output the settings for the buttons for switching between areas.
HiddenArea table	-	Serves to output the settings for hidden areas.
RestData table	-	Serves to output the general settings in the OS Project Editor.

12.3 Introduction to Runtime Documentation

Introduction

A number of options are available for logging runtime data in WinCC. Some WinCC controls for displaying runtime data have a configurable option for the log functions. If this option is selected in a WinCC control, the current view of this control is output directly by means of the integrated print job. The following WinCC controls have a configurable option for the log functions:

- WinCC Alarm Control
- WinCC Online Trend Control
- WinCC Online Table Control
- WinCC User Archive Table Control

For the independent logging of runtime data the corresponding layouts must be configured and associated with a print job. The data for output is specified and the output options set in the layout. Scheduling and how much is to be printed is specified in the print job associated with the layout.

A layout can also be used by several print jobs in order, for example, to output to different media or to take into account different time selections for the data. The print jobs can be called in different ways. The following calls are configurable:

- Call via a configurable option in one of the above-mentioned WinCC controls
- Call via a configurable option in a WinCC picture
- Call via the print job list application window in a WinCC picture
- Call via a configured cycle in the print job
- Call via configured start parameters in the print job
- Call via a script function

Components for Runtime Documentation

You can document runtime data from the following components of WinCC:

Alarm Logging Runtime

Log Object	Function
Message sequence report	The message sequence report starts at the same time as WinCC runtime and outputs a chronological list of all the messages that occur.
Message report	The message report outputs all the messages in the current message list in Alarm Logging.
Archive report	The archive log serves to output messages that are stored in the message archive.

Tag Logging Runtime

Log Object	Function
Tag table	The information and contents of the tags from process value and compressed archives are logged in the Tag Logging tag table. The data is output in tabular form.
Trend	The trend outputs the contents of the tags from process value and compressed archives in the form of a trend.

User Archives Runtime

Log Object	Function
User Archives table	The User Archives table outputs the information and contents of user archives and views.

Further Runtime Documentation Options

Logging of CSV Files

Log Object	Function
CSV provider table	The CSV provider table logs data from a file in CSV format. The data is output in tabular form.
CSV provider trend	The CSV provider trend logs data from a file in CSV format. The data is output in the form of a trend.

Logging Data by Means of ODBC

Log Object	Function
ODBC database field	Serves to output data from a field of a data source by means of ODBC. The data is output as text.
ODBC database table	Serves to output data from a table of a data source by means of ODBC. The data is output in tabular form.

Logging of Own COM Servers

Log Object	Function
COM server	The COM server object allows you to integrate your own COM object in WinCC and log the data from this object. The COM object can be of the type text, table, or picture.

Output of a Hard Copy

Log Object	Function
Hard copy	Serves to output the current contents of the screen, a subarea of it, or a currently selected picture window.

A hard copy can also be output by the "PrintScreen" application. You will find more information in the section "Output of a Hard Copy".

Dynamic Log Parameters

To enable the logging of runtime data to be adjusted before output, dynamic log parameters were introduced in WinCC. In addition to a configured value, WinCC tags to be specified from the parameters taken for output at runtime.

To change the parameters for output immediately before starting printing, a dialog for changing the log parameter can be displayed. The dialog display can be controlled by the relevant print job. The "Configuration dialog" option must be set in the "Dialog" field for this.

12.3.1 Logging Messages at Runtime

Overview

The WinCC reporting system allows all the messages that occur in the system to be logged. The following options are available:

- Output of runtime data from the message archive
- Output of runtime data from the message list
- Output of a message sequence report

The messages from the message archive and message list can be printed out directly via a configurable option in the WinCC Alarm Control. The messages are output in accordance with the selection made in the WinCC Alarm Control.

To output messages, you have to do the following things:

- Configure the button in the WinCC Alarm Control for starting the log at runtime.
- Select Alarm Logging Runtime and Graphics Runtime in the startup list.
- Select the project and start the log.

It is also possible to log messages by making a selection independently of the WinCC Alarm Control. The reporting system provides the log objects required for this. You will find more information in the section "Changing Output Options for Message Reports from Alarm Logging".

Points to Note About the Message Sequence Report

The message sequence report outputs the messages that occur in the system in chronological order in accordance with the specified selection criteria. To this end, the message sequence report in the startup list of the WinCC project must be activated. The message sequence report is generally output to a line printer, but it can also be output page by page.

12.3.2 Logging Process Values at Runtime

Overview

The WinCC reporting system allows process values from Tag Logging and the data from user archives to be logged. The following options are available:

- Output of tag data from process-value archives and compressed archives in tabular form
- Output of tag data from process-value archives and compressed archives in the form of a trend
- Output of data from user archives in tabular form

The data can be output directly in the relevant WinCC control by means of a configurable button. The logs are output with the data selection of the relevant WinCC control. To output runtime logs, you have to do the following things:

- Configure the button in the relevant WinCC control for starting the log at runtime.
- Select Tag Logging Runtime and Graphics Runtime in the startup list.
- Select the project and start the log.

It is also possible to log the process values from Tag Logging and the data from user archives with a selection that is independent of the WinCC control. The reporting system provides the log objects required for this. You will find more information in the sections "Changing Output Options for User Archive Tables", "Changing Output Options for Tag Logging Tables" and "Changing Output Options for Tag Logging Trends".

12.3.3 Logging Data from Other Data Sources

Overview

The WinCC reporting system also allows you to log data that do not come from a WinCC project. There are various log objects available for integrating this kind of data in a WinCC log. In addition, WinCC provides a hard copy log object, by means of which the screen or a section of the screen showing the current runtime status can be output in a report.

The following log objects are available for output:

ODBC database field	Serves to output data from a database field in a WinCC log by means of ODBC.
ODBC database tables	Serves to output data from a database table in a WinCC log by means of ODBC.
CSV table	Serves to output data from a file in CSV format in a WinCC log. The data is output in tabular form.
CSV trend	Serves to output data from a file in CSV format in a WinCC log. The data is output in the form of a trend.
Hard copy object	Serves to output a view of the screen or a section of the screen in a WinCC log.
COM server object	Permits the integration of a user-specific COM server to output its data in a WinCC log.

Detailed knowledge is required in order to configure these logs. Creating a COM server, in particular, requires appropriate background knowledge. Information on database accesses by means of ODBC, COM servers, and CSV files is not provided here. You will find more information on these subjects in the relevant technical literature.

13 Creating Page Layouts

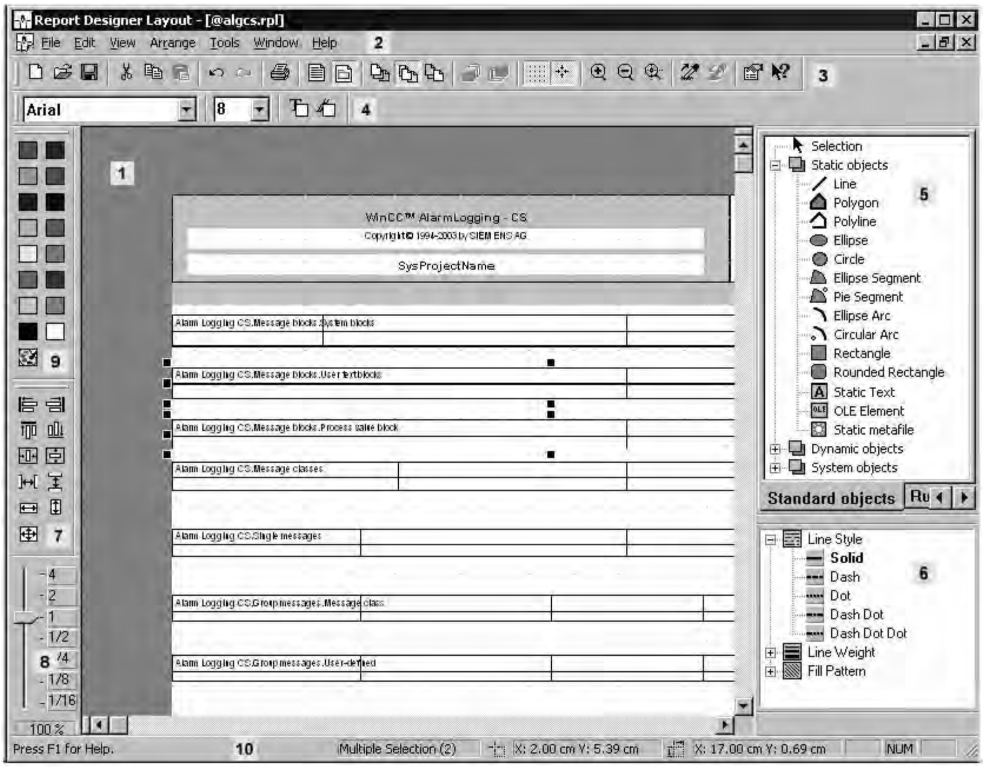
13.1 The Page Layout Editor

Introduction

The page layout editor offers objects and tools for creating page layouts. Start the page layout editor in WinCC Explorer.

Structure of the page layout editor

The page layout editor is constructed according to the Windows standard. It has a work area, toolbars, menu bar, status bar and various palettes. When you open the page layout editor, the work environment appears with the default settings. You can move and arrange the palettes and toolbars as you like and show or hide them.



(1) The work area

The printable area of the page is displayed on the gray area and the rest of the page on the white area. Every image in the work area represents a layout and will be saved as an independent RPL file. The layouts can be enlarged and reduced according to Windows standard.

(2) The Menu bar

The menu bar is always visible. The functions on the different menus are either active or inactive, depending on the situation.

(3) The Toolbar

The toolbar contains buttons for the most important menu commands so that the page layout editor can be used quickly and easily. It can be hidden or moved with the mouse anywhere on the screen, as required.

(4) The Font palette

The font palette is used to change the font, size and color of text objects, as well as the line color of standard objects.

(5) The Object palette

The object palette contains standard objects, objects for runtime documentation, COM server objects and objects for the project documentation. These objects can be used to construct a layout.

(6) The Style palette

The style palette is used to change the appearance of a selected object. Depending on the object, the line type, line weight or the fill pattern can be changed.

(7) The Alignment palette

The alignment palette enables you to change the absolute position of one or more objects, the position of selected objects relative to one another or to standardize the height and width of several objects.

(8) The Zoom palette

The zoom palette offers two options for increasing or reducing the objects in the active layout: using buttons with default zoom factors or using a slide bar.

(9) The Color palette

The color palette is used to select object colors. Besides the 16 standard colors, you can define your own colors.

(10) The Status bar

The status bar is located on the lower edge of the screen and can be shown and hidden. It contains, for example, tips, information about the position of highlighted objects and the keyboard settings.

13.1.1 The Object Palette

Usage

The object palette contains object types that can be inserted into the page layout. Static objects and system objects are available for the visual construction of the page layout. Dynamic objects are available for data output.

Contents

The objects in the object palette can be summarized in the following 4 object groups:

- Standard objects: static objects, dynamic objects and system objects
- Objects in the runtime documentation: e.g. alarm logging RT, user archive-runtime, CSV provider etc.
- COM server Objects
- Objects for the project documentation: e.g. actions in the Graphics Designer, alarm logging CS, Global Script etc.

Operation

Click the mouse on one of the tabs and select the object you want.

Characteristics

The object palette can be shown and hidden. It can be moved anywhere on the screen with the mouse.

13.1.1.1 Standard Objects

Usage

Standard objects contain object types that are used to visually construct the page layout. Dynamic object types can also be used for this, unlike objects for runtime documentation and project documentation, which are not connected to the WinCC components.

Overview

Standard objects can be divided into three object classes:

- Static objects are used for the visual construction of a page layout. You can insert static objects into both the static and dynamic parts of a page layout.
- Dynamic objects can be connected with data sources which have a valid data format for the current object. This data can be output in a WinCC layout. You can only insert dynamic objects into the dynamic part of the page layout.
- System objects are used as placeholders for the system time, the current page number and the project and layout names. You can only insert system objects into the static part of the page layout. The required entries are described in the "Format" attribute in the "Miscellaneous" property of the system object.

General properties

- The object properties (e.g. geometry, color) of the individual standard objects are preset. These default settings can be changed, however. The objects are displayed with the default object properties.
- The properties of the displayed objects can be changed at a later time.

13.1.1.2 Objects for Runtime Documentation

Usage

The runtime documentation objects are for outputting logs of the runtime data. The provided objects are already linked with the current data sources. The output options can be configured using the "object properties" dialog. The data for the logs are taken from the linked data sources at the time of the output. You can only insert the runtime documentation objects in the dynamic part of the page layout.

Overview

Object	Description
Archive report	The "Archive report" object is connected to the message system and outputs the messages saved in the message archive to a table.
Message report	The "Message report" object is connected to the message system and outputs the current messages in the message list to a table.
User archive - runtime table	The "User archive - runtime table" object is connected to the user archives and outputs the runtime data from the user archives and views to a table.
CSV-Provider table	The "CSV provider table" object can be linked to a CSV file. The data contained in the file are output to a table. The data must be in a predefined structure.
CSV-Provider trend	The "CSV provider trend" object can be linked to a CSV file. The data contained in the file are output in a curve. The data must be in a predefined structure.
Tag table	The "Tag table" object is connected to the tag logging runtime and outputs the contents of the tags used in tag logging to a table. All of the processed values archived with the tags are output in the corresponding time period.
Trend	The "Trend" object is connected to the tag logging runtime and outputs the contents of the tags used in the tag logging in a curve. All of the processed values archived with the tags are output in the corresponding time period.

General properties

- The object properties (e.g. font) of the individual objects are preset. However, these default values can be changed. The objects are displayed with the default object properties.
- The properties of the displayed objects can be changed at any time.

13.1.1.3 COM Server Objects

Usage

In order to use a COM server object, a COM server projector must be integrated into WinCC. This COM server makes the object for logging data available. This way, it is possible to integrate user-specific data in a WinCC log. The form and properties of a COM server object are determined by the COM server writer. The description of the COM server object is delivered with the COM server writer. The options for selecting the output data are determined by the current COM server object. The COM server objects can only be inserted into the dynamic part of a page layout. Additional information can be found in the chapter "Working with COM server objects".

13.1.1.4 Objects for Project Documentation

Usage

The objects for project documentation are available for the report output of the configured data. The objects for project documentation can only be inserted into the dynamic part of a page layout.

The objects for project documentation are closely connected with the WinCC components. The object types are fixed. Depending on the type and size of the configuration data for the output, the "static text", "dynamic metafile" or "dynamic table" object types are used. A detailed description of the objects used and the output data can be found in the chapter "Outputting project documentation".

For some of the objects used with the "dynamic metafile" and "dynamic table" object types, you can change the selection of the configuration data for output. Additional information can be found in the chapter "Working with objects for project documentation".

General properties

- The object properties (e.g. font) of the individual objects are preset. However, these defaults can be changed. The objects are displayed with the default object properties.
- The properties of the displayed objects can be changed at any time.

13.2 Working with Layouts

Introduction

The difference between a layout in the sense of a file and a layout as an object must be clearly understood. In this chapter, we will describe dealing with a layout in the form of a file. A layout opened in the page layout editor is treated as an object. The layout object has object properties and can be edited accordingly. Additional information can be found in the chapter "Working with the layout object".

In this chapter we will show you how to create layouts, display the properties of layouts and how to work with several layouts. WinCC comes with several ready-made layouts for most standard applications. So it is often simpler and less time consuming to save one of these layouts under a new name and adapt it to your own requirements, than to create a new layout.

Configuration steps

- Create a new page layout.
- Design the outward appearance in the static part of the layout and save the newly designed layout as a template.
- Apply the design in the layout for the output.

Editing options

In order to configure quickly, use the diverse program functions such as:

- Save the layouts under new names
- Apply object properties to other objects
- Copy or transfer objects in other layouts
- Transfer ready designed objects and layouts from other projects

13.2.1 Changing Predefined Layouts

Introduction

You can change the predefined layouts in order to fit your needs better. It is advisable to save an existing layout with a new name first and only then to edit the layout. If you do this, you can always fall back on the layouts provided with WinCC if you need to.

The system layouts and system print jobs delivered with WinCC are used by the WinCC components when the report output is triggered (e.g. project documentation in the Graphics Designer). Therefore, neither the system layouts nor the system print jobs may be deleted or renamed.

The system files are located in the WinCC directory in the "..\WinCC\syslay" folder and are copied to the "..\<Project name>\Prt" folder in the corresponding project directory the first time they are accessed.

The system layouts and system print jobs are created for every new project.

Changes in the static part of a layout

The header and footer are defined in the static part of a page layout. You can also change the appearance of the report without changing the dynamic properties of the layout. The static part of a page in the page layout extends across the entire printable area of the report. You can individually define the static part for the cover sheet, the report contents and the closing page. Static objects and system objects are available to you for the design.

The same static part will be repeated on all the succeeding pages of the report contents.

Activating the static report part:

5. Select "Static part" from the "View" menu.
or

Activate the static report part using the toolbar.

6. Next add static objects or system objects.

Changes in the dynamic part of a layout

You put together the structure and content of the report for output in this part of the page layout. There are static, dynamic, and system objects available for defining the report contents. If necessary, the dynamic part of the contents of the report is spread across the various subsequent pages at output, since it is not known until the time of output how much data there is. The first object in the vertical direction in a layout is automatically moved to the upper dynamic edge during output.

Important:

During the definition of a page layout, you cannot predict with absolute certainty, how many pages the dynamic part will require. The size of the dynamic tables and text fields is determined by the data which is used at the moment of output. Succeeding objects are moved downward according to the dynamic size changes.

Generally no line breaks are provided for the output of data in table cells. However, you can output large amounts of data in a table cell without the data being cut off by:

6. Select the output of the report in landscape format.
7. Select the "Vertical" table format in the table object properties.
8. Set the relative column width of all columns as narrowly as possible.
9. Use the smallest font size possible.

Static objects which are pasted into the static part of a page layout are in a layer below the dynamic objects. Thus, it can happen that the static objects will be covered by the dynamic objects, especially by the dynamic extension of a table.

Enabling the report contents for editing:

1. Select "Dynamic part" from the "View" menu

or

Activate the static report part using the toolbar.

1. Next add static, dynamic or system objects.
2. For dynamic objects, select the data for output from the "Connect" tab in the "Properties" dialog.

Editing the objects

Objects in the layout are edited using the "Object properties" dialog. This can be opened using the toolbar, the context menu for the objects or by double-clicking on an object. You can find additional information about this under "Working with objects".

13.2.2 Working with several layouts

Introduction

The page layout editor offers your some options for efficient configuration. In order not to have to keep repeating the same configuration steps, you can:

- Save the layouts under new names
- Apply object properties to other objects
- Copy or transfer objects in other layouts
- Transfer ready designed objects and layouts from other projects

Save layout under new name

If you want to configure several projects, it is recommended that you first save a layout as a template. Set the elements which remain the same, such as headers and footers, in this layout. Set the page format and the orientation. Set the print margins and the dynamic margins. Save the finished layouts as often as you want under new names. Then you can add the log objects you want to the individual layouts and set their parameters. You will find more information in the section entitled "Layout file options".

Apply object properties to other objects

If you have an object with certain properties in a layout, you can easily apply these properties to another object. Use the eyedropper function for this. You will find more information in the section entitled "Transferring object properties".

Copy or transfer objects in other layouts

This is a very useful function in order to paste already configured objects or object groups into another layout. In this way, you can, for example, design a complete header and transfer it to other layouts.

In order to copy an object or object group, highlight the object(s) and copy the selection to the operating system clipboard. Next open the target layout and paste in the contents of the clipboard. The contents of the clipboard can be pasted into as many other layouts as desired. The next time an object is copied, the data in the clipboard will be overwritten.

In order to transfer an object or object group, highlight the object(s) and cut the selection out. The selection will be copied to the operating system clipboard. Then you can proceed with this selection as with copying.

Transfer ready designed objects and layouts from other projects

If you have already created a project with WinCC, then you can transfer existing layouts to the new project. You can also transfer objects from an existing layout to a layout in the new project.

There are two options for transferring a complete layout:

1. Navigate to the root directory of the old project using Windows Explorer. Open the "PRT" folder. Copy the layout you want and paste it into the "PRT" folder of the new project.
2. Open the page layout editor in the new project and click the folder button in the toolbar. The "Open" dialog will appear. Use this dialog to navigate to the "PRT" folder of the old project. Open the layout you want in the page layout editor. Save the layout in the "PRT" folder of the new project using the "Save" or "Save as..." function .

Requirements

In order to transfer a fully defined object from another project, open the layout which contains this object in one of the two ways described above. Select the object or group of objects that you want and copy the selection to the clipboard. Open the target layout and paste in the contents of the clipboard.

Note:
If you are copying system layouts, it is recommended that you create a backup copy of the new system layout before overwriting it in the "PRT" folder.

13.3 Working with Objects

13.3.1 The Properties of an Object

Introduction

The form, appearance, position and application linking of an object are set by the "Object properties". These properties can be changed in the page layout editor as needed.

The properties of an object are described by a large number of "attributes". An object property can be changed by assigning a new value to its attribute.

The "Object properties" window contains all the attributes of a selected object or a multiple selection of objects on the properties tab. The attributes are divided into property groups, such as "geometry" or "color". The type and number of the available property groups and attributes are dependent on the type of the selected objects. For example, the "Font" property group is only shown for the object types which can be displayed with text.

As an alternative to changing the attributes in the "Object properties" window, objects can also be adjusted with the mouse and keyboard or by using the toolbar and palettes. However, in this way only certain object properties, such as the basic geometric size, color and line style, can be changed.

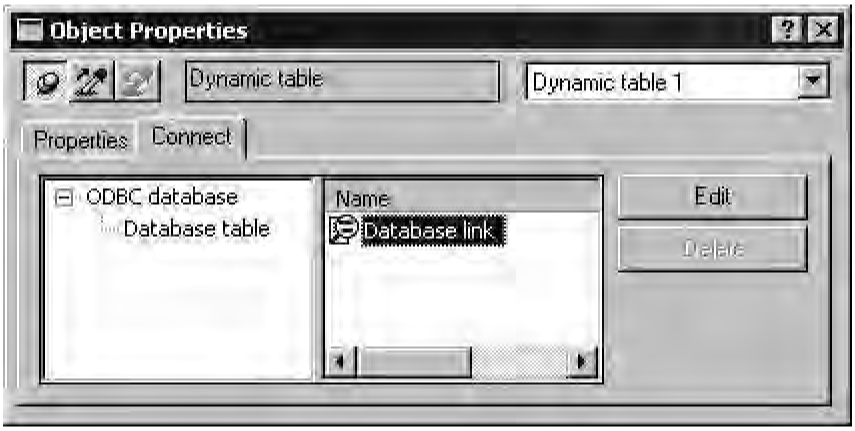
The Link Tab in the "Object Properties" window

Introduction

The Link tab represents the central dialog for configuring links in the "Object properties" window.

The dynamic objects are already linked with the corresponding applications. Some of the dynamic objects have one or more dialogs for selecting data for output.

For example, the Link tab in the "Object properties" window can have the following appearance for a database table with the object name "Dynamic Table 1":



The Link tab is divided into two areas:

- In the left-hand area, the linked applications from which the data is taken, are shown in the form of a directory tree.
- In the right-hand area, the editing options and, if they exist, the formatting options, are shown in the "Name" column. You can open the appropriate configuration dialog with the "Edit..." button. The configured selections are reset with the "Delete..." button. The selection options remain.

The width of the two window areas and the columns in the event display can be changed by moving the vertical dividing lines.

13.3.2 Working with Standard Objects

Introduction

Standard objects include geometric shapes, static text, OLE objects, static metafiles, the dynamic objects and the system objects. You can use static objects to design the layout for the output. System objects can be used, as an example, to inscribe the project name and page numbers in the static part of the report.

In the page layout editor, the various object types have predefined properties. When they are inserted the objects import these default properties, with the exception of individual geometric properties. After insertion the properties of an object can be modified. In the same way the default settings for the object types can be modified as required.

13.3.3 Working with objects for runtime documentation

Introduction

This chapter provides you with details regarding the available Page Layout Editor objects for the runtime documentation. You will learn how to work with the objects and which basic conditions apply for the individual objects.

Usage

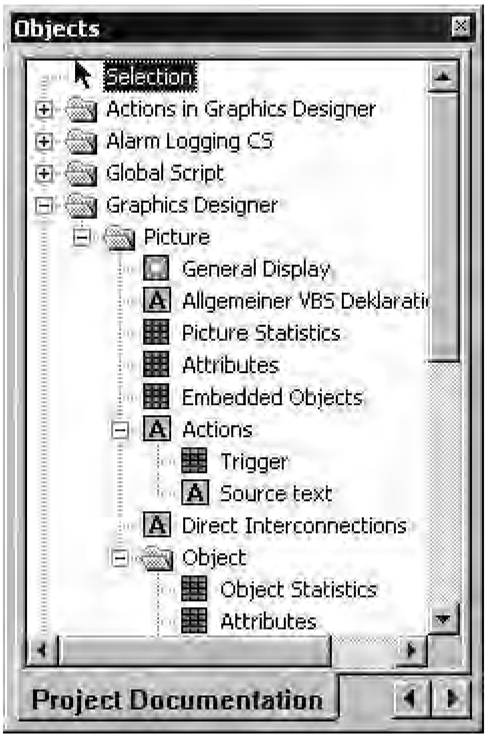
The runtime documentation objects are available for the output of runtime data in logs. The objects are linked with the pertinent WinCC applications and provide the output data in runtime. The objects provide dialogs for the selection of output data. The objects for runtime documentation can only be inserted in the dynamic part of a page layout.

13.3.4 Working with objects for project documentation

Introduction

In this chapter, you will find details regarding the available objects of the page layout editor for the project documentation. You will learn how to work with the objects, and what are the basic conditions for each object.

The project documentation objects are provided for the report output of configured data. The objects are linked to the corresponding WinCC applications and return the configuration data of the WinCC project for output in a report. The objects for project documentation can only be inserted in the dynamic part of a page layout.



Some objects provide an option to select output data. This allows you to restrict the output to the data you currently need. For some of these objects, the selection options are identical. For each type of selection, this chapter will describe the changes that can be made to the output options. The relevant objects are listed with the descriptions.

The other objects do not offer an option to select output data. There are 3 types of these objects. The editing steps to change the output options are identical for objects of the same type and are described once for each object type.

Objects without data selection

The following object types are available:

A	Dynamic text
☼	Dynamic metafile
▦	Dynamic table

The modification of output options is described for each of these object types.

- Changing output options for dynamic text
- Changing output options for dynamic metafiles
- Changing output options for dynamic tables without data selection

14 Creating Line Layouts

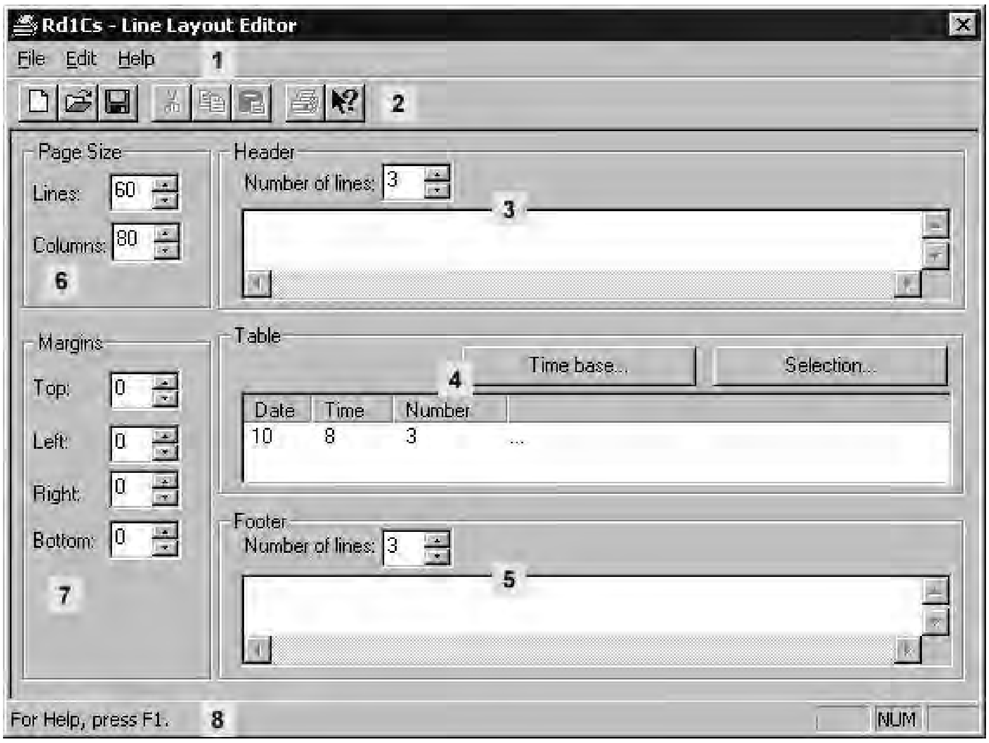
14.1 The Line Layout Editor

Introduction

The line layout editor is used only to create and edit line layouts for a message sequence report. Each line layout contains a dynamic table that is connected to the WinCC message system. Additional objects cannot be added to a line layout. Text can be entered in the header and footer. Start the line layout editor in WinCC Explorer.

Layout of the Line Layout Editor

The line layout editor has a toolbar, menu bar, status bar, and various different areas for editing a line layout. When opened, the line layout editor appears with the default settings.



Lines: 60

Columns: 80

Header – Number of lines: 3

Date	Time	Number	
10	8	3	...

Top: 0

Left: 0

Right: 0

Bottom: 0

Footer – Number of lines: 3

For Help, press F1.

(1) The Menu Bar

The menu bar is always visible. The functions on the different menus are either active or inactive, depending on the situation.

(2) The Toolbar

The toolbar is always visible in the line layout editor. There are different buttons on the toolbar that allow you to activate the functions of the menu commands quickly. The buttons are either active or inactive, depending on the situation.

(3) The Header Area

The Header area allows you to enter text in order to create the header of a line layout.

(4) The Table Area

The design of the table for output is displayed in the Table area. The configured column headings and the width of the columns (number of characters per column) are displayed. The table can be configured for output using the buttons in this area.

(5) The Footer Area

The Footer Area is used to enter text in order to create the footer of a line layout.

(6) The Page Size Area

The Page Size area is used to set the number of lines and columns for a line layout.

(7) The Margins Area

The Margins area is used to set the margins for the output of a line layout.

(8) The Status Bar

You will find the status bar along the lower edge of the screen. It contains tips on the buttons on the toolbar, on menu commands, and on keyboard settings, for example.

14.2 The Page Size and Margins Areas

Introduction

Line-by-line printing is used in practice for current and synchronous documentation. WinCC permits "line-by-line printing" on appropriate printers. The output to the printer is only possible via a local, parallel interface on the computer. You can define the page size and margins for each line layout within the permissible limits.

Page Size in the Line Layout

The page size of a line layout can be set within the following limits:

- The number of lines can be between 1 and 400.
- The number of columns can be between 20 and 400.

The value for the number of columns is defined as the number of characters per line.

Margins in the Line Layout

Because most printers cannot print right up to the edge of the page, the margins must be set accordingly in the line layout. The margins of a line layout can be set within the following limits:

- The value for the margins (left, right, top, bottom) can be between 0 and 30 characters.

Printer Settings

The number of columns set in a line layout under "Page Size" (= characters per line) must be supported by the printer used. If the number of columns is greater than the number of characters a printer can print per line, a line break is inserted. This overflow of the column text into a new line disturbs the column formatting. To remedy this, a narrower font or a smaller character width must be set for the printer.

Setting the font or character width depends on the printer used. Please check the printer manual for further information.

Note:
If the suitable font is not set as the default font on the printer, it will have to be set again each time the printer is reset or switched off and on again.

14.3 The Header and Footer Areas

Introduction

You can define a different header and footer for each line layout. The number of headers and footers can be between 0 and 10. You can enter any text you like in them. You can output as many headers as the number of lines you have set. You do not have to output the header and footer.

Usage

The headers and footers are output with each page of the message sequence report. The width of the headers and footers is based on the number of columns (= characters per line) set under "Page Size". Text in the header or footer that is longer than the configured number of columns is truncated at output.

14.4 The Table Area

Introduction

You can define a table for the output of reports and measured values for each line layout. The number and contents of the columns are defined using the "Selection..." button. The number of columns can be between 1 and 40 and depends on the number of selected message blocks.

Usage

The configured columns with the associated headings and the set column width are displayed in the Table area. You can configure the column heading and the column width for each column separately.

Use the "Selection..." button to open the "Alarm Logging Runtime: Report - Table Column Selection" dialog box. In this dialog box you can select the server(s) whose reports you want to log. You can assign the message blocks to the table columns and set the filter criteria for the alarm output. These changes are only effective for the output of the message sequence report and are not written back to the alarm logging. You can find additional information in the sections entitled "Outputting Runtime Data with the Message Sequence Report" and "Changing Output Options for the Message Sequence Report".

14.5 The Time Range

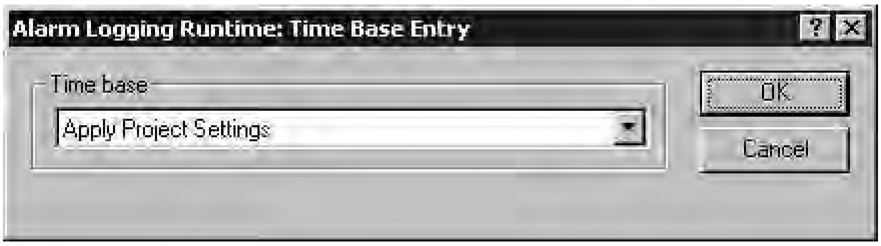
Introduction

By changing the base for the time settings you can convert the time specified in the time stamp of the report data to a different time base. The time settings are only converted for the output in the report and are not written back to the alarm logging. If, for example, you want to output the data of a remote computer in another time zone, you can use this function to ensure that the data is output with a comparable time. If the two computers in this case were working with the "Local Time" time base, you would receive reports on the two computers with different times. The results would not be directly comparable. You can find additional information in the section entitled "Adjusting the Time Settings in the Project".

If you change the time base for the output, ensure that the time reference is also output in the report. For example, specify an identifier in the alias for the "Time" message block. You can find additional information under "Changing the Output Options for the Message Sequence Report".

Usage

You can change the time base in the Table area. To do this, click the "Time base..." button. The "Details of the Time Base" dialog box appears. Select the time base you want from the list.



Alarm Logging Runtime: Time Base Entry

Time base: Apply Project Settings

15 Setting up Multilingual Projects

15.1 Language support in WinCC

Multilingual configuration in WinCC

WinCC enables you to configure your projects in multiple languages. For this purpose WinCC supports the multilingual configuration of almost all objects containing text that is visible in runtime.

In addition to direct text entry, WinCC provides you with user-friendly export and import options for translating your configuration. This is especially advantageous if you are configuring large projects with a high text content.

Supported languages

In principle WinCC allows you to create projects for every language installed on your operating system. During installation WinCC also offers you a group of languages in which you can set up your WinCC configuration interface.

The following languages can be installed with the standard version of WinCC:

- German
- English
- Spanish
- Italian
- French

Furthermore with the Asian languages version of WinCC you can install the following languages:

- Japanese
- Chinese (traditional and simplified)
- Korean
- English

You can set up the WinCC Configuration interface in the installed languages. In addition, once a language has been installed all standard texts are available in that language.

If a project should be used on another computer, then WinCC must be installed on the target computer in the same language as on the source computer. You should always install WinCC with all available languages.

Note:

Standard texts are all texts that already exist in WinCC's installed languages, e.g. default names of message classes in Alarm Logging, WinCC system messages, and the title bar captions and column titles in WinCC Control.

Configuration scenarios

When configuring in other languages the following scenarios can occur:

- You are configuring a project for a language with which you are unfamiliar. Example: You are an English speaking project engineer and configuring for an Asiatic market: Create the project user interface of WinCC in your preferred language. There are two procedures for configuration. Configure the project in a language with which you are familiar, export the texts, translate the texts externally and import the translated texts back into the WinCC project. Or you input all the text for your project directly in the language that is intended to be displayed later in runtime. Direct text entry is recommended only in cases where you have configured just a small amount of text in your project.
- You are configuring a project for a number of languages with which you are unfamiliar. Example: You are a French speaking project engineer and configuring a project for a system which must be operated by Russian, English and German speaking users. Display the project user interface of WinCC in your preferred language. Configure the project in a language you are familiar with and export all the text. Distribute the exported files to translators for each respective language. Following translation, import the translated texts back into your WinCC project. Also configure a change of language facility for the operators, so that they can use it in runtime to select the desired language for the operator interface.

15.1.1 Language expressions in WinCC

Principle of language expressions in WinCC

When using WinCC to create multilingual projects, you can set up languages at multiple system levels. It is therefore helpful to distinguish between these various levels by using clearly defined language expressions.

Operating system language

The operating system language (system locals) is the language environment set in the operating system in which "non Unicode-enabled" applications such as WinCC run. This setting defines the used "codepage" in character sets.

Note

Set the operating system language corresponding to your WinCC user interface so that the correct codepage is used and all character sets are correctly displayed. Alternatively you can use an operating system in the corresponding language.

Note:

Information about the operating system language can be found in your Windows documentation.

Operating System User Interface Language

The operating system user interface is the language in which the GUI of operating system is displayed. All Windows menus, dialogs and Help text are displayed in this language. During configuration, some system dialog boxes are displayed in WinCC Configuration in the operating system GUI language (e.g. "Open" and "Save As").

The operating system GUI language can only be switched with multiple language operating systems.

WinCC GUI Language

The WinCC GUI language is the language of the project interface in WinCC Configuration, i.e. the language in which the WinCC menus, dialog boxes and help files are displayed during configuration. You can set any language as the WinCC GUI language installed during setup. In the European version you have a choice between German, English, Italian, Spanish and French. The WinCC GUI languages available in the Asian version are Japanese, Korean, Chinese and English.

The language that you selected for the installation of WinCC is set as the WinCC GUI language when you first start WinCC. The next time you start WinCC the interface is displayed in the WinCC GUI language that you set up last.

You set the WinCC GUI language centrally in the WinCC Explorer. Select a language you are familiar with as the WinCC GUI language for configuration.

Note:
If the current WinCC GUI language is not the same as the operating system GUI language, the standard dialog elements of certain WinCC editors remain unchanged and continue to be displayed in the operating system GUI language. This applies to standard buttons such as "Cancel" and standard system dialogs such as "Save As", "Open file" or "Print".
In order to display these elements in the desired language, you need to set the operating system GUI to the same language as you are using for the WinCC GUI language.

Project language

The project language is the language intended for the project you are creating. You can create a project containing multiple project languages in order to make several languages available in runtime.

In addition to the languages installed with WinCC you can select any other language supported by your operating system as the project language.

Note:
With regard to operating system languages, WinCC supports only the primary languages, for example English (USA) but not English (GB), or traditional Chinese (Taiwan) but not simplified China (Peoples' Republic).

In each case set up the project language directly in the editor you are using for configuration, e.g. in Alarm Logging or the Graphics Designer. After switching the configuration language, the user interface remains the same in WinCC Configuration. Only the language of the texts which you create for your project is switched. The current project language is displayed for you in the status bar of the respective editor.

The following figure shows a section from the Graphics Designer with English as configuration language and WinCC with a German user interface:

Language

The Runtime language is the language in which the project is displayed in Runtime. In WinCC you can set a runtime language in which the project will start by default in runtime.

Note:
Always select one of your project languages as the runtime language, otherwise only "???" will be displayed in runtime.

When creating a project containing multiple languages, configure operating elements which will allow the operator to change the runtime language.

Default runtime language

The default runtime language is English. The default runtime language is the language that is used if no translation exists in the respective runtime language for the text of a WinCC object (e.g. AlarmControl, TableControl). In the main this affects the title bar captions and column titles in WinCC Control if the runtime language is not one of the languages installed with WinCC. In the languages installed with WinCC the controls are displayed correctly.

Unlike nonexistent WinCC texts, any texts that you have configured yourself are displayed as "???" if they do not exist in the current runtime language.

15.1.2 Configuring for multiple languages

Operating system requirements

When configuring projects for multiple languages, your operating system must meet the following requirements:

- The project languages must be installed on your operating system.
- The correct system regional setting (operating system language) must be specified as default on your operating system. This is particularly important if your are configuring for languages which are not Western European, e.g. Asian.
- Any special fonts that you use must be available on your operating system. This applies particularly to non-Latin fonts such as Cyrillic or Asian.
- You must install input methods on your operating system to input e.g. Asian fonts. For each running application you can then select the input methods independently of one another.

Note:
How to make the corresponding settings or installation in your operating system is described in your Windows documentation.

Language combinations

Due to the various language setting options in WinCC and your operating system, a number of different language combinations can occur, for example:

- Configure a single-language project in the language preferred: Operating system language, operating system user interface language, WinCC user interface language and configuration language are all the same.
- Configure a single-language project in but not in your preferred language: Operating system user interface language and WinCC user interface language are in the preferred language. The project language is the language in which you wish to subsequently display the project in runtime. If you are configuring for Asian languages, set the operating system language correspondingly so that the used character sets are displayed in the correct codepage. If all languages are from one regional area (e.g. Western European), this setting is not necessary because it was already made implicitly.

- Configure a multilingual project. One of the languages is your preferred language: Operating system user interface language and WinCC user interface language are in the preferred language. The project languages are the languages in which you wish to subsequently display the project in runtime. Configure the project in your preferred language and hand over the text for translation when the project is complete.
- Several project engineers who prefer different languages and configure on one computer: Select a neutral language as the operating system user interface language, e.g. English. Each configuration engineer can set the WinCC GUI language to his preferred language. The languages that will subsequently be displayed in runtime are set as the project languages. If you are configuring for Asian languages, set the operating system language correspondingly so that the used character sets are displayed in the correct codepage. If all languages are from one regional area (e.g. Western European), this setting is not necessary because it was already made implicitly.

Note:
If you are using a multilingual operating system, the configuration engineer can also set the operating system GUI language to the his preferred language.

Configuring for multiple languages - the main editors

When configuring for multiple languages you need to take the following editors into account:

- Text Library: All project texts are managed centrally in the Text Library except for texts from Graphics Designer. The Text Library is where you centrally set up fonts, translate text directly, or use the export and import functions in order to translate text externally.
- Graphics Designer: This is used to configure pictures for the project. Pictures can contain different text elements, such as Static Text, Tooltips or Labels for ActiveX Controls. Text from the Graphics Designer is stored in the picture to which it refers. Texts can be exported in table format for translation and can then be reimported into the pictures concerned.
- Alarm Logging: Alarm Logging is used to configure messages that are issued in runtime. The texts for the message system are managed centrally in the Text Library. You can translate text directly in Alarm Logging or via the Text Library. If there is a large amount of message text it is advisable to translate it via the Text Library. Configuring with SIMATIC STEP7: Alarm logging texts from the SIMATIC Manager are stored in the Text Library when transferred and must be translated there.
- Report Designer: Report Designer is used to configure layouts for reports issued in runtime and to create templates for the project documentation for projects.

- User Administrator: The authorizations configured in User Administrator are language-dependent and managed centrally in the Text Library. Translate these texts via the Text Library. So that the texts are created in the project text library, you must open the User Administrator in the respective language.
- User Archive (Option): All the texts in the user archives are managed centrally in the Text Library. Translate these texts via the Text Library.
- Picture Tree Manager (Option): Texts (e.g. container names) from this WinCC option are managed centrally in the Text Library. Translate these texts via the Text Library.

With multilingual configuration, you only have to intervene in the Graphics Designer and if necessary in the Alarm Logging. Text from the other editors is managed centrally via the Text Library (centralized text management).

Note:
The project language can be set up separately for the Graphics Designer and Alarm Logging. By default the editors start with the set runtime language as the project language.

Note:
WinCC elements such as archive names, tag names and script functions are unique and cannot be configured for multiple languages. In such elements do not use any national special characters or Asian characters because after switching language, these can otherwise no longer be read or can lead to WinnCC not working properly.

Displaying untranslated text in the project

Untranslated texts are displayed in different ways in runtime:

- WinCC objects (e.g. WinCC AlarmControl): If these texts are not available in specific languages, they are displayed in the default runtime language (English).
- Individually configured objects: If these are not available in specific languages, texts in graphic objects are displayed as "???" and texts from the alarm logging system are not displayed.

15.1.3 How to create a multilingual project

Introduction

The following section gives you an overview of the steps you need to take in order to configure multilingual projects. The procedure described here is the optimum. If alternative methods are possible and desirable in certain cases, they are described in the relevant sections of this Help text.

The procedure in principle

- Install all required fonts and input methods on your operating system. If you are configuring in non-Latin fonts, the relevant fonts must be available as Small Fonts.
- Activate the languages in your operating system that you wish to configure. Your Windows documentation will tell you exactly how to do this.
- Install WinCC complete with all the languages that need to be available as WinCC GUI languages. If you install languages later, the standard texts for these languages are not automatically transferred into the Text Library.
- When you create a new project, the WinCC GUI language is the language you selected for the installation of WinCC. When launched again, WinCC opens in the WinCC GUI language most recently set.
- Open Alarm Logging in WinCC. Do not open the Text Library beforehand. If you proceed in this order, all standard texts in the installed WinCC languages will be available in the Text Library. If the project language you are using is not one that is provided by WinCC, note that standard texts for such a language is not available in translated form in WinCC although it is entered into the Text Library in the current WinCC GUI language when you change to the project language. You can translate these texts later or directly enter the language in the Text Library before changing to the project language.
- Configure your project in a language with which you are familiar. This project language later serves as a basis for translation of the texts.
- Export the texts from the Graphics Designer using WinCC EasyLanguage tool. Export the texts from the Text Library via the Text Library export function. You then have two files to translate. Note that EasyLanguage is not suitable for exporting and importing texts in Asian languages. Use the Graphics Designer's options to compile texts or use another tool to export and import the texts.
- Translate the texts in an external editor.
- Re-import the translated texts.
- Test your translated project in runtime.

15.1.4 How to change languages in WinCC

Requirements

When setting up languages in WinCC, note the following requirements:

- If you are configuring with non-Latin fonts, the necessary fonts and input methods must be installed in your operating system.
- All the project languages that you wish to use must be installed in your operating system.
- If you are configuring Asian languages, the proper system input locale (operating system language) must be set to change the WinCC GUI language in the operating system control panel so that the used character sets are displayed in the right codepage. If all languages are from one regional area (e.g. Western European), this setting is not necessary because it was already made.

Note:
For information on how to enter the appropriate settings and installation in your operating system, please refer to your Windows documentation.

Recommendations

In order to make your configuring efficient, please note the following:

- As far as possible always configure for one language at a time.
- Set the runtime language to be the language for which you are configuring. The editors always start with the set runtime language as the project language.

Changing the WinCC GUI language

To change the WinCC GUI language, the language must be installed with WinCC.

To change the WinCC GUI language, in the WinCC Explorer select "Tools" > "Language". Displayed are all installed WinCC languages with the exception of the current WinCC GUI language in the selection window.

After the change the operating elements, menus and dialogs in WinCC Configuration are changed to the selected language.

Note:
If you are configuring for Asian languages and the current WinCC GUI language does not correspond to the operating system language, a large part of the Asian interface will not be displayed correctly.

Note:
If the current WinCC GUI language is not the same as the operating system GUI language, the standard dialog elements of certain WinCC editors remain unchanged and continue to be displayed in the operating system language. This applies to standard buttons such as "Cancel", and standard system dialogs such as "Save As", "Open file" or "Print".
In order to display these elements in the desired language, you need to set the operating system language to the same language as you are using for the WinCC GUI language.

Changing the project language in the editors

You can set up a separate project language in the Graphics Designer and in Alarm Logging.

To change the project language, open the editor concerned and select the menu command "View" > "Language". A list box shows you all the languages available on your system (primary languages only).

After changing, all configured texts are displayed in the selected project language. If you have not yet configured the language, texts are displayed in the Graphics Designer as "???", and no text is displayed in Alarm Logging.

The set project language is displayed in the status bar of the editor.

Setting Runtime language

Set the runtime language in WinCC in the course of centralized configuration in the starting configuration of the runtime computer. For further information please see the topic "Setting the runtime computer starting configuration".

15.2 Multilingual pictures in Graphics Designer

Languages in Graphics Designer

In the Graphics Designer you can create text for graphic objects in all the languages that are installed in your operating system.

Note:
With regard to operating system languages, WinCC supports only the primary languages, for example English (USA) but not English (GB), or traditional Chinese (Taiwan) but not simplifed China (Peoples' Republic).

The text that you configure in the Graphics Designer is saved with each picture. You can export text from pictures with the aid of the WinCC SmartTool "EasyLanguage", translate it outside WinCC and then re-import it. Texts are automatically assigned to their corresponding graphic objects on import. If you have configured only a few texts for your pictures, you can enter the target-language texts in each picture as required.

The object known as Text List is a special case. Here you can select whether you want to store the configured text in the picture itself or in the Text Library. By default the text is saved in the picture, but not acquired by the EasyLanguage export function. If you want to use the Text Library export function, store the text in the Text Library.

Note:
EasyLanguage is not suitable for exporting and importing Asian languages. When configuring for Asian languages, translate the text directly in the picture. Alternatively you can use another tool for export and import or you can use the WinCC programming interfaces to access the graphic objects.

Changing languages in Graphics Designer

When you open the Graphics Designer, the selected runtime language is set as the project language. If you want to check a translation or enter text directly in another language, you can change the project language in the Graphics Designer.

The desired project language must be installed in your operating system.

To change the project language, select the menu command "View" > "Language". A list box shows you all the languages available on your system (primary languages).

If the newly selected language originates from another linguisitic area as the previous language, the operating system language (system locale) must also be changed. Then the operating system must be restarted.

After changing, all configured texts are displayed in the selected language. If you have not yet configured a language, its texts are displayed as "???",

The current project language is displayed in the Graphics Designer status bar.

Restriction

National special characters from different linguistic regions (e.g. Western European and Asian) should not be used together in the same picture because when saving the picture these characters are changed according to the codepage under a "foreign" operating syatem language.

Tip

Option 1: Use different pictures for different linguisitic areas. Edit and save the pictures only under the appropriate language setting.

Option 2: Select an Asian operating system language also for Latin fonts. By modifying the Registry, automatic font association can be manipulated in such as way that Latin national special characters can also be displayed. Further information can be found on the Microsoft support pages.

Change the Registry entry from [HKEY LOCAL MACHINE\System\CurrentControlSet\Control\fontassoc\Associated CharSet] "ANSI(00)="yes" to "no".

Caution
Always carry out the changes in the Registry carefully because this can otherwise lead to unwanted problems with the operating system!

Language input options for picture objects

Depending on the picture object, you have different options for entering text:

- In the object itself, such as in the case of static text.
- In the Configuration dialog, for example for the label on a button.
- In the Object properties dialog box, you enter the greater part of the text directly or in other dialogs such as Tooltips, Output Text, Text Attributes.

For further information on configuring graphic objects, please refer to the online Help on "Creating process pictures".

Language-depenent object properties

The objects of the Graphics Designer have different object properties with different language properties. With the exception of the window titles of the smart object, picture windows/application windows and certain WinCC controls, you can change the language of all object properties.

"Text" property

Objects: Static Text, Text List, Button, Checkbox, Radiobox

"Font" property

Objects: Static Text, I/O Field, Bar, Group Display, Text List

"Tooltip" property

Objects: All objects except Picture Window and Application Window

Special characteristic: Tooltips can certainly be configured in multiple languages, but without any special setting for the font. When configuring tooltips in non-Latin fonts, you must set the corresponding font in your operating system's Control Panel in runtime:

- Windows 2000: Go to "Start" > "Settings" > "Control Panel" > "Display" > "Appearance" and then select the ToolTip item from the "Item" list and set the font.
- Windows XP: Go to "Start" > "Settings" > "Control Panel" > "Display" > "Appearance" and select the "Extended" button, then select the "ToolTip" item from the list and set the font.

Note that when you change the operating system language, the tooltips in all the other Windows applications are also changed.

"Windowheader" property

Objects: Picture Window, Application Window

ActiveX Controls

Certain ActiveX Controls have text properties such as Column Name, Window Title or Axis Label for which the language cannot be changed. In WinCC these properties are stored in all languages installed with WinCC and are correctly displayed in runtime. If you are configuring in a language that was not installed via WinCC, these elements are displayed in runtime in the default runtime language (English).

When configuring language changes to languages with non-Latin fonts, use a neutral language for these ActiveX Control properties, such as English without a special font. If you have configured a non-Latin font for these object properties, this font is used for each displayed language.

15.3 Multilingual messages in Alarm Logging

Languages in Alarm Logging

In the case of multilingual projects you use Alarm Logging to configure all message system texts in all languages. The texts for the message system are stored centrally in the Text Library.

The Text Library holds not only the texts that you yourself configure, but also the preset WinCC standard texts, e.g. default names for message classes and message blocks.

When creating a new project, you can transfer standard texts in all languages installed with WinCC into the Text Library by opening Alarm Logging first. The language is then created in the Text Library and the standard texts are entered. Standard texts also include the WinCC system messages.

If configuring with SIMATIC STEP7: When text for the message system is transferred from the SIMATIC Manager it too is stored in the Text Library.

Language change in Alarm Logging

When you open Alarm Logging, the selected runtime language is set as the project language. If you want to check a translation or enter text directly in another language, change the project language in Alarm Logging.

The desired project language must be installed in your operating system.

To change the project language, select the menu command "View" > "Language". A list box shows you all the languages available on your system (primary languages).

After changing, all configured texts are displayed in the selected language. If you have not yet configured the language, the text fields are displayed empty or with the entry "not in use".

The set project language is displayed in the Alarm Logging status bar.

Language-dependent message objects

Alarm Logging is where you can find user texts and standard texts, both of which are stored in the Text Library.

- You configure user texts yourself. These are the names of the message classes, the message blocks and the message types as well as the message text, point of error and info texts about the message. With the exception of Help text, user texts are entered in the Text Library immediately following input. Help text is not stored in the Text Library and its language cannot be changed. Only use info texts if you are configuring for just one language or enter the texts in a "neutral" language, e.g. English.
- Standard texts are delivered in the languages installed with WinCC and entered in the appropriate language column in the Text Library when Alarm Logging is first opened or when the WinCC system messages are created. Standard texts are default occupations for names of message classes, message types and message blocks. You can change standard texts in the Alarm Logging or in the Text Library.

Note:
As much as possible, set the same language for the WinCC GUI and the project to avoid "language confusion" in the Text Library.
If you set up a language not installed with WinCC as the project language, the standard texts are not entered in the selected project language, but in the current interface language. You must therefore take special care that these texts are translated.

Options for language input in Alarm Logging

Since the language-dependent message objects behave differently, there are a number of options for language input:

- You can input user texts and standard texts in the Text Library or in Alarm Logging. In Alarm Logging you can input texts directly in the input field in table view or in the Properties dialog for the message concerned.
- You enter Help text in a dedicated input window.

15.4 Language administration using the Text Library

Principle

All text for your project is managed centrally in the Text Library except for text from the Graphics Designer. Each text entry is assigned a unique ID, by which the texts in WinCC are referenced.

If there is a large number of texts stored in the Text Library, you can export the texts via a user-friendly export and import function, translate them outside WinCC and then re-import them.

If you set another project language in an editor, such as Alarm Logging, the corresponding language column is automatically created in the Text Library.

If you create a new project and open the Alarm Logging before opening the Text Library, language columns are created for each of the languages installed with WinCC. At the same time, all WinCC standard texts (default names of message classes, WinCC system messages, etc.) are entered in the languages installed with WinCC.

Note:
With languages not installed with WinCC, note that the standard texts are entered in the current WinCC GUI language because these other languages are not stored in WinCC. If the project language and the WinCC GUI language are different, you must also translate the standard texts.

Links between the WinCC editors and the Text Library

The following WinCC editors store your texts in the Text Library:

- Alarm Logging: All user defined texts (messages, points of error, etc.) and standard texts (default names for message classes, message status, etc.)
- Graphics Designer: Only texts for the Text List, if so configured
- User Administrator: Authorizations
- User Archive (optional): All configured texts (field names, etc.)
- Picture Tree Manager (optional): All configured texts

Font administration

You can set a custom font for each language created in the Text Library. This is particularly important if you are configuring in non-Latin fonts and need to create, say, a Cyrillic font. The set font is applied at the referenced locations in WinCC.

15.4.1 Working with the Text Library

The principle of the Text Library

The majority of the texts for your project are managed centrally in the Text Library. You can export and import the texts for external translation or translate them in the Text Library. The translated texts are then available within WinCC in the target language concerned.

You can also edit Text Library texts:

- Add Line: You add lines into the Text Library so that you can insert new terms and expressions that you prefer not to create in the WinCC editors. This may be the case, for instance, if you wish to link a term by means of a text reference or if you want to create terms for Alarm Logging once only and then use them in multiple locations in Alarm Logging. For further information see "Add Line".
- Copy and Past Text: If texts occur repeatedly in the Text Library, you can copy individual terms and then paste them at other locations. For further information see "Copy Text".
- Setting fonts: You can define a custom font for each language you have created in the Text Library. This is particularly important for non-Latin fonts. For further information see "Setting the language font".
- Creating a new language: Before configuring a new language or handing it out for translation, you must create a corresponding column in the Text Library. You can use all the languages that your operating system supports (primary languages). For further information see "Creating a new language".
- Deleting a language: If you are no longer intending to use one of the languages in your project, you can delete that language together with all of its associated entries in the project by means of a centralized operation in the Text Library. For further information see "Deleting a language".

15.4.2 How to translate texts from the Text Library

The procedure in principle

You can translate texts that are managed in the Text Library either within the Text Library itself or externally:

- If you only manage a few texts in the Text Library, you can translate them straight in there.
- If you are managing large quantities of text in the Text Library, for instance if you have configured a project containing many messages, translate the texts outside the Text Library.

Translating texts in the Text Library

When translating directly in the Text Library, input the texts as follows:

- Select the line for the term concerned in an existing language: The ID of the source text and the translated texts must agree.
- Select the column for the target language: All entries for the given language are entered in that column.

Caution
Do not delete or move any lines or fields during translation, otherwise text references in your project will be lost.

Translating texts outside the Text Library

To have texts from the Text Library translated externally, proceed as follows:

1. Export the texts from the Text Library (for further information see "Exporting Text Library text").
2. Translate the text in an external editor such as Excel (for further information see "Translating Text Library text externally").
3. Import the translated texts into the Text Library (for further information see "Importing text into the Text Library").

15.5 Reports for multilingual projects

Principle

There are two types of report in WinCC:

- Reports output during configuration to provide an overview of the configured data (project documentation). These reports can be output in all languages installed with WinCC. In the project documentation you can document all multilingual texts you configured.
- Reports that are output in runtime, such as regular printouts of measurement data. These reports are designed to be output in the set runtime language.

How to create multilingual project documentation

You use project documentation in WinCC to document your configured data. You can create project documentation with data from the following editors:

- WinCC Explorer
- Graphics Designer
- Alarm Logging
- Tag Logging
- Text Library
- User Administrator

Rules for multilingual project documentation

The following rules apply to multilingual reports involving project documentation:

- Headings and table names in the report are output in all installed WinCC languages.
- Object properties are output in the current WinCC GUI language.
- Configured texts are output in the runtime language that you have set up in the computer properties. If runtime is active when project documentation is being created, configured texts are output in the current runtime language.

Examples

1. Your WinCC GUI language is German, the Runtime language is English, Runtime is not active:

- Headings and table names are output in all installed WinCC languages.
- Object properties are output in German.
- Texts configured in English are output in English. If configured texts do not exist in English, the string "???" is output.

2. Your WinCC GUI language is French, the Runtime language is English, Runtime is active and after changing the language runs in Italian:

- Headings and table names are output in all installed WinCC languages.
- Object properties are output in French.
- Texts configured in Italian are output in Italian. If configured texts do not exist in Italian, the string "???" is output.

15.6 Displaying regional date and time

Principle

Regional date and time displays are language-dependent. In some countries the European format is displayed (Day.Month.Year) and in some the American format (Year/Month/Day). You must also pay attention to this format in the case of multilingual objects.

You configure date and time displays in Alarm Logging and the Graphics Designer.

Alarm Logging

In Alarm Logging you can select the format of the date and time in the system blocks "Date" and "Time", choosing between the European and American formats. The configured format applies to the whole project and is not affected by changing the runtime language.

If you insert an AlarmControl in Graphics Designer, it accepts the date and time setting that you configured in Alarm Logging.

Graphics Designer

The following objects, which you can configure in Graphics Designer, display the date and time:

- WinCC AlarmControl: You configure the format of the date and time in the Alarm Logging. The configured setting applies to the total project.
- WinCC TrendControl, WinCC TableControl: You can configure format of the date and time directly in the Properties of the Control concerned. The settings apply to the current picture.
- D/A Clock: The WinCC ActiveX Control Digital/Analog Clock accepts the local computer-specific settings from your operating system settings. They cannot be changed in the Control.

Note:
When configuring a Chinese project, you must set China (Taiwan) as the input locale and then adjust the date and time display format in the Control Panel of your operating system.

Report Designer

In the Report Designer you can adapt the format for the date and time by setting the corresponding parameters for the "Format" attribute in the "Object Properties" dialog box of the "Date/Time" system object in the "Miscellaneous" group. You can get an overview of the possible parameters and settings from the Direct Help on this attribute.

Display in runtime

In runtime the date and time displays are not changed in the event of a language change. Therefore always configure for a primary language.

For further information, please refer to the online Help on the Graphics Designer, Alarm Logging and the Report Designer.

15.7 Languages in Runtime

Runtime language

You set the language in which your project is displayed in runtime centrally, in the computer properties in the WinCC Explorer.

You can select as the runtime language any language for which a language column has been created in the Text Library.

Displaying untranslated texts in runtime

Untranslated texts from graphic objects are displayed in runtime as "???". If message system texts are not translated, nothing is displayed. By this means you can test your project in runtime to check whether all texts are present.

If you have configured a change to a language that has not been created in the Text Library, the previously set language continues to be displayed.

Changing languages in runtime

When you have configured a project in multiple languages, configure an operating element which will allow the operator to change the language in runtime. WinCC offers you several preconfigured operating elements that you can use in your project. You can of course also use dedicated picture objects for a language change, e.g. buttons.

If the new language originates from a different language area than the previous one, you must also change the operating system language (system locale). This is necessary so that the character set works with the proper codepage. Restart your operating system after changing the language.

16 Setting up User Administration

Functional principle

The "User Administrator" editor is used to set up a user administration system. The editor is used to assign and check authorizations which allow users to access the individual editors of the configuration system and to access the functions during runtime. Access rights to WinCC functions, the "authorizations", are assigned in the User Administrator. These authorizations can be assigned to individual users or to user groups. Authorizations can also be assigned during runtime.

When a user logs on to the system, the User Administrator checks whether the user is registered. If the user is not registered, he has no authorizations whatsoever. This means that he cannot call up or view data, or perform control operations.

If a registered user calls up a functionality protected by access authorization, the User Administrator checks whether the user has the appropriate authorization to permit this. If not, the User Administrator denies access to the desired functionality.

The User Administrator also provides the configuration functions for the "Variable Login" function which allows a user to log on to the workstation by means of a tag value which is set using a key-operated switch, for example. The automatic logout of a user after a certain time is also configured in the User Administrator.

If the WinCC "Chipcard" option is installed, the User Administrator provides functions for maintaining chipcards.

Restrictions in the User Administration system

Object	Maximum number
Authorizations	999
Users	128
User groups	128
Ranges	256

16.1 Project window

Introduction

The user administration data is displayed in the project window. The window consists of:

- the navigation window (left)
- and the multi-segment table window (right).

16.1.1 Navigation window

The navigation window contains a tree structure showing the configured groups and the associated registered users. The selected user name is displayed in the field above the navigation window. The "User" menu can be opened as a context menu for the selected user or user group.

16.1.2 Table window

Introduction

In the table window, the login name and the associated settings are displayed for the selected user and group.

"Login Only Via Chipcard" checkbox

If a user is to be able to log in by means of his chipcard only, this can be set by activating the "Login Only Via Chipcard" checkbox.

Note:
The "Login Only Via Chipcard" checkbox is displayed if the WinCC "Chipcard" option has been installed. A chipcard reader does not have to be connected to the configuration computer. As a result, it is possible to set up the "Chipcard" function and use it on other WinCC computers without a card reader needing to be available during configuration.

"Automatic Logout" area

If the automatic logout function is to be activated for the selected user, the time and starting point for this time can be entered in this area. The automatic logout prevents unauthorized persons having unlimited access to the system.
If "0" is entered in the input field, the function is deactivated and the user remains logged in until the system is shut down or another user logs in.

If the "Absolute Time" option button is set, the configured time for the automatic logout begins to run down when the user logs in, regardless of whether control operations are performed by the user in the meantime.
If the "Idle Time" option button is set, the configured time begins to run down from the point at which the user last performed a keyboard or mouse operation. The automatic logout only occurs after this pause in operator control.
When a user has been logged out automatically, the same or a different user can log in during runtime.

"Web Navigator" checkbox and "Web Options" area

If the checkbox is set, the "Web Options" area is displayed. In this area, the settings for the start screen and language are configured and then applied if the user dials into the WinCC project from the web. Only images which are available on the Web Navigator can be selected as the start screen.
The checkbox can also be activated using the "Web Navigator" button in the toolbar.

Authorizations table

The lower part of the table window shows the configured authorizations. Each line contains one authorization.
The number of displayed authorizations depends on the installed options, e.g. the "Basic Process Control" option.
Authorizations with the numbers 1000 - 1099 are system authorizations which cannot be set, modified, or deleted by the user.

Authorization 1 "User Administration" is set by default for users in the "Administrator" group. This authorization cannot be deleted.

In the "Unlock" column, an authorization can be assigned to the selected user by double-clicking in the desired line.

Each authorization must be assigned separately. Multiple authorizations can only be transferred according to the group assignment when you add a new user to a group. The table can be edited by selecting the "Table" menu item.

16.2 Overview of User Administration system structure

Introduction

The following basic steps are necessary to set up a user administration system:

1. Add the required groups.

2. Select the appropriate authorizations for the groups.

3. Add the users and assign the respective login names and passwords. The properties of the group can be copied when new users are added. In this case, it is advisable to assign the users to groups with authorizations which you want the users to have.

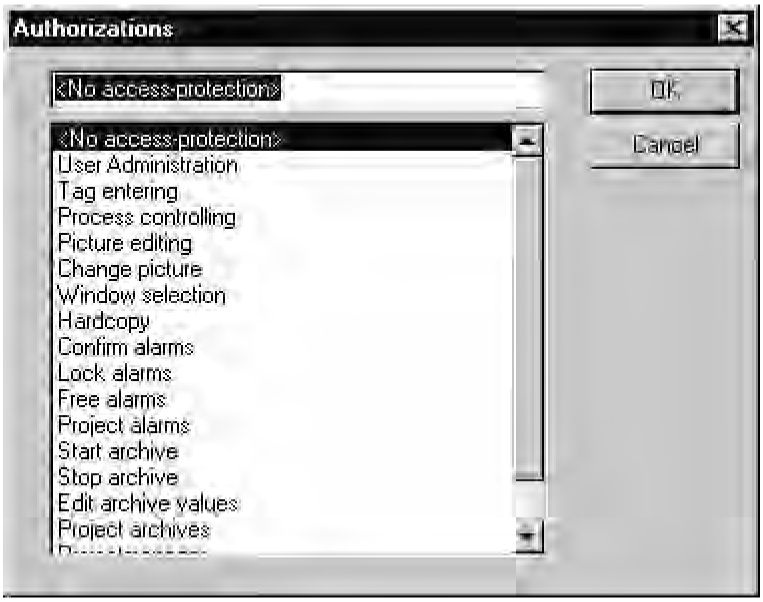
4. Select the specific authorizations for the various users. It is also possible to set a time here after which the system is to automatically log out the user in order to protect the system from unauthorized entries. It can also be determined whether the user should be able to log in by means of the chipcard only and which user-specific settings apply if the user should dial into the system from the web.

The data is applied without being stored.

16.2.1 Selecting an authorization in other editors

Introduction

If you want to assign an authorization in a different editor, e.g. an attribute of an object, the following dialog appears.



The authorizations are displayed in numerical order. If you select an authorization here, the function or control operation will be unlocked during runtime only for users who have the appropriate authorization.

You can open the dialog in various ways, e.g. by means of the "Authorization" attribute in the properties for an option button in the Graphics Designer.

16.2.2 Operation during runtime

Introduction

The following steps are necessary for a user to log in during runtime:

1. Start the runtime system.
2. Now open the password dialog using the shortcut keys for "Login" defined in the WinCC Explorer (Project Properties - Hotkey tab).
3. Enter your login name and the password in the dialog.

The system checks the authorizations you have been assigned with those of the editors and the installed components. If the authorizations match, the system unlocks the editors and components so that they can be used.

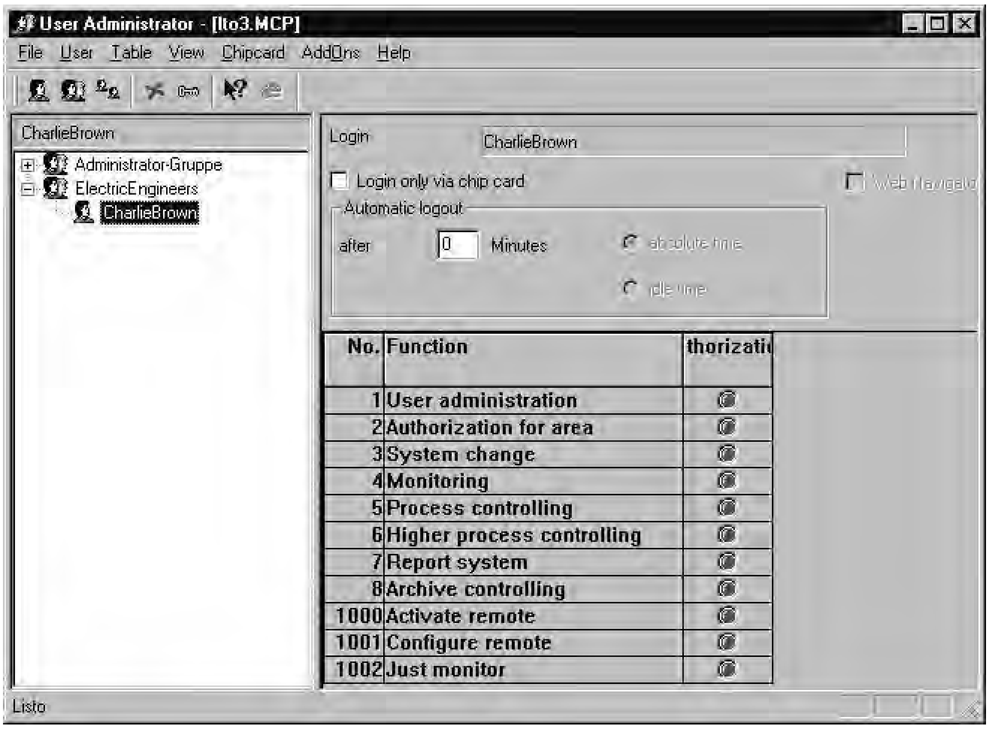
Note:
If a large number of authorizations have been assigned to a user, i.e. around the maximum number of 999 authorizations per user, several minutes may be required to log in the user.

16.3 WinCC options for the User Administrator

Introduction

Installing WinCC together with the options will extend the functionality of the User Administrator.

- The OS Project Editor in the "Basic Process Control" option changes the number and function of the authorizations in the table window. The authorizations then correspond to the PCS7 user hierarchies.
- The "Chipcard" option inserts the "Chipcard" item together with the associated functions in the menu bar and the "Login Only Via Chipcard" checkbox can then be activated in the table window.


User Administrator - [lto3.MCP]
File User Table View Chipcard AddOns Help
CharlieBrown
Administrator-Gruppe
ElectricEngineers
CharlieBrown
Listo

Login CharlieBrown

□ Login only via chip card

□ Web Navigato

Automatic logout

after 0 Minutes ● absolute time ○ idle time

No.	Function	thorizati
1	User administration	●
2	Authorization for area	●
3	System change	●
4	Monitoring	●
5	Process controlling	●
6	Higher process controlling	●
7	Report system	●
8	Archive controlling	●
1000	Activate remote	●
1001	Configure remote	●
1002	Just monitor	●

16.3.1 Extended "Chipcard" menu

Introduction

The User Administrator provides functions for controlling a chipcard reader. You can write to and check chipcards in the configuration system. The "Chipcard" menu is deactivated during runtime.

Requirements

The following requirements must be met before WinCC with the "Chipcard" option can be used:

- the "Chipcard" option must be installed and
- an interface (e.g. COM1 or COM2) must be assigned to the chipcard reader.

When these requirements have been met, the "Chipcard" menu becomes available and the "Login Only Via Chipcard" checkbox is displayed in the table window.

Note:
No Windows Administrator rights are required to write to and check chipcards in the configuration system or to use the chipcards during runtime.
To be able to write to and read a chipcard, the hardware connection between the chipcard reader and computer must be connected before the computer is started.

16.3.2 Chipcard reader during runtime

Introduction

To log into WinCC, the user inserts his chipcard into the chipcard reader and the required data is read out. If the chipcard is inserted, it is not possible to log on at the system manually. The user remains logged on at the system until he removes the card from the chipcard reader. The "Automatic Logout" function whereby the user is logged out automatically after a set time is only possible in conjunction with chipcard operation.

Note:
The "Chipcard" menu is deactivated during runtime since the functions can only be used in the configuration system.

17 VBA for Automated Configuration

17.1 Introduction: Using VBA in WinCC

Introduction

In the Graphics Designer there is a VBA editor available to you with which you can automate the configuring of pictures. The VBA editor is identical to the one from the products in the Microsoft Office family. You can make direct use of your VBA programming experience.

Principle

With VBA you extend the functionality of the Graphics Designer and automate configuring. You can use VBA in the Graphics Designer in the following ways:

- Create user-defined menus and toolbars
- Create and edit Standard, Smart and Windows objects
- Add dynamics to properties of pictures and objects
- Configure actions in pictures and objects
- Access products that support VBA (e.g. products in the MS Office family)

A description of the VBA object model for the Graphics Designer is provided in this documentation in the section headed "VBA reference".

17.1.1 Differentiation: Deployment of VBA

Introduction

You can use VBA exclusively for configuration and functional enhancement in the Graphics Designer. The following indicates where there are better options available for efficient configuration or where VBA cannot be used.

VB and C Scripts

VB and C scripts are active only at runtime and are used to dynamize picture and object properties as well as in action configuration.

Dynamic Wizards

The dynamic wizards are not replaced by VBA. However, VBA allows you to enhance the functionality of the dynamic wizards with ease.

ODK

ODK comprises function calls that enable access to all the functionality of WinCC both in the configuration system and in runtime. In contrast to ODK, VBA offers simple object-oriented access to the objects of the Graphics Designer.

17.1.2 Organizing VBA code in a WinCC project

Introduction

You organize the VBA code for your WinCC project in the VBA editor. This is where you specify whether the VBA code is to be available in only one picture, in the entire project or in all projects. Depending on where you place the VBA code, the term used to refer to the code is:

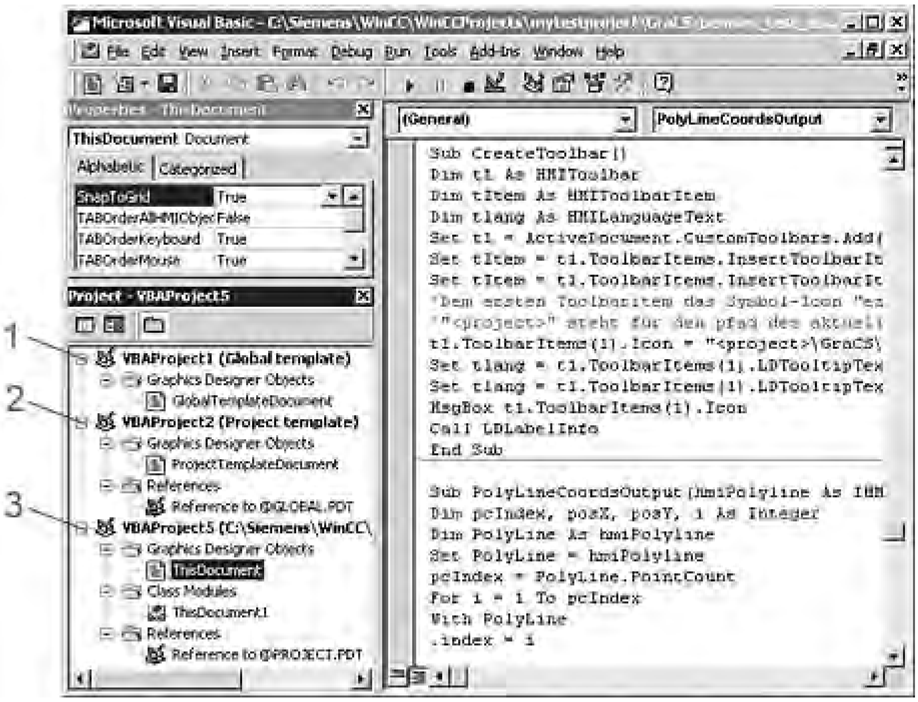
- global VBA code,
- project-specific VBA code or
- picture-specific VBA code.

Note:
A picture in the Graphics Designer is known as a document in the "VBA Object Model".

The VBA editor

To start the VBA editor in the Graphics Designer, press <ALT + F11 or choose > "Tools" > "Macros" > "Visual Basic Editor". If you have not yet opened a picture in the Graphics Designer, you can only edit the global or project-specific VBA code.

The global and project-specific data and all open pictures are displayed in the VBA editor's Project Explorer:



Microsoft Visual Basic - C:\Siemens\WinCC\WinCCProjects\mybestproject\GraCS\
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Properties - ThisDocument
ThisDocument Document
Alphabetic
Categorized
Project - VBAProject5
1
2
3
VBAProject1 (Global template)
Graphics Designer Objects
GlobalTemplateDocument
VBAProject2 (Project template)
Graphics Designer Objects
ProjectTemplateDocument
References
Reference to @GLOBAL.PDT
VBAProject5 (C:\Siemens\WinCC\
Graphics Designer Objects
ThisDocument
Class Modules
ThisDocument1
References
Reference to @PROJECT.PDT
(General)
PolyLineCoordsOutput

```
SnapToGrid              True
TABOrderAllHMIObjec     False
TABOrderKeyboard        True
TABOrderMouse           True
```

```
Sub CreateToolbar()
Dim tl As HMIToolbar
Dim tItem As HMIToolbarItem
Dim tlang As HMILanguageText
Set tl = ActiveDocument.CustomToolbars.Add(
Set tItem = tl.ToolbarItems.InsertToolbarIt
Set tItem = tl.ToolbarItems.InsertToolbarIt
'Dem ersten Toolbaritem das Symbol-Icon "ez
'"<project>" steht für den pfad des aktuel
tl.ToolbarItems(1).Icon = "<project>\GraCS\
Set tlang = tl.ToolbarItems(1).LDTooltipTex
Set tlang = tl.ToolbarItems(1).LDTooltipTex
MsgBox tl.ToolbarItems(1).Icon
Call LDLabelInfo
End Sub

Sub PolyLineCoordsOutput(hmiPolyline As IHM
Dim pcIndex, posX, posY, i As Integer
Dim PolyLine As hmiPolyline
Set PolyLine = hmiPolyline
pcIndex = PolyLine.PointCount
For i = 1 To pcIndex
With PolyLine
.index = i
```

Global VBA code (1)

Refers to VBA code that you write to the "GlobalTemplateDocument" in the VBA editor. This VBA code is saved in the "@GLOBAL.PDT" file, which is located in the WinCC installation directory.

The VBA code that you put in the "GlobalTemplateDocument" is the code that you want to be made available in all WinCC projects on your computer. If you need the VBA code on a different computer, use the export and import functions in the VBA editor.

A WinCC computer uses only the @GLOBAL.PDT stored locally in the WinCC installation directory.

Project-specific VBA code (2)

Refers to VBA code that you write to the "ProjectTemplateDocument" in the VBA editor. This VBA code is saved in the @"PROJECT.PDT" file, which is located in the root directory of each WinCC project.

The "@PROJECT.PDT" file has a reference to the "@GLOBAL.PDT" file. Functions and procedures which you have saved in the "@GLOBAL.PDT" file can be called up directly in the "ProjectTemplateDocument".

The "ProjectTemplateDocument" is where you put VBA code that you want to use in all pictures in the open project. If you need the VBA code on a different computer, use the export and import functions in the VBA editor.

You can open and edit the "@PROJECT.PDT" file in the same way as a PDL file. This means that you can use the "@PROJECT.PDT" file as a template file. For example, you can create the basic picture of your plant here, which is then automatically copied to every new PDL file in the project. Picture properties such as layers or zoom are not copied to the PDL file, nor is the VBA code.

Picture-specific VBA code (3)

Refers to VBA code that you write to the document "This Document" relating to the corresponding picture in the VBA editor. This VBA code is saved as a PDL file together with the picture.

The PDL file has a reference to the "@PROJECT.PDT" file. Functions and procedures which you have saved in the "@PROJECT.PDT" file can be called up directly from the PDL file. However, you do not have access to functions or procedures that are stored in the "@GLOBAL.PDT" file.

Note:
You can create modules, class modules and user forms in each document.

Note:
You can protect the VBA code of a module against unauthorized access by setting a password. To do this, select the "Tools" > "VBAObject Properties" menu item in the VBA editor.

Special features during the execution of VBA macros

The following rule applies when VBA macros are executed: first picture-specific VBA code is executed, then project-specific VBA code. If therefore you call a VBA macro that is contained for example both in the picture and in the project-specific VBA code, only the VBA macro from the picture is executed. This has the effect of preventing VBA macros and functions from being executed twice, which can lead to errors.

In connection with event handling the forwarding of events is activated by default. You can prevent events from being forwarded if you respond to an event in the picture-specific VBA code only.

Further information on this topic is given under "Event handling".

Testing with the Debugger

You can test your VB scripts at runtime with the VBA editor's debugger. You will find further information in the help system of the VBA editor.

17.2 VBA in the Graphics Designer

Introduction

You use VBA in the Graphics Designer in order to automate frequently recurring steps during configuring. You can create user-defined menus and toolbars in order to make it easier to execute the VBA macros that you have created.

Basically, in the Graphics Designer you can replace all configuring work that you would otherwise perform with the mouse with VBA macros. This applies in particular to the GUI (layers and zoom) and the editing of objects in pictures including dynamics.

Adapting Graphics Designer with VBA

The Graphics Designer is represented by the Application object in VBA. With VBA you can carry out configuring in the Graphics Designer in several languages, create user-defined menus and toolbars and access the component library.

Editing pictures with VBA

A picture in the Graphics Designer is represented by the Document object.

With VBA you can access the properties of the picture and edit settings for layers and the zoom factors. As well as this you can create picture-specific menus and toolbars. These are only visible, however, for as long as the picture is active.

Editing objects with VBA

An object in the picture is represented by the HMIObject object. With VBA you can create and delete objects and access the object properties. For example, with VBA you can very quickly create a large number of objects with identical properties for your plant display.

Creating Dynamics with VBA

VBA enables you to add dynamics to properties and events of pictures and objects.

Event handling

With VBA you can respond to events that occur in the Graphics Designer or in a picture (e.g. when you insert a new object into a picture). You use event handling in order to execute VBA macros in certain program situations.

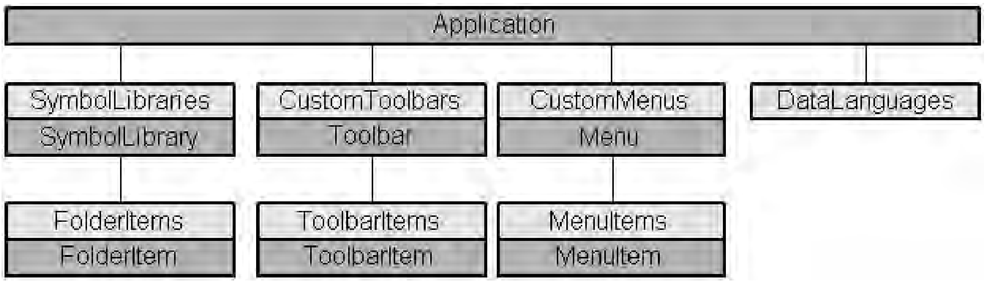
Access to external applications

You can use VBA to access programs which support VBA (e.g. products in the Microsoft Office family). This enables you, for example, to read out values from an Excel worksheet and then assign these to object properties.

17.2.1 Adapting Graphics Designer with VBA

Introduction

In VBA the Application object represents the Graphics Designer:



Access to the component library

VBA gives you full access to the component library. You can extend the component library with VBA by for example creating and deleting folders or copying objects and inserting them into a picture.

User-defined menus and toolbars

You can create user-defined menus and toolbars in order to execute VBA macros in the Graphics Designer. In this way you can extend the functionality of the Graphics Designer to suit your particular requirements.

Language-dependent configuring

With VBA you can carry out configuring in the Graphics Designer in more than one language. You therefore have access to the language-dependent object properties and you can create the user-defined menus and toolbars in different languages.

17.2.2 Language-dependent configuring with VBA

Introduction

With VBA you can carry out configuring in the Graphics Designer for several different languages. This gives you access to the language-dependent properties of objects in the Graphics Designer, while you can also make the user-defined menus and toolbars available in different languages. In VBA, foreign-language texts are stored in a list of the "LanguageTexts" type. The settings for language-dependent fonts are stored in a list of the "LanguageFonts" type.

Further information about language-dependent configuring is also provided in the WinCC documentation "Setting up multilingual projects".

Desktop language

You can only switch to a different desktop language in WinCC, not with VBA. When you switch desktop language in WinCC, the "DesktopLanguageChanged" event is triggered. You can adapt the user-defined menus and toolbars to suit the user by replacing language-dependent tool icons, for example.

The following objects and the associated language-dependent properties respond to a change to the desktop language:

- FolderItem object
- Menu object and MenuItem object
- ToolbarItem object
- Further information about the desktop language is provided in the WinCC documentation "Setting up multilingual projects" under "Language terms in WinCC".

Configuring language

You can change the configuring language with VBA using the "CurrentDataLanguage" property.

In this example the configuring language is changed to "English":

```
Sub ChangeCurrentDataLanguage()
'VBA1
Application.CurrentDataLanguage = 1033
MsgBox "The data language has been changed to English"
Application.CurrentDataLanguage = 1031
MsgBox "The data language has been changed to German"
End Sub
```

All language-dependent properties such as ToolTipText are affected by the change.

Configuring for more than one language in VBA

There are two possible ways for you to carry out configuring for several languages with VBA.

- Language changeover: text properties of objects.
- Language Texts listing: text properties of user-defined menus and toolbars, and objects.

Language changeover

You can change the language-dependent properties (e.g. "Text") of objects with VBA. To do this, assign the text to the corresponding property and then change the configuring language in order to assign the text in the other language.

LanguageTexts listing

You can save the multilingual texts for the respective object directly in the associated listing of the "LanguageTexts" type. To do this, enter the language ID for the language and the associated text.

You can find a list of the language identifiers in the WinCC documentation (>Index Language Identifier)

In this example a German label and an English label are assigned to the button "myButton":

```
Sub AddLanguagesToButton()
'VBA2
Dim objLabelText As HMILanguageText
Dim objButton As HMIButton
Set objButton = ActiveDocument.HMIObjects.AddHMIObject("myButton",
"HMIButton")
'
'Set defaultlabel:
objButton.Text = "Default-Text"
'
'Add english label:
Set objLabelText = objButton.LDTexts.Add(1033, "English Text")
'Add german label:
Set objLabelText = objButton.LDTexts.Add(1031, "German Text")
End Sub
```

17.2.3 Access to the component library with VBA

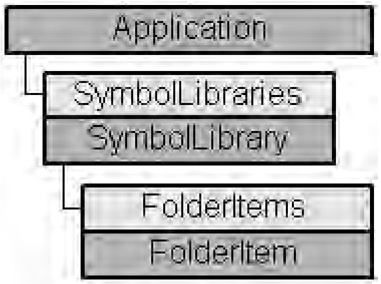
Introduction

The component library contains a large selection of ready-made objects which you can use to design your screens efficiently. The component library consists of a global library and a project-related library:

- The "Global Library" contains prepared objects that are supplied with WinCC. The objects are filed in folders, sorted according to subjects, such as valves, motors, cables and many others.
- The "Project Library" contains neither objects nor folders when you have created a new project. You can create objects which you need only in this particular project in the "Project Library".

With VBA you can access the component library without restriction. You can create and delete folders, and store objects in the component library or insert them into a picture.

Access to the component library with VBA



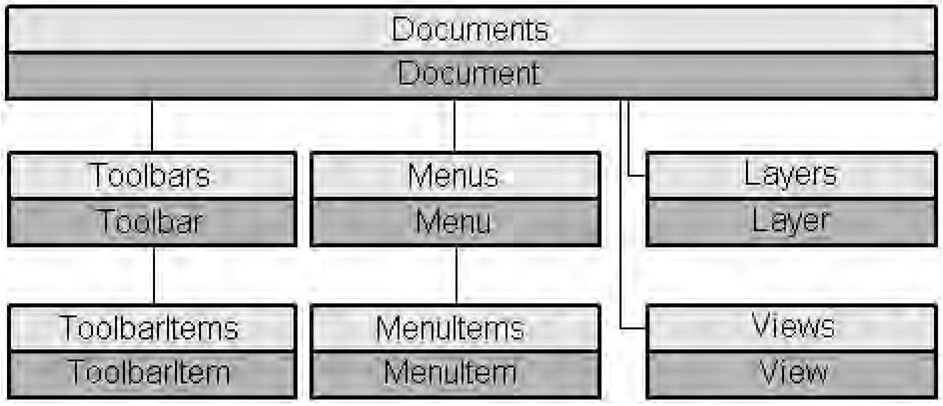
The component library is represented in VBA by the "SymbolLibraries" listing. The listing contains two elements, which represent the "Global Library" and the "Project Library". The "FolderItems" listing contains elements representing both folders and objects.

Note:
To address an object in the "SymbolLibraries" listing you use either the index number or the internal name.
You can find out the internal name by clicking the right mouse button on the relevant object in the component library and then choosing the "Copy path" command in the pop-up menu.
The path to the object within the component library is then copied to the clipboard.

17.2.4 Editing pictures with VBA

Introduction

Pictures visualize the process that is being monitored and operated. They show important process steps or parts of the plant and are a schematic representation of the production sequence. In VBA the picture is represented by the Document object.



Picture-specific menus and toolbars

In contrast with the application-specific menus and toolbars, the picture-specific menus and toolbars are coupled to a specific picture. The picture-specific menus and toolbars remain visible for as long as the picture is active.

You should use picture-specific menus and toolbars when the called VBA macros are only used in that picture.

Layers

You can access the layers in the Graphics Designer with VBA. Each layer is represented by the Layer object. By changing the properties of the Layer object you can specify among other things the layer names and the zoom settings.

You control the visibility of the RT layers via the Document object. You control the visibility of the CS layers via the View object.

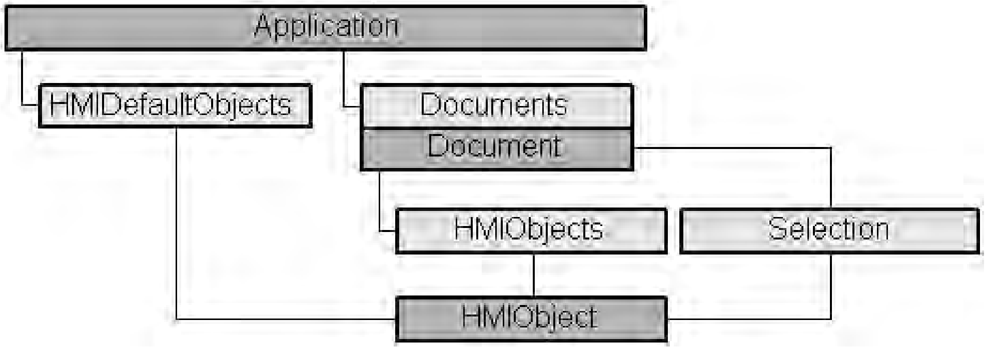
Copies of the picture

You can create copies of a picture with VBA in order to display different views of a picture. The copy of a picture is represented in VBA by the View object.

In the properties of the View object you can among other things set the zoom factor and specify which picture section is to be displayed.

17.2.5 Editing objects with VBA

Access to objects in the Graphics Designer



In VBA all object types of the current picture are contained in the "HMIObjects" listing. They are not divided according to object type (Standard, Smart, Windows and Controls objects) as in the Graphics Designer. With VBA you can therefore run through all objects in one or more pictures with a loop.

When you have selected objects in the picture, these objects are contained in the "Selection" listing. Use the "HMIDefaultObjects" listing if you want to change the default settings of the properties of an object.

To address an object in a picture with VBA, use either the object name, e.g. "ActiveDocument.HMIObjects("Circle1"), or the index number. "ActiveDocument.HMIObjects(1)" references, for example, the first object in the active picture.

Editing objects with VBA

You have the following possibilities for editing objects with VBA:

- Create a new object in a picture
- Delete an existing object
- Copy an existing object
- Group existing objects or cancel the grouping
- Search for objects
- Display or change object properties

When you insert a new object into a picture with VBA, the object behaves in the same way as if you double-clicked it in the Graphics Designer object palette. The object is given the predefined property values and is inserted in the top left-hand corner of the picture.

Access to the object properties is dependent on how you created the object. Two examples illustrate this:

Example 1:

In this example a circle of the type "HMIObject" is inserted into the current picture. You can use a VBA object of the "HMIObject" type or all objects in the Graphics Designer. However, you have to address individual properties of the respective object explicitly via the "Properties(Index)" property:

```
Sub AddObject()
'VBA30
Dim objObject As HMIObject
Set objObject =
ActiveDocument.HMIObjects.AddHMIObject("CircleAsHMIObject", "HMICircle")
'
'standard-properties (e.g. the position) are available every time:
objObject.Top = 40
objObject.Left = 40
'
'non-standard properties can be accessed using the Properties-collection:
objObject.Properties("FlashBackColor") = True
End Sub
```

Example 2:

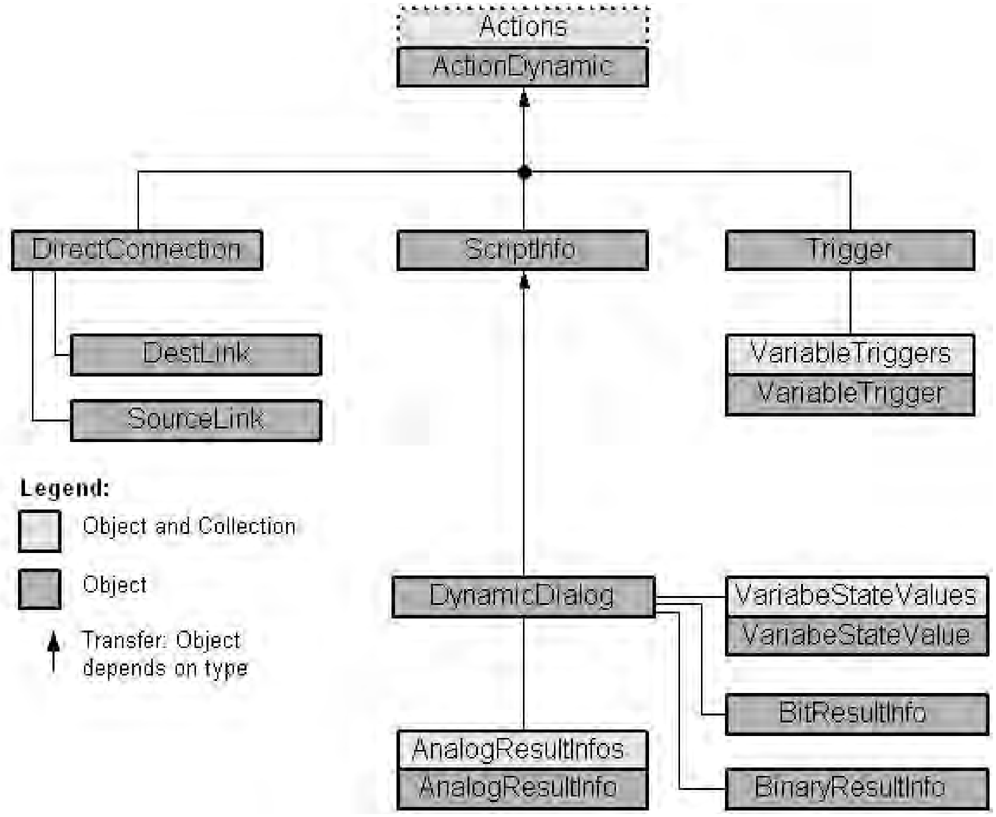
In this example a circle of the type "HMICircle" is inserted into the current picture. In contrast with Example 1 you can only use the "objCircle" object for objects of the "HMICircle" type, however:

```
Sub AddCircle()
'VBA31
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("CircleAsHMICircle",
"HMICircle")
'
'The same as in example 1, but here you can set/get direct the
'specific properties of the circle:
objCircle.Top = 80
objCircle.Left = 80
objCircle.FlashBackColor = True
End Sub
```

17.2.6 Creating Dynamics with VBA

Introduction

VBA allows you to add dynamics to properties of pictures and objects and to configure event-controlled actions. VBA provides you with the ActionDynamic object for this purpose:



The ActionDynamic object represents an interface that is dependent on the object type:

- When you configure a dynamic for a property (Property object), the ActionDynamic object inherits the properties of the ScriptInfo, Trigger and DynamicDialog objects.
- When you configure an event-controlled action (Event object), the ActionDynamic object inherits the properties of the ScriptInfo and DirectConnection objects.

Adding dynamics to properties of pictures and objects

VBA enables you to add dynamics to properties of pictures and objects. You can use variables, scripts or the Dynamic dialog to add dynamics. Using dynamics enables you, for example, to configure a color change for an object in runtime when the value of a variable changes.

Configuring event-controlled actions

You can configure event-controlled actions with VBA. An action (script or direct connection) is triggered then the defined event occurs in runtime. An event may be a change to an object property, for example, or the clicking of a button.

Editing triggers

You can edit triggers with VBA. Triggers are required when you use dynamics. They determine when a dynamic value is updated in runtime. This may occur at regular intervals, for example, or in the event of a picture change.

When you configure event-controlled actions, the event is the trigger.

17.2.6.1 Adding dynamics to properties of pictures and objects

Introduction

VBA enables you to add dynamics to properties of pictures and objects. Dynamic object properties can be changed as a function of a variable value in runtime. The following methods of adding dynamics are possible:

- Variable connection
- Dynamic dialog
- Scripts

Principle

The following example illustrates the principle of the procedure for adding dynamics to an object property:

```
Sub CreateDynamicOnProperty()
'VBA57
Dim objVariableTrigger As HMIVariableTrigger
Dim objCircle As HMICircle
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle1",
"HMICircle")
'
'Create dynamic with type "direct Variableconnection" at the
'property "Radius":
Set objVariableTrigger =
objCircle.Radius.CreateDynamic(hmiDynamicCreationTypeVariableDirect,
"'NewDynamic1'")
'
'To complete dynamic, e.g. define cycle:
With objVariableTrigger
.CycleType = hmiVariableCycleType_2s
End With
End Sub
```

Note:
Note that a variable is not created simply by specifying the variable name. Use the variable selection dialog to create the variable.

Variable connection

Use the VariableTrigger object to add a dynamic to a property with a direct or indirect variable connection. The dynamic property then responds in runtime to a change in value to the specified variable. To allow this in VBA, you need to specify the variable name (VarName property) and the cycle (CycleTime property).

Dynamic dialog

Use the DynamicDialog object to add dynamics to a property with the aid of the Dynamic dialog. The property with the dynamics responds in runtime to a variable's value ranges. The following objects are available for specifying the value range:

- AnalogResultInfos object: use this object to assign a fixed value to value ranges of a variable or a script. The fixed value is assigned to the dynamic property when the variable value or return value of the script is within the specified value range.

- BinaryResultInfo object: use this object to assign a fixed value to binary value ranges (zero and non-zero) of a variable or a script. The fixed value is assigned to the dynamic property when the variable value or return value of the script returns one of the two values.
- VariableStateValue object: use this object to assign a fixed value to the state (e.g. "High limit exceeded") of a specified variable. The fixed value is then allocated to the dynamic property when the state occurs.

Scripts

Use the ScriptInfo object to add a dynamic to a property with a C or VB script. The property with the dynamic responds to a script in runtime and is controlled via a trigger. Use the Trigger object for configuring the trigger.

17.2.6.2 Configuring event-controlled actions with VBA

Introduction

With VBA you can configure actions for pictures and objects which are triggered when predefined events occur. For example, when the mouse is clicked on an object in runtime a C script is called whose return value is used for the dynamics of an object property. The following methods of adding dynamics are possible:

- Direct connection
- Scripts

The events that are used for configuring event-controlled actions occur only in runtime and have nothing to do with the VBA event handlers.

Principle

You use the Events property for configuring event-controlled actions with VBA. The way this property is used depends on whether you are configuring an action on an object or picture or a property.

Configuring an action on an object or picture

An action that you configure on a picture or object is triggered when a predefined event occurs, for example when the object is clicked on with the mouse. You configure an action on an object with VBA by using the "Events(Index)" property, where "Index" stands for the triggering event:

```
Sub AddActionToObjectTypeCScript()
'VBA63
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle. Click on object executes an C-action
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB",
"HMICircle")
Set objEvent = objCircle.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
'
'Assign a corresponding custom-function to the property "SourceCode":
objCScript.SourceCode = ""
End Sub
```

Configuring an action on a property

An action that you configure on a property of a picture or object is triggered when the property value changes. You configure an action on a property with VBA by using the "Events(1)" property, where the index "1" stands for the event "Upon change":

```
Sub AddActionToPropertyTypeCScript()
'VBA64
Dim objEvent As HMIEvent
Dim objCScript As HMIScriptInfo
Dim objCircle As HMICircle
'Create circle. Changing of the Property
'"Radius" should be activate C-Aktion:
Set objCircle = ActiveDocument.HMIObjects.AddHMIObject("Circle_AB",
"HMICircle")
Set objEvent = objCircle.Radius.Events(1)
Set objCScript = objEvent.Actions.AddAction(hmiActionCreationTypeCScript)
'
'Assign a corresponding custom-function to the property "SourceCode":
objCScript.SourceCode = ""
End Sub
```

Direct connection

Use the DirectConnection object to configure a direct connection.

Scripts

Use the ScriptInfo object if you want an event to trigger a C or VB action.

17.2.6.3 Editing triggers

Introduction

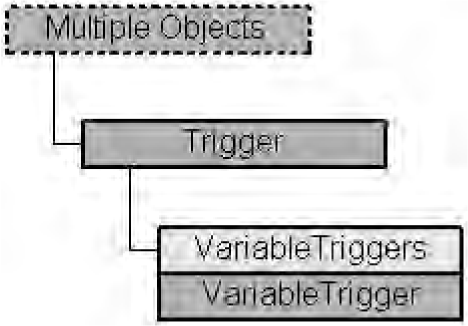
You use triggers in connection with dynamics for graphics objects and for triggering actions on object properties. Examples of triggers include:

- Variables: when the value of a variable is changed or its limit is exceeded at either extreme
- Standard cycle: cyclic execution of the action. The length of the cycles is selectable between 250 ms and 1 h. In addition, you can also use customized cycles that you define yourself.
- Picture cycle: a cyclic trigger is used as the trigger. This cycle provides the option of centrally defining the cycles of all the actions, tag connections and dynamic dialogs configured in a picture.
- Window cycle: a cyclic trigger is used as the trigger. This value applies to all actions, tag connections and dynamic dialog boxes used in the picture window which are configured with the trigger type "Window Cycle".

When you configure an action that responds to an event on a graphics object, the triggering event is the trigger.

Configuring triggers with VBA

Use the Trigger object to configure a trigger with VBA. If you intend to use a variable as the trigger, use the VariableTrigger object:



You determine the type of trigger with the Type property. Use the VariableTriggers property when you configure a variable as the trigger.

17.3 VBA in Other WinCC Editors

Introduction

VBA allows you to access other WinCC editors such as Tag Logging. The following editors can be automated with VBA in addition to Graphics Designer:

- Tag Management
- Tag Logging
- Text Library
- Alarm Logging

The functions for accessing the editors are contained in the HMIGO class.

Prerequisite

The "HMIGenObjects.dll" file is referenced. This happens automatically during WinCC installation.

Principle

To enable you to have access to the HMIGO class with VBA, you must reference the "HMI GeneralObjects 1.0 Type Library" in the VBA editor ("Project" > "References"). You must create a new instance of this class in the program code, e.g.:

```
 Dim HMIGOObject As New HMIGO
```

Create several different objects of this class if accessing several objects at the same time. For example, you need two instances of the HMIGO class in Tag Logging: the first to access the archive tags and the second to access the process value archive.

Usage

The functions provided by the HMIGO class provided you with access to Tag Management, Tag Logging, the Text Library, and Alarm Logging. To enable you to use the functions in VBA, you must have opened a project in WinCC. In addition, you can also access the properties of the class directly.

In this way, directly from the program code you can, for example, created several tags and change their values, edit text entries in the TextLibrary, or customize messages.

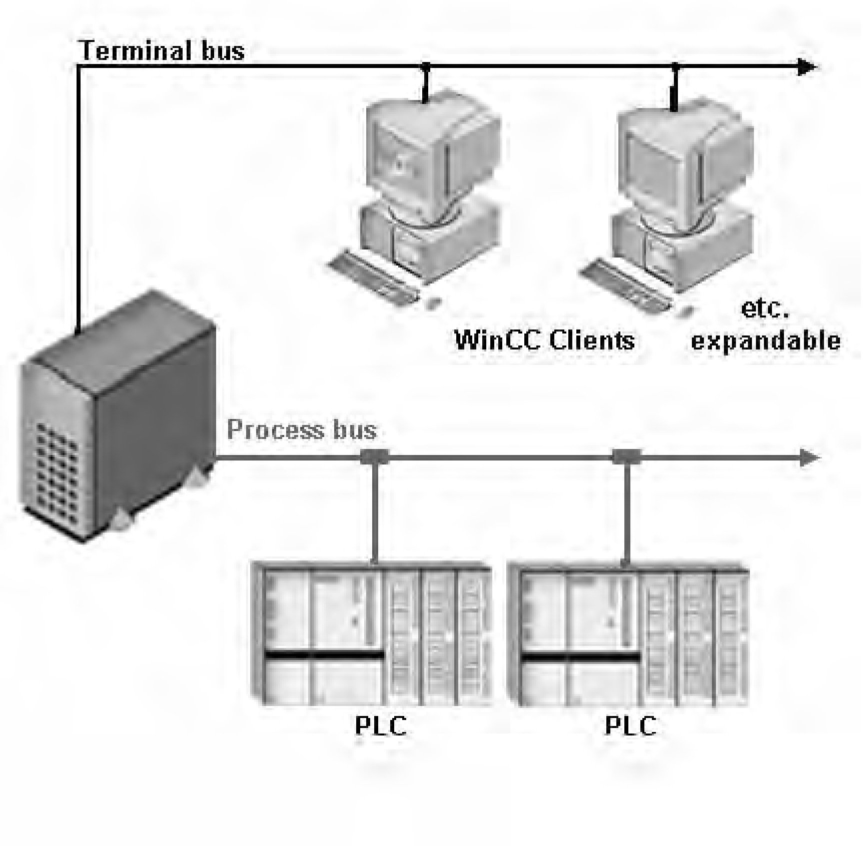
18 Communication

18.1 Basics of communication

Communication Tasks

Communication between WinCC and the automation systems is effected via the respective process bus, e.g. Ethernet or PROFIBUS. Communication is managed by specialized communication drivers known as channels. WinCC has channels for the automation systems SIMATIC S5/S7/505 and manufacturer-independent channels such as PROFIBUS DP and OPC. Moreover there is a variety of optional channels available for all common controls as an option or add-on.

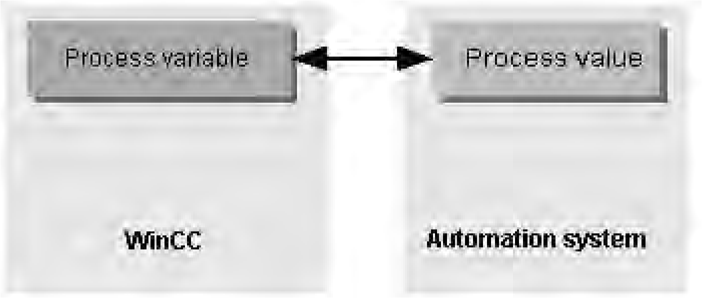
Communication with other applications, e.g. Microsoft Excel or SIMATIC ProTool, is effected with the assistance of the OPC (OLE for Process Control). When the WinCC OPC server is used, the data will be made available to other applications by WinCC. The data of other OPC servers can also be received by WinCC via the OPC client.



Communicating with the automation systems

Process tags form the link for data exchange between WinCC and the automation systems. Each process tag in WinCC corresponds to a certain process value in the memory of one of the connected automation systems. In runtime, the data area in which this process value is saved will be read out of the automation system by WinCC thus allowing the value of the process tags to be ascertained.

WinCC is also able to write data back into the automation system. This data is then processed by the automation system. In this manner, you can use WinCC to control process.



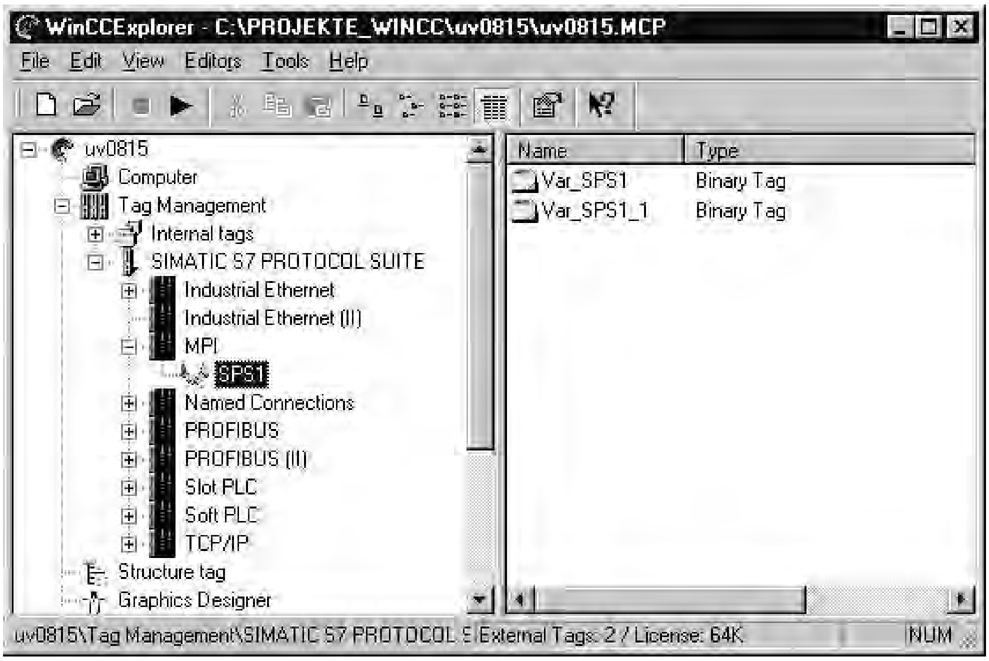
Communication by means of OPC

OPC clients can access WinCC data through the integrated OPC servers. The following type of access are possible:

- Access to WinCC tags through the WinCC OPC DA server.
- Access to the archive system through the WinCC OPC HDA server.
- Access to the message system through the WinCC OPC A&E server.

Channel units, logical connections, process tags

Communication between WinCC and the automation systems is effected via logical connections. These logical connections are arranged in a hierarchical manner over several levels. These individual levels are reflected in the hierarchical structure of WinCC Explorer.



Name	Type
Var_SPS1	Binary Tag
Var_SPS1_1	Binary Tag

uv0815\Tag Management\SIMATIC S7 PROTOCOL S External Tags: 2 / License: 64K

The communication drivers can be found at the highest level. These are also referred to as channels (e.g. the channel "SIMATIC S7 PROTOCOL SUITE").

One or more protocols have been provided for communication via a channel. The protocol is used to determine the channel unit that is to be used (e.g. "MPI"). This channel unit is then used together with a certain protocol to access a certain type of automation system.

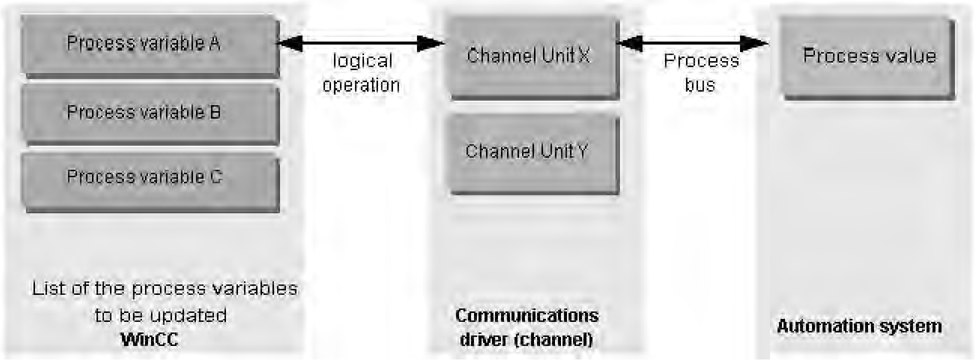
A channel unit can be used to establish logical connections to several automation systems which communicate via this channel unit (e.g. the automation system "SPS1"). A logical connection therefore represents the interface to an individual, defined automation system.

The process tags of the automation system are shown to the right of the data window in the case of each logical connection (e.g. the process tag "MyTag1").

The communication process in runtime

Up-to-date process values are required in runtime. Thanks to the logical connection, WinCC knows which automation system the process tags are located in and which channel is being used to process the data traffic. The process values are transferred via the channel. The data that is read will be stored in the working memory of the WinCC server.

The necessary communication steps are optimized by the channel in such a way that data traffic via the process bus is reduced to a minimum.



18.2 External Tags

Introduction

WinCC tags are required to access specific data in an AS. These tags, which require a connection to an AS, are referred to as external tags. Tags, which do not have a process connection, are called internal tags.

Data Types and Type Conversion

When configuring external tags, in addition to the tag name, you must also specify a data type and - for some data types - a type conversion:

The data type agrees with the data format in WinCC. Type conversion determines the conversion from AS format to WinCC format. The type conversion applies in both transmission directions:

- in the AS: e.g. for certain functions (such as timer values / BCD displays) or via the information to be addressed (such as byte address, word address in data block or I/O range)
- in WinCC: e.g. for analog values - for processing or calculating.

As a rule, the AS data format is fixed. There are then the following options for selecting the WinCC format:

- The WinCC data format can match the AS format. For this purpose, select the type conversion, which uses the same format on both ends and considers the sign of the WinCC data type, e.g. "WordToSignedWord". If this cannot be achieved with the selected data type, change it in WinCC.
- The WinCC format follows the value processing in WinCC.

You must observe the following points when selecting the data type and - if necessary - type conversion:

- Sign: Should it be considered in the conversion? Can negative tag values occur during operation? (for example, when differences in regulation values are presented in percent)
- Value range: Do the tag values, which can occur in operation, all fall within the value range available in each format or is it possible that the value could overflow the range in WinCC or on the AS? In the event of an overflow, a value may not be possible to display on the end or may lead to an error in the later processing.

- Different type conversions with the same value range: It is possible that several conversions of a data type may have the same value range, e.g. "ByteToUnsignedDword" and "ByteToUnsignedWord" with the value range [0...127]. f In such cases you should always determine the format of the data on the AS and check whether this format is not perhaps over-dimensioned and thus wasting resources. (e.g. DWord instead of Word).

If the value range required in the AS is not covered by the selected type conversion, change the data type in WinCC.

WinCC Data Types and Type Conversion

The following table shows which WinCC data types support type conversion.

Data Type	Type Conversion
Binary Tag	No
Unsigned 8-Bit Value	Yes
Signed 8-Bit Value	Yes
Unsigned 16-Bit Value	Yes
Signed 16-Bit Value	Yes
Unsigned 32-Bit Value	Yes
Signed 32-Bit Value	Yes
Floating-point Number 32-Bit IEEE 754	Yes
Floating-point Number 64-Bit IEEE 754	Yes
Text tag 8-bit character set	No
Text Tag 16-Bit Character Set	No
Raw data type	No

Note:
When using type conversions, make certain that the data sent from the AS can be interpreted by WinCC within the selected type conversion. If the data cannot be interpreted by WinCC, an error will be logged in the file "WinCC_sys_0x.log" in the directory "..\Siemens\WinCC\Diagnose".

Linear Scaling of Numerical Tag Types

Linear scaling can be used with numerical data types. The value range of a value in a process can be represented linearly in a specific value range of a WinCC tag.

As an example, a process could require that a nominal value be entered in the unit [bar], while WinCC should permit the user to enter this value in [mbar]. Using linear scaling, the value range in the process [0 ...1] can be converted to the value range [0 ...1000] of the WinCC tag.

Setting Text Tag Lengths

You must set the length of tags with the data type "Text Tag 8-Bit Character Set" or "Text Tag 16-Bit Character Set". A text tag, which should later take 10 characters, must be assigned a length of 10 in the case of an "8-Bit Character Set" or, in the case of a "16-Bit Character Set", a length of 20.

Addressing in an Automation System

WinCC tags are assigned to data areas in the AS. They must addressed in a particular manner in the AS. The type of addressing depends on the type of communication partner.

18.3 OPC - OLE for Process Control

18.3.1 Functionality

Introduction

OPC (OLE for Process Control) refers to a standard, manufacturer-independent software interface. The OPC interface is based on Microsoft Windows COM (Component Object Model) and DCOM (Distributed Component Object Model) technology.

COM

COM is the standard protocol for communication between objects located on the same computer but which are part of different programs. The server is the object providing services, such as making data available. The client is an application which uses the services provided by the server.

DCOM

DCOM represents an expansion of COM functionality to allow access to objects on remote computers.

This foundation allows standardized data exchange between applications from industry, administrative offices and manufacturing.

Up to that point, applications which accessed process data were tied to the access protocols of the communication network. The standard software interface OPC allows devices and applications from various manufacturers to be combined with one another in a uniform manner.

The OPC client is an application which accesses process data, messages and archives of an OPC server. Access takes place using the OPC software interface.

An OPC server is a program which provides the applications from various manufacturers with a standard software interface. The OPC server is the middleware between the applications for handling process data, the various network protocols and the interfaces for accessing these data.

Only devices with operating systems based on Windows COM and DCOM technology can use the OPC software interface for data exchange. At the present time, Windows NT, Windows 98, Windows 2000 and Windows XP use this software interface.

18.3.2 OPC Specifications

Introduction

The standard software interface OPC was defined by the OPC Foundation. The OPC Foundation is an alliance of leading companies in the field of industry automation. The OPC server from WinCC supports the following specifications.

- OPC Data Access 1.0a and 2.0
- OPC Historical Data Access 1.1
- OPC Alarm & Events 1.0

OPC Data Access (OPC DA)

OPC Data Access (OPC DA) is the specification for managing process data. The WinCC OPC DA Server from WinCC V 6.0 conforms to OPC DA specifications 2.0 and 1.0a.

OPC Historical Data Access (OPC HDA)

OPC Historical Data Access (OPC HDA) is the specification for managing archive data. The specification is an extension of the OPC Data Access specification. The WinCC OPC HDA Server as of WinCC V 6.x corresponds to OPC HDA specification 1.1.

OPC Alarms & Events (OPC A&E)

OPC Alarm & Events is a specification for transmitting process alarms and events. The WinCC OPC A&E Server as of WinCC V 6.x corresponds to the OPC A&E specification 1.0.

18.3.3 Using OPC in WinCC

Introduction

WinCC can be used as both an OPC server and as an OPC client. During installation of WinCC, the following WinCC OPC servers may be selected for installation:

- WinCC OPC DA Server
- WinCC OPC HDA Server
- WinCC OPC A&E Server

The following OPC components are installed automatically:

- OPC Communication Driver (OPC DA Client)
- OPC Item Manager

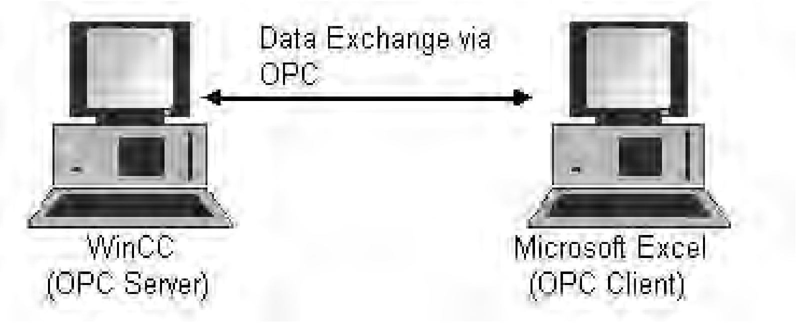
Licensing:

In order to use the WinCC OPC HDA server and WinCC OPC A&E server, it is necessary to apply for a license, a so called authorization. The "Connectivity Pack" authorization must be installed on the WinCC server which is implemented as the WinCC OPC HDA server or WinCC OPC A&E server. Further information is available under "Authorization".

Possible applications

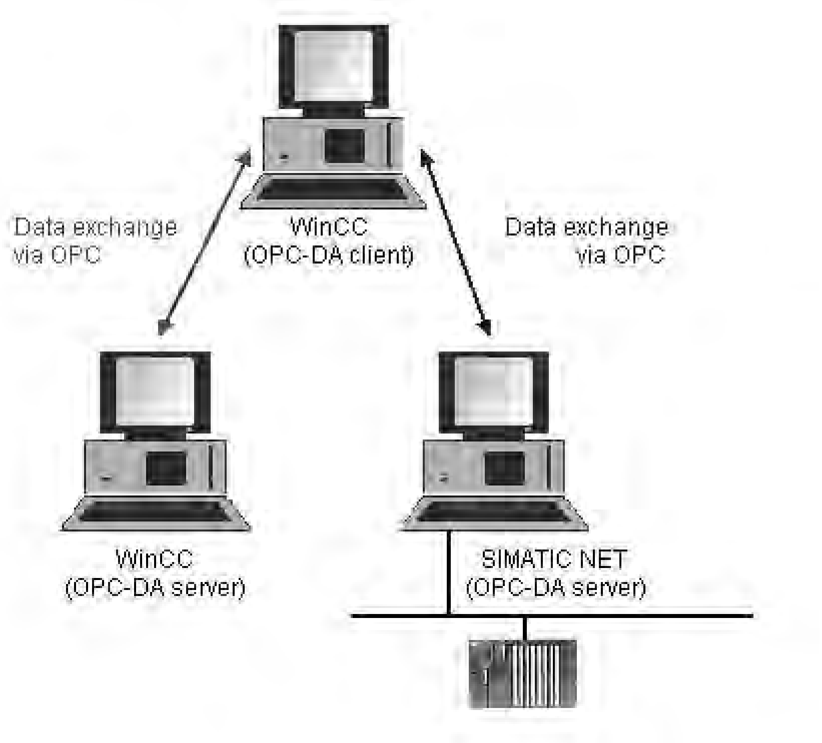
WinCC as an OPC DA server

The WinCC OPC DA server makes the data from the WinCC project available to other applications. These applications may be running on the same computer or on computers in the networked environment. In this way for example, WinCC tags may be exported to Microsoft Excel.



WinCC as an OPC DA client

If WinCC is to be used as an OPC DA client, the "OPC" channel must be added to the WinCC project. A connection for data exchange will then be created in the WinCC project of the WinCC OPC DA client; this will be used to handle access to the WinCC tag variables of the WinCC OPC DA server. To simplify the process, the OPC Item Manager is used. A WinCC OPC DA client can access multiple OPC DA servers. This requires that a connection be created for each OPC server. In this way, the WinCC OPC DA client may be used as a central operation and monitoring station.



WinCC OPC servers in a distributed system

In a distributed system, each WinCC server can monitor the entire installation. However, an individual WinCC server assumes only a single specific task type, such as message processing or archiving.

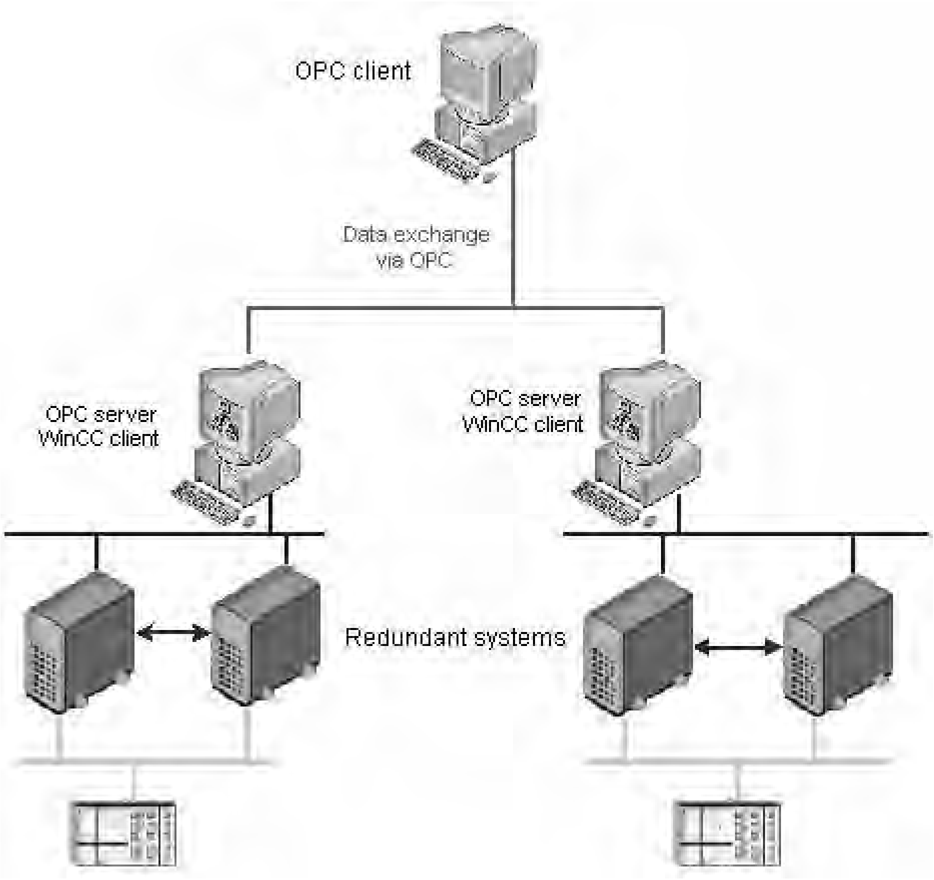
The WinCC OPC server enables access to the WinCC runtime data via the OPC software interface. WinCC OPC servers support the complete range of function found in the OPC specification.

Any software based on a particular OPC specification may be used as an OPC client. Thus for example the OPC client for analysis may be drawn from a variety of sources. Proprietary OPC clients may be created to best meet specific requirements.

WinCC OPC servers in a redundant system

In a redundant system, the WinCC servers monitor one another at runtime for early recognition of a server outage. The WinCC OPC servers make the WinCC runtime data available to the OPC client via the OPC software interface.

Any software program based on the corresponding OPC specification may be used as an OPC client. Thus for example the OPC client used for central monitoring may come from any of the various redundant systems. Proprietary OPC clients may be created to best meet specific requirements.



18.3.3.1 Functionality of the WinCC OPC DA Server

Introduction

The WinCC OPC DA server can be selected during the installation of WinCC. It is immediately usable after installation without further need of configuration.

The WinCC OPC DA server conforms to the OPC Data Access 1.0a and 2.0 specifications. This has been verified by compliance testing.

The WinCC OPC DA server is a DCOM application. This interface is used by the WinCC OPC DA server to make the required information about WinCC tag variables available to the WinCC client.

The WinCC OPC DA server is active if the WinCC OPC DA client is accessing it via a connection. To establish successful OPC communication, the following must be observed:

- The WinCC project of the WinCC OPC DA server must be enabled.
- The computer on which the WinCC OPC DA server runs must be accessible via its IP address.

18.3.3.2 Functionality of the WinCC OPC DA Client

Introduction

The channel "OPC" does not require a separate communication group. The channel "OPC" is an application which employs the OPC software interface to use an OPC DA server to access process data.

If WinCC is to be used as an OPC DA client, the "OPC" channel must be added to the WinCC project.

Data exchange between the WinCC OPC DA server and the WinCC OPC DA client occurs via WinCC tags. To do this, a connection is set up in the WinCC project of the WinCC OPC DA client; it is used to handle access to the WinCC OPC DA server.

For the WinCC OPC DA client to access multiple OPC DA servers, a connection for each of the OPC DA servers must be set up in the WinCC project.

More information about troubleshooting channels and tag variables may be found under "Troubleshooting".

18.3.3.3 Functionality of the WinCC OPC HDA Server

Introduction

The WinCC OPC HDA server is a DCOM application which makes data needed from the archive system available to the OPC HDA client. Access the data using Item Handles. The data can be read and analyzed.

The WinCC OPC HDA server supports the OPC Historical Data Access 1.1 specification. This has been confirmed by the compliance test.

The following chapter explains the design of the data structure, as well as the attributes, aggregates and functions supported by the WinCC OPC HDA server. This is not a detailed description, but rather a summary of the specific information. Additional information may be found in the "OPC Historical Data Access 1.1" specification.

Installation

The WinCC OPC HDA server can be selected during the installation of WinCC. After installation, the WinCC OPC HDA server is immediately available for use.

The WinCC OPC HDA server can only be implemented on a WinCC server.

Authorization

In order to operate the WinCC OPC HDA server, each WinCC server, to be implemented as an OPC HDA server, must have the Connectivity Pack installed in addition to the license for the WinCC basis system.

OPC HDA client

All OPC HDA clients which conforms with the OPC Historical Data Access 1.1 specification can access the WinCC OPC HDA server. Custom-developed OPC HDA clients can also be used. Using custom OPC HDA clients is the best way to fulfill specific requirements.

Examples of how an OPC HDA client may be used include:

- Analysis and evaluation of archived data.
- Statistical process control of archives from different OPC HDA servers.

OPC Historical Data Access 1.1

18.3.3.4 Functionality of the WinCC OPC A&E server

Introduction

The WinCC OPC A&E server is a DCOM application. The OPC A&E client is kept informed of status changes for WinCC messages by means of subscriptions. The OPC A&E client can apply a filter to the subscription. This filter determines which messages and attributes are displayed.

The WinCC OPC A&E server supports the specification OPC Alarm&Event 1.0. This has been confirmed by the compliance test.

The following chapter explains the display of the WinCC message system on OPC A&E, as well as the attributes supported by the WinCC OPC A&E server. This is not a detailed description, but rather a summary of the specific information. Additional information may be found in the "OPC Alarms & Events 1.0 specification.

Installation

The WinCC OPC A&E server can be selected during the installation of WinCC. After installation, the WinCC OPC A&E Server is immediately usable without configuration required.

The WinCC OPC A&E server can only be implemented on a WinCC server.

Authorization

In order to operate the WinCC OPC A&E server, each WinCC server, to be implemented as an OPC A&E server, must have the Connectivity Pack installed in addition to the license for the WinCC basis system.

Server types

The WinCC OPC A&E server is a conditional related event server. There are also simple event and tracking event servers.

Condition related event server

With a condition related event server, the event is associated with a condition. The condition, for example, might be the exceeding of a tag's bounding value. A message occurs in WinCC as soon as the bounding value is exceeded. This message is shown as an alarm in OPC A&E.

Simple event server

Simple events are messages that inform the OPC A&E client about events. Simple events include launching and exiting programs.

Tracking event server

If a change in a process occurs, the OPC A&E client receives a message. This might be a regulator adjustment for example.

18.4 WinCC "PROFIBUS FMS" Channel

Introduction

The communication driver is used to read and write automation system process tags that can be accessed using the PROFIBUS FMS protocol.

Channel Unit

The communication driver has a channel unit. Therefore, the following applications are possible:

- PROFIBUS FMS Channel Unit for SIMATIC NET PROFIBUS (Communication Module CP 5613).

Note:
A logical connection can be configured in either Configuration Mode or Runtime. Both approaches are described below.
In the case of a "PROFIBUS FMS" channel, only WinCC can request data from the connected AS. The AS cannot initiate the sending of data.

18.5 WinCC "SIMATIC S5 Ethernet TF" channel

Introduction

The communication driver is used for connection to the automation systems SIMATIC S5-115U/H, SIMATIC S5-135U and SIMATIC S5-155U/H over the Industrial Ethernet with the TF protocol (engineering functions).

Channel unit

The communication driver has a channel unit for operating a CP1613 communication module.

The logical device name can be modified by means of the system parameters of the channel unit S5 application (CP 1413-1).

The following option exists with regard to its use:

- Channel unit S5 application (CP 1413-1) - SIMATIC S5 Ethernet TF for the communication modules for SIMATIC NET Industrial Ethernet (such as CP 1613).

18.6 WinCC "SIMATIC S5 Ethernet Layer 4" Channel

Introduction

This communication driver is used for the connection to the SIMATIC S5 -115U/H, SIMATIC S5 -135U and SIMATIC S5 -155U/H PLCs by means of the ISO transport protocol or the TCP/IP protocol.

Depending on the communication driver employed, the following communication partners are used:

Communication Protocol	WinCC Side	SIMATIC S5 Side
ISO Transport Protocol	CP1612 (3Com-compatible) CP1613	CP1430 TF
TCP/IP (to RFC1006)	CP1612 (3Com-compatible) CP1613	CP1430 TCP

No local data basis is required to use this channel.

Channel Units

The communication driver possesses two 'CP1413-x' channel units, with which a maximum of two CP1413s or CP1613s can be operated. The functionality of the channel units is identical. They differ only in their logical device names.

With the third channel unit "TCP/IP", it is possible to use the TCP/IP protocol with a CP1612 or CP1613.

It is possible to set the logical device name in the system parameters of a channel unit. Here you can also set the parameters for the protocol used.

The following applications are available:

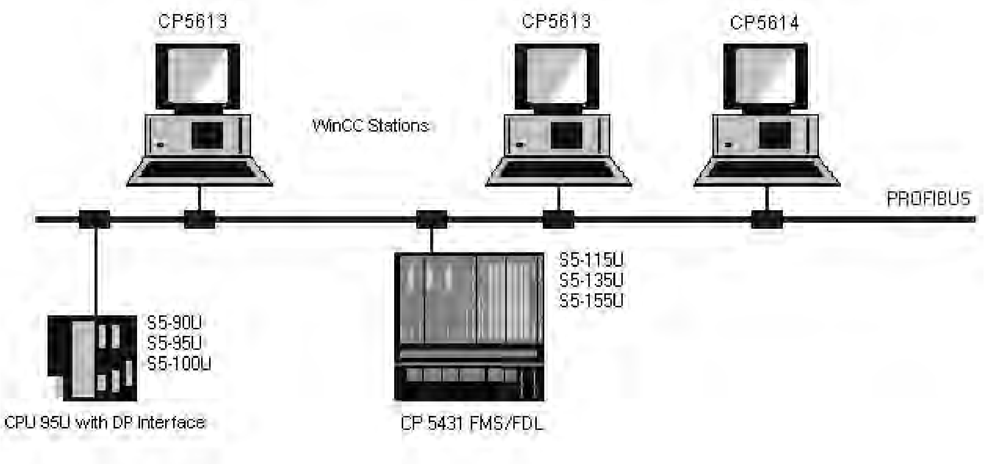
- "S5-Transport (CP 1413-1)" channel unit for the SIMATIC Industrial Ethernet communication modules (CP 1612/1613).
- "S5-Transport (CP 1413-2)" channel unit for the SIMATIC Industrial Ethernet communication modules (CP 1612/1613).
- "S5-Transport (TCP/CP)" channel unit for the SIMATIC Industrial Ethernet communication modules (CP 1612/1613).

18.7 WinCC channel "SIMATIC S5 Profibus FDL"

Introduction

The "SIMATIC S5 Profibus FDL" channel is used for communication between a WinCC station and a SIMATIC S5 automation system. In this instance the network type process field bus (PROFIBUS) and the field data link (FDL) are used.

PROFIBUS is the network for small to medium-sized quantities of data. With a maximum number of 127 connectable nodes, a wide range of automation tasks can be accomplished.



The reading and writing of tags over PROFIBUS, using the FDL protocol, is implemented using request and reply telegrams. The request telegram is sent by WinCC to the PLC. The automation system responds with the reply telegram.

An FDL connection is specified by the local and remote service access points.

Channel unit FDL (CP5412/A2-1)

Regardless of whether the communication processor being used is a CP 5613 or a CP 5614, the connection facility exists on the SIMATIC S5 via the "FDL (CP5412/A2-1)" channel unit. This channel unit supports as many as 24 connections. A channel unit and a connection have to be created for the channel to function.

Service access point

SAPs are local data ports within a PROFIBUS node. The SAPs have to be configured in WinCC and on the automation system. A unique identifier is defined with the service access point. This unique identifier is required for communication between WinCC and the automation system.

Active connection

An active connection is also referred to as the fetch connection. A fetch connection is a connection with which an active peer fetches data from a communication peer. The communication peer from which the data are fetched is termed the passive peer.

Passive connection

A passive connection exists when the active automation system asynchronously sends data to the passive WinCC peer without receiving a request telegram.

18.8 WinCC "SIMATIC S5 Programmers Port AS511" Channel

Introduction

The communication driver is used to set up a serial link to a SIMATIC S5 automation system via the TTY interface.

Channel Unit

The communication driver has a channel unit that supports the use of a COM-Port for the serial link.

The following application is available:

- S5-AS511 channel unit for serial communication via a "Siemens-specific" protocol.

18.9 WinCC "SIMATIC S5 Serial 3964R" channel

Introduction

The communication driver is used for the serial connection to the SIMATIC S5 automation system.

Channel unit

The communication driver has a channel unit for operating a COM port for the serial connection.

The following option exists with regard to its use:

- Channel unit S5-RK512 (3964R) for serial communication by means of the 3964R or 3964 protocol.

18.10 WinCC Channel "SIMATIC S7 Protocol Suite"

Mode of Operation

The Channel "SIMATIC S7 Protocol Suite" is used to link SIMATIC S7-300 and SIMATIC S7-400 automation systems.

Depending on the communication hardware used, the system supports connections via the following channel units:

- Industrial Ethernet and Industrial Ethernet (II): to communicate via a communication processor (e.g. CP 1612; CP1613) using SIMATIC NET Industrial Ethernet.
- MPI: For communication via the external MPI port of a programming device (e.g. PG 760/PC RI45), via an MPI communications processor or a communications module (e.g. CP 5511, CP 5613).
- Named Connections: to communicate with STEP 7 via a symbolic connection. These symbolic connections are configured using STEP 7 and are needed e.g. to provide high-availability communication with the AS S7-400 in conjunction with the redundancy in H/F systems.
- PROFIBUS and PROFIBUS (II): For communication via a communications processor (e.g. CP 5613) using the SIMATIC NET PROFIBUS.
- Slot PLC: to communicate with a Slot PLC (e.g. WinAC Pro), which is installed as a PC card in the WinCC computer.
- Soft PLC: to communicate with a Software PLC (e.g. WinAC Basis), which is installed as an application on the WinCC computer.
- TCP/IP: to communicate with networks using the TCP/IP protocol.

Further information regarding the diagnosis of the channel, connection and tags can be found under "Communication Diagnosis".

Detailed Procedure

Communication Manual: Here you will find additional information with detailed examples of channel configuration. This manual can be downloaded from "www.ad.siemens.de/meta/support/html_00/support.shtml". In the menu on the left, select the link "Service and Support". Under "Product Support", select "Find Manual". In the field "Search Term" on the left on the "Expert Search" tabbed card, enter the order number "6AV6392-1CA05-0AA0" for Volumes 1 and 2 of the manual.

Channel Unit Selection

Introduction

To setup a communication link to an existing or planned network, a selection must be made for:

- one of the channel's channel units
- a suitable communication processor for the WinCC station
- a suitable communication module for a specific automation system

This section provides an overview of the various options.

There are two different types of communication processors for WinCC:

- communication processors for the so-called Hardnet. They have their own microprocessors and reduce the load on the system's CPU. It is possible to use two different protocols at the same time (multi-protocol operation).
- communication processors for the so-called Softnet. They do not have their own microprocessors. Only one protocol can be used at a time (mono-protocol operation).

Channel Unit Assignment

The table below presents the assignment of the channel units of the Channel "SIMATIC S7 Protocol Suite" to networks and automation systems.

Channel's Channel Unit	Communication Network	Automation System
MPI	MPI	S7-300 and S7-400
PROFIBUS + PROFIBUS (II)	PROFIBUS	S7-300 and S7-400
Industrial Ethernet + Industrial Ethernet (II)	Industrial Ethernet	S7-300 and S7-400
TCP/IP	Industrial Ethernet via TCP/IP	S7-300 and S7-400
Named Connections	Industrial Ethernet or PROFIBUS	S7-400 H/F Systems
Slot PLC	"Soft K-Bus" (internal)	PC (internal)
Soft PLC	"Soft K-Bus" (internal)	PC (internal)

MPI

For communication with the S7-300 and S7-400 ASs via MPI, Channel "SIMATIC S7 Protocol Suite" includes the "MPI" channel unit.

The MPI network corresponds in largest part to the PROFIBUS network with preset parameters and limitations with regard to the number of subscribers and the transfer rate. The same communication processors and modules are used for communication via MPI as for the PROFIBUS network. The same communication protocols are also used.

Automation System Communication Links

The S7-300 and S7-400 programmable controllers can communicate via a MPI network using their internal MPI port or a suitable communication module. The table shows the recommended components.

System	CPU or Communication Module (recommended)
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communication Processors for WinCC

The following table shows the communication processors recommended for connecting a WinCC station to a MPI network. Only one communication processor per WinCC computer can be used for the MPI communication. Each card must also have a driver suitable for the respective communication protocol.

Communication Processor (WinCC)	Assembly/Type
CP 5613	PCI Card/ Hardnet
CP 5511	PCMCIA Card / Softnet
CP 5611	PCI Card / Softnet

PROFIBUS

For communication with the S7-300 and S7-400 ASs via PROFIBUS, Channel "SIMATIC S7 Protocol Suite" includes the "PROFIBUS" and "PROFIBUS II" channel units.

The channel units support communication via the Hardnet and Softnet modules.

Automation System Communication Links

The S7-300 and S7-400 programmable controllers can communicate via a PROFIBUS network using their internal port or a suitable communication module. The table shows the recommended components.

System	**CPU or Communication Module**
S7-300	CPU 31x CP 342-5 CP 343-5
S7-400	CPU 41x CP 443-5 Ext. CP 443-5 Basic

Communication Processors for WinCC

The following table shows the communication processors recommended for connecting a WinCC station to the PROFIBUS. The "PROFIBUS" channel unit supports communication via Hardnet and Softnet cards. A WinCC station supports the use of up to two of these modules. Each communication processor must also have a driver suitable for the respective communication protocol.

Communication Processor (WinCC)	**Assembly/Type**
CP 5613	PCI Card/ Hardnet
CP 5511	PCMCIA Card / Softnet
CP 5611	PCI Card / Softnet

Industrial Ethernet and TCP/IP

In WinCC, the Channel "SIMATIC S7 Protocol Suite" supports multiple channel units for communication via Industrial Ethernet:

- "Industrial Ethernet" and "Industrial Ethernet (II)" Channel Units for "ISO" Protocol with S7 Functions
- "TCP/IP" Channel Unit for "ISO-on-TCP" Protocol with S7 Functions

The channel units support communication via the Hardnet and Softnet modules.

Automation System Communication Modules

If the S7-300 or S7-400 AS should communicate via an Industrial Ethernet using the "ISO" or "ISO-on-TCP" protocol, it must be equipped with a suitable communication module. The table shows the recommended components.

System	Communication Module for Industrial Ethernet	Communication Module for TCP/IP Protocol
S7-300	CP 343-1	CP 343-1 TCP
S7-400	CP 443-1	CP 443-1 TCP CP 443-1 IT

Communication Processors for WinCC

A WinCC station uses Industrial Ethernet to communicate via the "ISO" or "ISO-on-TCP" protocol with one of the communication processors recommended in the table below.

Each communication processor also has a driver suitable for each of the respective communication protocols.

Communication Processor (WinCC)	Assembly/Type
CP 1612	PCI Card / Softnet
CP 1613	PCI Card/ Hardnet
CP 1411	ISA Card / Softnet
CP 1512	PCMCIA Card / Softnet

18.11 WinCC "SIMATIC TI Ethernet Layer 4" Channel

Introduction

The communication driver is used for example to connect a SIMATIC TI505 automation system via the ISO transport protocol.

Channel Units

They have two channel units, which support a maximum of CP 1613s. The channel units are functionally identical. They differ only in the logical device names of the two CP 1613s.

The logical device name can be changed via the System parameter of the channel unit. It is also possible to set the parameter for the ISO transport protocol here.

The following applications are supported:

- Channel Unit 505-Ethernet (CP 1413-1) for the communication module for SIMATIC Industrial Ethernet (e.g. CP 1613).
- Channel Unit 505-Ethernet (CP 1413-2) for the communication module for SIMATIC Industrial Ethernet (e.g. CP 1613).

18.12 WinCC "SIMATIC TI Serial" channel

Introduction

The communication driver is used for the serial connection to the SIMATIC TI505 automation system.

Channel units

The communication driver has a channel unit for operating a COM port for the serial connection.

The following option exists with regard to its use:

- Channel unit "505 Serial Unit #1" for serial communication, either by means of the TBP protocol or by means of the NITP protocol

Note

It is possible to operate several logical connections (using different COM ports) by means of one channel unit.

18.13 WinCC "System Info" Channel

Mode of Operation

The "System Info" channel is used to evaluate system information such as the time, date, drive capacity and provides functions such as timers and counters.

Possible applications are:

- Display of the time, date and day of the week in process pictures.
- Triggering of events through evaluation of system information in scripts.
- Display of the CPU load in a trend graph.
- Displaying and monitoring the available drive space on different servers of a client system.
- Monitoring of the available drive capacity and triggering of a message.

The channel requires no hardware, since it directly accesses the system information of the computer on which it has been installed. In order for the channel to function, a connection must be set up. Additional connections are possible, but not required for the proper operation.

For further information regarding the diagnosis of channels and tags, please see "Communication Diagnostics".

18.14 Diagnosis of Channels and Tags

18.14.1 General Information about Error Detection

A fault or error in establishing a communication link is generally first detected in Runtime.

Objects dynamized using WinCC tags, which cannot be supplied with current process values, will be displayed in the process picture as inactive. These could be e.g. I/O fields, slider objects or bar graphs.

If the fault does not effect some of a connection's WinCC tags, this indicates that one of the WinCC tags is the source of the trouble. In this case, you should for example check the addressing of the tags as well as their spelling when used in Graphics Designer.

If the fault effects all of a connection's WinCC tags, this indicates a fault in the connection itself.

The following sections describe which measures and means can be used to pinpoint the source of the error.

18.14.2 Channel Diagnosis

The following are available to support the diagnosis of channels and their connections:

- The Function "Status - Logical Connections"
- WinCC "Channel Diagnosis"

18.14.2.1 The Function "Status - Logical Connections"

With the function "Status - Logical Connections", WinCC Explorer offers a simple way to display the current status of all configured connections. However, the status can only be displayed in Runtime.

The function is started in WinCC Explorer via the "Extras" menu.

Note
Detailed information regarding the connection status can be had with "WinCC Channel Diagnosis".

18.14.2.2 Diagnosis of Channels with Channel Diagnosis

Introduction

WinCC's "Channel Diagnosis", enables the WinCC user to gain a quick overview of the status of the active connections in Runtime. On the one hand, "Channel Diagnosis" provides status and diagnostic information regarding channel units and, on the other hand, it serves as a user interface for the configuration of the diagnostic output:

- The output of statistics or information on the status of the communication e.g. in a process picture
- Text output in a logbook file for fault analysis and correction by Service.
- Text output in a trace file to assist the Hotline in pinpointing the cause of communication problems.

The Diagnostic module can be inserted into a process picture as an ActiveX control or be started as an independent application from the Windows Start menu. The module will only display status information for channels that support diagnosis.

The diagnosis of a channel's tag can be found in the description of the channel-specific diagnosis.

Logbook File

"Channel Diagnosis" will create a logbook file named <Channel_Name.log> for every configured WinCC channel. The system will record important information and errors in these. The exact text content depends on the channel.

The creation of the file and the output text cannot be configured.

The logbook files contain information such as the start and end messages, version information and information regarding communication errors.

Each entry in the file consists of a data and time stamp, the flag name and a description. The file will always be saved immediately after an entry to ensure that e.g., in the event of a voltage drop, all of the information possible will be available.

Trace File

A trace file named <Channel_Name.trc> can be created to hold error messages and any additional information for every configured WinCC channel. You can select in Runtime whether to use a trace file. When the trace function is activated, a message will be displayed warning that the link's propagation time will be affected.

Each entry in a trace file has a time stamp followed by a flag name and description.

When the trace function is enabled all information recorded in the logbook will also be written to the trace file.

The information recorded in a trace file is intended to assist the Hotline in pinpointing the cause of communication problems.

Note

The trace and logbook file entries are only recorded in English.
Both files are saved in the "Diagnostics" directory in the WinCC directory structure.
The current Counter values are not recorded in these files.

Index